

Додаток А.

Корисні ресурси при розробленні дизайну

RESOURCE CARDS

<https://resourcecards.com/>

DESIGN / THEMES

<https://www.behance.net/>

<https://dribbble.com/>

<https://www.pinterest.com/>

<http://www.awwwards.com/>

<https://themeforest.net/>

<https://www.templatemonster.com/>

PSD / VECTOR

<http://www.freepik.com/>

<https://psdrepo.com/>

<https://freebiesupply.com/>

<https://photoshop-master.ru/>

<https://www.vecteezy.com/>

FONTS

<https://fonts.google.com/>

<https://fontstorage.com/>

ICONS

<http://www.flaticon.com/>

<https://thenounproject.com/>

<https://www.iconfinder.com/>

TOOLS

<https://snazzymaps.com/>

<https://online-generators.ru/>
<http://quirktools.com/screenfly/>

STOCK PHOTO

<https://unsplash.com/>
<https://www.pexels.com/>
<https://stocksnap.io/>
<https://pixabay.com/>

<http://pngimg.com/>

HTML / CSS

<http://codepen.io/>
<https://tympanus.net/codrops/>

COLORS

<https://color.adobe.com/ru/create/color-wheel/>
<https://colorscheme.ru/>
<https://colorhunt.co/>
<https://flatuicolors.com/>
<https://colorscheme.ru/color-names.html>

BLOGS

<https://medium.com/topic/design>
<https://habr.com/ru/>
<https://telegraf.design/>
<http://deadsign.ru/>
<https://ux.pub/>
<https://say-hi.me/>
<https://prodesign.in.ua/>
<http://www.dejurka.ru/>
<https://designpub.ru/>
<https://infogra.ru/>

WEB AGENCY

<https://thegradiant.com/>

<https://brcl.ru/>

<https://zipl.pro/>

<https://galverstka.com.ua/ru/>

<https://chulakov.ru/>

<http://www.theglyph.studio/#home>

<https://fantasy.co/>

<http://banda.agency/>

<https://cpeople.ru/>

<https://m2hagency.com/>

<https://aic.ru/>

<http://otdelx.ru/>

<https://vintage.com.ua/>

<https://tubikstudio.com/>

<https://otakoyi.com/uk/>

<https://dops.digital/uk>

Додаток Б.

Керівництво користувача засобом прототипування та проектування дизайну FIGMA.

План

1. Установка Фігми
2. Файли, проекти и команди
3. Зверху: тулбар и темна панель
4. По центру: робоча область
5. Зліва: панель шарів и сторінки
6. Справа: панель властивостей
7. Фрейми
8. Шейпи
9. Перо
10. Векторні мережі
11. Булеві групи и флетен
12. Режим Outline: векторні контури
13. Режими колірного кодування
14. Заливка і градієнти
15. Режим заливки Image
16. Кольорокорекція
17. Обводка
18. Маски
19. Адаптивність и обмежувачі
20. Текстові шари
21. Вирівнювання и розподіл
22. Стилi кольорів
23. Компоненти

1. Установка Фігми

Клиенты

У Фігми є клієнт для Mac і Windows, а також безкоштовний додаток Figma Mirror для iOS і Android. Mirror - обов'язковий інструмент дизайнера мобільних додатків і сайтів.

Все це можна завантажити на офіційному сайті: [figma.com/downloads](https://www.figma.com/downloads).

WebGL

Також Фігму можна використовувати в браузері, якщо той підтримує технологію WebGL. Розробники Фігми рекомендують використовувати браузер Google Chrome.

Перевірити, чи підтримує браузер WebGL можна на webglreport.com.

Локальні шрифти

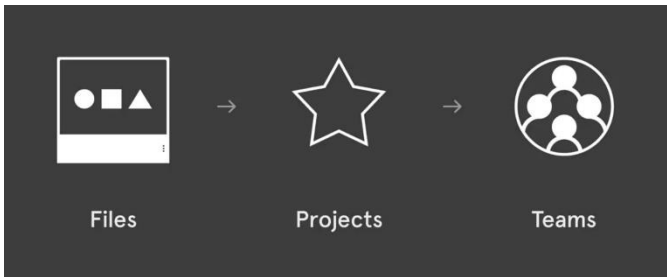
Фігма може використовувати шрифти, встановлені в операційній системі як в редакторі, так і в браузері.

Якщо використовується клієнт для Мака, після установки нового шрифту його не видно в шрифтах. Щоб це виправити, потрібно вийти і заново увійти в додаток Фігми. Щоб шрифти відображалися в браузері, потрібно додаток **Font Helper**. Посилання на його скачування доступне в меню **Help and Account** → **Account Settings**.

Font Helper повинен постійно бути в пам'яті під час роботи.

Практика: сетап

1. Встановити клієнт Фігми.
2. Зареєструвати безкоштовний акаунт.
3. Увійти в Фігму в браузері.
4. Завдання на п'ятірку: Знайти в настройках, як прив'язати номер телефону до акаунту Фігми, і включити двухфакторну авторизацію. Це сильно ускладнить зловмисникам можливість зламати твій акаунт в Фігмі.



2. Файли, проекти и команди

Файли — аналог скетч-проектів. У них можна створювати макети.

За замовчуванням створюються в загальній категорії **Drafts**. На файли можна давати такі посилання. Також файли можна вивантажувати з Фігми в форматі **.fig**. Можна створити необмежену кількість файлів.

Проекти групують файли. Також файли можна створювати відразу в проектах. У безкоштовному акаунті може бути до трьох проектів.

Команди - групи проектів, у які даємо доступ учасникам.

Приклад: Є команда **/designer**, в ній є проект **Figma Guide**, а в ньому є файл **Figma-Guide-Rus**.

Також є секція **Deleted Files**, де зберігаються видалені файли. Там же можна видаляти файли безповоротно.

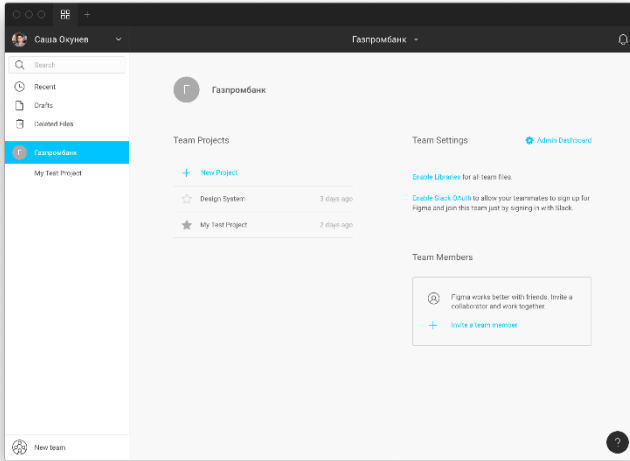
Імпорт

У Фігму можна імпортувати файли в розширеннях **.fig** або **.sketch**. Щоб імпортувати, натискаємо кнопку **Import**, або перетягуємо файли у вікно Фігми.

Команди

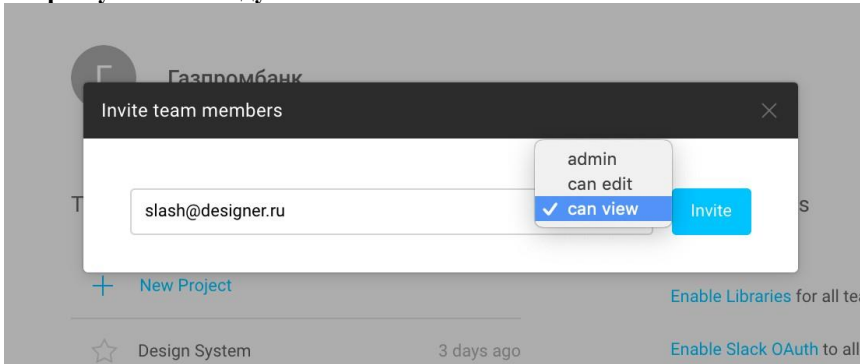
Команди потрібні, щоб організувати роботу групи людей, які працюють над кількома проектами.

Щоб створити команду, натискаємо **New team**.



На сторінці команди доступні проекти, які до неї належать, а також налаштування.

Запрошуємо команду



Натискаємо **Invite a team member** і визначаємо рівень доступу. За кожного редактора або адміністратора команди Фігма вимагає 15 \$ в місяць. Кількість глядачів необмежено.

У запрошеного користувача в його вікні Фігми в верхньому правому куті

з'являється повідомлення, яке можна прийняти або відхилити.

Enable Libraries

Функція **Enable Libraries** дозволяє зробити доступними бібліотеки для всіх файлів команди.

Slack OAuth

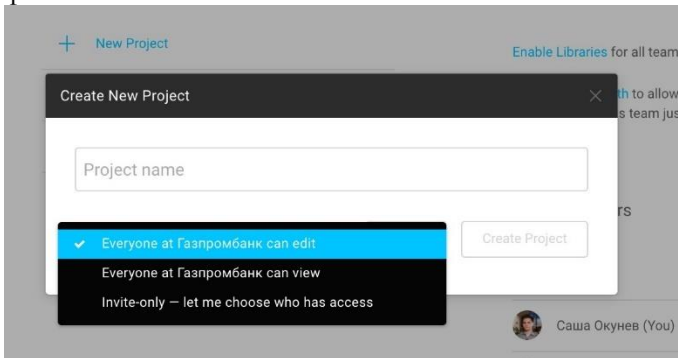
Функція **Enable Slack OAuth** дозволяє авторизуватися в Фігмі і приєднатися до команди, якщо авторизований в Слак.

Створення проекту

На сторінці команди ми можемо створювати проекти, натиснувши **New Project, Cmd + N**.

При створенні проекту можна визначити **рівень доступу**:

- **Everyone at Team can edit** – будь-який член команди може редагувати файли всередині неї.
- **Everyone at Team can view** – спочатку всі можуть тільки дивитися файли.

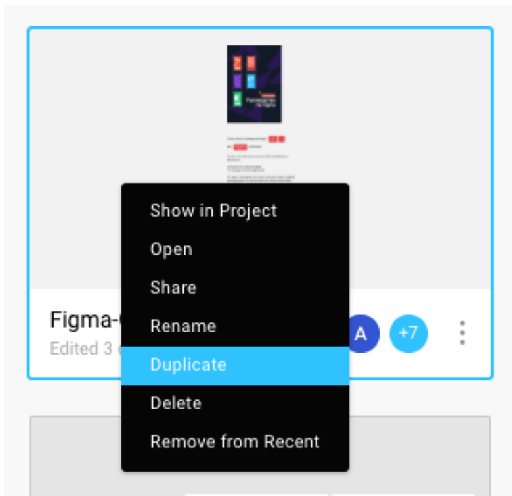


Invite-only – доступ в проекти за індивідуальними запрошеннями.

Проект для вивчення: як скачати

В процесі написання цієї книги я робив ілюстрації в Фігмі і знімав скріншоти, працюючи в проекті **Figma-Guide-Rus**.

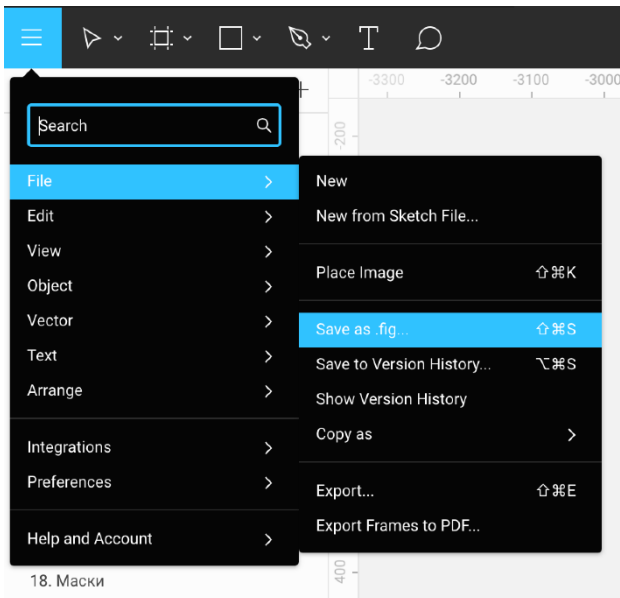
До нього можна отримати доступ за посиланням bit.ly/figma-examples.



Варіант 1, через дублювання файла

Коли ти пройшов по посиланню, проект з'являється у вкладці **Recent**.

Там його можна дублювати в свій простір. Копія буде доступна для редагування.



Варіант 2, зі збереженням .fig-файла

Його можна відкрити в Фігмі, зберегти як .fig-файл і скопіювати в свій акаунт для подальшого розбору:

File → **Save as fig...**, **Shift + Cmd + S**.

У кожному розділі підручника я даю посилання на відповідні сторінки цього проекту.

Практика: перший файл

1. Відкрити десктопний клієнт Фігми.
2. Створити команду, вибрати для неї аватар.
3. Створити в ній проект.
4. Створити в ньому перший файл.

3. Зверху: тулбар и темна панель

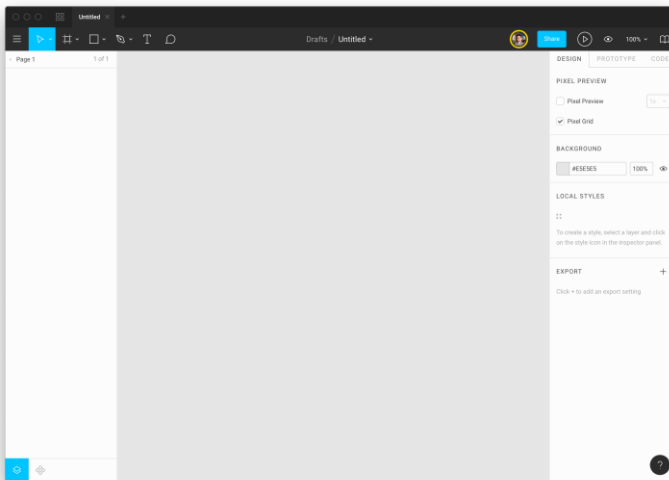
У цьому розділі розглянемо головне меню і темну верхню панель, в якій ми вибираємо інструменти і налаштовуємо доступ.

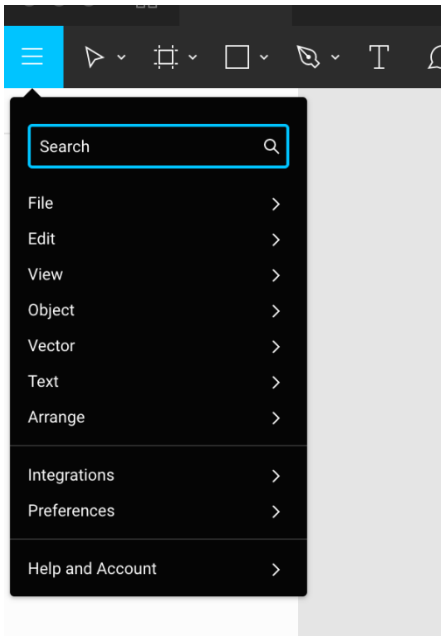
Відкрили новий файл. Закрити — **Cmd + W**. Зліва від відкритої вкладки відкритого файлу видно вкладку повернення на сторінку проекту. Можна створити ще один файл, натиснувши +. Відкриється ще одна вкладка.

Перемикається між вкладками можна як в браузері:

Shift + Cmd + [— Попередня вкладка.

Shift + Cmd +] — Наступна.

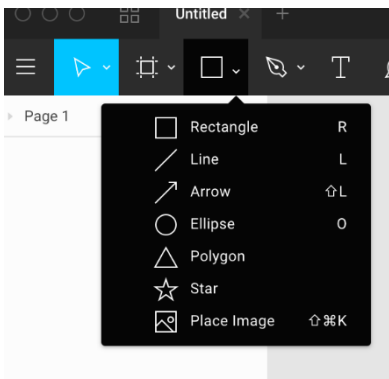




Головне меню

У верхньому лівому кутку - головне меню програми, яке можна відкрити клавішею **Cmd + /**.

В меню є пошук по пунктам. Розгорнувши меню клавішею, можна продовжити друкувати потрібну команду.



Тулбар

Панель інструментів в Фігмі (**Toolbar**) працює так само як в інших редакторах. Ми розберемо кожен інструмент в подальших главах.

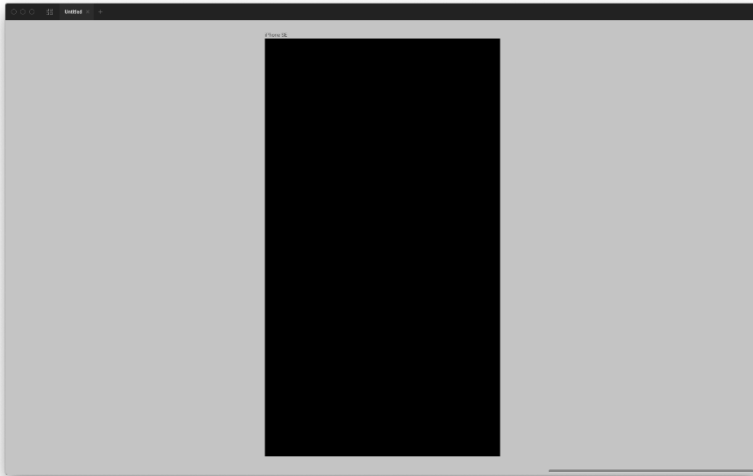
Спочатку активний інструмент **Move**. Якщо вибрати інший, в **Move** можна повернутися, натиснувши **Esc**.

V,Esc	Move
K	Scale
F,A	Frame
S	Slice
R	Rectangle
L	Line
Shift + L	Line with Arrow
O	Ellipse, Oval
P	Pen
Shift + P	Pencil
C	Comment

Shift + Cmd + K Place Image

Приховування інтерфейсу

Cmd + | приховує бокові панелі. Чи не плутаємо символ |[пайп] буквой I [ай]. На клавіші |намальовані Ë і \.



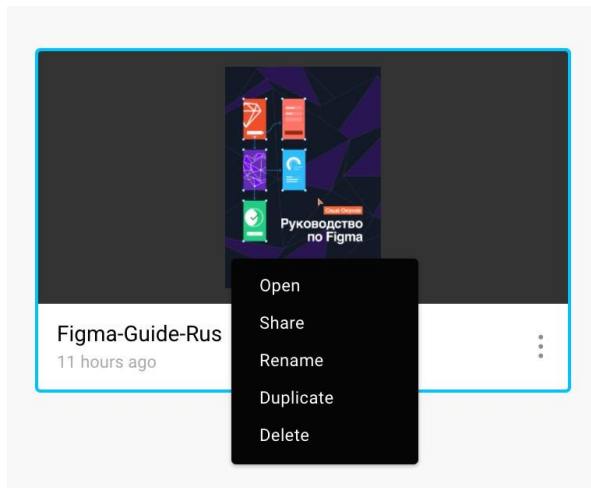
Бокові панелі приховані:

Перейменуємо файл

У центрі екрана вказані **назва проекту / ім'я поточного файлу**.

Зараз файл називається **Untitled**. Якщо клікнемо за назвою, ми зможемо редагувати його.

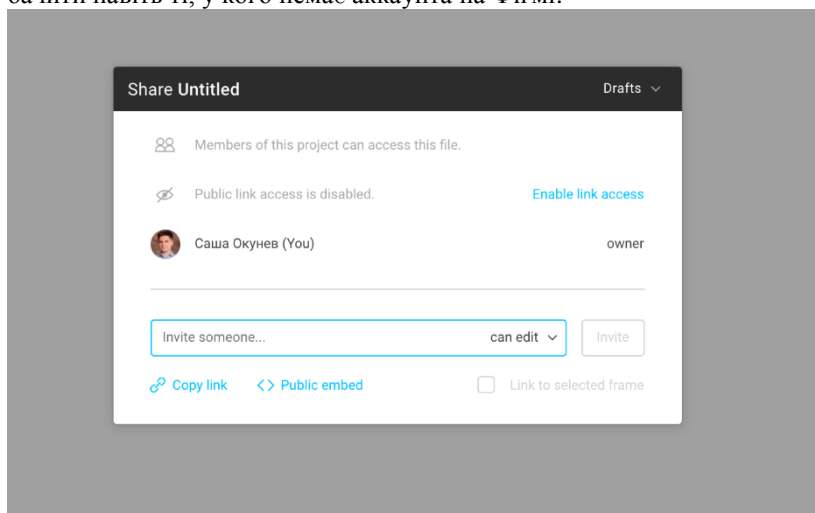
Також перейменовувати файл можна, перебуваючи на екрані проекту. У контекстному меню є пункт **Rename**:



Як давати доступ по посиланню

Блакитна кнопка **Share** в правій частині верхньої панелі дозволяє налаштувати, хто буде бачити проект.

Кнопка **Enable Link Access** дозволяє створити публічне посилання, яке можуть бачити навіть ті, у кого немає аккаунта на Фігмі.



Даючи доступ за посиланням, потрібно пам'ятати, що той у кого є таке посилання має можливість зберегти вихідний файл проекту собі. Це може бути небажано, для тих дизайнерів, які переживають, що їх роботу вкрадуть, не

заплативши. Фігма працює над вирішенням цієї уразливості.

Практика: тулбар и посилання на проект

1. Розібратися, як перемикає інструменти і повертатися в **Move**.
2. Зробити публічну посилання на файл.
3. Відкрити її в браузері, в якому виконаний вхід в Фігму.
4. За посиланням в браузері знайти курсор, який залишає десктопний клієнт. На ньому має бути твоє ім'я. Переконайся, що вийшло синхронно дивитися файл з двох різних вікон.

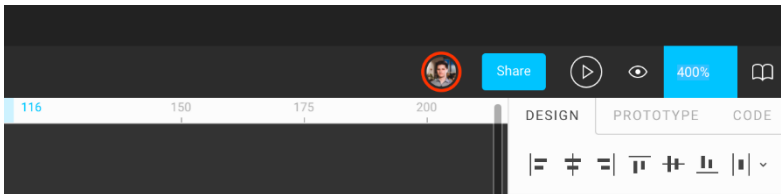
4. По центру: робоча область

У центрі екрана ми бачимо сіре безрозмірний полотно. Це наш всесвіт, в якому ми можемо творити. Полотно може бути величезне і його досить для створення проекту будь-якого розміру, хоча насправді він не нескінченний.

По полотну можна повзати, затиснувши **пробіл** і перетягуючи простір лівою клавішею. На Маках для навігації по полотну зручно використовувати трекпад. На Windows замість пробілу також можна затискати **середню кнопку миші**.

Існують нульові координати: **x: 0, y: 0**, це центр всесвіту, щодо якого позиціонуються всі верстви, у яких немає батьківських шарів.

Шар будь-якого типу має базові властивості: координати по **x, y, ширину, висоту і кут нахилу**.



Працюємо змасштабом

У віконці **Zoom** в правому верхньому куті відображається коефіцієнт масштабування у відсотках. Можна клікнути його і ввести потрібне.

Plus — наближає вдвічі.

Minus — віддаляє вдвічі.

Shift + 1 **Zoom To Fit**, аналог **Cmd + 1** в Скетчі.

Виставляє такий коефіцієнт масштабування і кадрування, щоб весь контент сторінки вліз в робочу область. Зручна команда, щоб охопити поглядом всю сторінку.

Shift + 2 **Zoom To Selection**, аналог **Cmd + 2** в Скетчі.

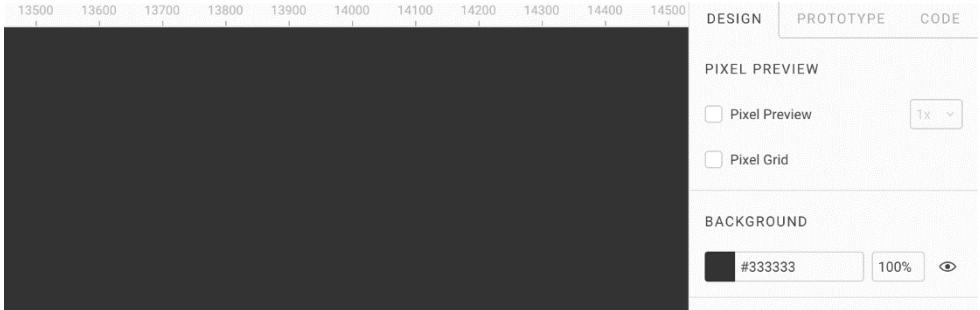
Виставляє такий зум і кадрування, щоб виділений об'єкт зайняв максимум простору робочої області. Команда, яка економить багато сил на налаштування масштабу вікна під певний шар на полотні. Чи виділена маленька іконка або група екранів, ця команда покаже виділене під весь доступний розмір робочої

області.
Shift + 0

Zoom to 100%. Фактичний проектувальний масштаб.

Для гіків

Шари можуть перебувати в діапазоні від -66 400 px до +66 000 px. Вони можуть мати і ще більш екстремальні значення, але перестає працювати **Zoom to selection, Shift + 2**. Це робить такий далекий космос марним.



Фон робочої області

Можна задати його колір у вкладці **Design**.

Варто врахувати, що він повинен бути таким, щоб дрібні написи в заголовках екранів читалися. Фігма може писати назви фреймів блакитним, а компонентів фіолетовим кольором і це слід врахувати.

Око навпроти рядка з кольором дозволяє робити прозорий фон робочої області з канонічними шашечками.

В якості темного фонового зручний сірий # 333333, а в якості світлого - # F2F2F2. Щоб встановити ці кольори, досить ввести в поле **Background** з клавіатури **33** або **F2**.

Робити фон темним зручно коли довго довго працюєш над проектом, особливо в темній кімнаті. Чим менше світла потрапляє в очі, тим комфортніше.

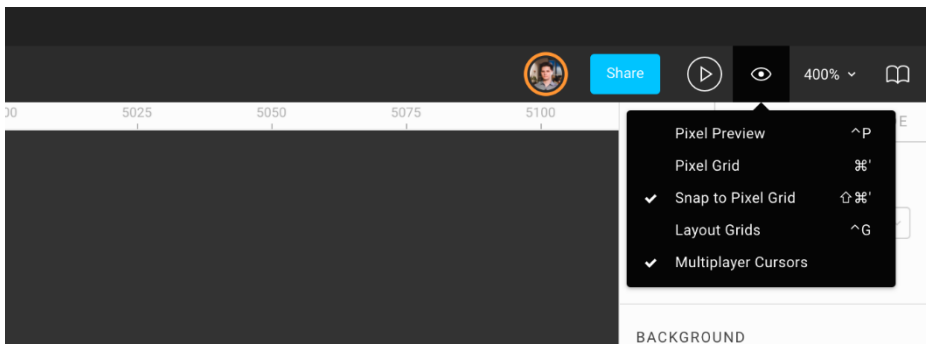
Налаштування кольору в певній сторінці глобальна для всіх учасників команди, тому слід домовитися про ті кольори, які комфортні для всіх.

Не варто робити полотно білим, інакше фрейми будуть недостатньо контрастні, навіть якщо їх контури видно.

Щоб виявляти і приховувати контури фреймів (екранів дизайну), використовуємо команду **Frame Outlines, Cmd + /**, потім продовжуємо друкувати: **ou**.

Недостатньо контрастний фрейм буде важко знайти на полотні, а фон екрану в ньому може злитися з фоном глобального полотна:

Frame 3



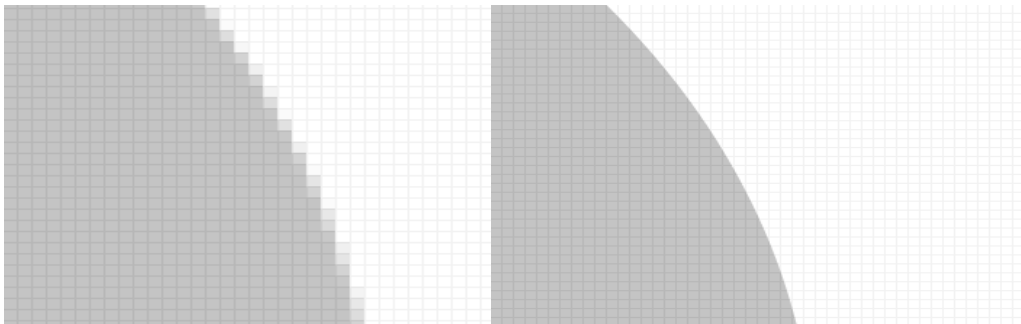
Режими відображення робочої області

Іконка ока приховує меню режимів відображення. У ньому можна налаштувати, що видно в робочій області.

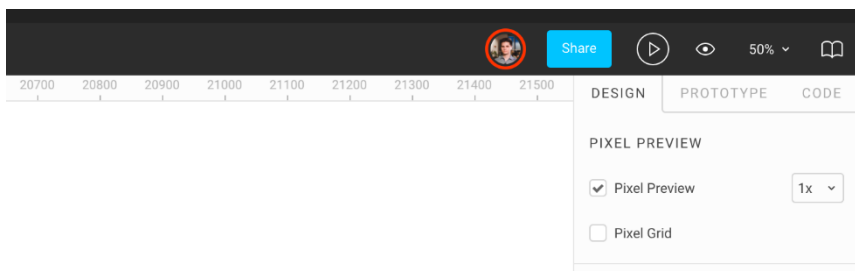
Ctrl + P

Pixel Preview — в якому видно пікселі в щільності 1x або будь-якому іншому, налаштованому в випадаючому полі.

На Windows для цієї команди використовується клавіша **Ctrl + Alt + Y**.



Зліва: включений, праворуч: вимкнений.



Pixel

Grid, Cmd + ' — тогл-режим¹, в якому показується піксельна сітка. ' на тій самій клавіші, що і Э.

Також режими **Pixel Preview** і **Pixel Grid** доступні у вкладці **Design**.

¹ Тогл-режим — режим роботи, який може бути або увімкнений, або вимкнений.

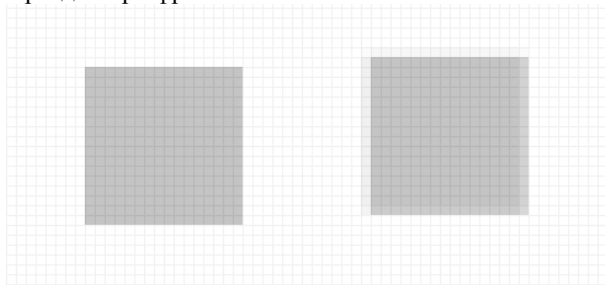
Snap To Pixel Grid, Shift + Cmd + ' —

— тогл-режим прив'язки шарів до сітки під час створення або переміщення. Якщо активний, всі нові або рухомі шари ставляться на рівні пікселі. Якщо відключений, вони можуть потрапити на дробові значення пікселів (напівпікселі). В цьому випадку координати матимуть числа після точки: 2.29. При щільності пікселів 1x, контури такої фігури, будуть відображатися нечітко.

Щоб створити на полотні сірий квадрат, натиснемо **R** і клікнемо в будь-якому місці.

Ctrl + G Layout Grids

— тогл-режим, який включає і відключає сітки для всіх фреймів. Розберемо сітки в розділі про фрейми.

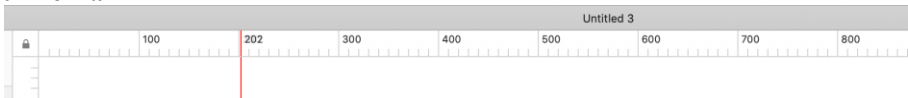


Щоб не відволікатися на курсори інших користувачів, які одночасно працюють у

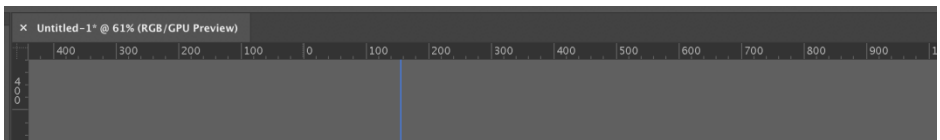
файлі, відключимо їх:

View settings → **Multiplayer Cursors**

Лінійка



Щоб включити лінійку в Скетчі, використовуємо **Ctrl + R**, команда **Show / Hide Rulers**. Напрявні ставляться кліком на перпендикулярній осі, в даному випадку, **X**:



У Ілюстраторі і фотошопі команда називається так само і має клавішу **Cmd + R**. Направляючі потрібно витягати з паралельної осі, в даному випадку, **Y**.



У Фігмі команда називається **Rules** і має клавішу **Shift + R**. Напрявні витягуються так само, як в Ілюстраторі, витягуванням.

Практика: робоча область

1. Освоїтися з навігацією по полотну: навчимося наближати-видаляти, перемикатися на загальний і близький вид.
2. Створити фрейм.
3. У ньому створимо сірий квадрат. Наблизимо його до великого масштабу.
4. Включимо **Snap To Pixel** і **Pixel Preview**.
5. Пересунемо квадрат і відчуємо, як він липне.
6. Відключимо **Snap to Pixel**.
7. Пересунемо квадрат і побачимо, як попливли його краї.
8. Знову включимо прив'язку. Пересунемо квадрат, побачимо, що він знову стоїть по піксельній сітці.
9. Включимо лінійку, якщо її немає. Приховуємо якщо є.
10. Поставимо гайд (направляючу) по лівому краю нашого квадрата.
11. Дублюємо квадрат, затиснувши **Opt / Alt** і відтягнувши отримавший другий квадрат від першого. Поставимо його нижче першого, вирівнюючи по

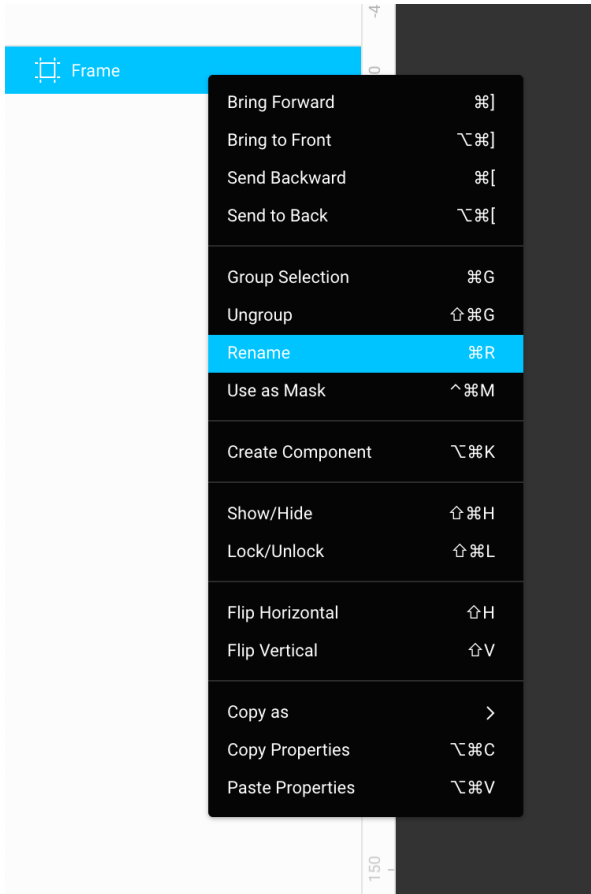
гайду.

5. Зліва: панель шарів і сторінки

Панель шарів складається з двох вкладок:

Layers, Opt + 1 Components, Opt + 2

У першій вкладці ми бачимо список всіх шарів, які існують в робочій області. У кожного шару є контекстне меню, де можна підглянути більшість гарячих клавіш:



Корисні клавіші для роботи із шарами

Cmd + D

Дублювати шар. Новий з'явиться в списку шарів нижче першого. На полотні він буде правіше свого батька.

Якщо натискати кілька разів, шари будуть дублюватися на тому ж абзаці, який був між оригіналом і першим дублем.

Також дуже потужна техніка - затиснути **Opt** і тягнути шар, який ми хочемо дублювати.

Cmd + R

Перейменувати шар, **Enter** для збереження імені

Tab

Виділити наступний

Shift + Tab

Виділити попередній

Enter

Розгорнути поточний фрейм, компонент або групу

Shift + Enter

Виділити батьківський фрейм

У режимі перейменування можна змінювати фокус на сусідні шари за допомогою **Tab** і **Shift + Tab**. Шар, з якого ми прибрали фокус, зберігається.

Shift + Cmd + L Заблокувати или разблокувати слой. Заблокованный остаётся видимым, но его нельзя двигать и выделять.

Shift + Cmd + H Приховати / показати шар. На відміну від Скетча, в Фігмі можна приховувати екрани.

Сортировка

Cmd +] або **Opt + Cmd + ↑** Перетягнути шар на одну позицію вгору

Cmd + [або **Opt + Cmd + ↓** Вниз.

Opt + Cmd +]

Перетягнути шар в самий верх списку.

Opt + Cmd + [

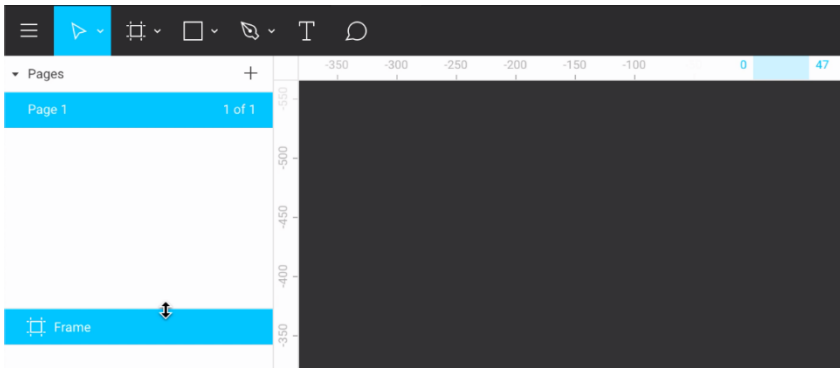
Поставити останнім

Сторінки



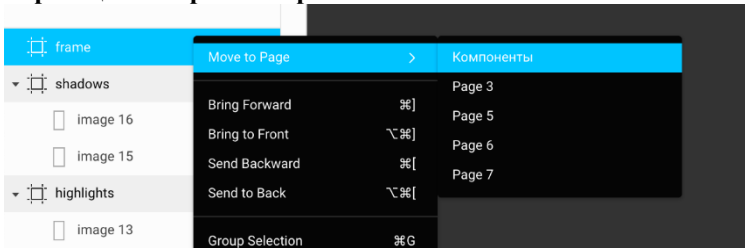
Як і в Скетчі, сторінки в Фігмі - секції, в яких містяться наші екрани. Зручні для групування їх за змістом. В кожній сторінці своя нескінченна координатна площина. Є де розвернутися!

fn + ↑↓ Гортаємо сторінки з клавіатури. Windows—**PageUp, PageDn**.



Клік на рядок **Page 1** розгортає список сторінок. Кнопка + створює сторінки. Якщо у файлі занадто багато сторінок, можна розтягнути вниз і дати їм більше простору:

Переміщення шарів між сторінками



На відміну від Скетча, в Фігмі не працює перетягування фрейма в іншу сторінку. Для переміщення слід використовувати команду **Move To Page** контекстного меню.

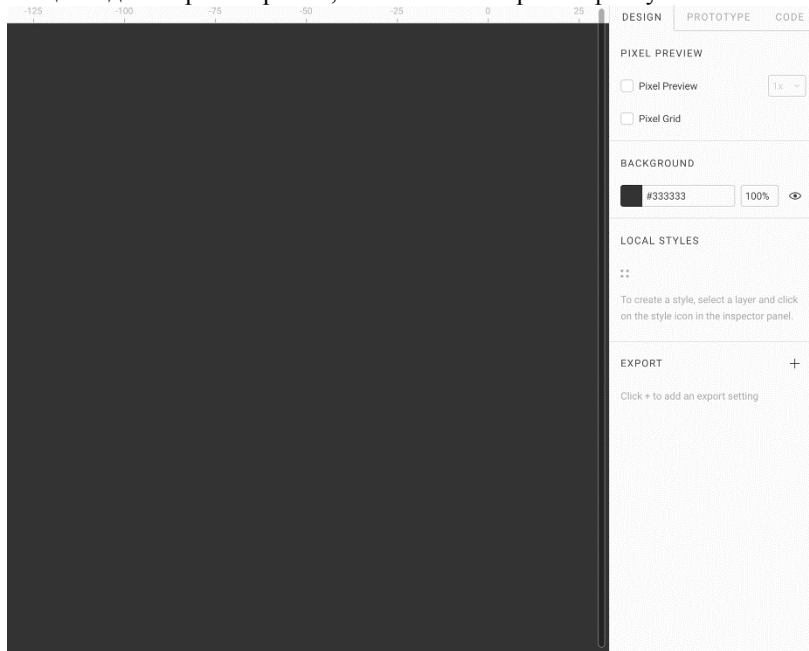
Практика: список шарів и сторінки

1. Створимо фрейм.
2. Дублюємо його 3 рази, щоб вийшло 4 фрейми.
3. Пронумеруємо їх від верхнього до нижнього: 1 → 4.
4. Поставимо фрейм 1 нижнім.
5. Другий заблокуємо.
6. Третій приховуємо.
7. Створимо сторінку Page 2. З'ясуємо, як перейменовувати сторінки.
8. Надіслати фрейм 4 на неї.

6.Справа: панель властивостей

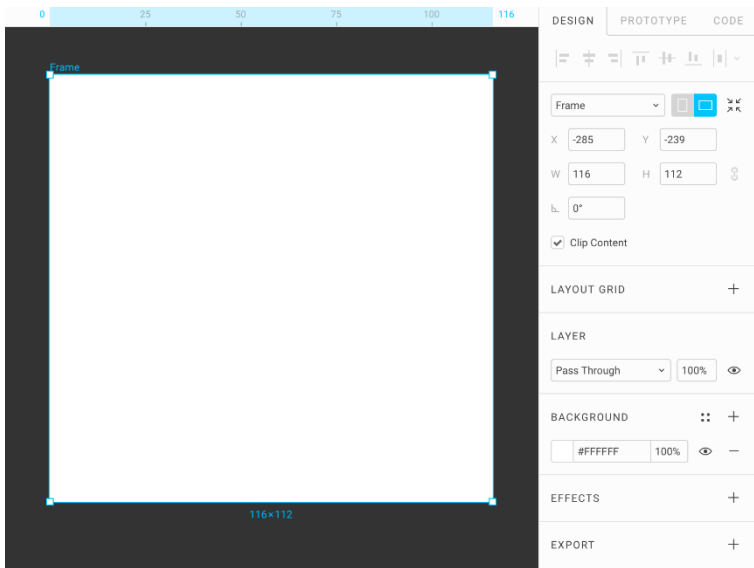
У правій частині інтерфейсу видно велику білу панель у всю висоту вікна - панель властивостей. Залежно від виділеного типу шару в ній може бути різний набір полів і кнопок.

Якщо жоден шар не обраний, ми бачимо настройки файлу.



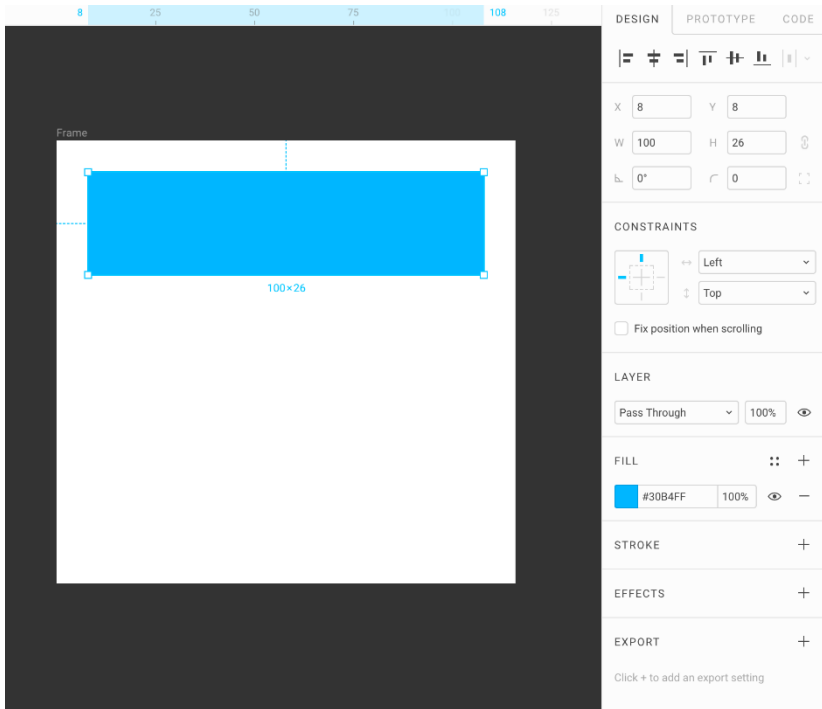
У такій панелі видно вже розглянуті нами режими перегляду **Pixel Preview**, настройка кольору робочої області, спочатку порожній блок **Local Styles** для палітри, а також блок **Export** для масового експорту.

В панелі властивостей є три вкладки: **Design**, **Prototype** і **Code**. Для створення дизайну ми працюємо з першою.



Для прикладу створимо шар типу фрейм, **F**. Якщо виділити його, панель праворуч відобразить його властивості:

Зверху ми бачимо настройки вирівнювання, спочатку недоступні, тому що в порожньому фреймі нічого вирівнювати. Можна налаштувати розмір фрейма під розмір екрану певного пристрою. Для цього виберемо потрібне з довгого списку доступних пристроїв.



У блоці **Layout Grid** налаштовується сітка, в **Layer** - режим накладення, в **Background** - фон, в **Effects** тіні і блюр. У блоці **Export** можна налаштувати, в які формати шар буде експортуватися.

У фреймі створимо прямокутник, **R**.

Якщо виділена векторна фігура (шейп), бічна панель має характерні елементи для цього типу шарів: наприклад, поле заокруглення кутів, якого не було у фреймів. Блок **Constraints** для визначення, як буде вести себе шейп, якщо будемо розтягувати його батьківський фрейм. Детальніше про цю тему поговоримо в розділі **Адаптивність і обмежувачі**. Блок **Stroke**, щоб виконувати завдання обведенням.

Загальні властивості шарів

Шар будь-якого типу має базові властивості:

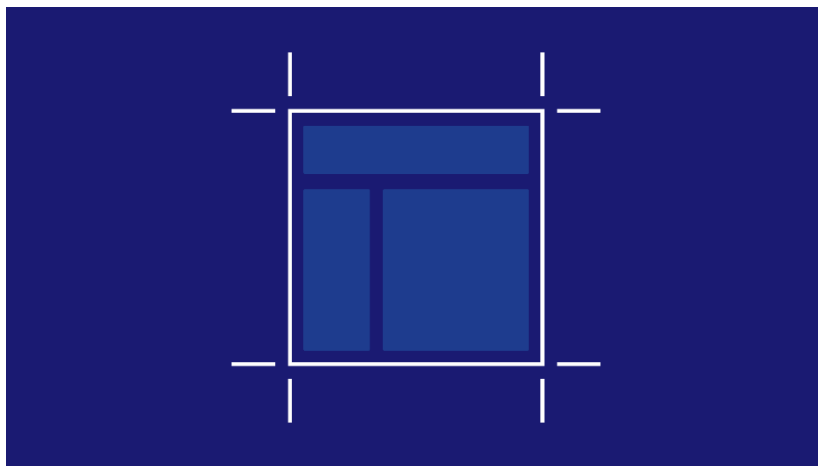
- **X, Y** – координати в пікселях. Якщо шар на нескінченному полотні, абсолютні. Якщо всередині фрейму - відносні.
- **Width, Height** — ширина і висота в пікселях
- Праворуч ширини - кнопка **Constrain Proportions**, дозволяє зберегти пропорції ширини і висоти, якщо вони змінюються в одному з полів.
- **Angle** — кут нахилу. Будь-шар можна нахилити на цей кут.

- **Opacity.** Будь-який шар, включаючи фрейми, може бути напівпрозорим. Можна задавати з клавіатури: **1** — 10%. **5** — 50%. **0** — 100%. Якщо натиснути дві клавіші досить швидко, введемо точне значення: **0,5** — 5%. **5,5** — 55%.

- **Layermode**—режим накладення. За замовчуванням **PassThrough**. У всіх векторних типів: Шейп, текстових шарів і векторних мереж є такі властивості як обведення **Stroke** і заливка **Fill**. **Corner Radius** - радіус заокруглення кутів у шейпів.

Практика по властивостям шара

1. Створимо фрейм.
2. Задамо йому ширину 1600, висоту 900.
3. Задамо йому координати: 0, 0.
4. Нехай фон його буде чорний.
5. Усередині нього створимо білий квадрат розміром 100 x 100 і радіусом заокруглення 10.
6. Вирівняємо квадрат по центру фрейму.
7. За допомогою клавіатури задамо опасіті квадрата в 20%.
8. Експортуємо фрейм в PNG-файл.



7. Фрейми

Проект в Фігмі →

Про те, як працювати з панеллю шарів почали говорити в розділі 5. Розглянули також, що кожен тип шарів має свої унікальні налаштування в панелі праворуч. Ближче знайомство з шарами почнемо з фреймів.

Фрейми - аналог артбордів в Скетчі. Як ми дізналися в минулому розділі, створюються клавішею **F** в режимі інструменту **Frame**.

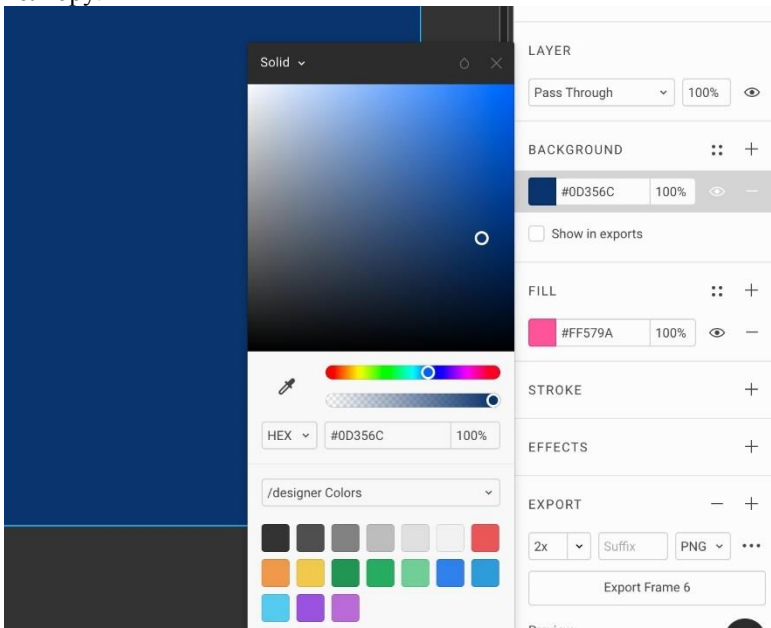
Фрейми дуже важливі при роботі з сіткою. Їм можна задавати єдині стилі сіток, які можна використовувати по всьому файлу.

Мають фоновий колір — **Background**, спочатку білий. Галочка **Show in exports** дозволяє відображати фоновий колір при експорті фрейма в файл. Якщо її зняти, фрейм буде вивантажено в PNG або інший формат з прозорим фоном.

Галочка **Clip Content** дозволяє вибрати, чи буде видно шар, частини якого вилазять за межі фрейма. У Скetchі завжди ріжеться.

Фоновий колір фрейма

Фрейми можуть мати фоновий колір, який задається в блоці **Background**. Спочатку заливка фону стоїть в режимі **Solid**, що дає рівномірне накладення кольору.



Як і в Скetchі, в Фігмі діє клавіша **Ctrl + C**, яка викликає **ColorPicker**—піпетку. На Windows використовується **I**. Пикер дозволяє зняти потрібний колір в будь-якому місці проекту.

Фрейми: Робота з сіткою



В фреймах можна налаштувати сітку. Для цього потрібно виділити фрейм і в блоці **Layout Grid** натиснути +. Це створить в фреймі нову сітку.

Ctrl + G показує або приховує сітку по всіх фреймах в проєкті.

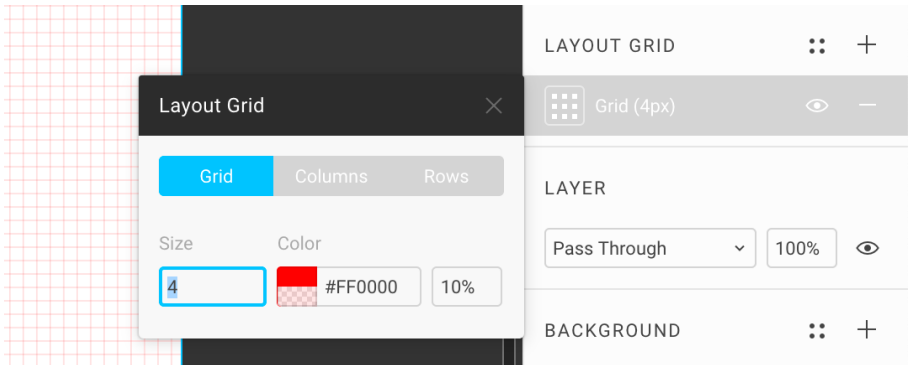


Додали другу сітку:

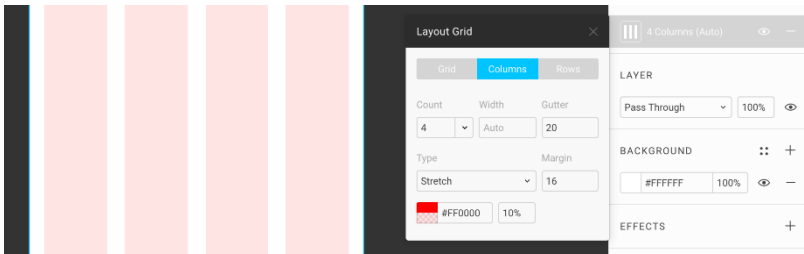
Сітка може бути трьох типів: **Grid**, **Columns**, **Rows**. Вони можуть накладатися один на одного і вибудовуватися в список в блоці **Layout Grid** подібно верствам. Можна включати потрібні сітки, натиснувши око.



Щоб налаштувати тип сітки, натискаємо на іконку:



Grid—базова регулярна сітка.

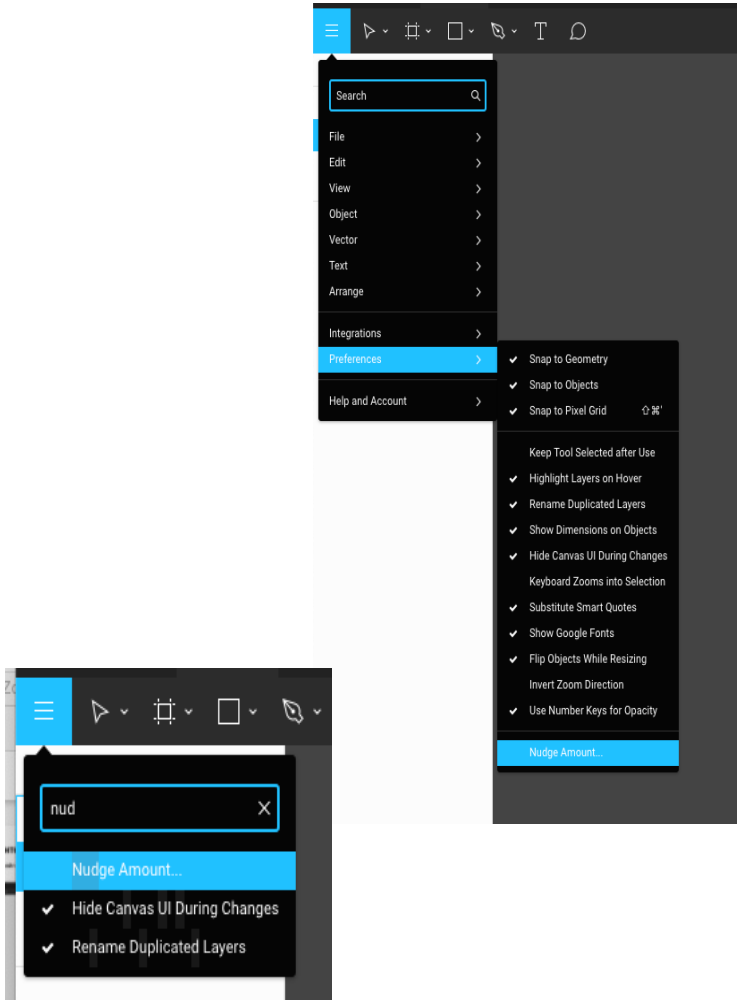


Columns—модульна вертикальна сітка.

Rows—модульна горизонтальна сітка.

Крок зсуву по сітці

У Фігмі можна налаштувати, на яку відстань буде зрушуватися об'єкт, якщо затиснути **Shift** + **стрелки**.



За замовчуванням **10**,

Я рекомендую використовувати **4** або **8** - значення, рівне кроку базової сітки.

Через пошук буде швидше:

Cmd + /, nud

Фрейми для профі

Межі фреймів

Фрейми можуть мати явний кордон, який допомагає розуміти їх розмір в ситуації, коли у фрейма немає фону. Для цього використовується налаштування **View** → **Frame Outlines**, **Cmd + /, ou**.



Зліва **Frame Outlines** вимкнений, праворуч включений. Також у визначенні кордонів фреймів допоможе режим **Outline**, **Cmd + Y**.

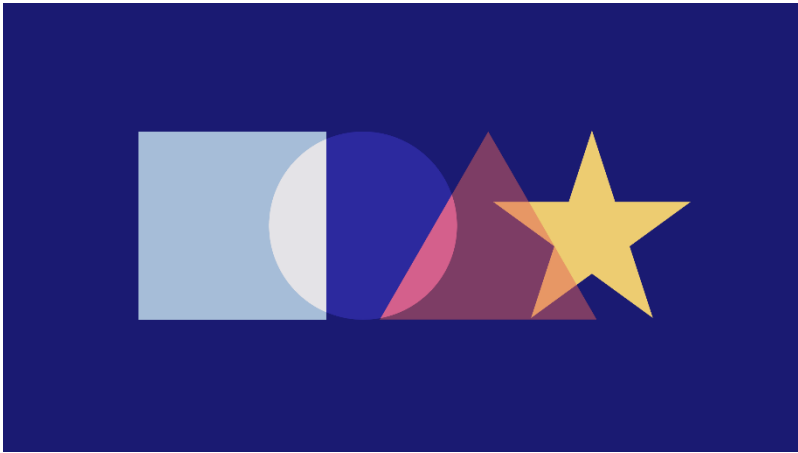
Огортання фреймом

Будь-який об'єкт можна за один клік обернути фреймом. Це буває іноді потрібно, щоб згодом його експортувати або перетворити в компонент.

Для цього використовується команда **Frame Selection**, **Opt + Cmd + G**.

Як і артборди в Скетчі, фрейми - це різновид груп. Тому, їх можна розв'язувати за допомогою **Shift + Cmd + G**.

Якщо фрейм навколо шару вже є і потрібно стиснути його до розмірів цього шару, як і в Скетчі, використовується команда **Resize To Fit**. **Cmd + /, fit**.



8. Шейпи

Проект в Фігмі →

Шейпи (shapes) - векторні фігури з закритим периметром. Якщо периметр розірваний, шейп перетворюється в шлях (path).

Фігма дозволяє створювати такі типи шейпів:

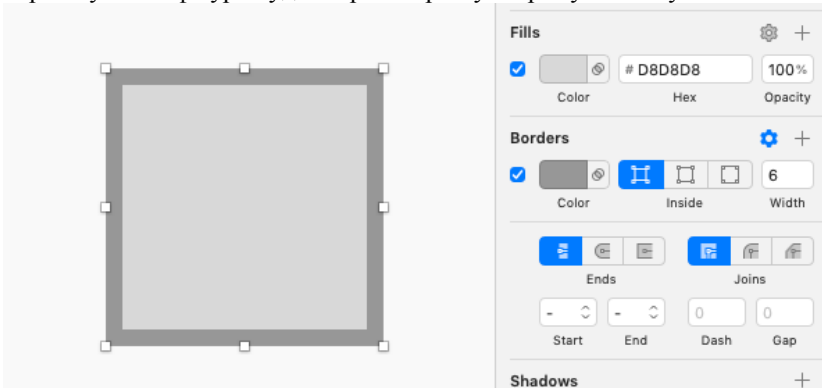
R Rectangle

O Ellipse, Oval Polygon Star

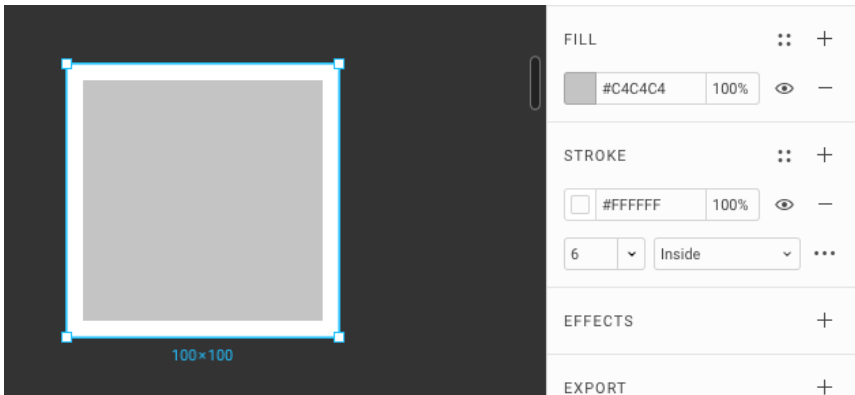
Якщо обраний інструмент будь-якого шейпа, при кліці в робочу область буде створена фігура розміром 100x100.

Прямокутник

Малюємо квадрат: **R**, **Shift** + **тянем**. Якщо затиснутий **Shift**, під час перетягування фігура буде зберігати рівну ширину і висоту.



Скетч:



Фігма:

Чим відрізняються фрейми і прямокутники

Ціль фреймів в Фігмі така ж як у артбордів в Скетчі - бути межами макета. Тому вони заточені під роботу з сіткою, мають встановлені пропорції популярних пристроїв і фонову заливку.

Ціль прямокутних шейпів — бути видимою формою всередині фрейму. Вони більше підходять для малювання кнопок і блоків. Тому вони крім заливки мають обведення і заокруглення кутів.

Ще прямокутники служать для відображення растрових зображень, які накладаються на них як заливка.

Лінії, які їх формують, можна розривати і спотворювати, йдучи від початкової форми прямокутника.

Зсув і масштабування

Shift и зсув мишкою

Фігуру можна переміщати на інше місце. Якщо під час руху затиснути **Shift**, ми зафіксуємо горизонтальну або вертикальну вісь, по якій вона зможе переміщатися.

Shift и масштабування

Фігуру можна тягнути за квадратні ручки. Щоб зберегти її пропорції під час масштабування, потрібно затиснути **Shift**.

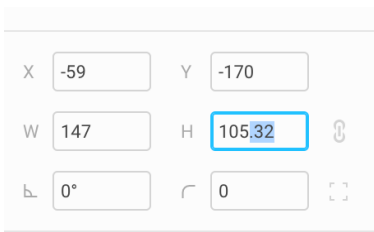
Товщина ліній при такому масштабування мінатися не буде.

Масштабування зі збереженням товщини ліній Якщо потрібно збільшити або зменшити об'єкт так, щоб лінії обведення теж змінилися, треба використовувати інструмент **Scale, K**.

На відміну від Скетча, в Фігмі немає вікна, яке дозволяє задавати в



відсотках, наскільки точно потрібно збільшити об'єкт.



Режим збільшення **Scale** небезпечний тим, що лінії можуть отримувати значення повз пікселів. Якщо потрібно оптимізувати іконки в 1x, лінії повз пікселів в них будуть мильні.

Заливка и обведення шейпів: перший погляд

Як і в будь-якому векторному редакторі, в Фігмі є можливість заливати векторні фігури кольором, а також накладати на їх контур обведення.

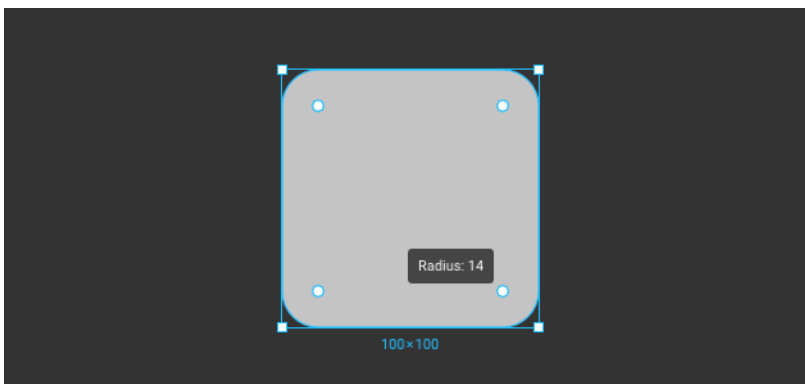
Для роботи з заливкою використовуємо блок **Fill**.

За обведення відповідає наступний за ним блок **Stroke**.

У Скетчі та Фігмі у шара-шейпа можна налаштувати необмежену кількість обводок і заливок, натискаючи +.

В **Fill** і **Stroke** колір накладається на фігуру по одним і тим же принципам, які ми розберемо в окремому розділі **Заливка и градієнт**.

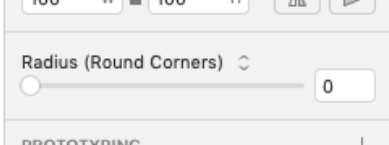
Про детальні настройки обведення поговоримо в розділі **Обводка**.



Заокруглення

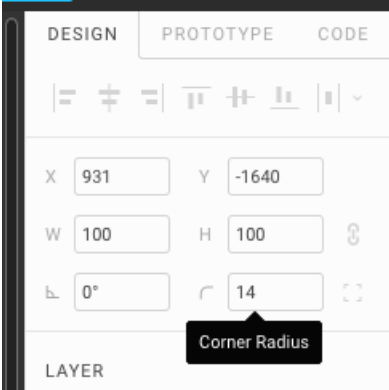
Заокруглення кутів в Фігмі і Скетчі можна міняти.

У Фігмі, як і в Ілюстраторі, якщо навести на виділений квадрат, всередині нього буде видно круглі ручки для управління заокругленням. Тягнемо їх до центру, всі 4 кута скругляються.

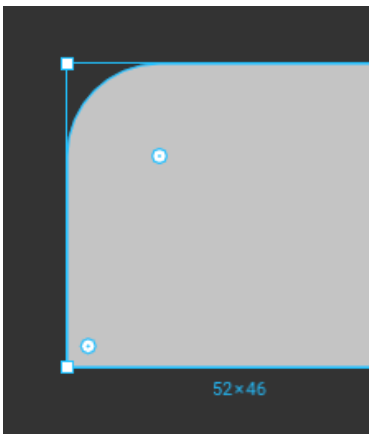


У Скетчі для цього доводиться використовувати фейдер **Radius**.

Ця ж настройка буде в блоці координат справа.



Щоб прибрати заокруглення, виставляємо в **Corner Radius** 0 або відсуваєм ручки від центру.



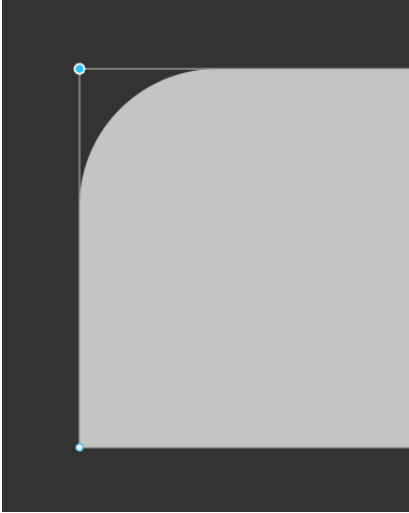
Заокруглення одного кута

Також можна округляти кожен кут окремо, якщо затиснути **Opt** і тягнути за

потрібний кут.

Це рекомендований спосіб заокруглювати кути. В цьому випадку, якщо потрібно буде копіювати CSS-код, рядок **border-radius** потрапить в нього:

```
width: 52px; height: 46px;  
background: #C4C4C4;  
border-radius: 14px 0px 0px 0px;
```



Другий спосіб заокруглення Через режим редагування: виділити точку і індивідуально для неї заокруглити кут. Спосіб поганий тим, що при копіюванні CSS Фігма НЕ пропише рядок з **border-radius**. Верстальники будуть не раді. Однак, це непоганий варіант, якщо наш прямокутник стане не кнопкою, а частиною ілюстрації. CSS- властивості з нього ніхто копіювати НЕ буде.

Прямокутник і сітка

Прямокутники - потужний пластилін для створення блоків, кнопок, отбівок і будь-яких інших об'єктів, які повинні лягати в сітку. Якщо потрібна точність, мишею або стилусом з прямокутниками працювати набагато важче, тому корисно вивчити клавіші нижче.

Скетч і Фігма дозволяють контролювати прямокутники і точки в них з клавіатури з точністю пікселя.

Рухаємо фігуру стрілками

Виділяємо фігуру, натискаємо **стрілки** — рухаємо з кроком 1px.

Shift + стрілки — рухаємо на відстань, налаштовану в **Nudge amount**. Як правило, такий крок дорівнює тій сітці, в якій ти працюєш. У нашому випадку це 4px.

Shift значно прискорює швидкість зсуву, дозволяючи залишатися в прийнятій сітці.

Плющимо или тягнемо фігуру кóммандом

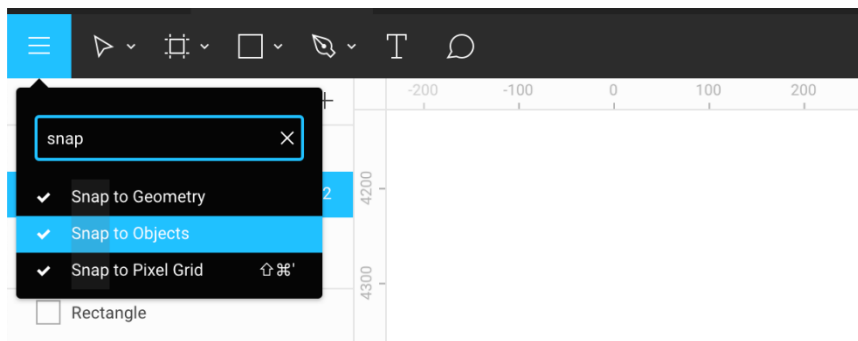
Cmd + стрелки змінюють ширину або висоту фігури на 1px.

Shift + **Cmd** + стрелки — тягнемо ширину і висоту на **Nudge amount**.

Аналогічно зрушенню, що не з'їжджає з сітки.

Плющимо декілька фігур

Ще одна перевага перед мишею: можна виділити відразу декілька фігур, потім з клавіатури міняти їх розмір стрілками, а не кожну по черзі мишею.

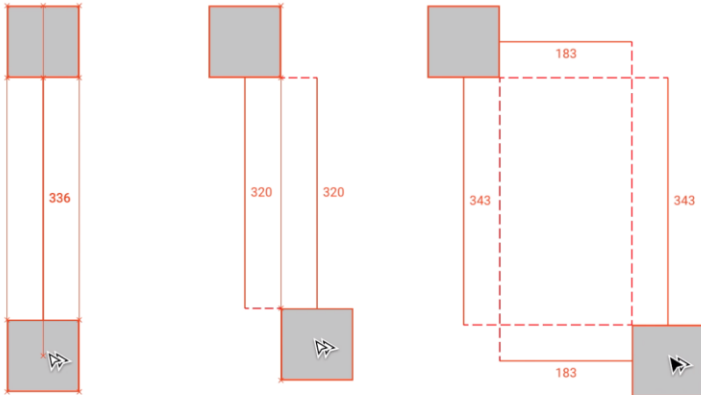


Зрушуємо і липнемо до об'єктів

Спочатку активний режим **Snap to Objects** дозволяє рухати фігури більш точно.

Під час зсуву фігура магнітиться до точок інших фігур. Навіть працюючи мишею, дизайнер може розставляти об'єкти по полотну з точністю до пікселя.

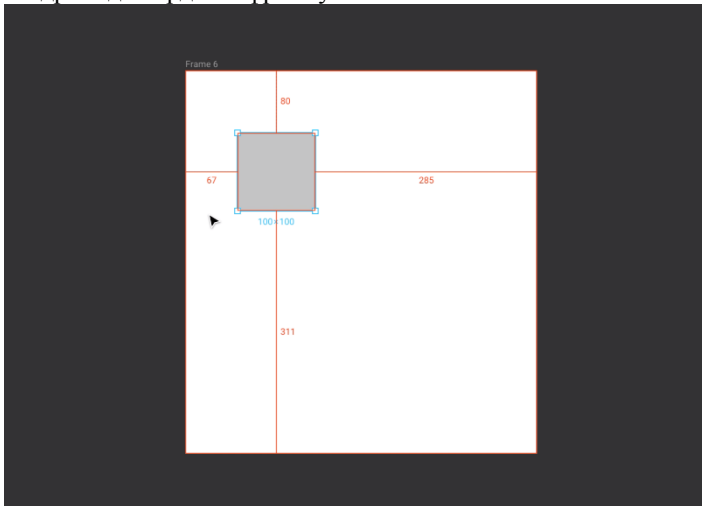
Як в Скетчі та Ілюстраторі зажmemo **Opt** і потягнемо квадрат вниз, щоб дублювати його. Під нижнім квадратом буде видно червоні лінії і відстані в пікселях, які підказують, як його можна вирівнювати по верхньому:

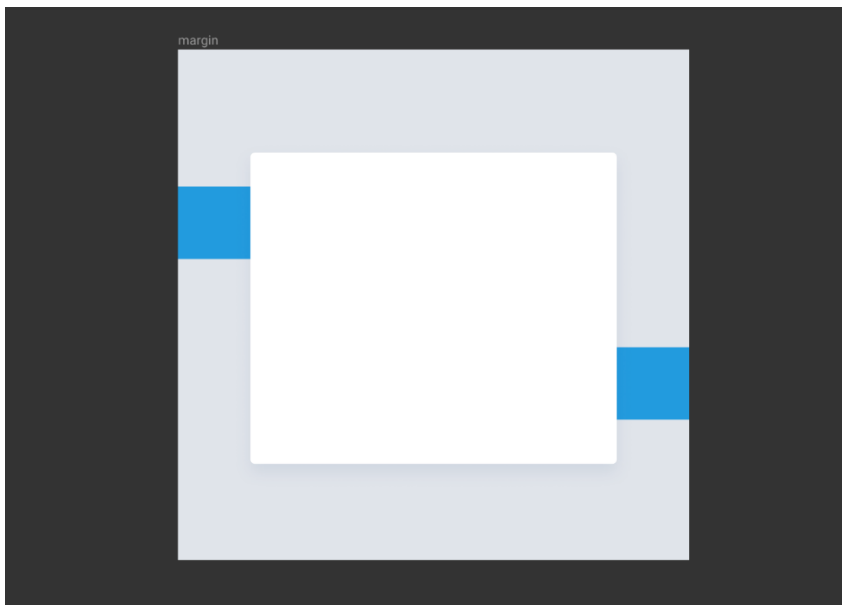


Opt для визначення відстаней

Вимірювання відстаней через **Opt** стало найпотужнішою технікою, яка вперше з'явилася в Скетчі. Потім ідею підхопили Фігма і Фреймер.

1. Накреслимо фрейм, **F**.
2. Усередині нього клікнемо квадрат, **R**.
3. Змістимо показник з квадрата і зажmemo **Opt**. Поки показчик знаходиться усередині фрейму, буде видно відстані в пікселях від кордонів квадрата до кордонів фрейму:





Ще один спосіб міряти відстані

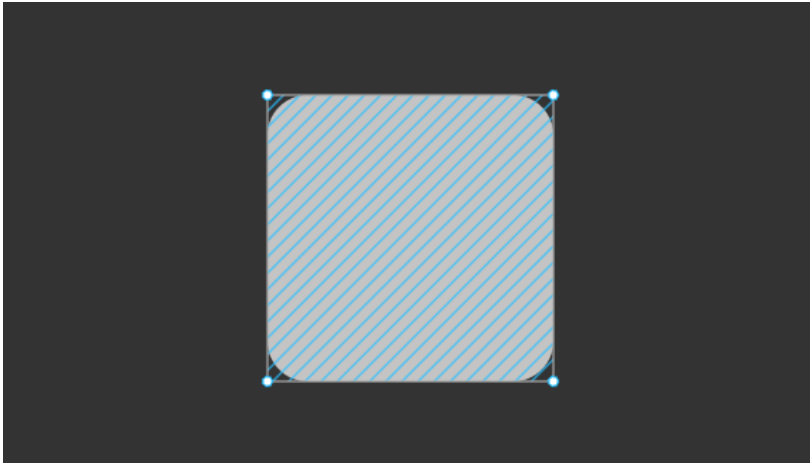
Прямокутники самі по собі - ідеальний вимірювач. Їх зручно плющить до потрібної форми, а потім міряти ними відстані. Такі прямокутники я називаю рівнялками.

Наприклад, найшвидший спосіб зробити відступ в 64 пікселя - зробити допоміжні прямокутники і по ним підлаштувати ширину основної фігури. Потім, видалити їх або винести за межі фрейму.

Ще більш витончений спосіб рівняти об'єкти - обгорнути рівнялки в компоненти і вставляти потрібні величини з панелі компонентів.

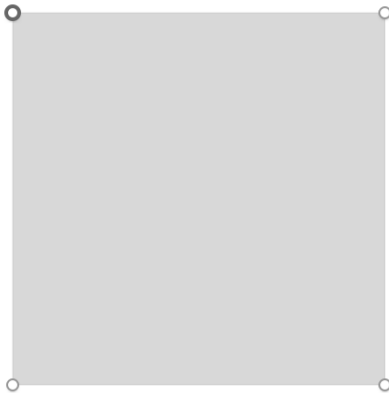
Наприклад, я знаю, що від основного блоку до нижнього краю макета у мене 120 пікселів. Я ввожу 120 в пошуку по компонентах і вставляю квадрат такого розміру, яким міряю відстань до краю макету.

Режим редагування



Як і в Скетчі, якщо виділити фігуру і натиснути **Enter**, ми перейдемо в режим редагування. Використовуємо **Esc** або **Enter** щоб вийти з нього. При ховері на фігуру, Фігма наочно показує штрихуванням, що ми знаходимося в цьому режимі.

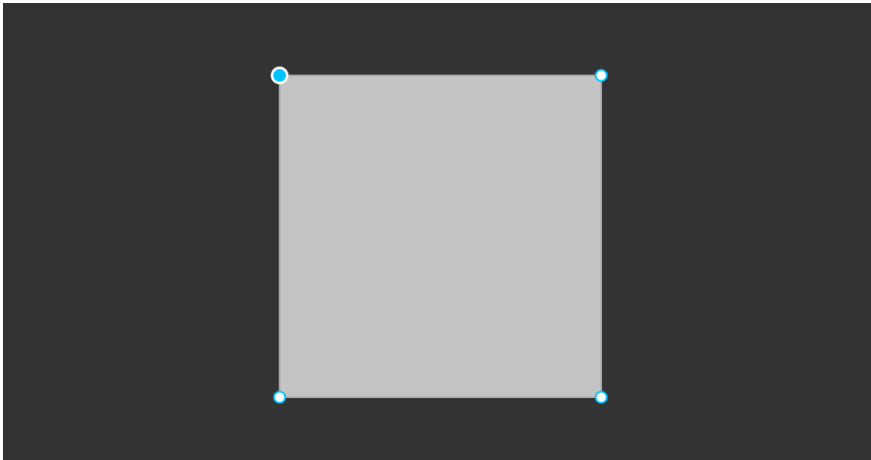
Коли переходимо в аналогічний режим в Скетчі, одна з точок, які формують фігуру, виділена. **Tab** дозволяє перемикатися між ними і рухати їх з клавіатури.



Виділена точка в Скетчі має більш жирне обведення:

У Фігмі точки треба виділяти вручну, на жаль, через **Tab** як в Скетчі вони не перемикаються.

Якщо точка виділена, її можна рухати через **стрілки** і **Shift + стрілки**.



Виділена точка залита кольором:

Прямокутник можна перетворити в іншу фігуру, наприклад, в трапецію. Для цього пересуваємо його точки в режимі редагування. В цьому випадку у нього безповоротно зникнуть ручки заокруглення. Це відбувається, тому що Фігма перестане сприймати його як об'єкт типу **Rectangle**, а вважатиме його векторною фігурою довільної форми.

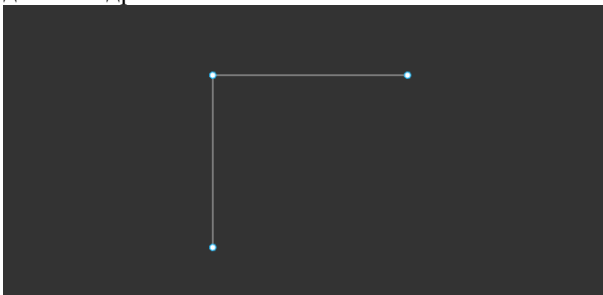
У режимі редагування можна додати прямокутнику нових точок і виліпити з нього нову фігуру.

Також прямокутник можна розірвати в шлях. Для цього потрібно видалити з нього будь-яку точку.

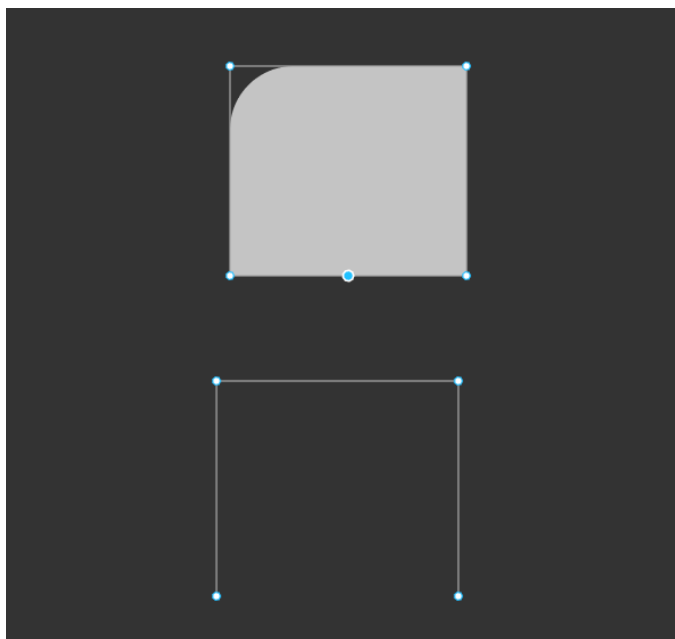
У Скетчі для розриву шляху використовується інструмент **Scissors**.

Скетч: видалення точки не розриває фігуру, а зменшує її площу. Квадрат, якому видалили точку в Скетчі, стане трикутником.

Фігма: Видалення точки розриває фігуру. Разом з нею втечуть прилеглі до неї відрізки.



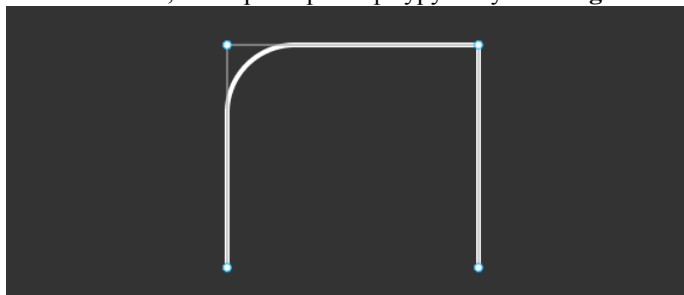
Виділяємо точку, видаляються точка і дві сторони фігури:



У Фігмі не потрібен інструмент **Scissors**. Якщо потрібно видалити тільки одну сторону, створюємо точку по її центру. Фігма при ховері підказує, де центр відрізка:

Видаляємо точку, **Delete**. Залишився векторний скелет з точок і відрізків. При цьому у верхньому лівому куті точки залишилося заокруглення, якого не видно через те, що у векторної фігури немає обведення.

Таким чином, ми перетворили фігуру типу **Rectangle**

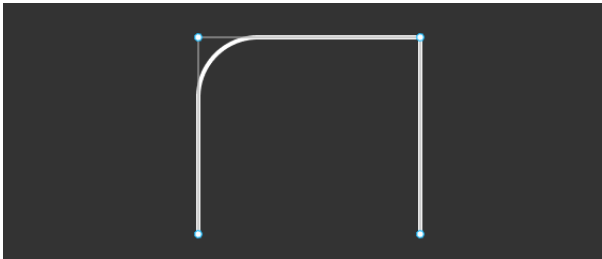
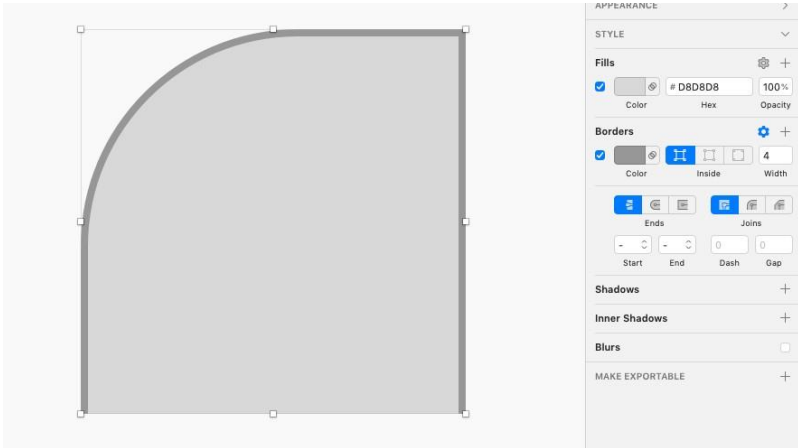


в розімкнутий шлях. Задаємо **stroke**, щоб побачити заокруглення:

Заокруглений кут залишився. Це означає, що кожна точка векторного шляху зберігає в собі інформацію про ступінь заокруглення, навіть якщо шейп розірваний. У Скетчі після розриву шейпа заокруглення перетворюється в

окрему криву, позбавляючи нас можливості редагувати її через значення **Corner Radius**. Робота із заокругленнями в Фігмі реалізована краще.

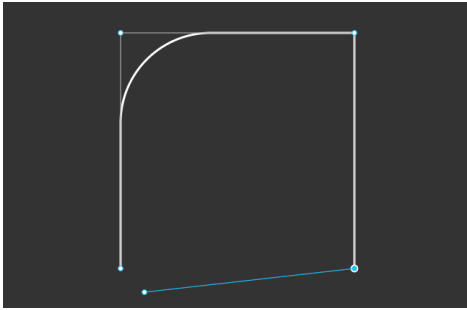
У Скетчі навіть у розімкнутих шляхів може бути заливка. Застосовуємо **Fill** в скетчі:



Аналогічна ситуація в Фігмі: заливка є в настройках, але не видно, оскільки фігура розімкнута.

Тепер Фігма сприймає такий об'єкт як векторний шлях, або векторну мережу. У шляху немає поняття внутрішнього і зовнішнього, тому обведення може бути тільки в режимі **Center**, поза залежністю від того, який режим виставлений в настройках.

Аналогічно Скетчу: Якщо взяти **Pen Tool, P**, навести на одну з кінцевих точок, в курсорі з'явиться коло. Це означає, що при натисканні ми будемо продовжувати малювати лінію з цієї точки.

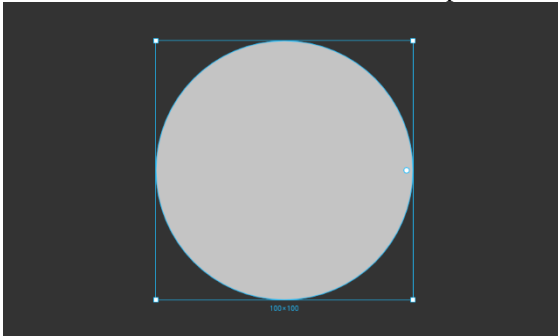


Клікаємо в точку, натискаємо в іншу навпроти і таким чином замикаємо фігуру назад:

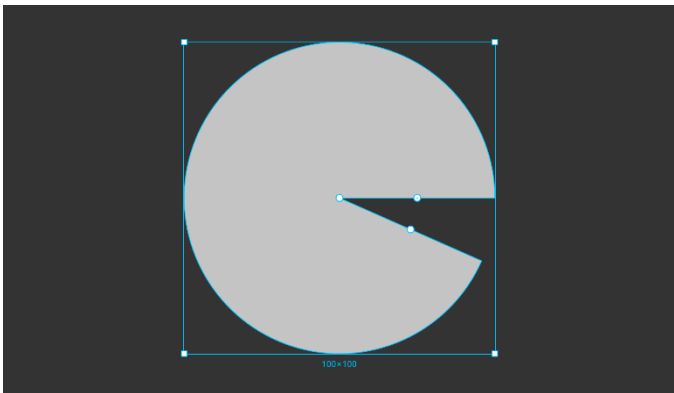
Окружність, овал и пайчарт

Клавiша: **O**

Ellipse, Oval



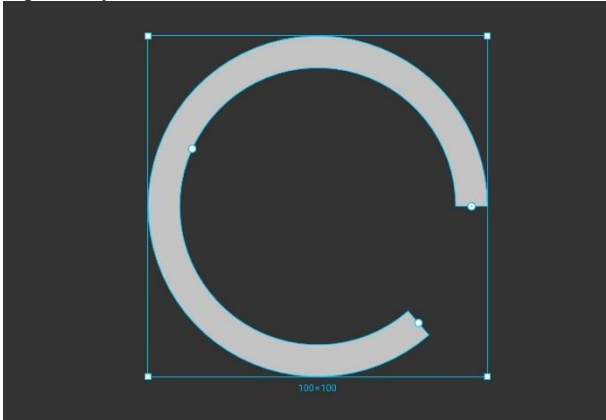
Малюємо коло із затиснутим **Shift**, або клікнемо в будь-яке місце:



На відміну від Скетча, в Фігмі окружності і овали можна легко розмикати, не використовуючи Pen Tool або ножиці. Для цього наводимо на фігуру і тягнемо круглу ручку в правій частині вгору або вниз. Так ми міняємо властивість

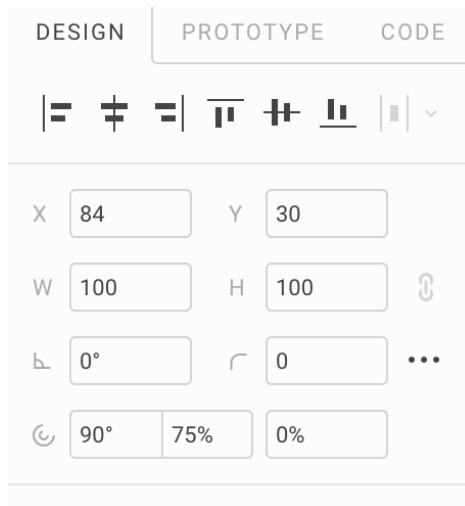
Sweep.

Коли ми задали значення, отриманий пайчарт має два радіуса з ручками, які можна підлаштовувати в потрібні значення, а також центральна точка, яка дозволяє вирізати серцевину.



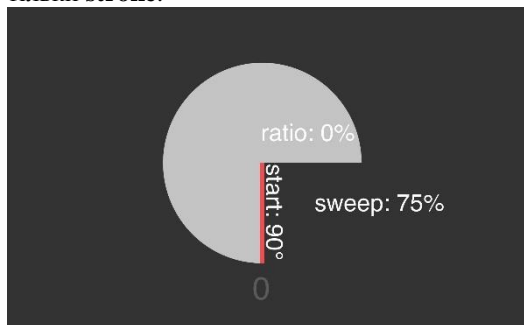
Перетягуємо центр вниз, вирізаючи з кола середину:

За роботу з пайчартами відповідають три властивості, які є в панелі виділеної окружності:



- **Start**— кут початку в градусах, спочатку 0. Вихідний градус - на 3 години.
- **Sweep**— відсоток заповнювання від 0 до 100%.
- **Ratio** [рэйшио] - розмір порожнечі по центру у відсотках від радіуса. При 0

пайчарт буде заповнений по центру, при 100 заливки видно не буде, залишиться тільки **stroke**.

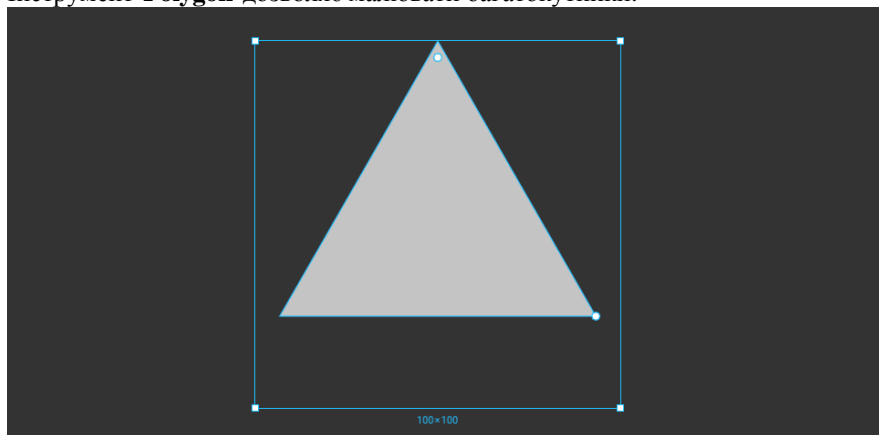


Ще один приклад: **start** з трьох годин повернутий на 90 °, **sweep** зайняв 3/4 окружності. Він вираховується з положення **start**. **Ratio** не використовується.

У Скетчі пайчарти можна малювати або через плагіни, або через обведення з пунктиром. У Фігмі значно легше задавати потрібні значення пайчартів в візуальному режимі.

Трикутник

Інструмент **Polygon** дозволяє малювати багатокутники.



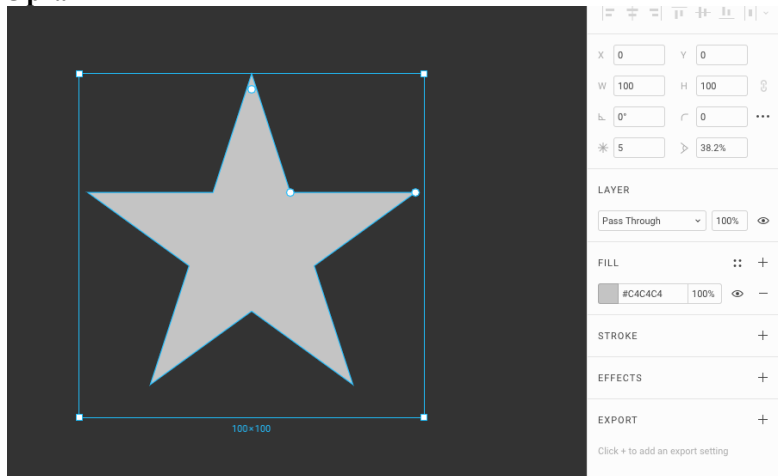
Не має рівної клавіші, можна викликати **Cmd + /**, **pol**. Щоб трикутник був рівностороннім, при його розтягуванні затискаємо **Shift**.

Одержаний рівносторонній трикутник має дві круглі ручки: всередині зверху і справа на розі.

Верхня ручка дозволяє **заокруглювати кути** трикутника і працює так само як на прямокутниках, проте якщо потрібно округляти тільки один кут, треба заходити в режим редагування.

Кутова ручка дозволяє задавати потрібну **кількість кутів** в інтервалі від 3 до 60. При 60 кутах багатокутник мало відрізняється від кола. Також задавати кути можна в **Count**.

Зірка



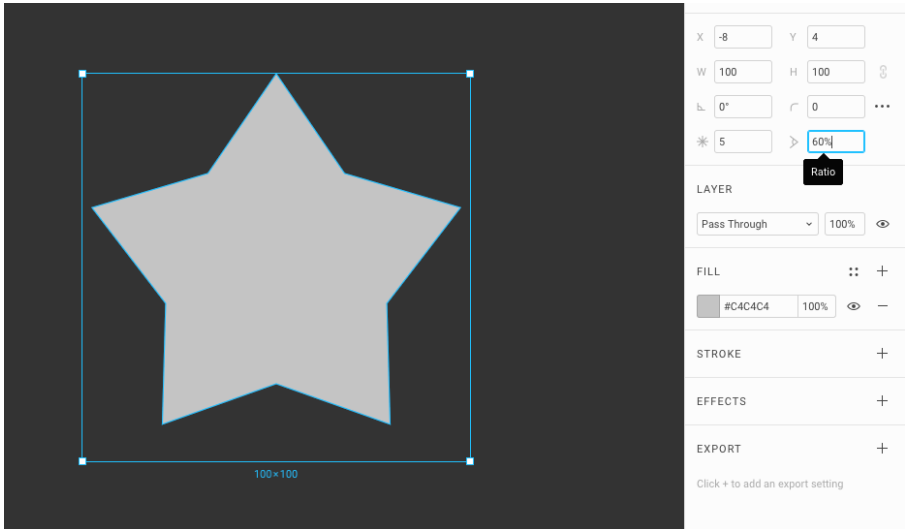
Інструмент **Star** дозволяє малювати променеві багатокутники. Не має рівної клавіші, можна викликати **Cmd + /, sta**.

Крім стандартних, у зірки є 3 параметри:

- **Corner Radius** — ступінь заокруглення кутів
- **Count** — кількість променів
- **Ratio** — ступінь гостроти променів

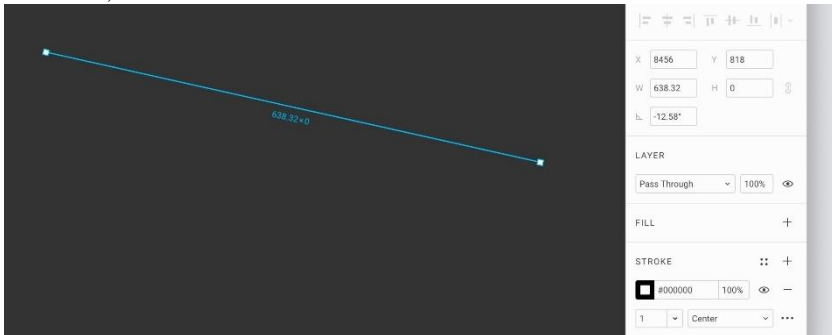
Ручка в верхньому промені дозволяє заокруглювати кути всіх променів.

Ручка в кутку між верхнім і правим променями дозволяє задавати **ratio**:



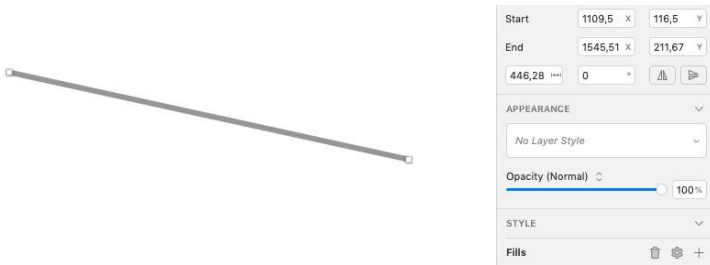
чи буде зірка гострою або пузатою.

Лінія Line Tool, L



Інструмент дозволяє за допомогою перетягування створити векторний відрізок. Щоб намалювати лінію, вибираємо місце початку, затискаємо ліву, тягнемо. Там де треба, відпускаємо.

У Скетчі лінія має **Start** - координати початку і **End** - координати кінця, а також довжину і кут нахилу, який не залежить від реального кута нахилу намальованої лінії. Якщо провести лінію на певний кут, в Скетчі в поле кута залишиться 0, а в Фігмі воно буде відповідати цьому кутку.



Дві пари координат ускладнюють роботу з такою лінією, якщо потрібна точність.

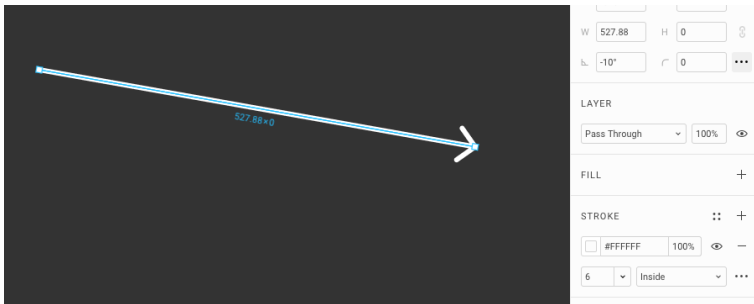
Розробники Фігми не наступили на граблі Скетча і зробили тільки одну пару координат для початку лінії. В поле **Width** лінії можна задати довжину, а в **Angle** кут нахилу.

Деякі використовують **Line Tool** для того, щоб робити відбиття між рядками в таблицях. Однак, їх простіше контролювати через **Rectangle**.

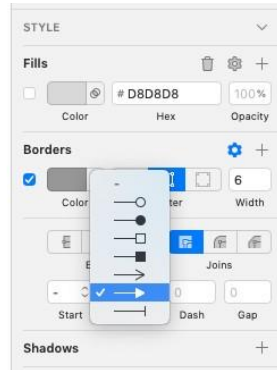
Стрілка

Arrow Tool, **Shift + L**

Інструмент працює аналогічно **Line Tool**, єдина різниця - на кінцевій точці є стрілка.

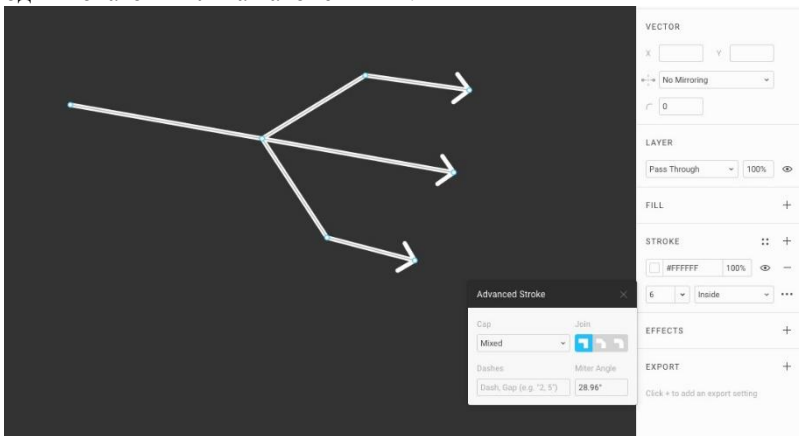


Як і в Скетчі, стрілка на кінці - це певна властивість. Однак, в Скетчі стрілка - це властивість всієї лінії, а в Фігмі - це властивість окремої точки.



Як налаштовується стрілка в Скетчі: вибираємо тип стрілки в полі **End**.

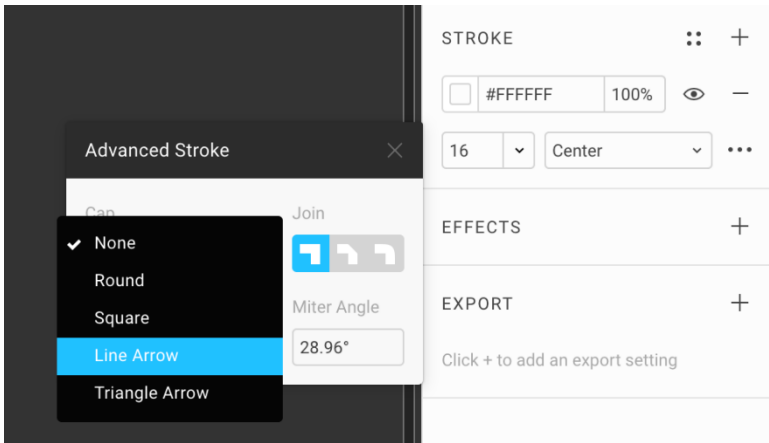
У Скетчі лінія може мати початкову і кінцеву стрілку, а в Фігмі стрілку може мати будь-яка кінцева точка векторної мережі. Отже, може бути стрілка, у якій один початок і кілька наконечників:



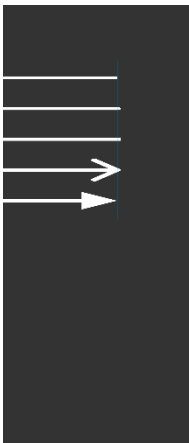
Наконечник точки можна відключати, перетворюючи стрілку в звичайну лінію. Або навпаки, можна намалювати криву пером **Vector Tool, V** і налаштувати їй стрілку на будь-якому з кінців.

У Скетчі можна копіювати стиль стрілок, **Opt + Cmd + C** і передавати їх іншим лініям, **Opt + Cmd + V**.

У Фігмі цього не можна робити, оскільки таким чином ми передамо тільки інформацію про колір і ширину лінії. Наконечники доведеться налаштовувати індивідуально.

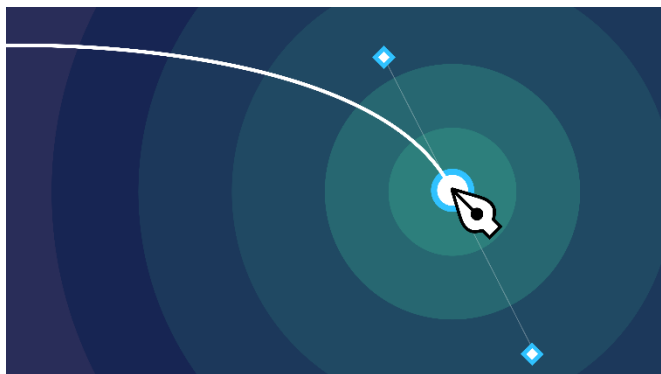


Щоб налаштувати стрілку, потрібно в режимі редагування **Enter** вибрати потрібну точку і розгорнути меню **Advanced Stroke**. Це меню виглядає як три крапки в блоці **Stroke**.



У ньому є меню **Cap**, де можна вибрати тип наконечника:

- **None**
- **Round**—закруглений
- **Square**—квадратний, схожий на **None**
- **ine Arrow** — контурна стрілка
- **Triangle Arrow** — трикутна стрілка



9. Перо

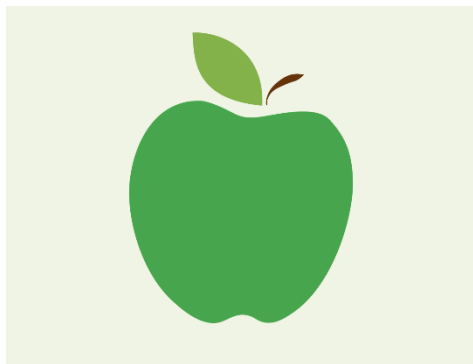
Проект в Фігмі →

Pen Tool, P — найважливіший інструмент для малювання кривих ліній і складних непередбачуваних форм за допомогою кривих Безьє.

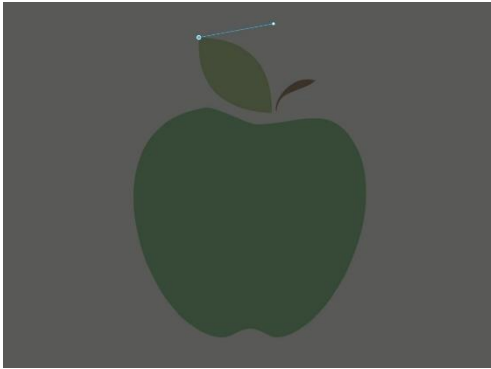
У Скетчі називається **Vector Tool, V**.

Робота пера в Фігмі більше схожа на роботу в Ілюстраторі, ніж в скетчі, і це хороша новина.

Щоб навчитися використовувати перо, обрисуємо яблуко.



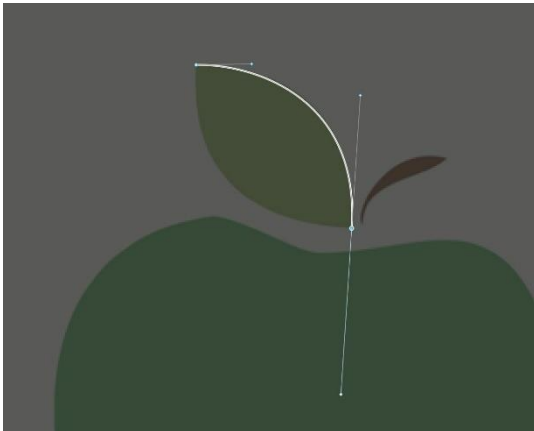
Джерело вектора: freevector.com/apple-vector-icon



Щоб зображення не завадило бачити лінію контуру, зробимо його на 20%, **2.** Також виберемо білий колір ліній пера.

1. Вибераємо Pen Tool, **P**.
2. Почнемо з листа. На верхньому куті листа лівим кліком створюємо точку, з якої буде тягнутися перша крива. Після того, як поставили крапку, можна підкоригувати її положення **стрілками**. Від точки до покажчика тягнеться шлейф, який підказує, як буде виглядати лінія, якщо її зафіксувати в цьому місці.

Наше завдання - провести криву від першої точки до того місця, де її доведеться перервати.



3. Клікаємо на нижній кут листа і тягнемо вниз. З другої точки з'являються тонкі білі важелі - вуса.

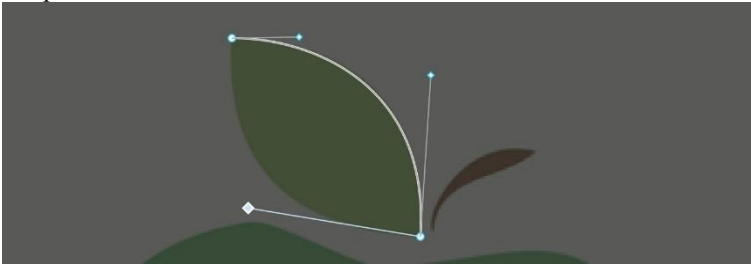
Вуса символізують швидкість, з якою лінія виходить з точки до центру лінії. Чим вуса

довші, тим сильніше вони впливають на викривлення лінії. Перший вус коротший і виходить горизонтально. Не відриваючи покажчика, тягнемо за другий вус вниз і трохи лівіше.

Таким чином, верхня його половина зміщується вправо, прогинаючи криву. Відпускаємо покажчик. Лінія фіксується. Від точки до покажчика знову тягнеться шлейф.

Цього разу він може викривлятися, тому що у початкової точки нової кривої вже є швидкість. За один клік ми встановили відразу два важелі: для закінчення першої кривої і для початку другої.

4. Тепер потрібно загнути вус так щоб він направив наступну криву туди ж, куди йде контур листа. Для цього ми беремо нижній вус і тягнемо його лівіше. Підбираємо потрібний кут і робимо вус тієї ж довжини, що і його вертикальний побратим.



Ламасмо вуса за допомогою Opt

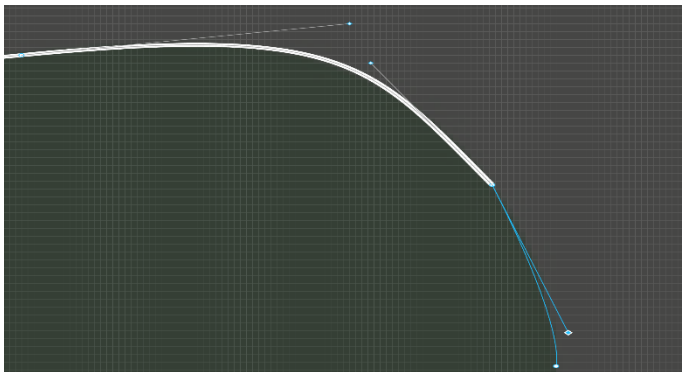
Ми дійшли до важливого переваги пера в Фігмі і Ілюстраторі перед **Vector Tool** в скетчі. На кроці 3 ми зафіксували вус, важелі якого стоять паралельно, а на кроці 4 скорегували нижній важіль. Це можна було б зробити зручніше: клікнути точку, розтягнути з неї вуса, підлаштувати верхній вус, тримаючись за нижній. Потім, не відриваючи покажчика, затиснути **Opt**. Поки клавіша затиснута, верхній перестане рухатися, а нижній можна тягнути далі. Таким чином, за допомогою клавіші ми за один клік можемо поставити поведінку двох кривих: як буде вести себе кінець першої і початок другої.

У Скетчі **Opt** так не працює, а значить, на кожному кроці потрібно коригувати вже зафіксований другий вус вручну.

5. Замикаємо контур листа, клікаючи в першу точку і витягаючи з неї вгору нову пару вусів. Підстроюємо вигин другої кривої за верхній вус.

6. Якщо потрібно підкоригувати будь-який з вусів, це можна зробити, якщо виділити будь-яку точку в фігурі.

7. Виходимо з режиму редагування, зафіксувавши фігуру, **Enter**.



Часта помилка: злами ліній

При змалюванні слід уникати тупих кутів там де вони не потрібні.

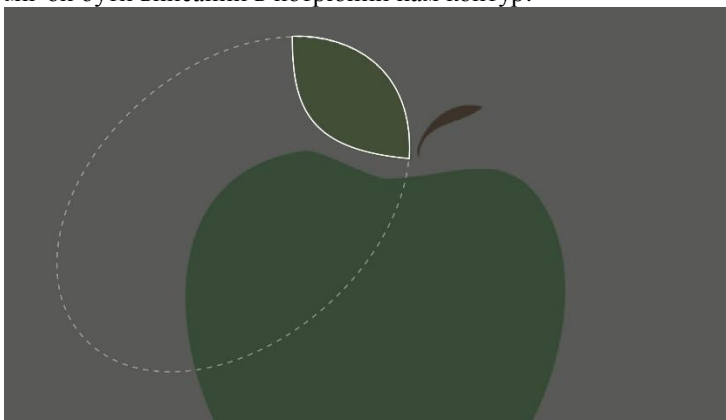
У скріншоті вище вуса утворюють тупий кут, який буде привертати увагу в фінальному контурі. Робимо лінії плавними. Для цього вуса повинні формувати розгорнутий кут.

Мінімалізм і точність

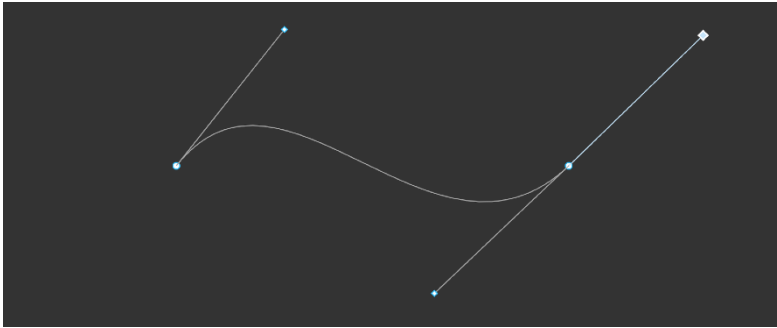
Намагаємося використовувати мінімальну кількість опорних точок: це підвищує швидкість роботи. При цьому криві повинні залишатися точними, щоб не спотворювати форму об'єкта, який ми описуємо.

Навчаючись роботі з пером, корисно уявляти, як еліпси можуть вписатися в криві.

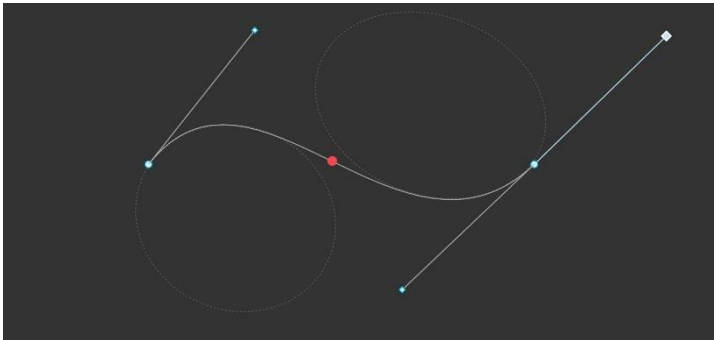
Ми вгадуємо, як може зігнутися лінія, представляючи її як частину еліпса, який міг би бути вписаний в потрібний нам контур:



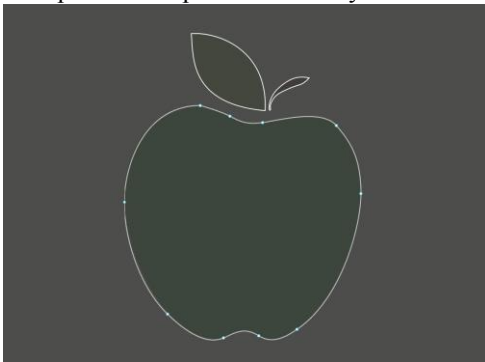
Це працює, тому що криві Безьє і еліпси будуються по одним і тим же математичним правилам.



Якщо у початкової точки є витягнутий вус, таку криву не вийде вписати в один еліпс, оскільки цей вус буде спотворювати її:



У цьому випадку ми можемо намітити на кривій умовне місце, яке розділяє дві найпростіших кривих. У кожному з них можна вписати еліпс.



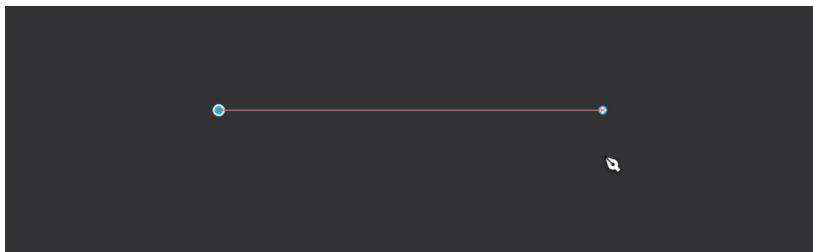
Shift і Snap to Geometry: креслимо прямокутник пером

Якщо потрібно, щоб лінія була строго вертикальною або горизонтальною, можна затиснути **Shift**. В такому режимі можна креслити геометрію. Намалюємо трикутник пером.

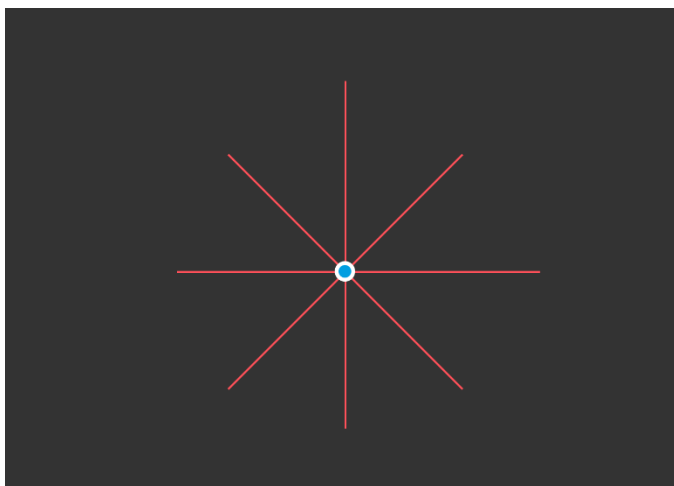
1. Виділяємо перо, **P**. Ставимо першу опорну точку. Зміщуємо

показчик вправо.

2. Затискаємо **Shift**. Шлейф від точки до показчика стане червоним і витягнеться горизонтальною лінією вправо.

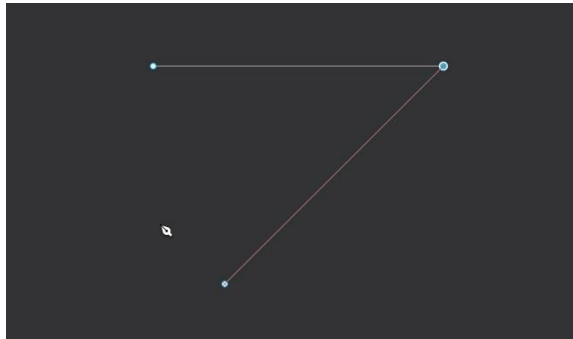


3. Не відпускаючи **Shift**, натискаємо, щоб зафіксувати лінію. Отримуємо новий шлейф від другої опорної точки.



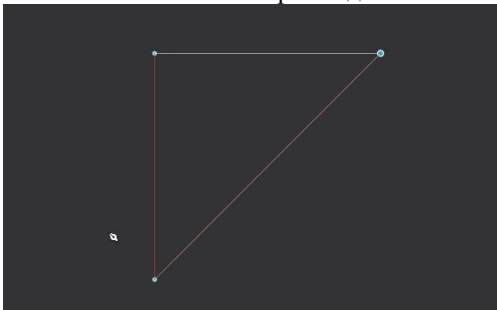
4. Шлейф з затиснутими **Shift** може бути направлений в одну з 8 сторін. Крім горизонтальних і вертикальних ліній, можна креслити лінії під кутом 45° .

Розуміння цієї концепції здорово підвищує точність креслення ліній і навіть градієнтів.

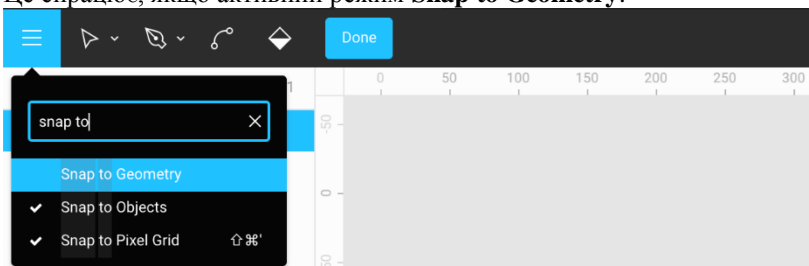


5. Тягнемо шлейф на південний захід до моменту, поки курсор не буде під першою опорною точкою.

6. Повинна з'явитися червона інформаційна лінія, яка нагадує, що якщо ми зафіксуємо лінію, яку малюємо, третя точка встане строго під першою. У Скетчі такі лінії називалися смарт-гайдами. Стаavimo третю опорну точку.

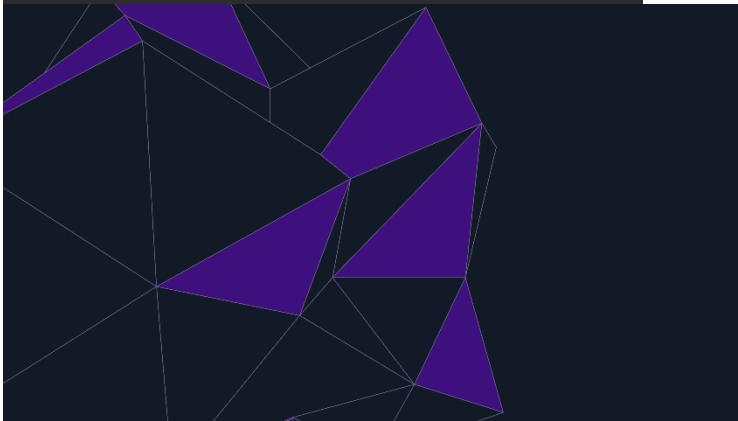
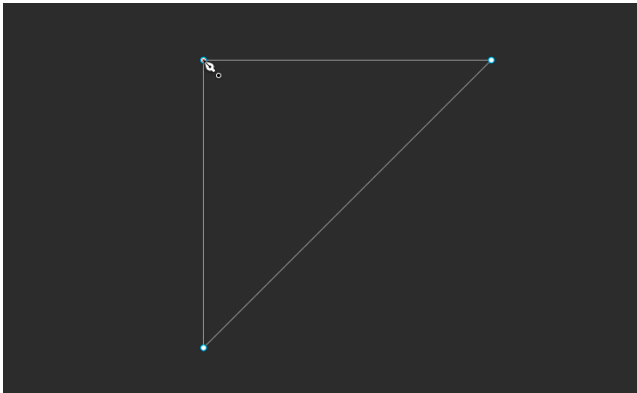


Це спрацює, якщо активний режим **Snap to Geometry**.



Він дозволяє липнути до допоміжних червоних ліній.

7. Тягнемо останню лінію від третьої точки до першої і натискаємо в першу, закриваючи трикутник. фіксуємо фігуру, **Enter**.



10. Векторні мережі

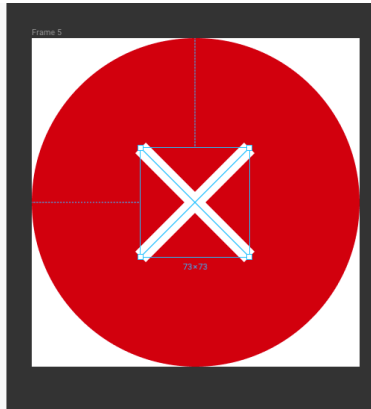
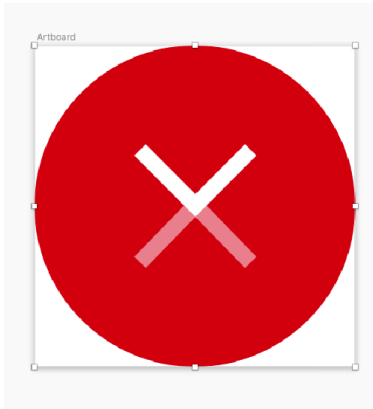
Проект в Фігмі →

Векторна мережа - це одношарова фігура, що складається з векторних точок і ліній.

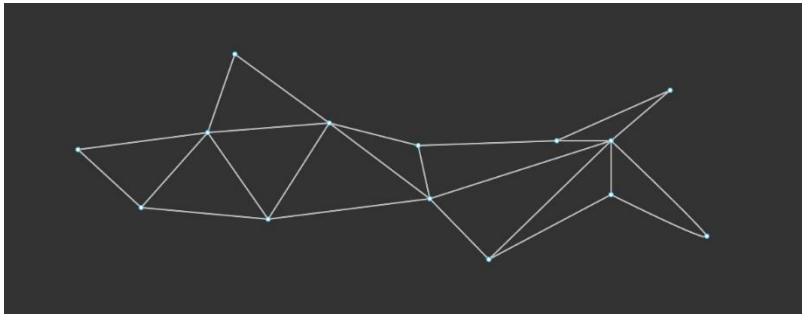
У більшості інструментів, включаючи Скетч і Ілюстратор, кожна векторна точка може мати максимум два відрізки, які до неї прилягають.

Важливо: в Фігмі в векторну точку може сходитися будь-яка кількість ліній. У цьому криються великі творчі можливості.

Наприклад, не потрібно накладати дві точки різних об'єктів один на одного, щоб імітувати один об'єкт.

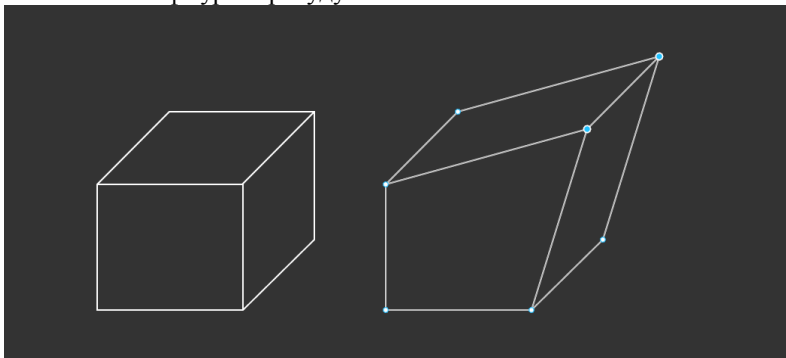


Зліва Скетч, праворуч Фігма. Для наочності приглушив нижню частину хреста в скетчі:

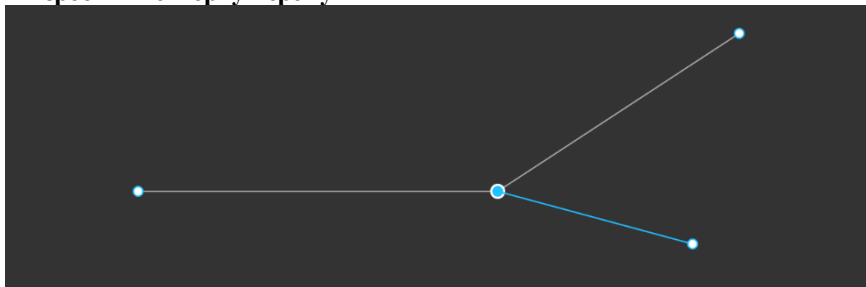


Ця можливість дозволяє швидко створювати складні фігури, які легко редагувати, не шаманячи із кутами, які з'їхалися. Приклад:

Всі точки можна рухати. Товщину всіх ліній налаштувати в один клік. Разом з точкою пересунути і всі лінії, які до неї тягнуться. Виділимо дві точки, потягнемо їх і фігура перебудується:

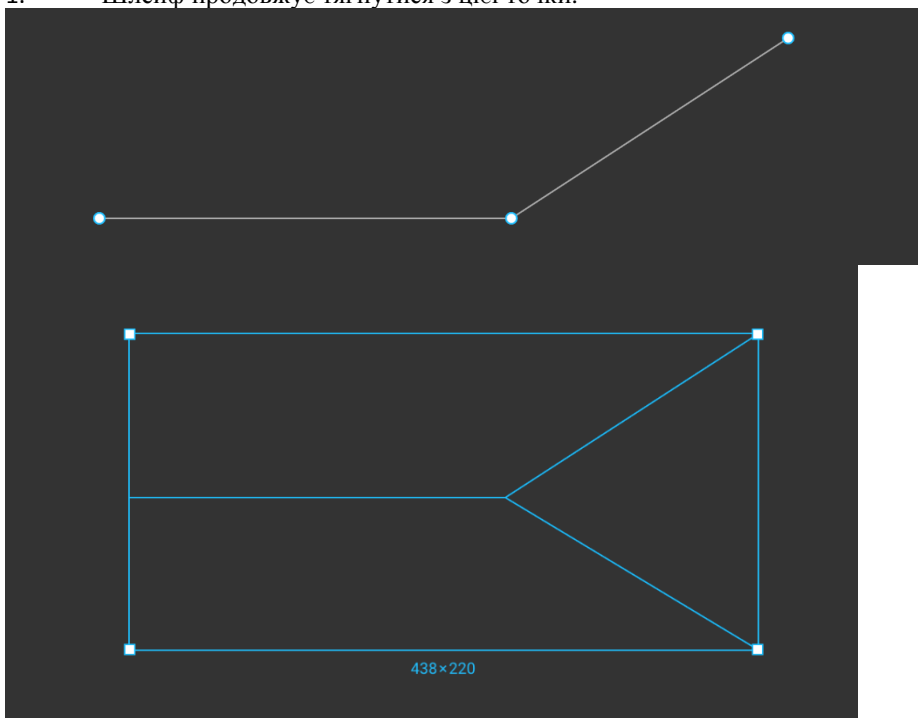


Як зробити векторну мережу



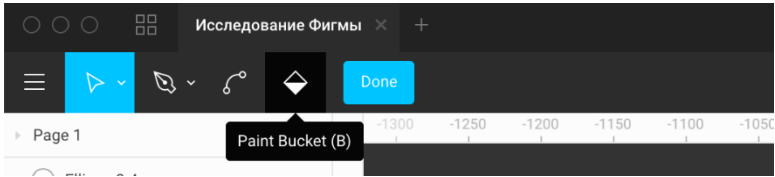
Вибираємо **перо, Р**. Малюємо два відрізки. Клікаємо в центральну точку між відрізками.

1. Шлейф продовжує тягнутися з цієї точки.



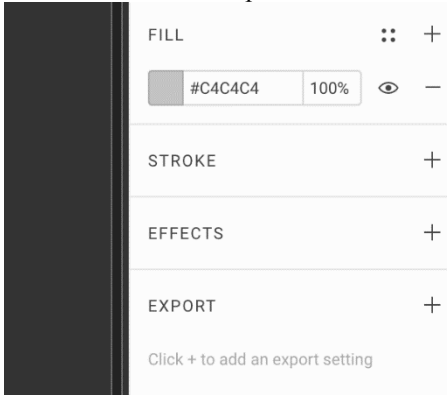
Вийшла рогатка, в якій друга опорна точка має три входящі до неї лінії. Фіксуємо, **Enter**. У нас вийшла найпростіша векторна мережа. Здавалося б, дрібниця, а саме в цій ідеї криється прорив при роботі з вектором.

Заливаємо фрагменти мережі



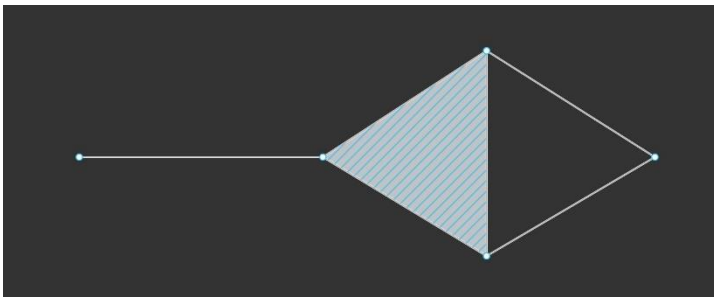
У векторних мереж можна заливати внутрішні площі. Для цього використовуємо добре знайомий інструмент **Paint Bucket, B**, доступний в режимі редагування. У народі його називають відерцем.

Він дозволяє задати ті фрагменти фігури, які будуть схильні до заливки **Fill**. Решта залишається порожніми.



Якщо в режимі редагування вибрати відерце, і клікнути на закритий фрагмент, він буде залитий. Якщо клікнути на вже залитий, заливка на ньому відбуватися не буде.

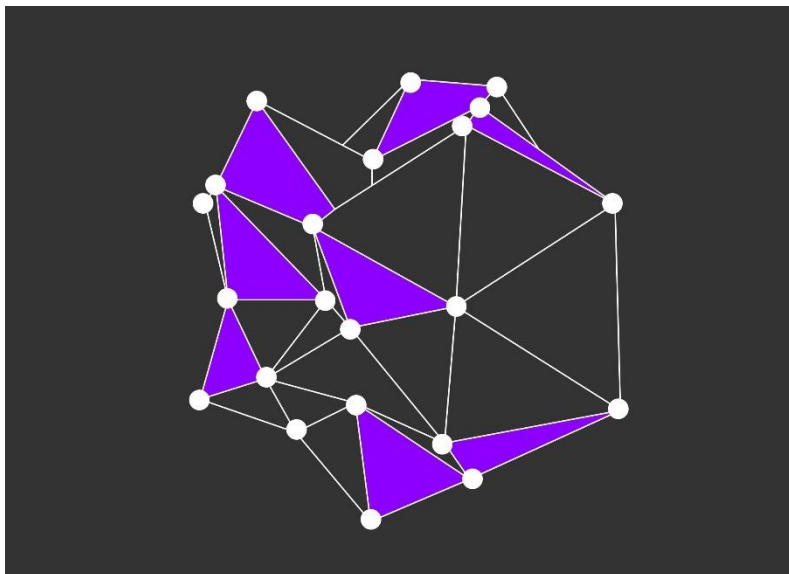
Якщо ж робити заливку через віконце **Fill** в панелі праворуч, залиті будуть всі фрагменти мережі.



Обмеження: оскільки векторна мережа - один об'єкт, колір заливки різних трикутників буде єдиним. Не можна в рамках роботи з однією фігурою один трикутник залити одним кольором, а інший іншим.

Bend Tool, Cmd

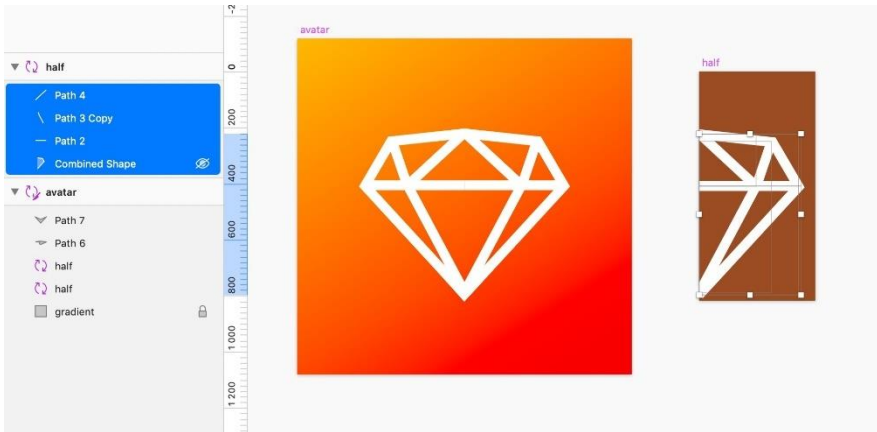
Інший цікавий режим роботи пера - **Bend Tool**. Він працює, якщо затиснути **Cmd**. і дозволяє гнути будь-які векторні лінії, перетягуючи їх за центр. Це дуже ефективно, тому що не потрібно налаштовувати вуса.



Для малювання такої картинки потрібно не більше 5 хвилин.

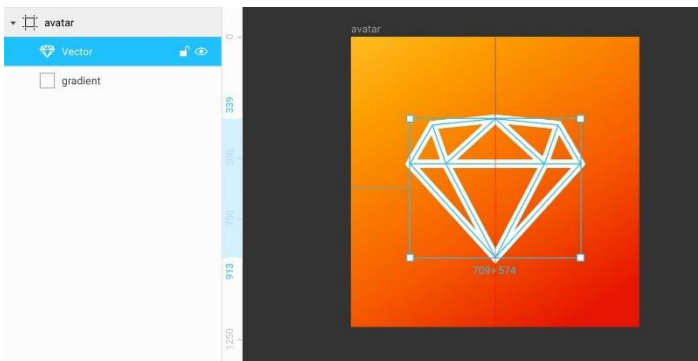


Малюємо іконки

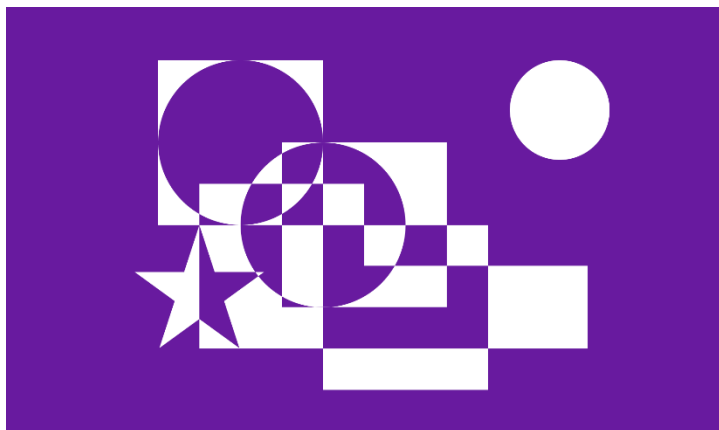


Векторні мережі дають велику гнучкість в роботі з товщиною лінії. Наприклад, щоб намалювати аватар мого каналу в Скетчі, мені довелося використовувати кілька ліній.

Якщо змінити товщину лінії, іконка розсіплеться:



Якщо намалювати ту саму фігуру в Фігмі, товщину ліній можна змінювати як завгодно.



II Булеві групи і флетен

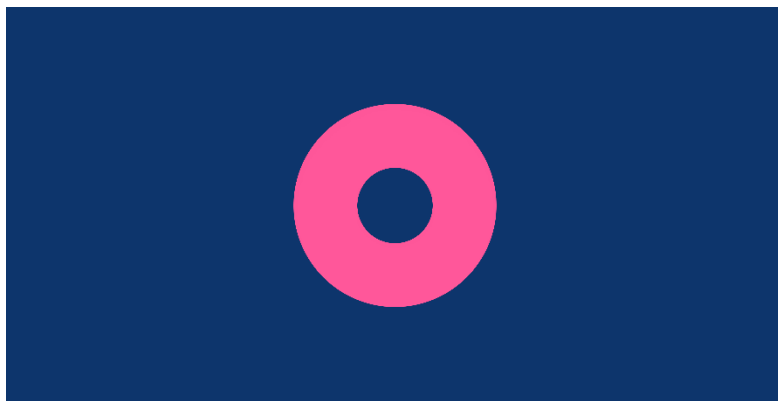
Проект в Фігмі →

Якщо потрібна більш складна форма, ніж стандартні прямокутники, кола або трикутники, її можна намалювати за допомогою пера.

Однак, якщо складну форму можна розбити на кілька базових геометричних фігур, буде швидше комбінувати їх в одну.

Булевими групами в Фігмі називаються принципи взаємодії шейпів один з одним.

У Скетчі вони називаються булевими операціями.



Приклад, в якому перо буде зовсім марно: якщо потрібно намалювати пончик. Ми можемо скласти його з двох кіл, наклавши синій на рожевий:

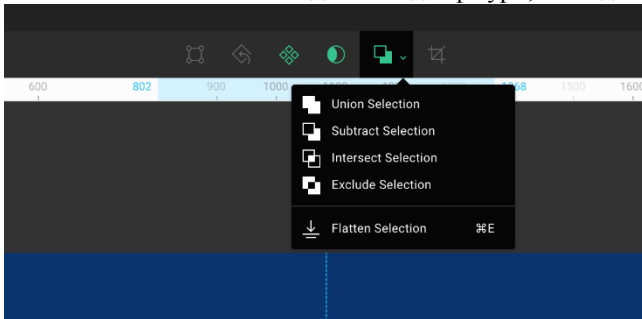
Виникає проблема: якщо зміниться фоновий колір, внутрішнє коло залишиться старого кольору. Щоб в центрі рожевого кола утворився отвір, потрібно відняти від нього форму синього. Для цього ми заключаємо обидва кола в булеву групу і ставимо її в режим віднімання.

Логічне значення може бути в двох режимах: **true** і **false**.

Слово *булевий* тут означає, що шейп може бути або видимим (true), або невидимим (false). Залежно від режиму, шейп поводитиметься певним чином.

Всього буває 4 режими булевих груп:

- **Union**— Об'єднання двох фігур в одну. Видимі обидві.
- **Subtract**— Віднімання верхньої з нижньої. Видима нижня.
- **Intersect**— Видима область перетину двох фігур.
- **Exclude**— Видимі обидві фігури, не видима область перетину.



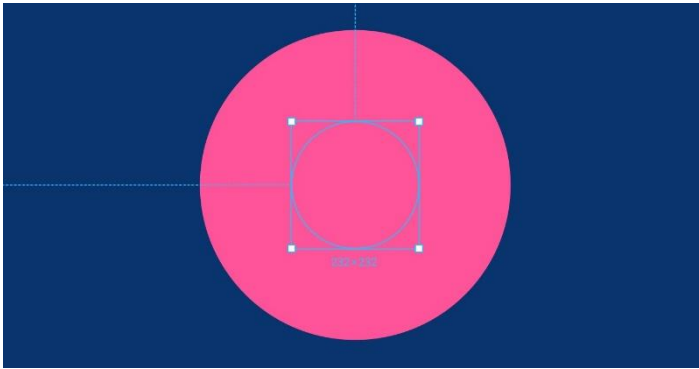
Задавши один режим, можна переключити групу на інший або розгрупувати її, **Shift + Cmd + G**.

Subtract на практиці: віднімаємо кола

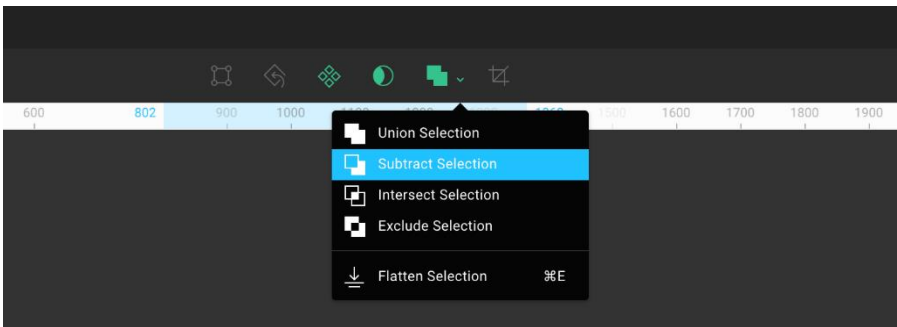
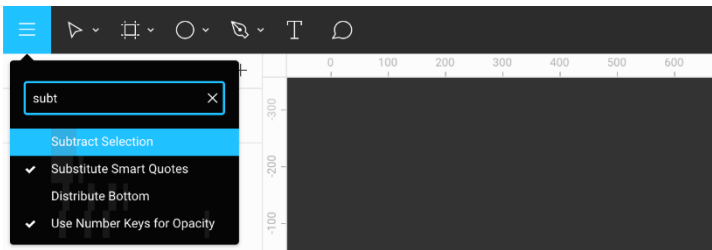
Найпростіший приклад з пончиком.

1. Малюємо коло, **O**. При бажанні заливаємо його кольором в поле **Fill**.
2. Дублюємо його клавішею **Cmd + D**. Тепер два однакових кола лежать стопкою.
3. Затискаємо нижню праву квадратну ручку, щоб зменшити верхній круг. Тягнемо вгору і вліво, до його центру.
4. Поки тягнемо до центру, великим пальцем затискаємо **Opt**, щоб опорою для масштабування кола була не ліва верхня точка, а геометричний центр.
5. До затиснутому **Opt** додаємо **Shift** мізинцем. Він фіксує пропорції кола, не дозволяючи нам малювати еліпс.

Зменшуємо коло до потрібного розміру.

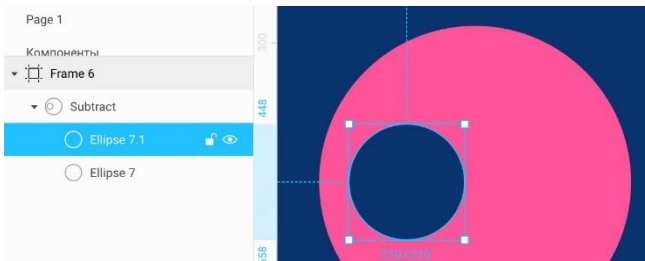
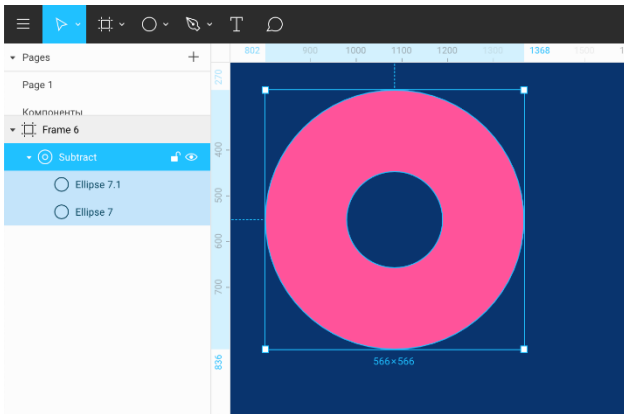


6. Виділяємо обидва шара, затиснувши **Shift** і послідовно клікнувши їх.
7. Застосовуємо команду **Subtract Selection**, яка доступна в панелі інструментів:



Також її можна викликати з клавіатури через пошук: **Cmd + /, subt**.

8. У результаті виходить булева група з назвою **Subtract**, в якій поєднані обидва кола. Верхній віднімається з нижнього.

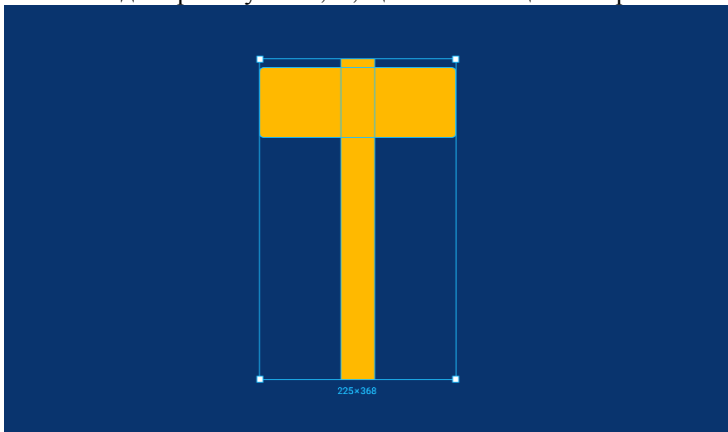


Кожен з кіл все ще можна редагувати всередині групи. Також групу можна розгрупувати, **Shift + Cmd + G**.

Union на практиці: з'єднаємо шейп

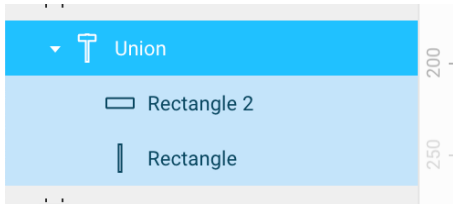
Приклад з молотом, аналогічний **Subtract**.

1. Малюємо два прямокутника, **R**, щоб вийшло щось на зразок молота:



2. Виділяємо обидва з **Shift**.

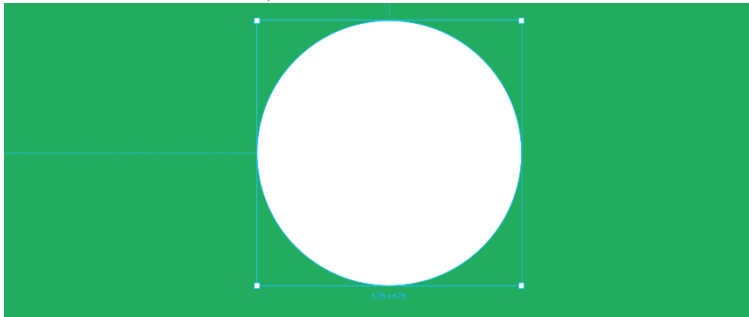
3. Застосовуємо команду **Union Selection, Cmd + /, uni.**



Intersect на практиці

Намалюємо м'яч для регбі. Проявимо перетин двох кіл.

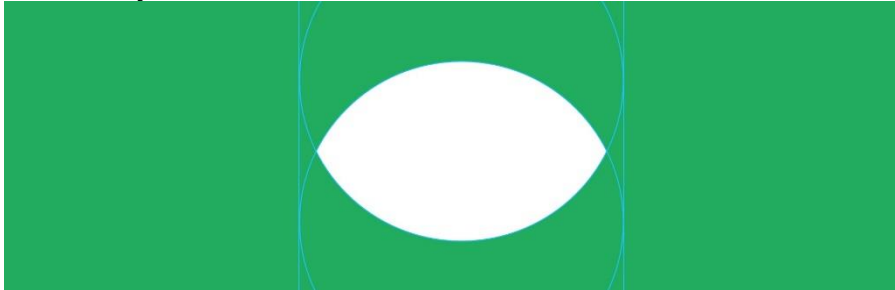
1. Малюємо коло, **O**.



2. Дублюємо його, затиснувши **Opt.** тягнемо вниз.

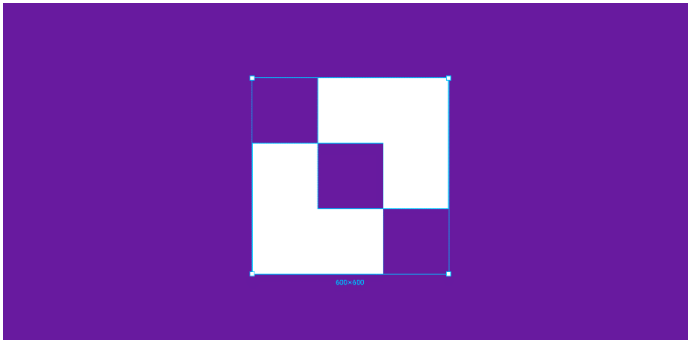
3. Виділяємо обидва, застосовуємо **Intersect Selection, Cmd + /, in.**

4. Результат:



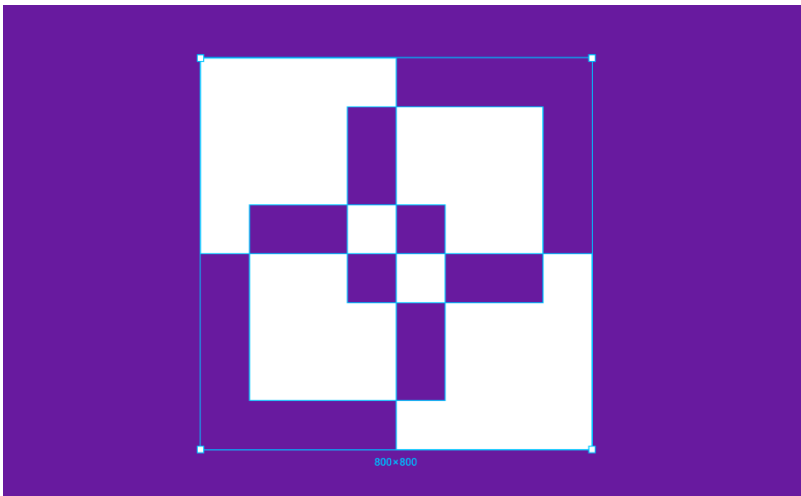
Exclude на практиці

Режим, який у випадку з двома фігурами приховує загальні фрагменти пересічних форм.

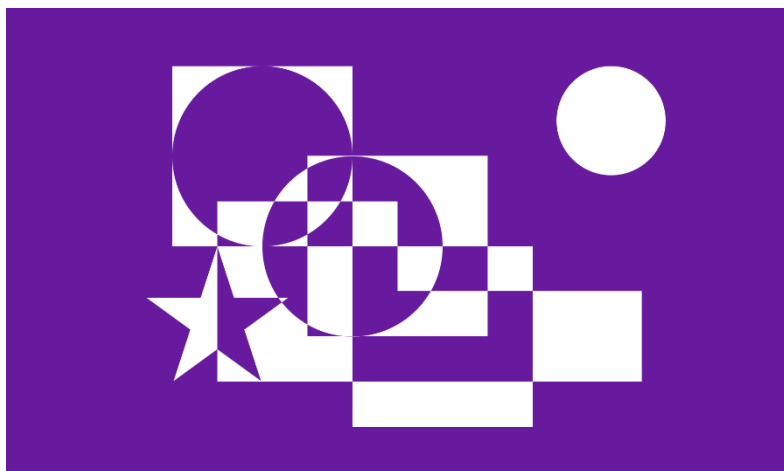


Намалюємо таку фігуру:

1. Малюємо квадрат, **R**.
2. Затиснувши **Opt**, витягуємо його дубль. Ставимо кут одного квадрата в центр іншого.
3. Виділяємо обидва, перемикаємо їх в режим **Exclude Selection**, **Cmd + /**, **ex**.



Exclude — інструмент імпровізаторів. Режим дозволяє досягати цікавих і непередбачуваних результатів. Якщо додати до початкової фігури з двох квадратів ще два і змішати їх, вийде більш складний геометричний візерунок:



Намалювати таку геометричну наркоманію в стилі 70-х за кілька хвилин без **Exclude** було б неможливо.



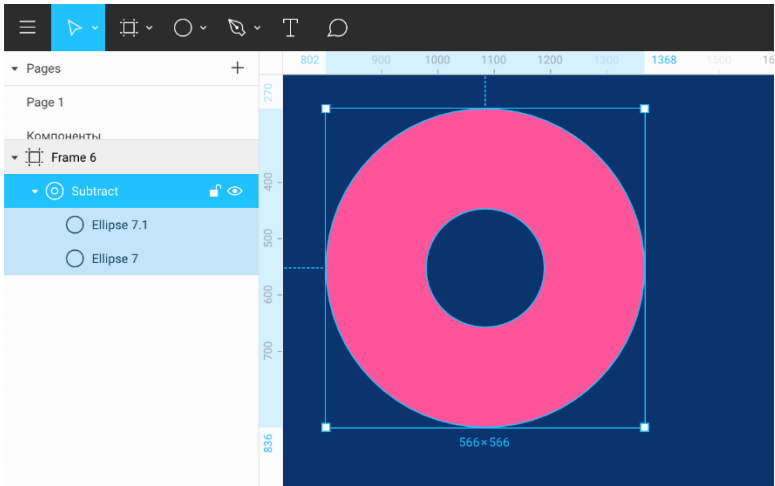
Забігаю вперед в тему текстових шарів: булеві режими працюють і з текстом. Два шара в режимі **Exclude** і з невеликим зсувом на геометричному шрифті **Gilroy ExtraBold** можуть давати такий ефект:

Подібності та відмінності булевих груп в Фігмі і Скетчі

Boolean Groups і **Boolean Operations** в цілому схожі, проте, є невеликі відмінності.

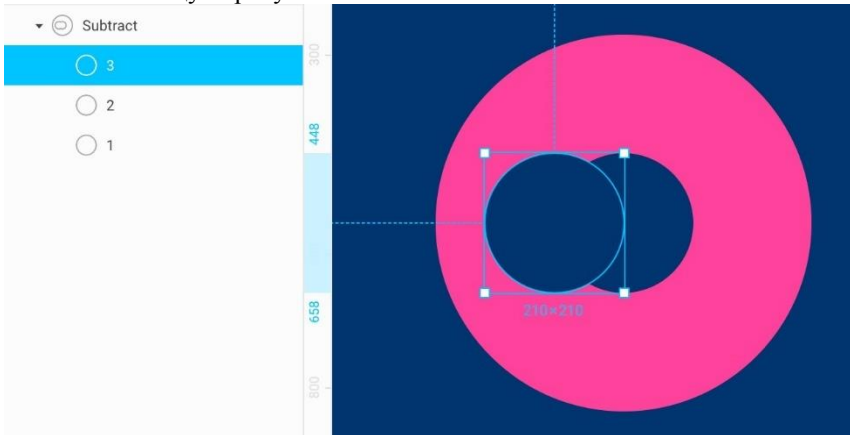
У скетчі логічний режим - це настройка шейпа. Два різних шейпа можуть бути в різних режимах. Повернемося до прикладу з пончиком і повторимо його в Скетчі. Є рожеве коло, а з нього віднімається менше коло в режимі **Subtract**. В шарі видна його іконка справа.

По кліку на неї можна перемикаєтьс в різних режимах у цього шару.



У Фігмі же логічний режим - це властивість цілої групи. Окремий шар не може бути в режимі **Subtract** або **Union**.

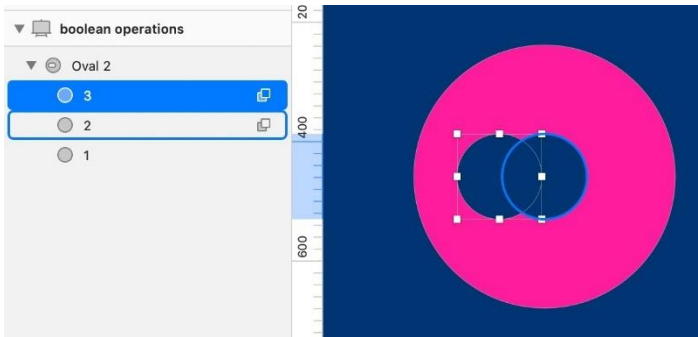
У Скетчі та Фігмі нижній шар є базовою фігурою, з якою відбувається віднімання. Якщо дублювати фігуру, яка віднімається, і зрушити її, вона збільшить площу вирахування.



Фігма:

Як ми бачимо, якщо у групі стоїть режим **Subtract**, все верхні фігури всередині цієї групи **2** і **3** віднімаються з нижньої **1**.

Оскільки ми дублювали шар **3** з шару **2**, він успадкував режим



Subtract. У правій частині списку видно його іконка.

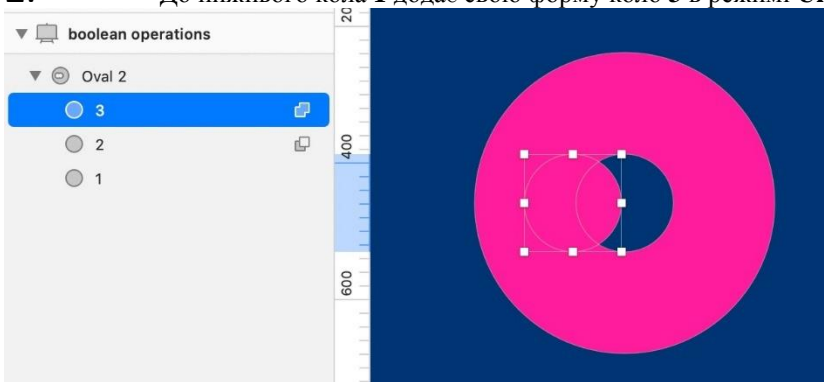
Принцип інверсії в Скетчі

Щоб віднімати з негативного простору, треба додавати.

Якщо потрібно, щоб верхнє коло **3** відрізало шматок від середнього кола **2**, перебуваючи всередині нижнього кола **1**, можна переключити коло **3** в режим **Union**, який складає форми.

Залежно від послідовності шарів, Скетч буде прораховувати підсумкову форму, рухаючись від нижнього шару до верхнього:

1. З нижнього кола **1** віднімає коло **2**. Виходить пончик.
2. До нижнього кола **1** додає свою форму коло **3** в режимі **Union**.



Коло **3** залишає від кола **2** форму півмісяця:

Вкладені булеві групи в Фігмі

Оскільки в Фігмі не можна окремого шару задати логічний режим, потрібно зробити вкладену булеву групу. Для цього всередині групи **Subtract** потрібно виділити шари **2** і **3** і знову натиснути **Subtract**.

Тепер базовим шаром всередині цієї групи став шар **2**, а з нього віднімається шар **3**. Виходить півмісяць. Потім, за правилом першої групи **Subtract**, він

віднімається з шару 1.

Треба визнати, що в Скетчі перемикати режими в списку шарів швидше, ніж налаштовувати вкладені групи в Фігмі.

В іншому реалізація дає рівні можливості.

Flatten складених фігур

Фігури, які ми створюємо через булеві групи, складаються з простих об'єктів. Однак, буває необхідність з'єднати багатшарову форму в одношарову, коли не хочеться відволікатися на структуру фігури, а потрібно працювати з нею як з єдиним цілим. Для цього в Фігмі використовується функція **Flatten Selection**, і у неї нелогічна стандартна клавіша **Cmd + E**.

Отож, коли ти робиш **Flatten**, бажано робити копію вихідної форми, щоб завжди мати можливість розібрати її на складові частини, а не мучитися, редагуючи плоску фігуру в кривих.

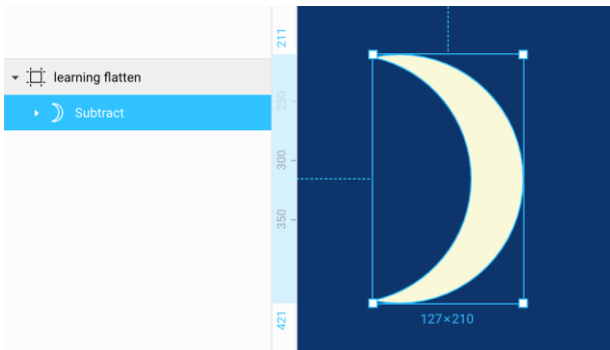
У Скетчі я призначив на неї клавішу **Ctrl + Opt + F**.

На Маке в Фігмі можна призначити на ту ж клавішу, яка використовується в скетчі.

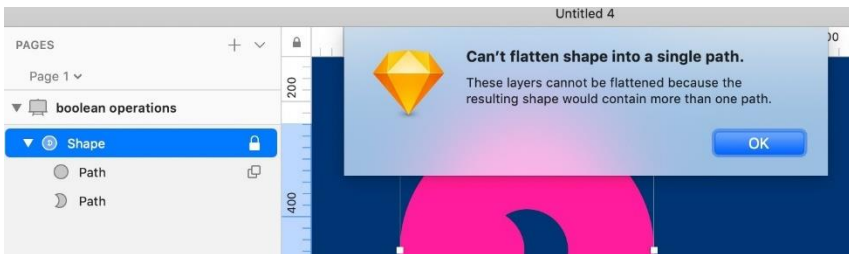
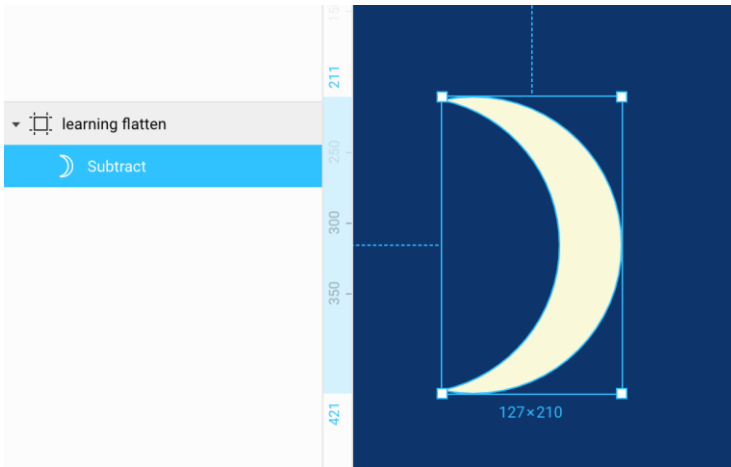
Викликати **Flatten** можна через пошук по меню: **Cmd + /, fl**.

Як зробити флетен

1. Виділяємо булеву групу. Звертаємо увагу на трикутник зліва, який її розгортає.



2. Натискаємо **Flatten**, яку призначили на **Ctrl + Opt + F**. Розкриваючий трикутник лівіше назви шару зник. Група перетворилася в шейп в формі півмісяця. Тепер його можна міняти тільки в режимі редагування, **Enter**.



Flatten в Скетчі має обмеження: не можна схлопнути фігуру, яка має отвори:

Причина в тому, що Скетч не може обробляти шейпи, які не можна промальовувати однією замкнутою векторною лінією.

У Фігмі такого обмеження немає, оскільки плоска фігура може складатися з будь-якої кількості незв'язаних ліній. Завдяки цій же особливості в Фігмі реалізовані векторні мережі.

12. Режим **Outline**: векторні контури

Проект в Фігмі →

У Фігмі є режим **Outline View**, якого нам багато років не вистачало в скетчі.

Outline зручно використовувати для роботи з розмірами фреймів і для розуміння, з чого складається векторна форма.



Кнопка та ж, що і в Ілюстраторі: **Cmd + Y**.

В цьому режимі видно тільки векторні контури шарів без урахування обводок і заливок.

- Фрейми показуються пунктиром. У шейпів видно заокруглення.
- Наконечники стрілок не показуються.
- У текстових шарів видно шейпи букв, але на жаль, не видно бокси.

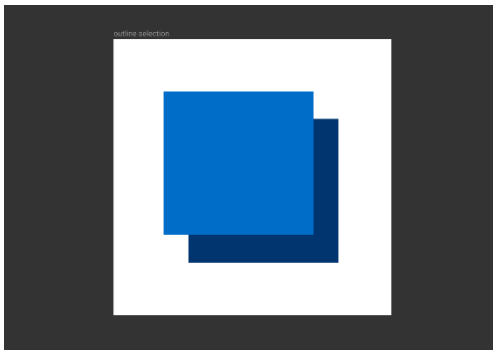
Знаходимо невидимі шейпи

Інший частий кейс, в якому зручно використовувати режим **Outline** - коли потрібно знайти невидиму фігуру, у якої немає ні обведення, ні заливки, але залишилася векторна форма.

Як виділяти важкодоступні шари

У Ілюстраторі режим **Outline** зручно використовувати для виділення важкодоступних шарів, загорджених іншими. Для цього ми натискаємо на контур потрібного шару. У Фігмі такий фокус не пройде: верхній шар зловить клік на себе. Але є рішення.

1. Припустимо, необхідно виділити нижній квадрат, який загорджений верхнім. І зробити це треба крізь верхній.



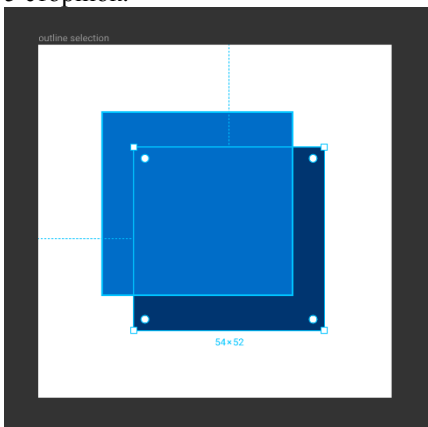
2. Перемикаємося в режим **Outline, Cmd + Y**.
3. Затискаємо **Cmd** і розтягуємо виділяючий прямокутник лівою кнопкою. У нього повинен потрапити контур нижньої фігури, при цьому треба не зачепити верхню.
4. Виходимо з **Outline, Cmd + Y**, виділено нижній квадрат.

13. Режими колірного кодування

Проект в Фігмі →

Там тільки ілюстрації.

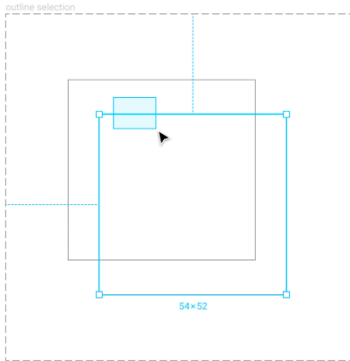
Ця базова інформація стане в нагоді, коли ми будемо вивчати заливки, обведення і градієнти. Якщо ти все це знаєш, не витрачай час і гортай наступні 5 сторінок.



HEX:шістнадцяткові кольори

Будь-який колір, який видно на екрані в рамках колірної системи **RGB** (Red, Green, Blue), може бути переданий за допомогою комбінації трьох колірних каналів: червоного, зеленого і синього. Хекс-коди складаються з трьох пар символів, кожна з яких відповідає за один з цих каналів. Використовується шістнадцяткові обчислення: 0 1 2 3 4 5 6 7 8 9 A B C D E F 00 - мінімальне

значення каналу, а FF - максимальне.



#000000 — відсутність кольору по всіх трьох каналах, чорний

[black box] — максимум по всіх каналах, білий

#FF0000 — максимум по червоному каналу, яскраво-червоний

#00FF00 — максимум по зеленому, яскраво-зелений

#0000FF — максимум по синьому каналу

Якщо по всім трьом каналам стоїть рівне значення, наприклад, # 808080, колір буде монохромним, тобто не матиме відтінку. Це відбувається, тому що канали один одного компенсують.

Хекс-коди - це основний стандарт кодування кольорів в інтернеті. Їх плюс в тому, що їх зручно копіювати і передавати.

RGB-цвета

Крім Хекс-кодів, кольори можна кодувати в схожій трьоканальній системі, де замість кодів використовуються десяткові значення.

0 — мінімальне значення каналу, 255 - максимальне.

0, 0, 0 — відсутність кольору по всіх каналах, чорний [black box] — максимум по всіх каналах, білий **255, 0, 0** — максимум по червоному каналу, яскраво-червоний

0, 255, 0 — максимум по зеленому, яскраво-зелений

0, 0, 255 — максимум по синьому каналу

Такий метод кодування кольорів застосовується у пресі нарівні з системою каналів **СМΥΚ** (Cyan, Magenta, Yellow, Black), а також послужив прототипом для системи **RGBa**, про яку поговоримо далі.

Плюс такої системи, що такі коди більш запам'ятовувані і їх зручно диктувати на слух.

RGBa-коди

В якості альтернативи неінтуїтивним Хекс-кодам розробили систему кодування, адаптовану для використання в коді веб-сторінок та інших програм.

Вона містить в основі RGB, а також додає до цих трьох каналів додатковий канал, який дозволяє задати опасіті. Він називається альфа-каналом.

Перші три канали задаються від 0 до 255, а альфа-канал від 0 до 1.

Значення 0.5 відповідає 50% опасіті. Як формується RGBA-код: `rgba(червоний, зелений, синій, альфа)` Приклади:

`rgba(0,0,0,0.5)` — чорний, з опасіті на 50%

`rgba(255,255,255,1)` — непрозорий білий

Така система має плюси від обох підходів: RGBA-код можна скопіювати і все ще легше запам'ятовувати.

Їх можна вставляти в CSS-код замість Хекса.



HSB-кольори

Існує також третя система кодування кольору – HSB.

У ній колір не ділиться на канали, а формується з трьох якостей: Hue [хью], Saturation [сетюрейшн], Brightness [брайтнес].

1. Hue — відтінок по колірному колі, який відповідає градусу. Від 0° до 359°. Значно наочніше ця шкала буде, якщо уявити її у вигляді кола. Шкала починається з червоного. Бірюзового відтінку, протилежного червоному відповідає кут 180. На жаль, це не є очевидним з шкали відтінку:



2. Saturation — насиченість відтінку від 0 до 100.

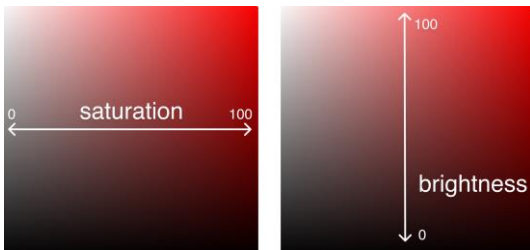
Передає, наскільки колір їдкий і інтенсивний.

Колір може бути блідим або концентрованим. Якщо у нього зовсім відсутній відтінок, значить, він монохромний і в шкалі кольорів буде прибитий до лівої границі. Чим більше насиченість, тим ближче до правої. Щоб змінювати насиченість, потрібно пересувати маячок по горизонталі.

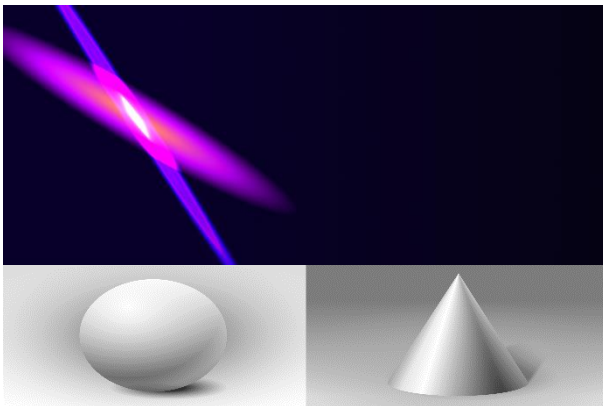
3. Brightness — яскравість відтінку. Яка частка чорного у відтінку.

Червоний може прагнути до бордового, і це означає, що в ньому підмішано багато чорного, отже, знижена яскравість. Навпаки, чистий червоний показує відсутність чорного. Щоб змінювати яскравість, потрібно пересувати маячок вертикально.

Шкала червоного відтінку:



На цьому з теорією все.

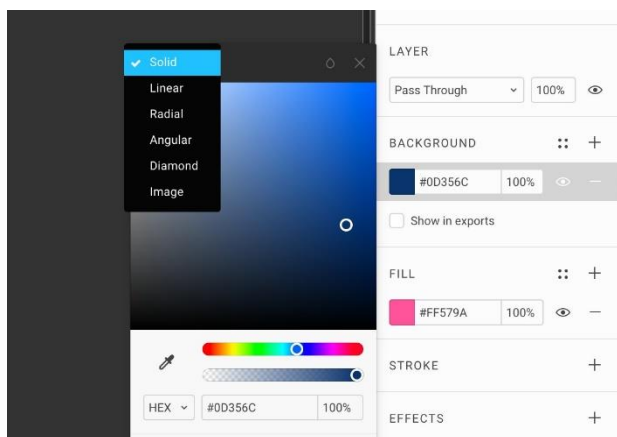


14. Заливка і градієнти

Проект в Фігмі →

Заливка **Fill** може бути у фрейма, шейпа або текстового шару. У цьому розділі розглянемо, які бувають режими заливки і навчимося їх застосовувати на

практиці.



У верхньому лівому кутку розгорнутої палітри відображається режим заливки фону, який спочатку обраний в **Solid**.

Лівіше райдужного фейдера розміщена кнопка з піпеткою, що дозволяє знімати кольори в будь-якому місці полотна. У піпетки клавіша як в Скетчі: **Ctrl+C**. Мнемоніка: **Color**. На Windows – **I**.

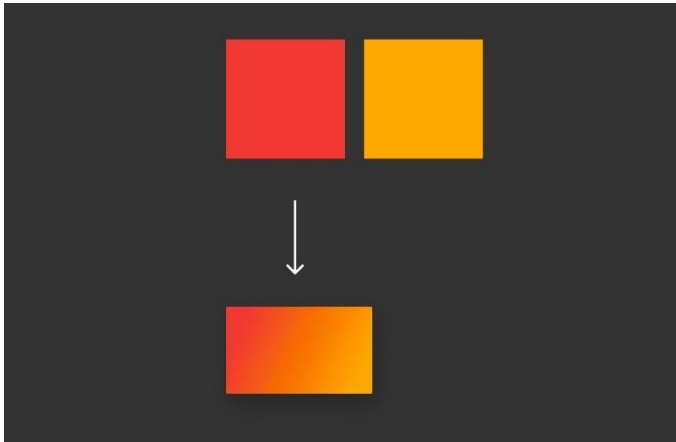
Робимо пробники для градієнтів

Я розповім про корисний прийом, який зробить роботу з градієнтами значно комфортніше.

Створюємо кілька прямокутників, **R**, яким налаштуємо кольори. Щоб налаштувати градієнт, будемо знімати піккером кольори з цих пробників.

Коли розгорнута панель градієнтів, в ній не дуже зручно оцінювати обраний колір, тому що віконця контрольних точок градієнта маленькі. Тому перехід кольору виходить недостатньо точним.

Якщо ми заздалегідь визначимо кожен колір, а потім знімемо їх в градієнті, результат буде більш передбачуваним.



Особливо це зручно при роботі зі складними градієнтами, що складаються з декількох кольорів.

Режими градієнтів

За винятком **Diamond**, режими заливки в Фігмі і Скетчі дуже схожі:

- **Solid** [солід] — режим заливки суцільним кольором, який можна задавати як шістнадцятковий код (HEX-код від слова *hexadecimal*) від білого **#FFFFFF**, до **#000000** чорного.
- **Linear** [лінійар] — режим заливки лінійним градієнтом, в якому можуть бути дві або більше контрольних точок кольору.
- **Radial** [редіал] — режим заливки радіальним градієнтом, в якому один кінець шкали градієнта асоціюється з центром кола, а інший з його краями. Від центру до країв можна налаштувати плавний перехід.
- **Angular** [енгюлар] — режим заливки градієнтом, в якому шкала від початку до кінця градієнта обертається навколо певної точки на заливається фігурі.
- **Diamond** [даймонд] — режим заливки радіальним градієнтом з 4 променями, який є тільки в Фігмі. Дозволяє налаштувати градієнти з прямокутною текстурою.
- **Image** — режим заливки растровим зображенням. Застосовується для вставки растрових шарів в Фігму. Якщо вставити PNG або JPG-файл у вікно редактора, він з'явиться на полотні як прямокутник із заливкою цього типу.

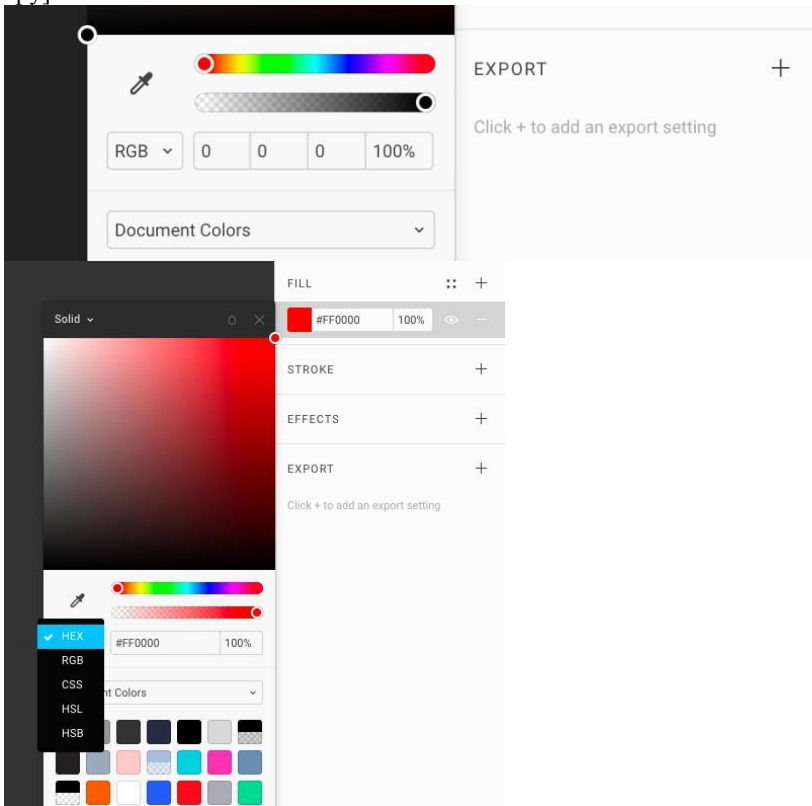
Режим заливки **Solid**: рівний колір

Найпростіший режим заливки суцільним кольором. Спочатку колір показаний у вигляді Хекс-коду. Такі коди не несуть інформації про те, що колір може бути прозорим.

Фейдер нижче райдужного дозволяє налаштувати опасіті заливки від 0 до 100.

Також його можна задати в полі праворуч від Хекс-коду.

Опасіті шару налаштовується в схожому полі в блоці **Layer** навпроти режиму накладення. Зараз встановлено режим за замовчуванням **Pass Through** [пестру]. Клавiші: **1 - 0**.

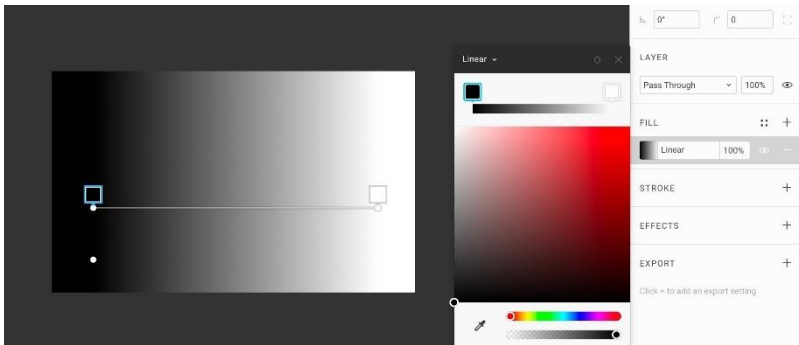


У режимі **Solid** можна перемкнути спосіб кодування кольору:

В режимі RGB ми бачимо три поля для кожного каналу і четверте поле для альфа-каналу.

Режим заливки **Linear**: лінійний градієнт

Градієнти - плавні переходи від одного кольору до іншого. Лінійний градієнт має вісь, уздовж якої змінюється колір.

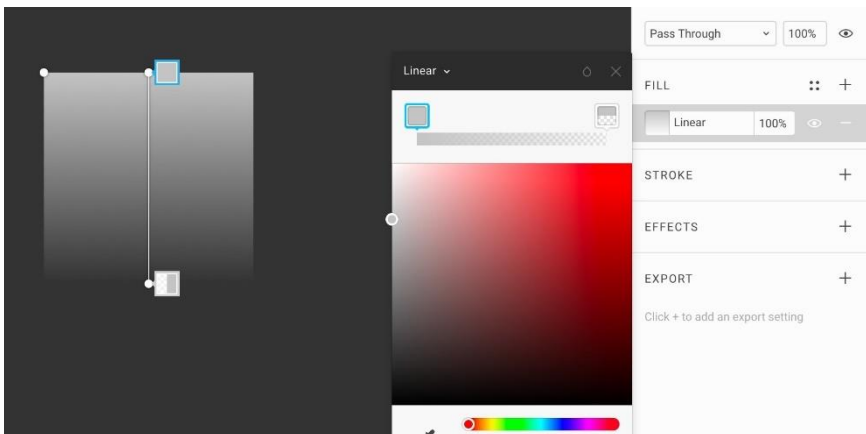
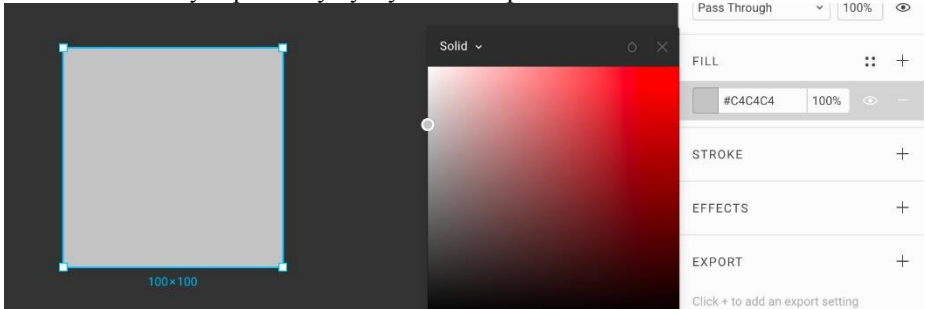


Задати, які кольори беруть участь в градієнті, можна за допомогою контрольних точок на його осі. У цьому простому градієнті їх дві: чорна і біла.

Створимо градієнт:

1. Створимо шейп квадрата, **R**. Розгорнемо вікно заливки. За замовчуванням її колір сірий #C4C4C4.

2. У лівому верхньому кутку міняємо режим заливки з **Solid** на **Linear**.



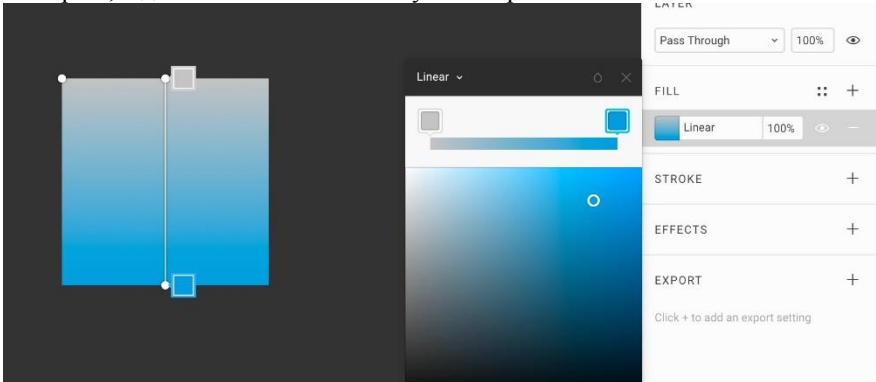
По центру фігури з'явилася вертикальна вісь і дві контрольні точки кольорів.

Верхня точка виділена.

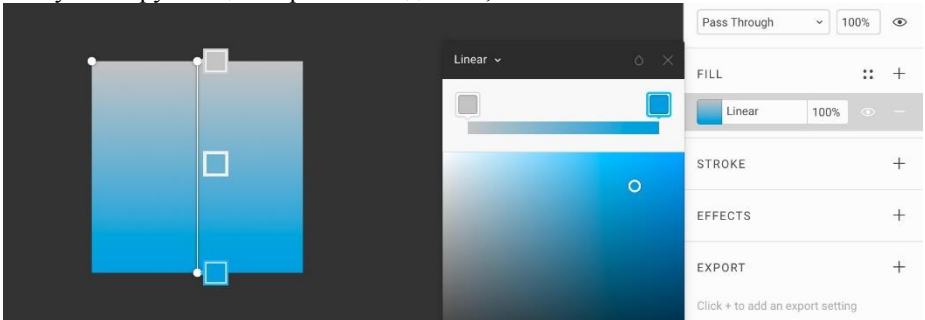
У нижній точці вибраний той же колір, проте його опасіті викручена в 0. Тому, виникає ефект плавного зникнення квадрата. Нижній квадратний пробник кольору розділений на дві частини. У лівій половині видно колір з урахуванням поточної настройки опасіті, праворуч повністю видимий.

Якщо в режимі **Solid** був спочатку вибраний інший колір, при переході в **Linear** він буде використаний в градієнті.

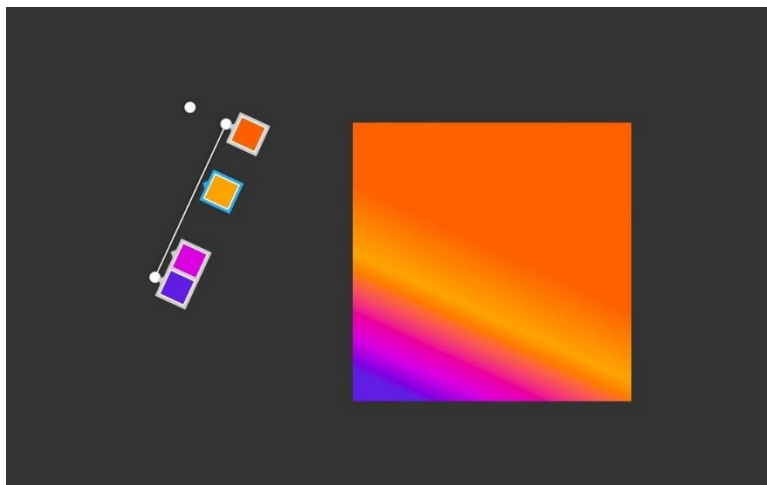
3. Виділимо нижню точку, повернемо опасіті і поміняємо колір. Стане явно видно перехід між кольорами. У момент підбору відтінку рекомендуємо покрутити райдужний слайдер відтінку (hue). Це допомагає підбирати такі відтінки, які не конфліктують один з одним. Якщо покажчик знаходиться над палітрою, відтінок можна налаштувати скролом миші.



4. У центрі осі можна клікнути і таким чином додати ще одну контрольну точку кольору. Якщо потрібно її видалити, натискаємо **Delete**.



Точки можна пересувати по осі, а саму її тягнути за кінці і ставити їй потрібний кут.



Корисні клавіші для роботи з градієнтами

Щоб зафіксувати градієнт і приховати панель заливки, натискаємо **Enter** або **Esc**. Якщо контрольна точка виділена, її можна зміщувати **стрілками** по осі градієнта. Затиснутий **Shift** прискорює рух стрілки на осі.

На жаль, на відміну від Скетча, між точками в градієнті можна переключатися з клавіатури через **Tab** і **Shift + Tab**.

Під час зміни кута затиснутий **Shift** дозволяє перейти в режим **Snap To Geometry** і фіксувати вісь під рівними кутами з кроком в 45° рівно так само, як ми це робили в розділі про перо.

Цікава можливість, якої немає в Скетчі: якщо схопити за один кінець лінії і затиснути **пробіл**, всю лінію разом з її контрольними точками можна буде зрушувати, не змінюючи кута. Це дає можливість швидко і точно налаштувати положення градієнта, не рухаючи кожен його кінець індивідуально.

Перпендикуляр градієнта

Ще одна унікальна особливість лінійних градієнтів в Фігмі: кут градієнта можна повертати щодо основної осі.

У градієнта є третя непримітна кругла ручка, призначення якої відразу незрозуміло. Якщо провести через неї пряму лінію до одного з кінців осі, вийде кут 90° . Це додаткова ручка перпендикулярній осі градієнта. Якщо затиснути **Opt** і потягнути за неї, колірний перехід буде відмальований перпендикулярно їй.

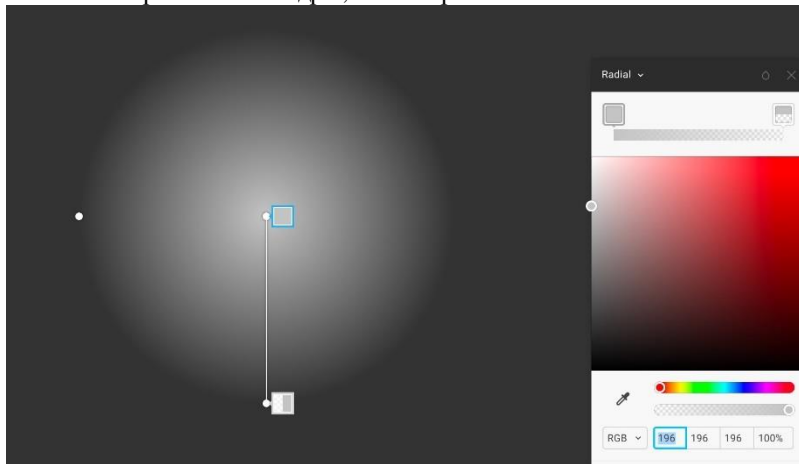
Режим заливки Radial: радіальний градієнт

Такий градієнт формується на основі еліпса. У нього є центр і зовнішній кордон. Вони з'єднані радіусом, на якому розміщуються контрольні точки кольору.

Приклад з двох накладених один на одного радіальних градієнтів. Для посилення ефекту світіння застосував режим накладення Color Dodge у верхнього.

Створимо простий радіальний градієнт:

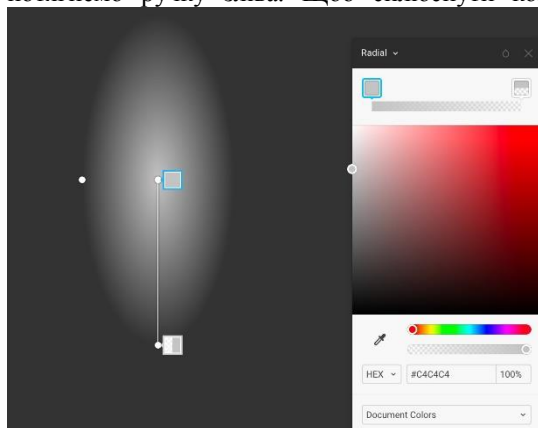
1. Створимо новий квадрат, **R**. Розгорнемо панель **Fill**.



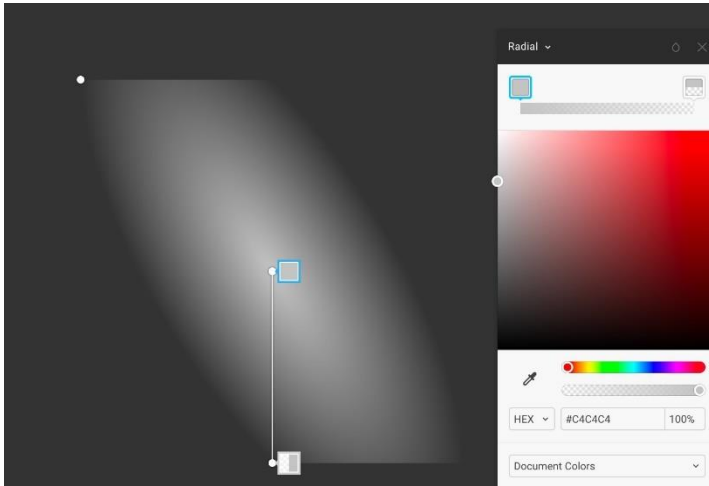
2. Переведемо заливку з режиму **Solid** в **Radial**. Результат:

3. Аналогічно лінійним градієнтам у нас є дві контрольні точки сірого кольору, одна з яких повністю прозора. Також видно радіальну вісь.

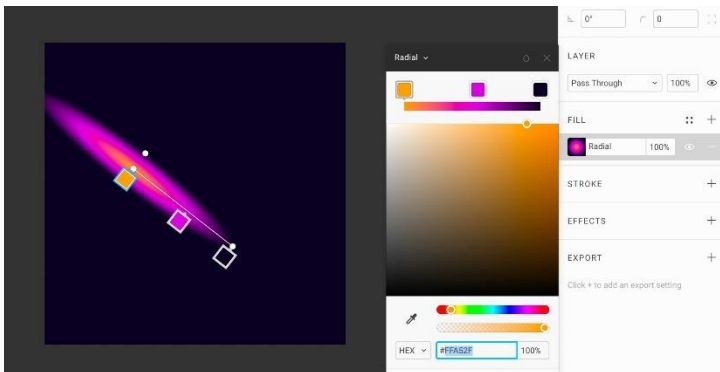
4. Щоб створювати радіальні градієнти не тільки на основі кола, потягнемо ручку зліва. Щоб сплюснути коло в еліпс, тягнемо до центру.



5. Працює також і трюк з **Opt**, як в лінійних градієнтах. Він дозволяє зміщувати ручку так, щоб еліпс спотворювався. Це зручніше, ніж налаштовувати загальний кут нахилу радіуса, а потім ужимати еліпс по малому радіусу.



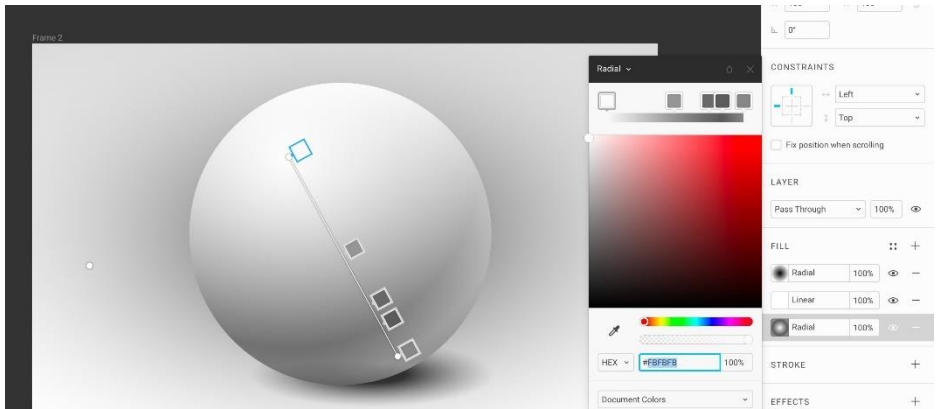
В даному прикладі я з затиснутим **Opt** повів ручку у верхній кут. Еліпс виявився нахилений, хоча основна вісь залишилася вертикальною.



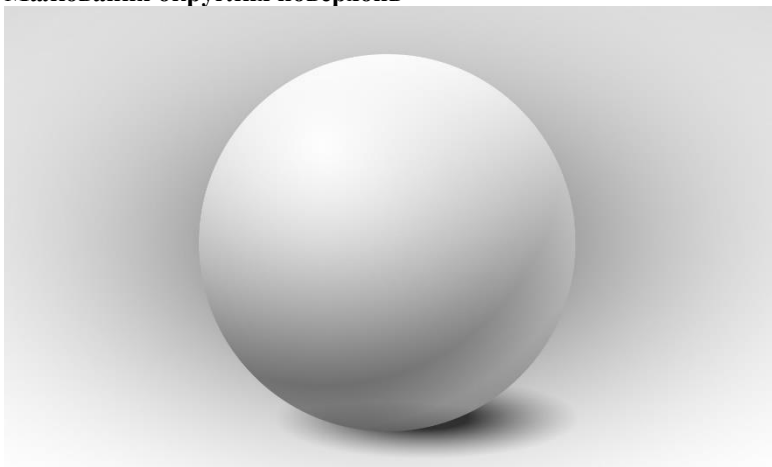
Повторимо приклад з фіолетовим білком.

1. Накладаємо нижній радіальний градієнт. Щоб зміщувати всю вісь, не змінюючи місце, затискаємо **Shift**.
2. Робимо радіус менше розміру основної фігури, так щоб зовнішній колір градієнта став фоновим для всієї фігури.
3. Робимо копію першого градієнта. Для цього виділяємо пункт в списку **Fill** і натискаємо **Cmd + D**. Таким чином на фігурі квадрата тепер два шари заливки. Схожим чином працюють шари і для обведення **Stroke**.
4. Зміщуємо другому градієнту кут і рухаємо перпендикуляр якомога ближче до центру.

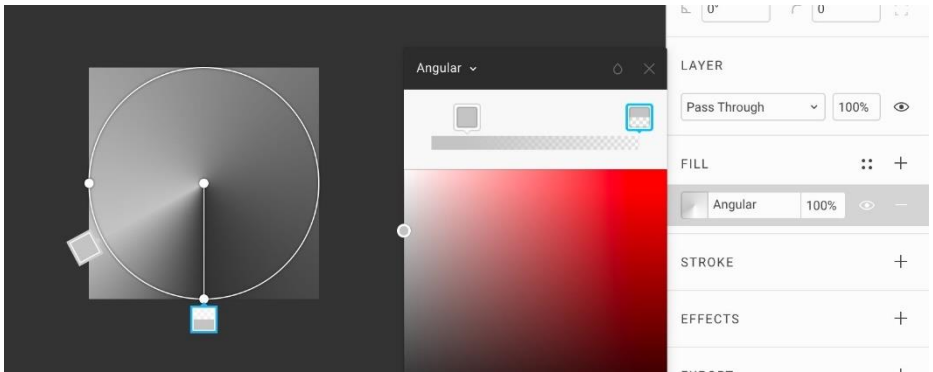
5. Ставимо іконку краплі справа - режим накладення цього градієнта в режим **Color Dodge**. Виходить ефект відблиску.



Малювання округлих поверхонь



За допомогою радіальних градієнтів можна промальовувати округлі поверхні, імітуючи обсяг.



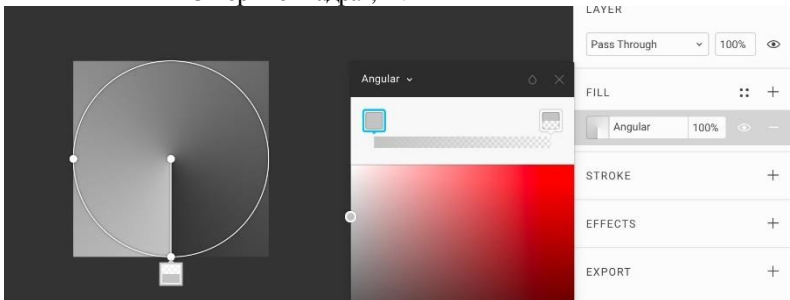
Режим заливки **Angular**: кутовий градієнт

Такий градієнт формується на основі кута в 360° , в якому перехід кольору йде від початкової точки до кінцевої. Оскільки цей кут складає повний круг, для управління таким градієнтом теж використовується еліпс. Однак, перехід кольору в ньому відбувається по зовнішньому контуру, а не по радіусу.

Пам'ятай, що кутові градієнти не підтримуються в форматі SVG, тому такі фігури доведеться експортувати в растр.

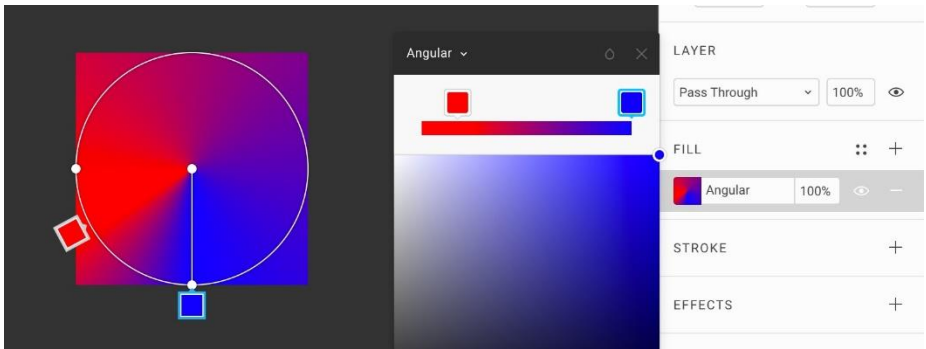
Задача: Зробимо простий радіальний градієнт.

1. Створимо квадрат, **R**.



Перемкнемо заливку в режим **Angular**. Буде видно коло, діаметр якого відповідає довжині квадрата.

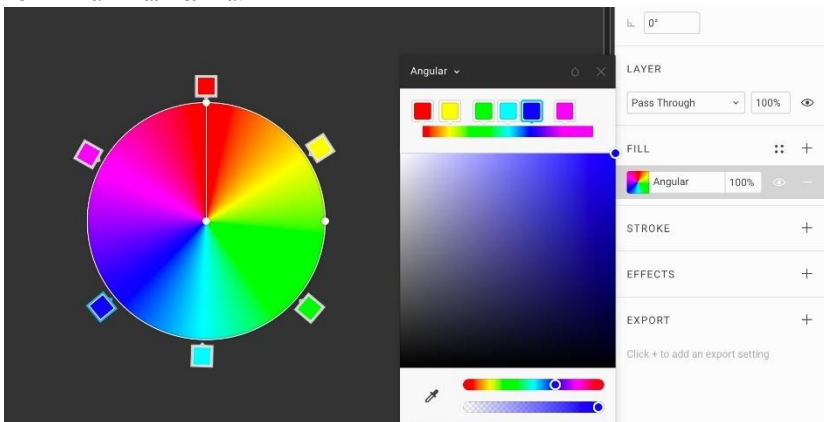
Початкова і кінцева точка стоять на 6 годин, а перехід відбувається за годинниковою стрілкою. Як і в будь-якому іншому режимі, градієнт спочатку складається з одного кольору з різною опасіті. Починається з повністю видимого кольору # C4C4C4, а закінчується повністю невидимим. Оскільки заданий темно-сірий колір полотна, здається, що градієнт йде від світлого до темного, але насправді один і той же сірий лише втрачає опасіті.



2. Щоб налаштування кутового градієнта стало наочніше, перетягнемо початкову контрольну точку кольору наліво, на 8 годин. Кінцева точка так і залишиться на 6 годин. Тепер вони не накладаються один на одного.

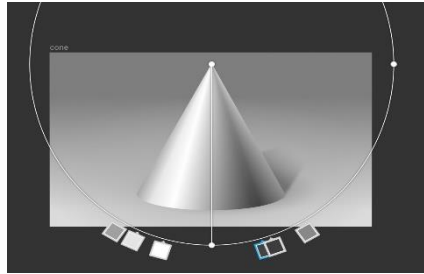
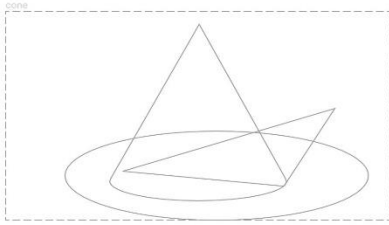
3. Міняємо кольори на більш помітні: перша отримає червоний, а друга синій. Тепер явно видно, як в обидві сторони від червоної крапки колір поступово переходить від червоного до синього.

4. Також як і в інших типах градієнтів можна задавати контрольні точки на шкалі зліва.

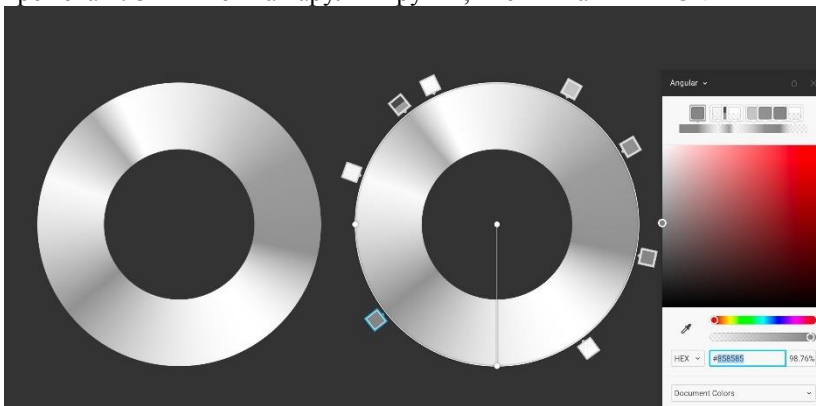


Використовуючи кутовий градієнт, можна зробити круговий спектр, розділивши окружність на 6 рівних відрізків:

Також за допомогою кутового градієнта можна намалювати конічні поверхні. Саму форму конуса можна намалювати за допомогою еліпса і пера. Потім, об'єднати їх в одну фігуру, розірвавши верхню точку еліпса.

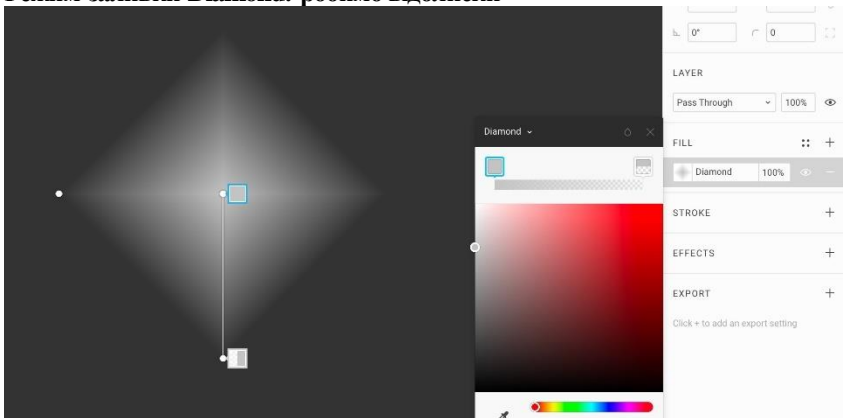


Ще одне застосування радіальних градієнтів - круглі поверхні, схожі на хромовані. З них можна накрулити ручки, кнопки та інший UI.



Тільки, будь ласка, не треба, це вже не модно.

Режим заливки Diamond: робимо відблиски

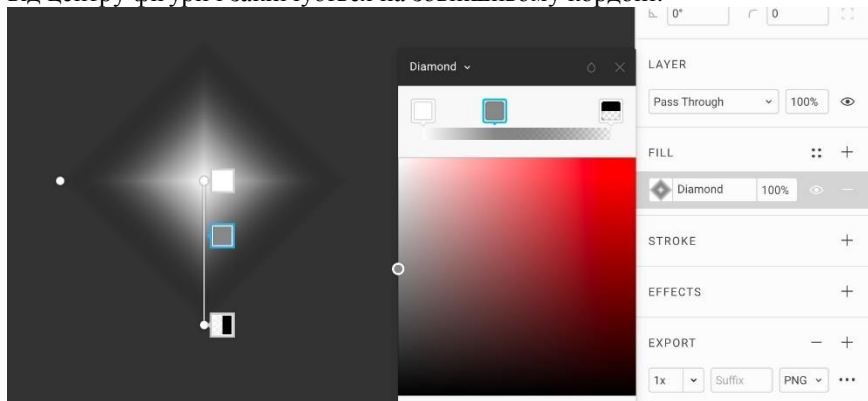


Новий режим, якого немає в Скетчі. Дозволяє робити градієнти, схожі на радіальні, з тією відмінністю, що зовнішній край переходу має форму ромба.

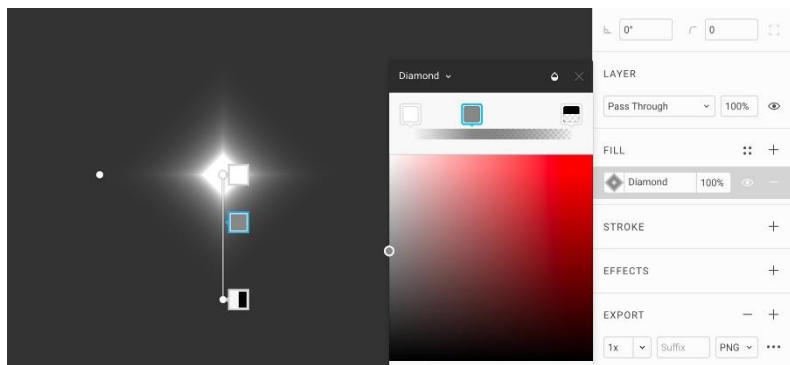
Застосовується рідко. Годен в основному для відблисків на безглузких хромованих ретро-футуристичних логотипах в стилі 80-х.
Але навіть в них слід бути обережним з силою відблиску, інакше вище неореалістично і дешево.

1. Створимо квадрат, **R**.
2. Переведемо його заливку **Fill** в режим **Diamond**.

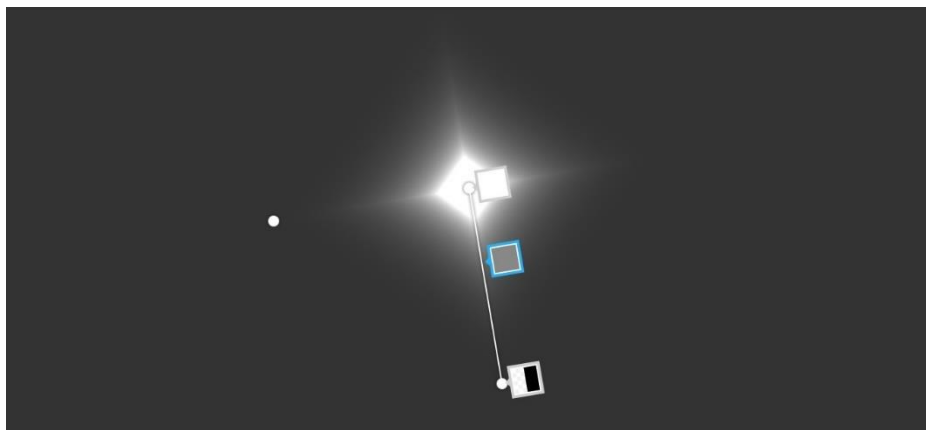
Як і в радіальному градієнті, в цьому типі вісь переходу кольору починається від центру фігури і закінчується на зовнішньому кордоні.



Виберемо першим кольором білий, посередині шкали поставимо сірий, а в кінці - чорний. Повністю приберемо йому опасіті.



Перемкнемо градієнту режим накладення (меню з краплею) в **Color Dodge**. Цей режим підсилює білий в центрі і приглушує сірий на краях, даючи відчуття «пересвета»:



Рухаючи сіру точку, можна налаштувати бажаний розмір білого засвіченого ромба. Бажано, щоб форма зовнішнього ромба взагалі була розмита. Залишиться тільки внутрішній білий ромб і чотири променя.

3. Загинаємо кут, тримаючись за нижню точку.
4. Виходимо з редагування градієнта, **Enter**.

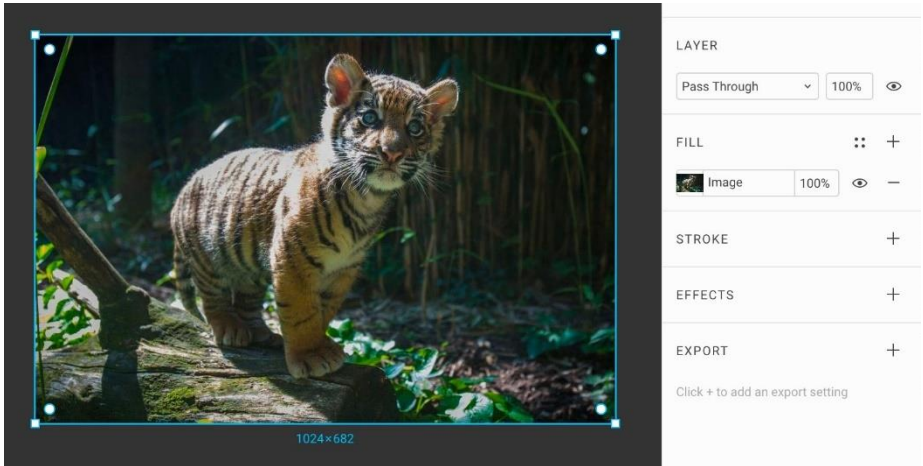


5. Ставимо відблиск на край фігури, де міг би відбиватися найяскравіший джерело світла.

15. Режим заливки **Image**

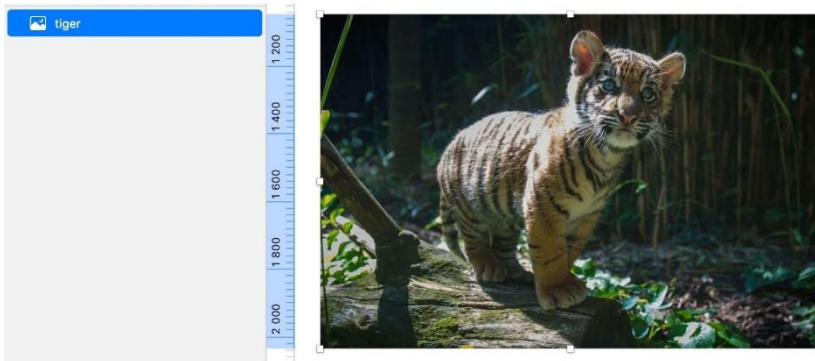
Проект в Фігмі →

Режим **Image** — останній з режимів в списку, але один з найважливіших. Дозволяє накладати на фігури растрову заливку: фотографії, патерни і текстури. Важливо зрозуміти цю концепцію. Якщо в Фігму вставити будь-який JPG або



PNG-файл, він з'явиться на полотні як векторний прямокутник. Растрове зображення буде накладено на нього в режимі **Image**. Вставляємо растровий файл в Фігму. Це можна зробити перетягуванням, або командою **Place Image, Shift + Cmd + K**.

На відміну від всіх інших режимів заливок, якщо виділити шар з растром і натиснути **Enter**, ми перейдемо в панель роботи з заливкою, а не в режим редагування векторної фігури. Так інтерфейс розставляє акценти, що в режимі **Image** користувачеві важливіше працювати з заливкою, а не з шейпом.



У Скетчі є окремий тип об'єктів - **Bitmap**. Увага на іконку зліва. Якщо вставити растровий файл, він з'явиться на полотні як бітмап, і в цьому ключова відмінність Скетча від Фігми при роботі з растром.

Бітмапи можна обрізати і заливати кольором. У Скетчі теж можна накласти на прямокутник растрову заливку, проте, ми не зможемо її редагувати і як завгодно кадрувати після вставки.

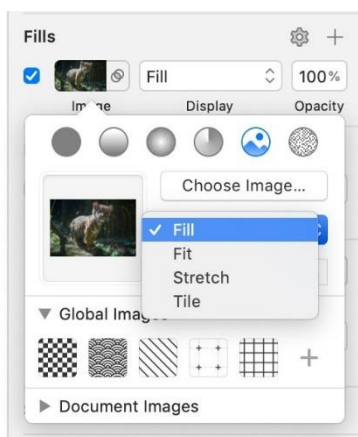
Оскільки в Фігмі немає бітмапів, ми редагуємо растрові об'єкти тільки через заливку.

Виділяємо шар з фото і відкриваємо панель заливки, **Enter**.



Сам режим заливки **Image** може бути в 4 варіантах.

- **Fill** — вся площа векторної рамки заповнюється растром.
- **Fit** — растр вписаний в векторну рамку.
- **Crop** — режим, в якому можна тонко налаштувати, як потрібно обрізати растр, змінюючи форму векторної рамки, нахиляючи або спотворюючи пропорції растра.
- **Tile** — режим, в якому растр використовується як патерн і заповнює собою площу фігури немов плитка.



У скетчі режими майже такі ж:

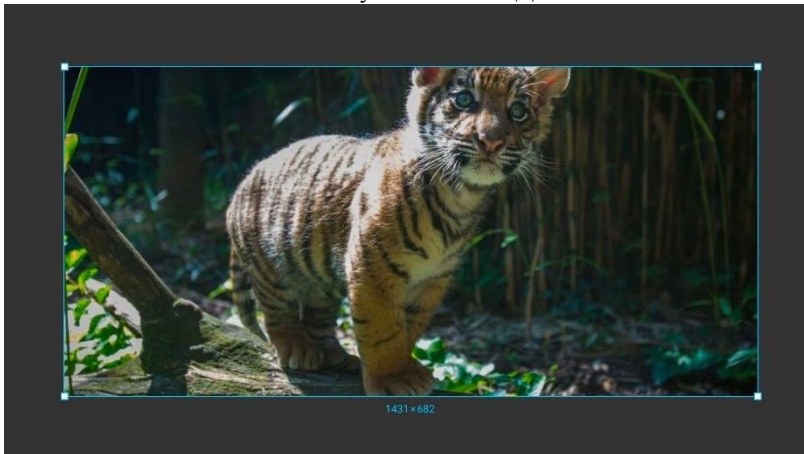
- **Fill**
- **Fit**
- **Stretch** — заповнення фігури растром з спотворенням пропорцій

- **Tile**

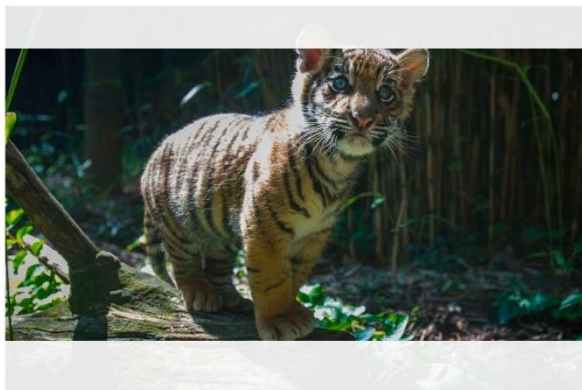
Режиму **Сtop** немає, оскільки ця логіка реалізується через обріз бітмапів.

Режим **Image / Fill**

Встановлено за замовчуванням. Діє як заливка растром



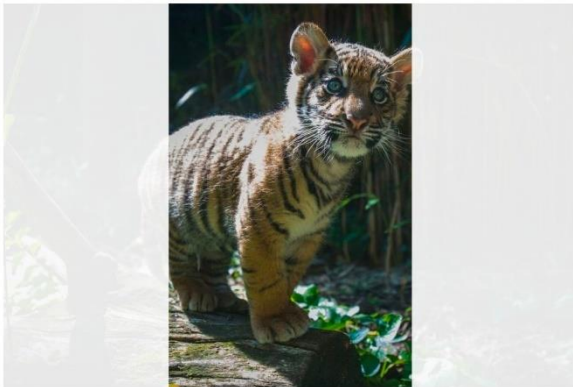
в Скетчі в однойменному режимі. Растрове зображення максимально заповнює площу фігури. Центр растра в тій же точці, що і центр фігури-рамки. Якщо розтягнути прямокутник, растр підлаштується по більшій зі сторін. Потягнемо ширину:



Растр заповнить всю доступну ширину. Фігма приховає надлишки зверху і знизу.



Якщо ужимати ширину так, що вона стане менше висоти, растр буде заповнювати всю доступну висоту.



Центр растра і прямокутника, як і раніше залишаються в одній точці. Зрушити зображення усередині прямокутника, змістивши центральну точку, не можна. Для цього використовуємо режим **Crop**.

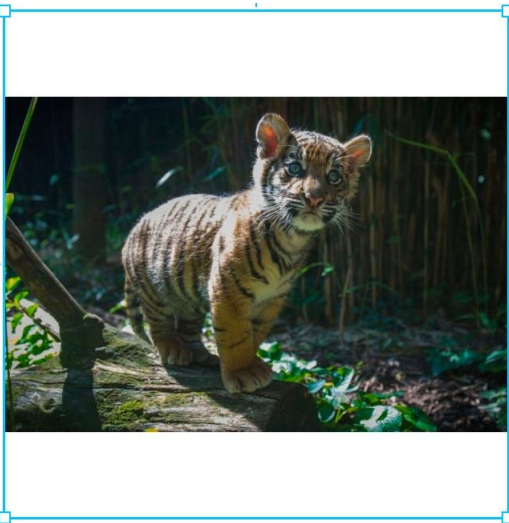
Режим **Image / Fit**

В цьому режимі растр вписується в фігуру по меншій зі сторін: його завжди видно і не обрізається прямокутником.



444x256

Якщо у прямокутника і растра в ньому різні пропорції, з боків з'являються порожні:



157x158

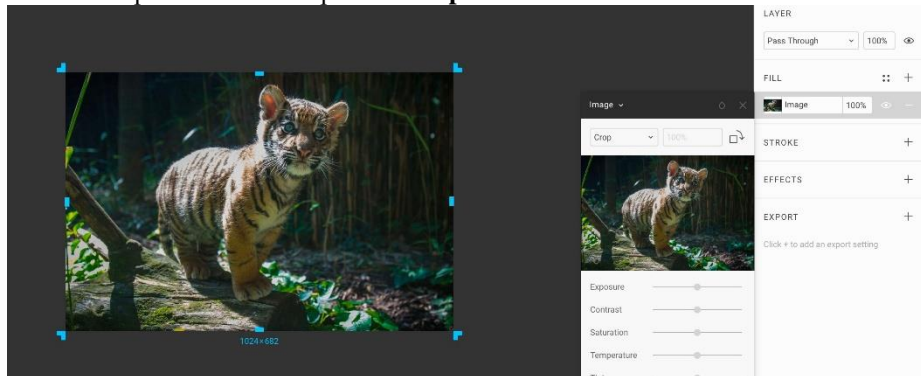
Режим Image / Crop

Коли ми вставляємо зображення в Фігму, дуже часто потрібно його обрізати. Режим **Crop** дозволяє змінювати пропорції векторної рамки, в якій є растрова заливка. При цьому, растр не зміщується.

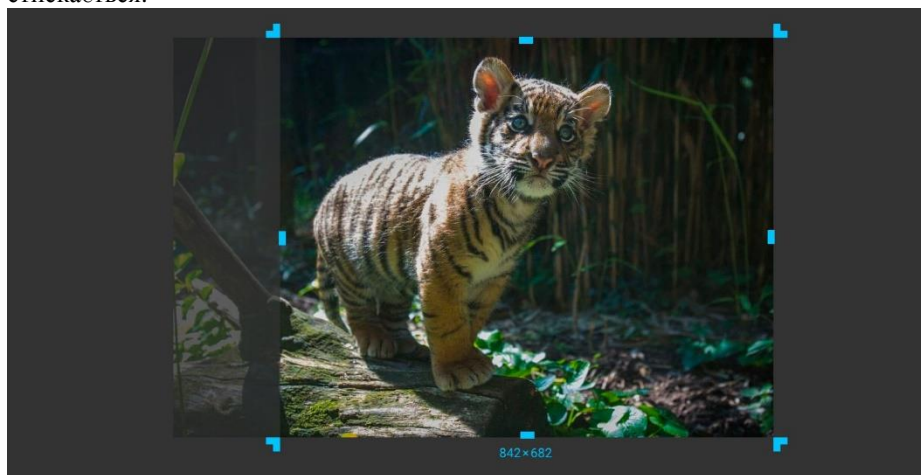
Якщо ми хочемо задати точний розмір обрізаного зображення в пікселях, це

краще робити в режимі **F3H**. Якщо в режимі **Crop** задавати значення ширини і висоти рамки з клавіатури, растр буде спотворюватися.

1. Перемикаємося на режим **Crop**.

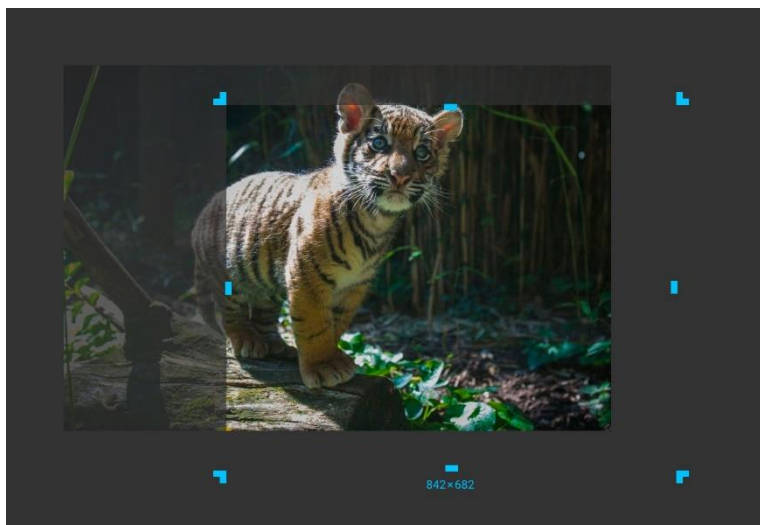


2. Тягнемо за будь-яку ручку до центру зображення. Рамка стискається.

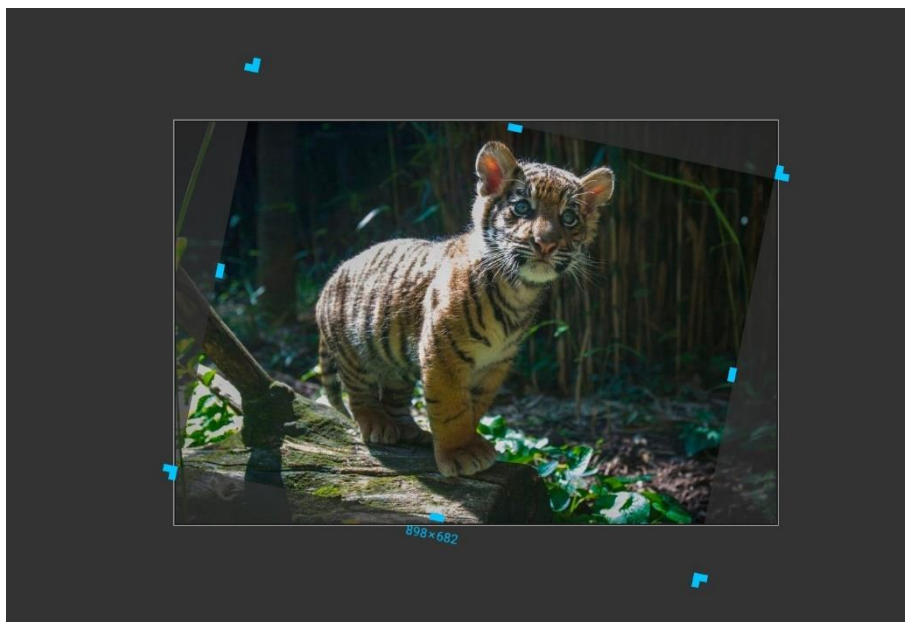


Щоб кадрувати від центральної точки, затискаємо **Opt**.

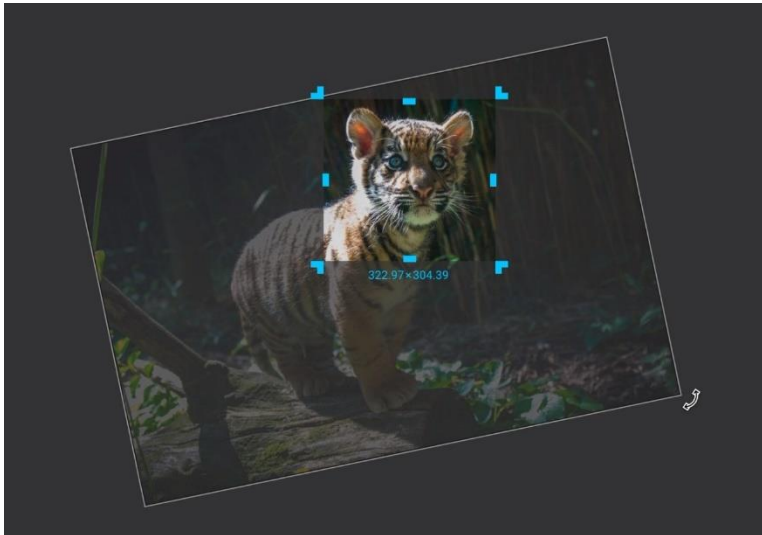
3. Також в цьому режимі можна зрушувати зображення в рамці. Для цього затискаємо **Cmd**, беремо за растр і тягнемо його в потрібну сторону.



4. Рамку кропа можна нахиляти і таким чином виправляти завалений горизонт. Для цього наведемо на будь-яку з кутових ручок, поки курсор не перетвориться в вигнуту стрілку.



Нахиляємо кроп-рамку:



5. Також можна нахилити і сам растр, тримаючись за його кут.

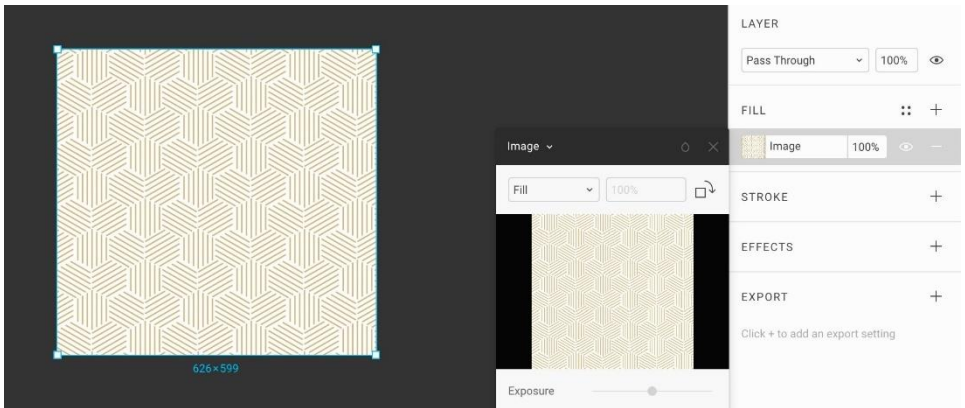
Якщо необхідно спотворювати його пропорції, тягнемо за білі краї.

6. Щоб зафіксувати зміщене чи обрізане зображення, двічі натискаємо в межах рамки або натискаємо **Enter**. При цьому вихідне зображення не змінюється, а лише маскується. До нього завжди можна повернутися, виділивши шар і натиснувши **Enter** ще раз.

Щоб скинути зсув і кут нахилу кропа, можна повернутися в режим **Fill**.

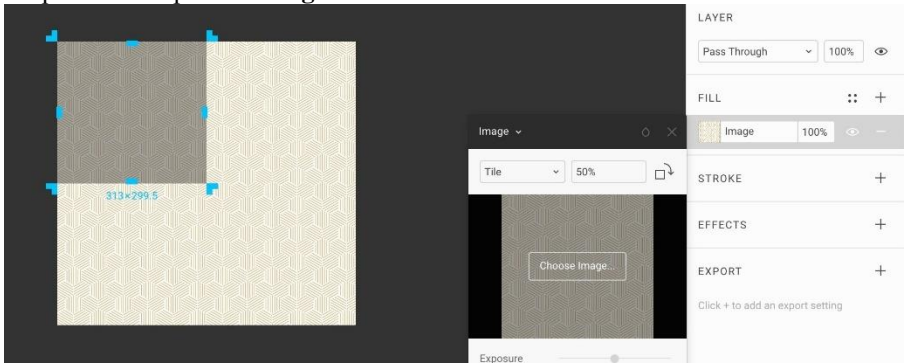
Режим **Image / Tile**

Режим **Tile** [тайл] використовується для заповнення фігури повторюваними фрагментами - безшовним паттернами. Зазвичай вони намальовані так, що якщо їх розмножити і поставити впритул один до одного, шви між ними будуть непомітні.



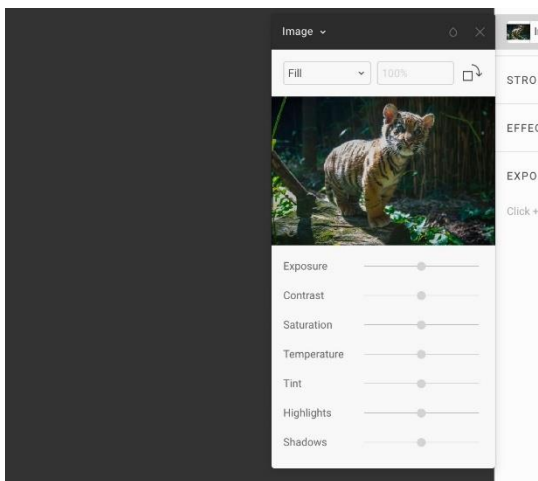
Вставляємо растровий патерн в Фігму:

Перемикаємо режим **Image** з **Fill** на **Tile**.



З'являється темна рамка з ручками:

Вона визначає, якого розміру буде сітка, по якій патерн замостить фігуру. Нижня права ручка дозволяє налаштовувати бажаний розмір плитки. Незалежно від розміру фігури, патерн буде поширюватися на всю доступну площу. Також його розмір можна задавати у відсотках в панелі **Image**.



16.

17. Кольорокорекція

Проект в Фігмі →

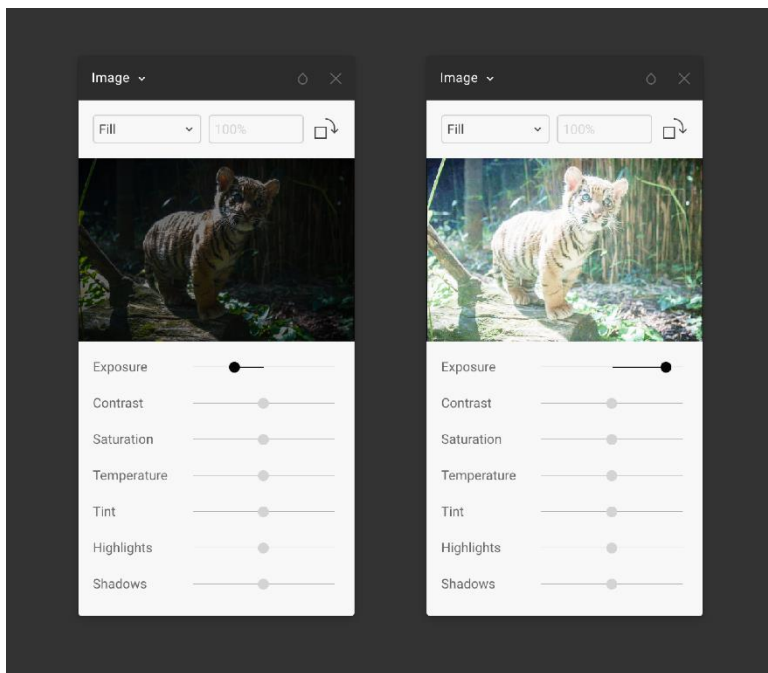
Там тільки скріншоти.

У Фігмі доступні прості, але потужні фільтри для швидкої корекції кольору фото без використання зовнішніх редакторів.

Їх цілком достатньо, щоб зробити ілюстрації і аватари в дизайні яскравішими або надати їм потрібний характер. Всі зміни корекції не змінюють вихідне растрове зображення.

Коригувати кольори можна в панелі **Fill** у тих шарів, де є растрова заливка. Якщо панель стоїть в режимі Image, доступні наступні фейдери:

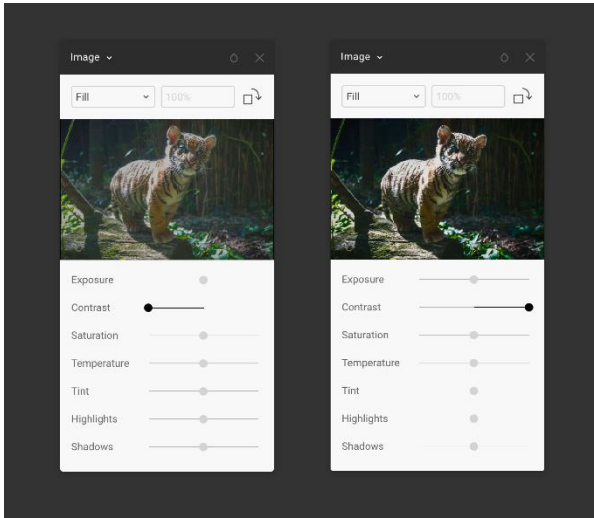
- **Exposure** [експоужа] — експозиція. Дозволяє імітувати різну кількість світла в кадрі. Темніше-яскравіше.
- **Contrast** — м'якше-драматичніше.
- **Saturation** [сетюрейшн] — насиченість кольору. Якщо відвести вліво, знебарвити.
- **Temperature** — температура світла. Синій холодніше, жовтий тепліше.
- **Tint** — відтінок світла, від зеленого до червоного. Стане в нагоді для імітації старих кольорових фото.
- **Highlights** [хайлайтс] — корекція світлих і пересвічених ділянок.
- **Shadows** — корекція темних і тіньових ділянок.



Exposure

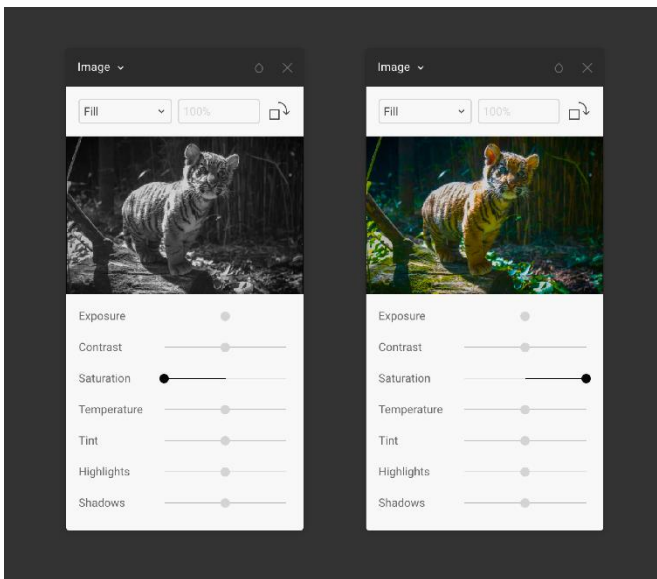
Експозиція - базове налаштування яскравості фотографії. Якщо кадр недодержаний, він темний і має чорні провали. Якщо світло занадто довго потрапляло в об'єктив, кадр занадто світлий і має переосвітлення - аналогічні білі провали. В фото з тигром експозиція була оптимальною, тому її варто було б залишити в нульовому значенні.

Якщо немає мети досягти певного артистичного ефекту, і провалених чорних тіней і пересвітів слід по можливості уникати.



Contrast

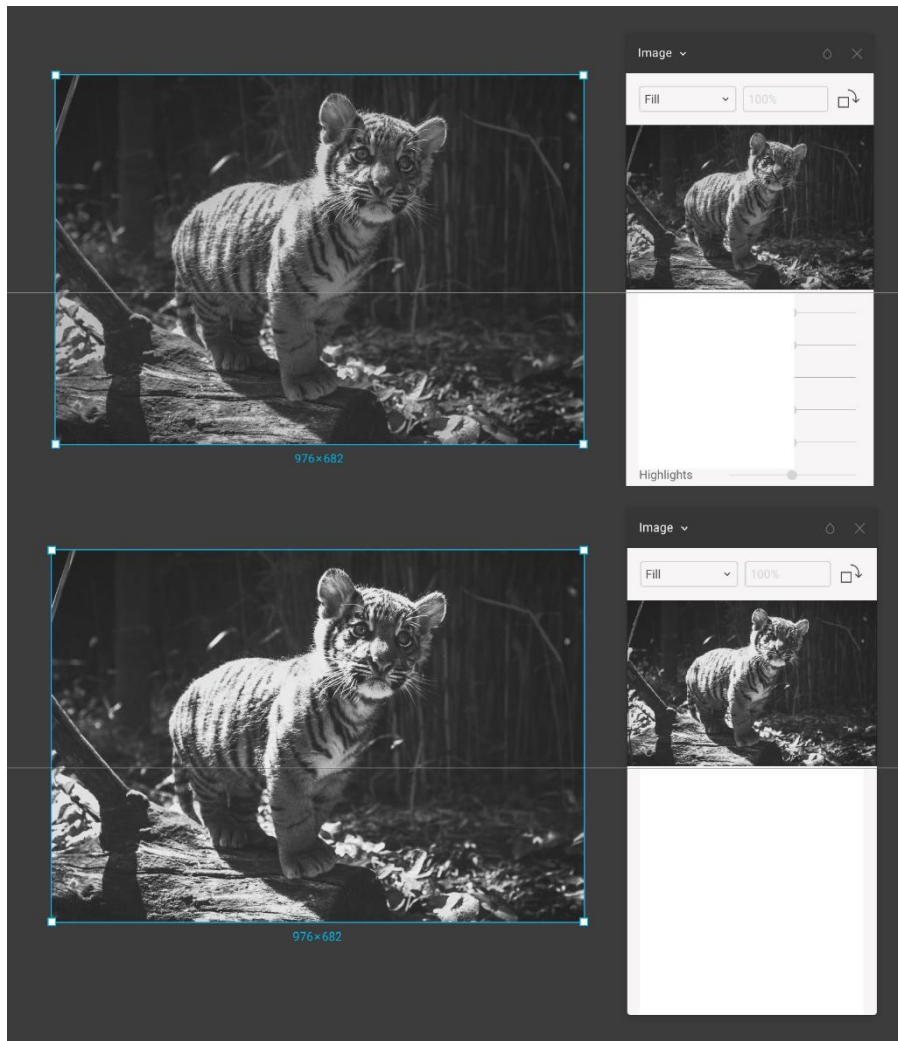
Збільшення контрасту дозволяє робити темні ділянки темніше, а світлі світліше. Зменшення приглушує між ними різницю. Якщо контрасту не вистачає, фотографія виглядає невиразно. Якщо його занадто багато, переходи кольору занадто різкі. На фото праворуч їх видно як чорні і білі плями.

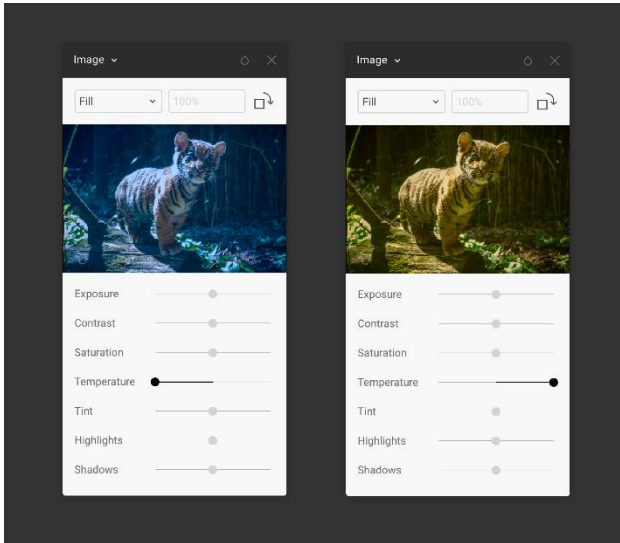


Saturation

Насиченість відтінків, дозволяє зробити кольори більш яскраві. Якщо відвести цю настройку в нуль, ми позбавимо кольору відтінків і таким чином зробимо фото чорно-білим. Максимальне значення насиченості робить кольори неприродно-кислотними.

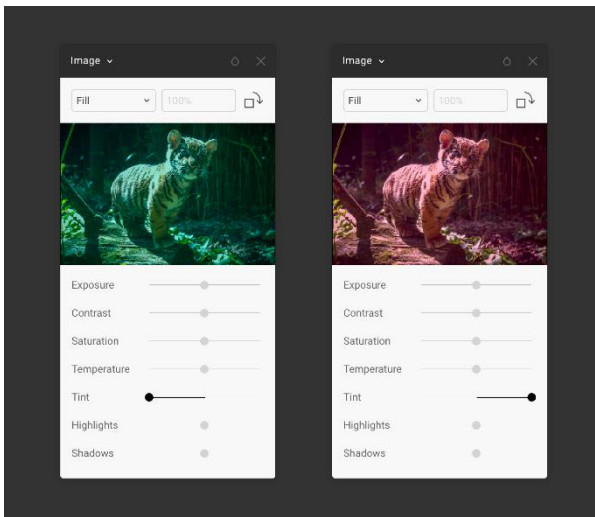
Щоб чорно-біла фотографія виглядала соковито і драматично, відвести контраст в нуль недостатньо. На фото залишиться багато сірого.





Temperature

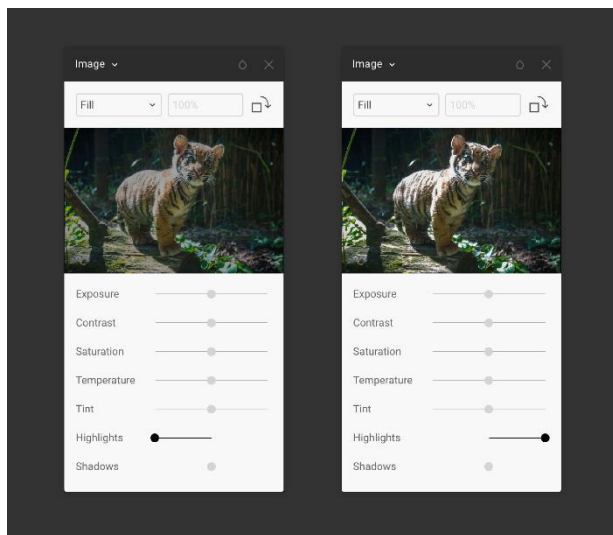
Налаштування температури світла дозволяє працювати з балансом білого. Якщо кадр виглядає занадто теплим або холодним, таке налаштування дозволяє коригувати його, відновлюючи нейтральне висвітлення. Сонячне світло вдень і під вечір має занадто теплий відтінок. Для відновлення нейтрального освітлення він компенсується холодним синім. Лампам люмінесцентного освітлення навпаки бракує температури. Такі фото компенсують жовтим.



Tint

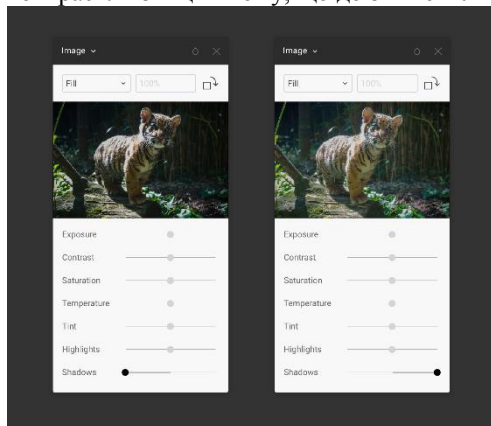
Налаштування відтінку дозволяє домогтися ефекту вицвілого фото. Фотографи використовують її, щоб приглушати зайву червоність осіб, ведучи відтінок трохи вниз.

Корисний також для компенсації балансу білого в складних умовах зйомки, коли не вистачає звичайної корекції температури. Наприклад, коли на об'єкт зйомки впливають сильно освітлені кольорові стіни.



Highlights

Додаткова настройка для корекції світлих і пересвічених ділянок. Її дія схожа на контраст. Різниця в тому, що до змін схильні переважно яскраві ділянки.



Shadows

Аналог **Highlights**, тільки для тінювих ділянок фото. Дозволяє витягнути

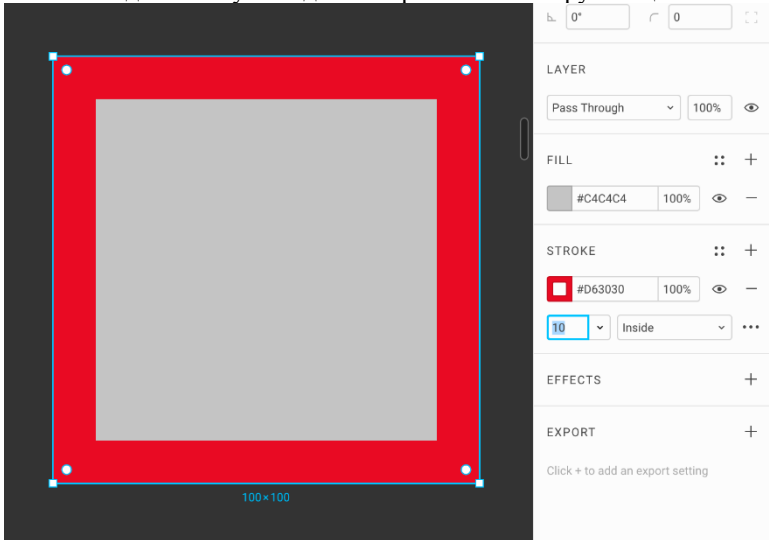
надмірно глибокі тіні.

18. Обведення

Проект в Фігмі →

В цьому розділі поговоримо про додаткові налаштування обведення шейпів, векторних ліній і текстових шарів, а заодно попрактикуємося їх створювати. Також можна порівняти, чим відрізняється реалізація обведення Фігми і Скетча.

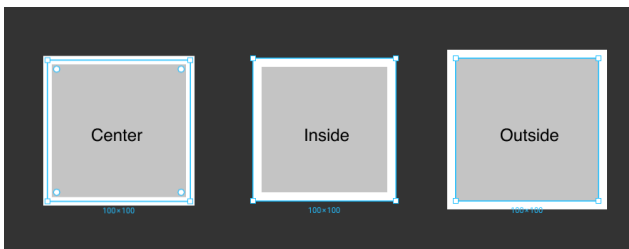
1. Створимо квадрат, **R**.
- 2.
3. Задамо йому обведення червоного кольору товщиною 10 пікселів.

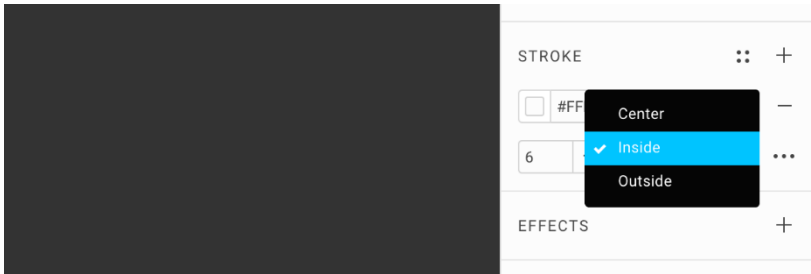


Режими обводки

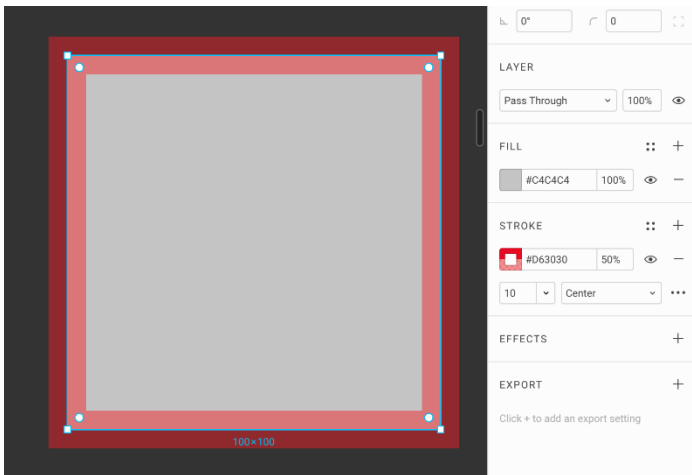
Як і в скетчі, обведення може бути накладене на фігуру трьома способами:

- **Center** — Центр обведення збігається з межами шейпа.
- **Inside** — Усередині. Обведення не виходить за межі шейпа.
- **Outside** — Зовні. Зовнішнє обведення як в полях введення Google Material.

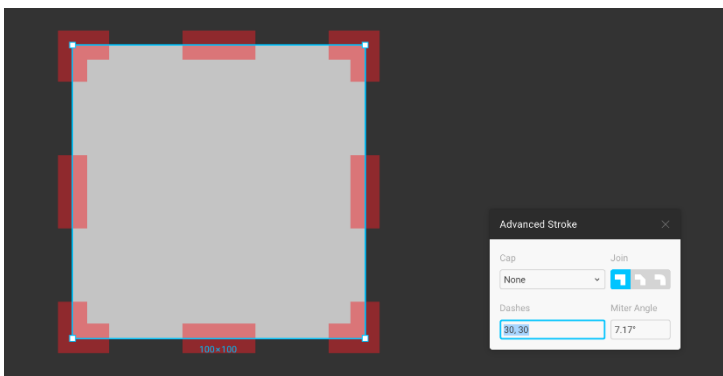




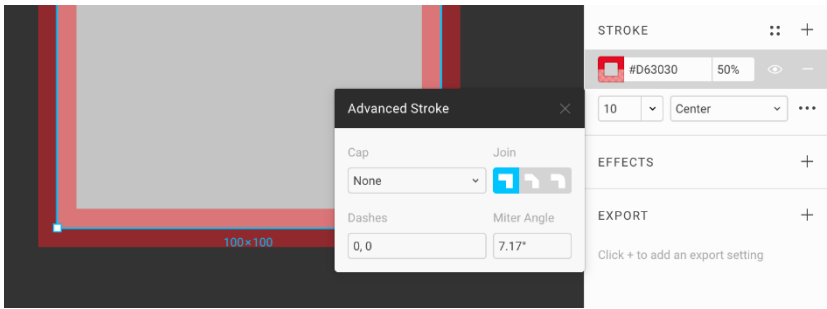
Це налаштується в меню, що випадає в блоці **Stroke**.



4. Поставимо режим обведення з **Inside** в **Center**, а саму її зробимо на 50%. Тепер явно видно, що червона лінія проходить по центру контуру фігури.

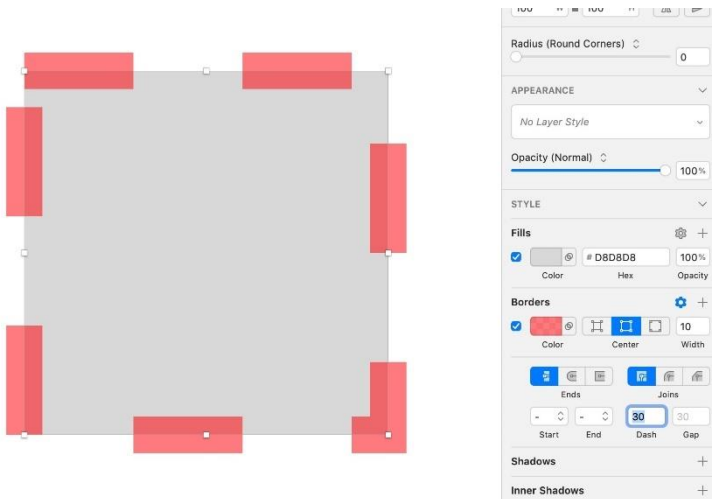


Відмінності пунктирного обведення в Скетчі та Фігмі



У правій частині блоку **Stroke** є меню у вигляді трьох точок. Розгортаємо:

У ньому в поле **Dashes** можна задати, щоб лінія стала пунктирною. Перше значення - довжина лінії в пікселях, друге - відступ в пікселях до наступної. задаємо **30**.



У Скетчі аналогічна фігура при таких налаштуваннях виглядає зовсім інакше:

Різниця в тому, що Скетч механічно відраховує лінію від верхнього лівого кута. Робота з таким пунктиром слабо передбачувана.

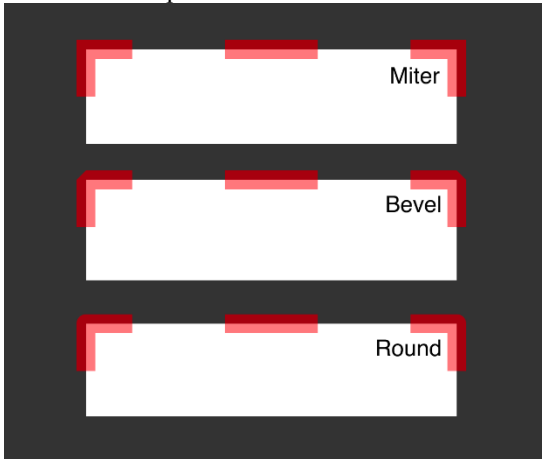
Фігма підлаштовує пунктир таким чином, щоб той був симетричний. Червона лінія обов'язково потрапляє на кути фігури своїм центром. Так пунктир виглядає найбільш естетично. При цьому Фігма нехтує математично-точними значеннями.

Стилі вигинів обведення

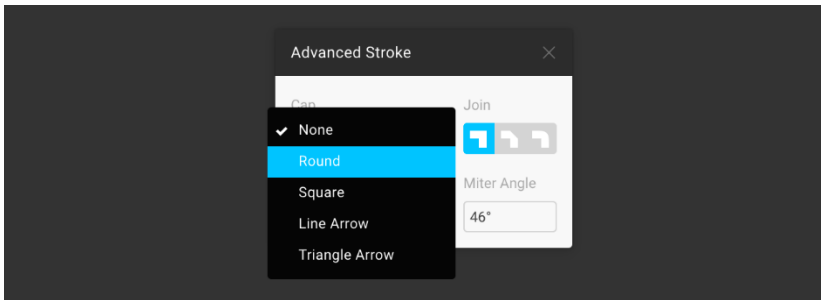
У обведення в квадраті можуть бути три стилі вигинів, які перемикаються в

сегменті **Join**:

- **Miter**—прямий кут, за замовчуванням.
- **Bevel**—зрізаний на 45° .

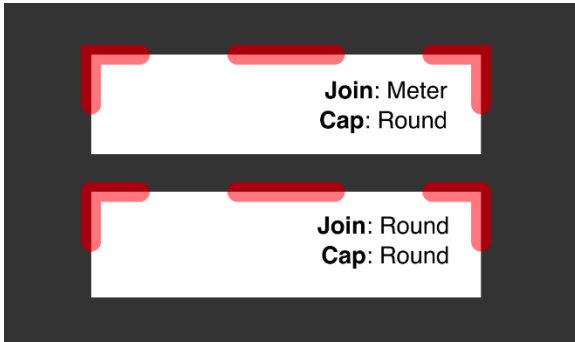


- **Round**—закруглений.

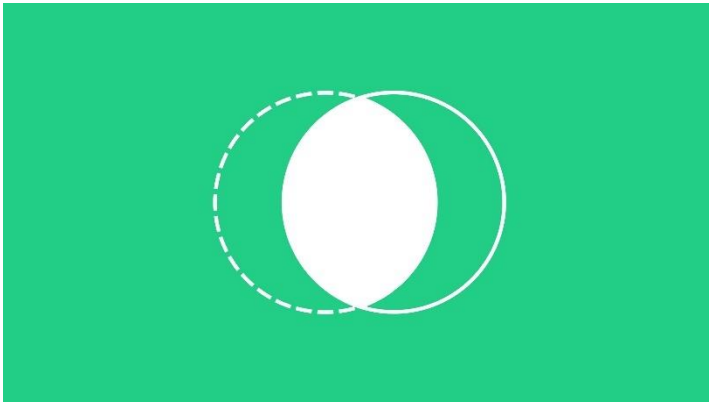


Типи закінчення лінії

Налаштування Cap відповідає за те, як буде оформлено закінчення лінії. Ми говорили про типи закінчень, коли вивчали стрілки в розділі Шейпи.



За замовчуванням **Cap** встановлений в **None** і переривається. До ліній в обведення можна застосувати тип **Round**, щоб округляти пунктир.



18. Маски

Проект в Фігмі →

Маски дозволяють брати одну векторну фігуру і по її формі робити видимої іншу.

Кадрування фото

У Скetchі найчастіше в якості маски використовуються прямокутник або коло, а в якості контенту - фотографії.

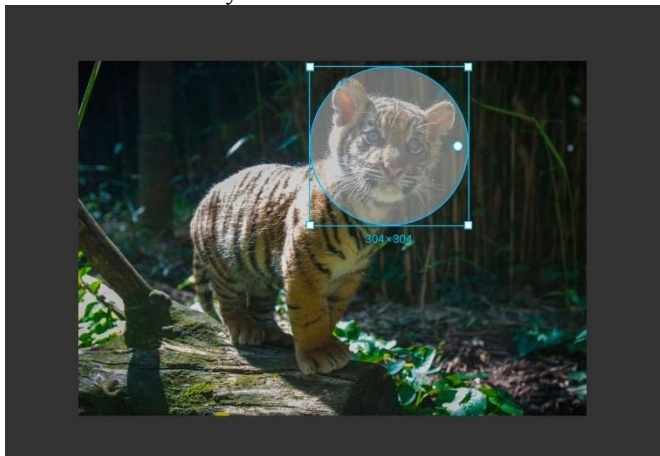
Щоб домогтися контролю над кадруванням без використання масок, потрібно заздалегідь підготувати всі зображення і вставляти їх в заливку в вигляді вже обрізаних бітмапами. Це костильний і негнучкий варіант.

Тому, маски стали у мене основним способом для створення круглих аватарів і інших ілюстрацій в Скetchі.

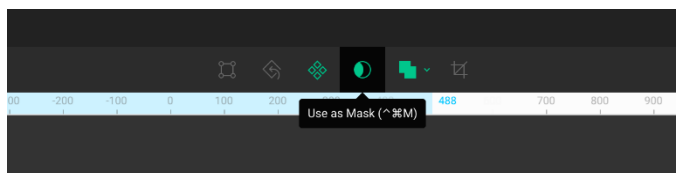
Оскільки в Фігмі є дуже розвинений режим кадрування заливки **Crop**, немає сенсу використовувати маски для кадрування аватарів і ілюстрацій. Однак, для

розуміння принципу їх роботи корисно створити найпростішу маску. Механіка повністю аналогічна Скетчу.

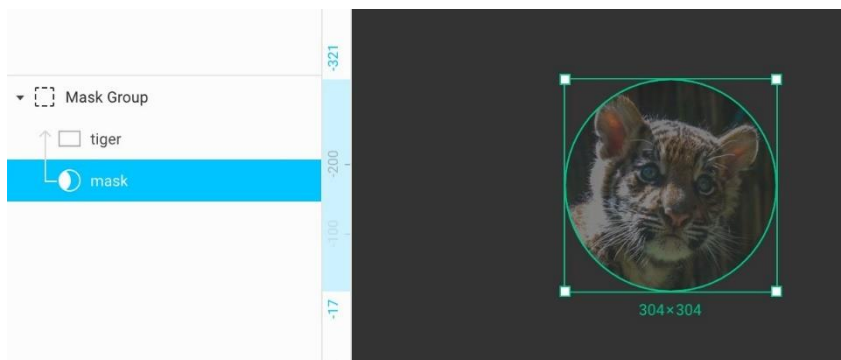
1. Вставимо в робочу область фотографію.
2. Поверх неї розтягнемо коло, **O**. Воно повинно загороджувати те місце в фотографії, яке ми збираємося залишити видимим. Для наочності можна знизити опасіті кола до 50%, **5**.
3. Коло виділено. Поміняємо його місцями з фотографією, **Cmd + [**. Коло повинно встати знизу.



- 4.
5. Затискаємо **Shift** і виділяємо обидва шари.



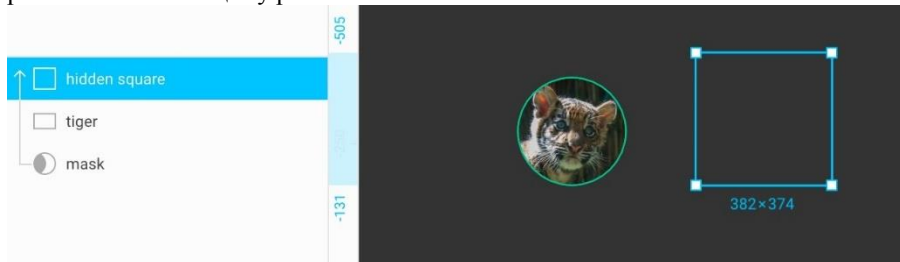
Викликаємо команду **Use as Mask**, **Ctrl + Cmd + M** або натискаємо тулбар:



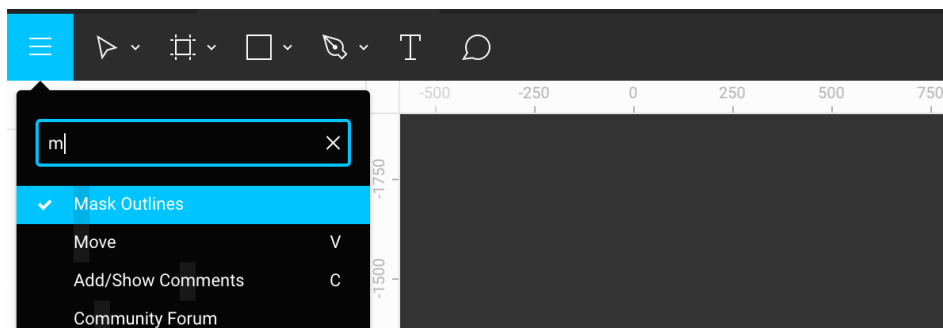
Була створена нова група з назвою **Mask Group**. Шар з колом тепер знаходиться в режимі маски і тому його бокс і контур кодуються зеленим кольором.

Поки шар фотографії знаходиться над маскою, він буде обрізатися по її формі. Оскільки у кола була задана опасіті 50%, фотографія теж її успадковує. Повертаємо опасіті в максимум, **100**.

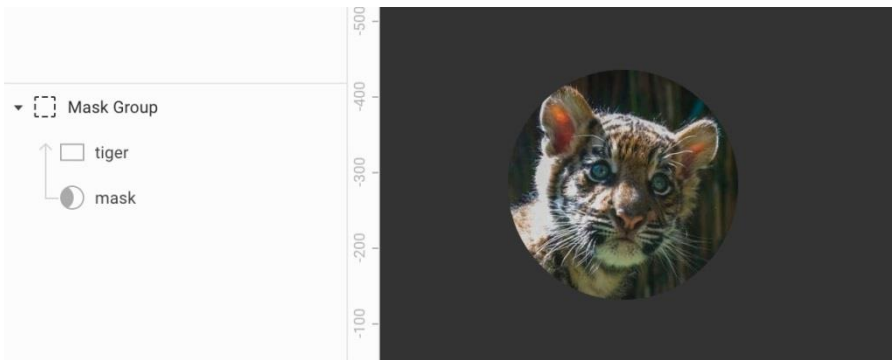
Шари шикуються в ієрархію за допомогою груп. Якщо груп немає, всі верстви розташовані на вищому рівні.



Маска діє тільки в рамках своєї групи, тому, щоб уникнути попадання в неї сторонніх шарів, маску і її контент інкапсулюють в групу. Якщо її розв'язати, **Shift + Cmd + G**, маска зжере всі шари вище себе, доступні на своєму рівні ієрархії.

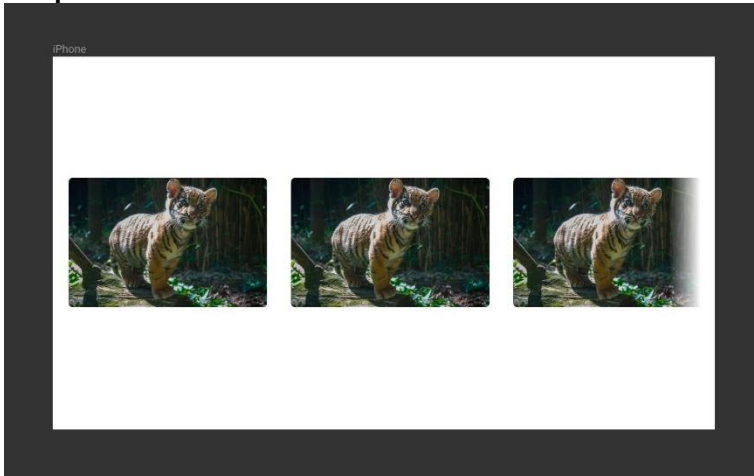


У Фігми є тогл-режим **Mask Outlines**, який показує кордони масок зеленим обведенням. Його можна відключити клавішею **Cmd + /, m** або через меню.



Зелена межа зникла:

Альфа-маска



Такий прийом використовується в дизайні, якщо треба показати, що справа або зліва ще є контент і його можна тягнути вправо-вліво

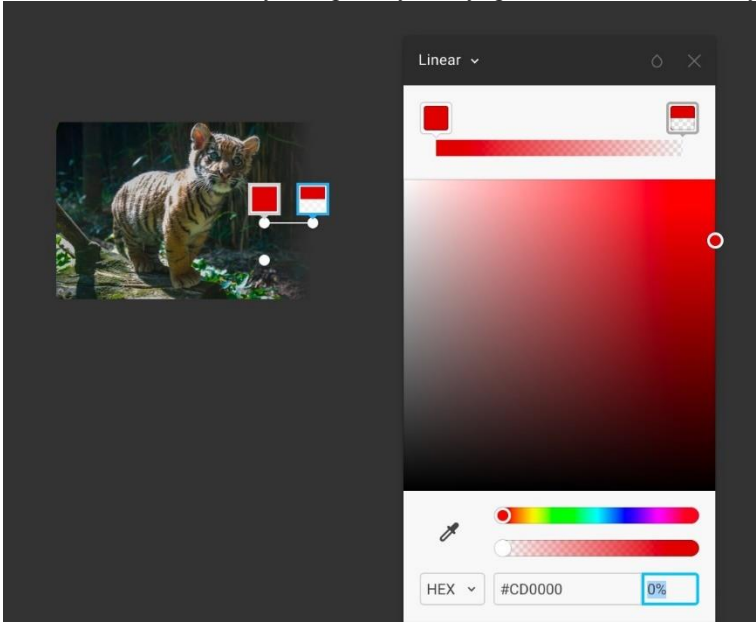
У Скетчі маска може бути в двох режимах: в базовому **Outline Mask** і в **Alpha Mask**. Другий режим дозволяє налаштовувати плавні переходи прозорості контенту. Опасіті шару-маски ні на що не впливає.

У Фігмі такого зайвого поділу немає, оскільки контент маски чутливий до її опасіті. Якщо в масці є градієнт або сама вона прозора, така маска буде відводити в прозорість свій контент без додаткового перемикання режим альфа-маски.

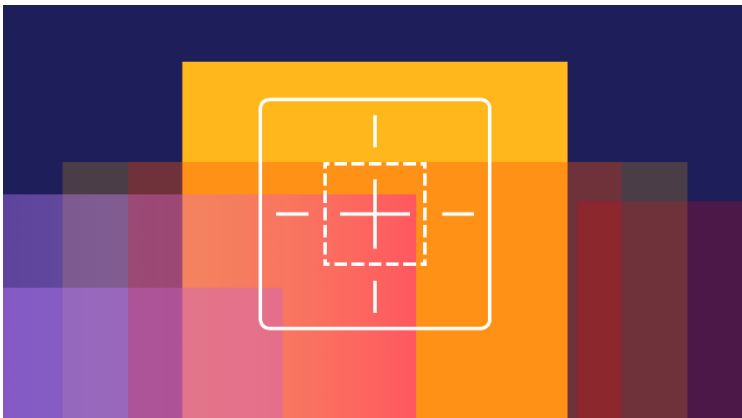
Задача: зробимо альфа-маску.

1. Зробимо звичайну маску по інструкції вище.

2. Виділимо шар маски.
3. Накладемо на нього лінійний градієнт. Колір не важливий, тому що якщо шейп з градієнтом використовується як маска, Фігма буде використовувати його градієнт для роботи з прозорістю контенту. Для наочності вибрав червоний.
4. Кінцеву контрольну точку градієнта відводимо в нуль.



Бачимо, як права частина фотографії плавно зникає.



19. Адаптивність і обмежувачі

Проект в Фігмі →

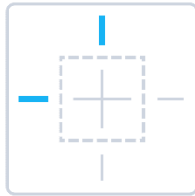
Дизайн для екранів рідко проєктують під один розмір. Щоб полегшити створення макетів в мінливому розмірі фрейму, в Скетчі та Фігмі використовуються обмежувачі.

Це важлива тема, яка стоїть в основі роботи з компонентами і типографікою.

Обмежувачі - це індивідуальні настройки кожного шару в фреймі, які допомагають контролювати, як він буде вести себе при зміні розмірів фрейма.

Ми задаємо поведінку шарів, щоб використовувати їх в різних макетах. Можна налаштувати шапку сайту або програми один раз, а потім перевикористати її компонент на макетах будь-якої ширини від самого маленького телефону з шириною 320 px до гігантського моноблока шириною 1920 px. Кожен шар буде мати набір налаштувань, який визначає, чи буде він спотворюватися, липнути до країв або вставати по центру.

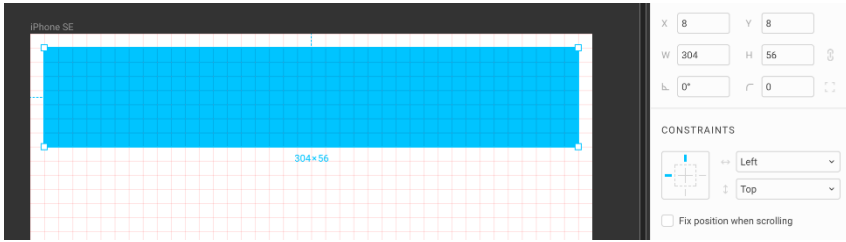
Якби не існувало обмежувачів, довелось робити шапки під кожен розмір екрану, а це неефективно.



Обмежувачі шейпа за замовчуванням

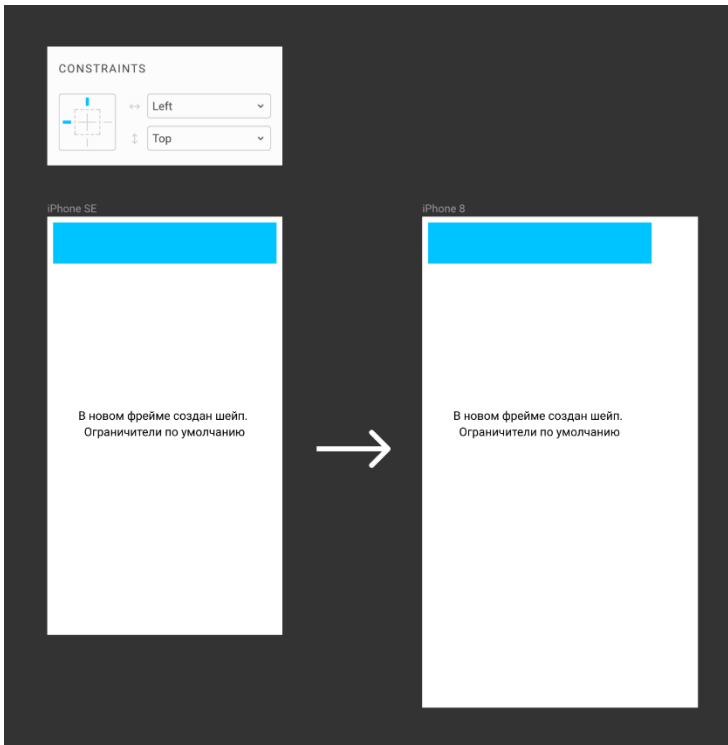
Коли ми створюємо шар в фреймі або переміщуємо існуючий шар в фрейм, у того з'являються два обмежувача: **Left** і **Top**. Їх видно в прицілі в блоці **Constraints**.

1. Створимо фрейм за розміром iPhone SE, **F**. Найменший айфон, який має сенс підтримувати. Він буде розміру 320 x 568.
2. Настроїмо йому сітку 8px в блоці **Layout Grid**. Зазвичай я використовую 4px, але для прикладу зручніше взяти сітку побільше.
3. Розтягніть прямокутник від верхнього лівого краю, зробивши його на відступі 8 з трьох сторін. Його координати повинні бути (8, 8), а розмір 304 x 56.

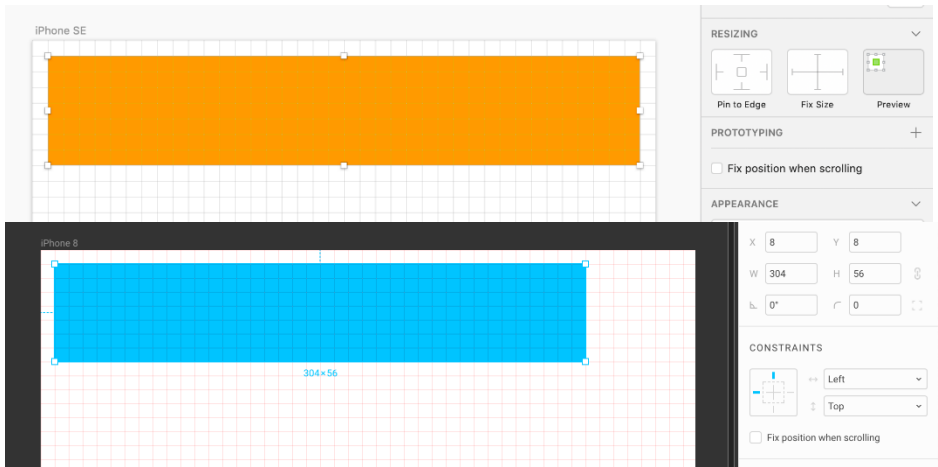


При зміні розміру рамки прямокутник з такими обмежувачами буде зберігати відстань від краю зліва і зверху, а не буде розтягуватися, з'їжджаючи з сітки. Розмір фрейму може змінюватися, але відступ від краю і розмір фігури залишаться фіксованими. Доведемо це на практиці.

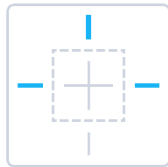
4. Виділимо фрейм і поставимо замість iPhone SE інший розмір: iPhone 8.



Ширина збільшиться до 375. Положення і ширина прямокутника залишаться колишніми.



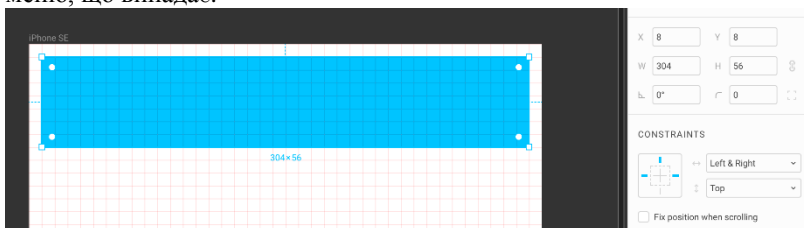
У скетчі така настройка відповідає відсутності обмежувачів і знятої галочки **Adjust content on resize** в налаштуваннях артборда.



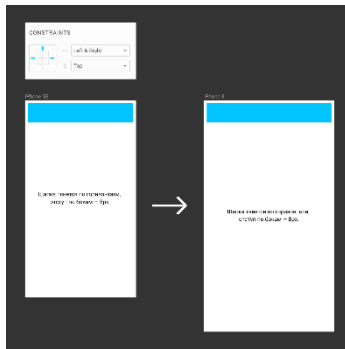
Як зробити гумову шапку

Задача: при зміні ширини блоку шапка повинна розтягуватися по довжині, зберігати відступи від країв кадру і свою висоту.

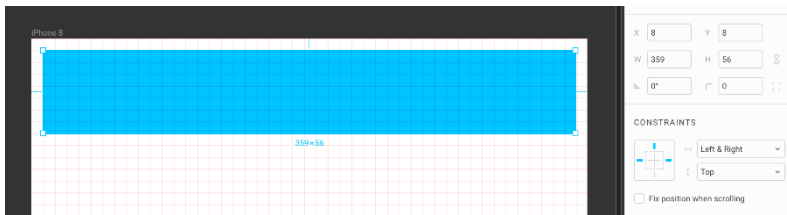
1. Повертаємо розмір фрейма на iPhone SE.
2. Виділяємо прямокутник, затискаємо **Shift** і клікнемо в праву риску, щоб закріпити обмежувачі зліва і справа. Також це можна зробити через меню, що випадає.



Тепер горять три з них, а замість **Left** в меню горизонтальних обмежувачів стало **Left & Right**:



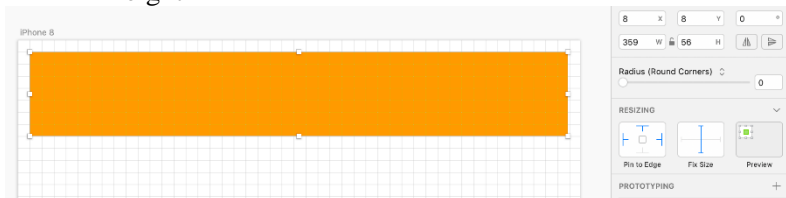
3. Знову ставимо розмір фрейма в iPhone 8, щоб перевірити, як поведе себе блок. Тепер правий відступ став фіксованим. Він дорівнював 8 при ширині фрейму 320 і залишився дорівнює 8 при ширині 375.



Як ми і хотіли, ширина прямокутника збільшилася: була 304, а стала 359. Висота збереглася в 56.

У скетчі цьому налаштуванню відповідає така комбінація обмежувачів:

- Pin to Left
- Pin to Top
- Pin to Right
- Fix Height

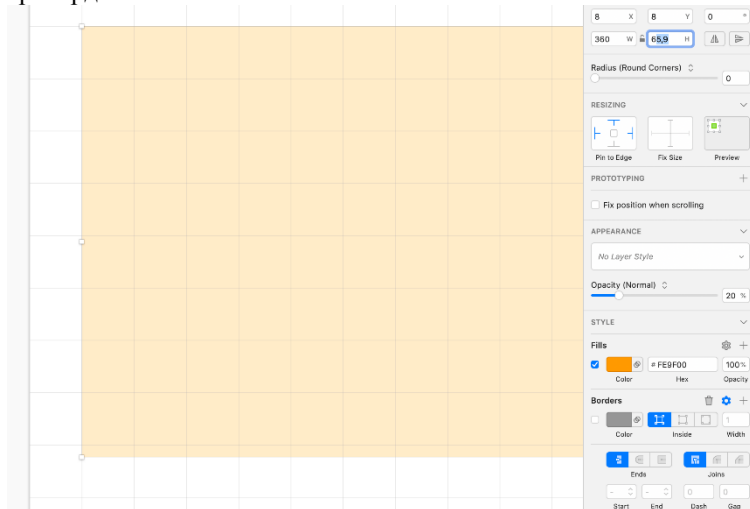


- В налаштуваннях артборда стоїть галочка **Adjust content on resize**.

Галочка **Adjust content on resize** в Скетчі

У вимкненому стані вона дозволяє ігнорувати обмежувачі і розтягувати

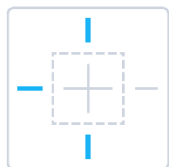
артборд.



Якщо залишити її в налаштуваннях артборда, при перемиканні з iPhone SE на iPhone 8 прямокутник збільшиться пропорційно ширині артборда і з'їде з сітки.

Щоб цього не сталося, перед розтягуванням артборда потрібно додатково зафіксувати його висоту обмежувачем **Fix Height**. Аналогічний обмежувач буває для ширини.

У Фігмі ж ширина і висота спочатку зафіксовані.

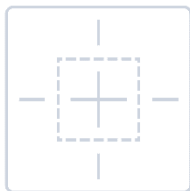


Як зробити сайдбар з гумовою висотою

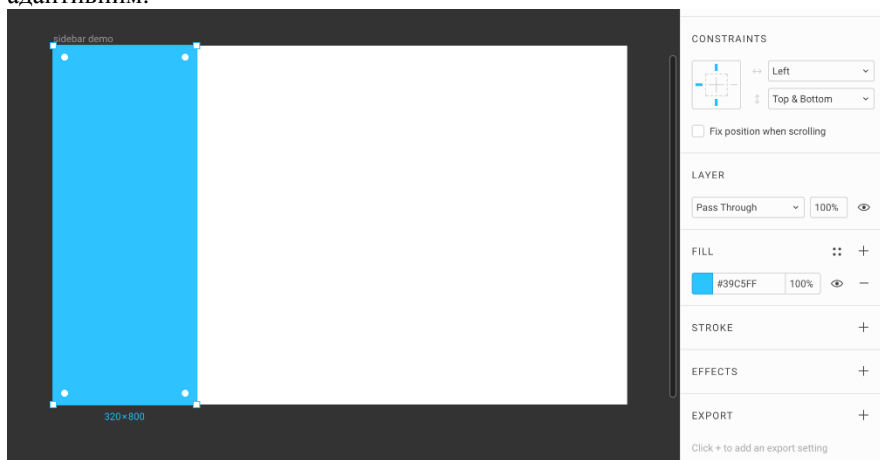
Інший класичний кейс використання обмежувачів - інтерфейс з бічною панеллю - сайдбарі. У дизайні для вебу висота сторінки часто залежить від кількості контенту.

Задача: Прив'язати висоту сайдбара до висоти фрейма, щоб можна було підлаштовувати висоту одним дією.

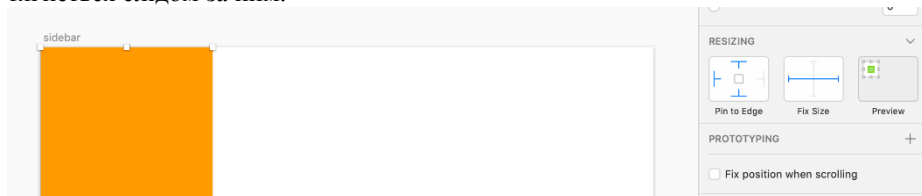
1. Створюємо фрейм розміром 1280 x 800, **F**.
2. Усередині нього розмічаємо сайдбар розміром 320 x 800, **R**. При створенні він отримує обмежувачі за замовчуванням: **Left** і **Top**.



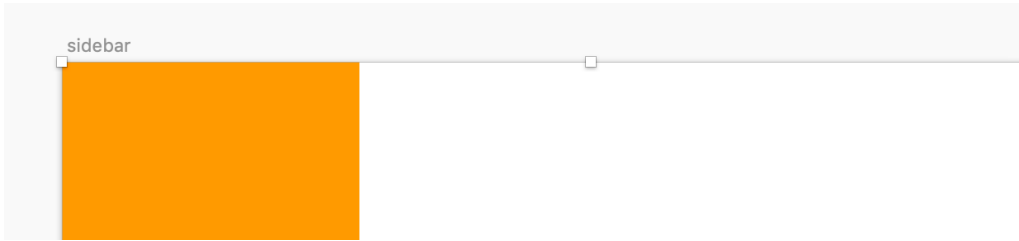
3. Затискаємо **Shift** і додаємо до них ще один обмежувач **Bottom**, клікнувши на нижню вертикальну риску в блоці **Constraints**. Горизонтальний **Left** можна залишити, він стане в нагоді, якщо будемо робити макет адаптивним.



4. Тягнемо фрейм за нижній край, щоб переконатися, що сайдбар тягнеться слідом за ним.



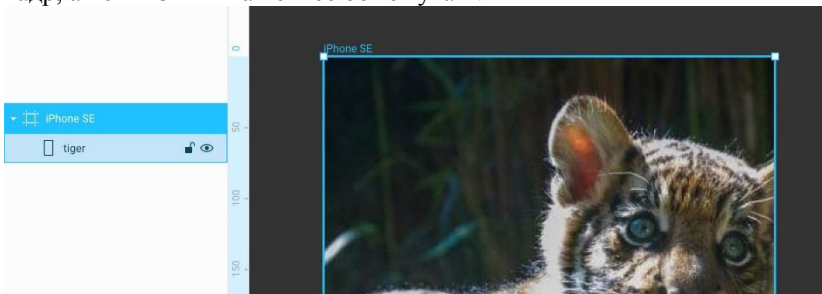
У Скетчі настройка аналогічної поведінки виглядає так. Виділено шейп:



Виділено артборд:

Як розтягнути фонове фото на весь фрейм

В якості повноекранного фону дизайнери часто використовують фотографії. Спочатку на думку спадає ідея залити растровим фоном сам фрейм, але на практиці це виявляється незручно, тому що часто виникає потреба з клавіатури копіювати стиль заливки фону і переносити його в інші фрейми. Набагато легше перетягнути з робочого столу фотографію в вікно Фігми і вписати її в кадр, а потім зняти на неї все обмежувачі.

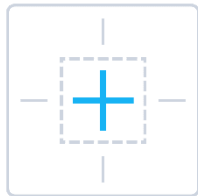


Задача: при зміні розміру рамки фото повинно підлаштовуватися.

1. Створюємо фрейм, **F**. Вибираємо розмір iPhone SE.
2. Перетаскуємо з робочого столу в Фігму файл фотографії в форматі JPG або PNG.
3. Вкладаємо шар зображення в фрейм, перетягнувши їх один на одного в панелі шарів:
4. Підлаштовуємо розмір фотографії під розмір фрейму, хапаючи за квадратні ручки в кутах. Координати фотографії — **0, 0**, розмір: 320 x 568.
5. Затискаємо **Shift** і знімаємо обмежувачі за замовчуванням: **Left** і **Top**. Праворуч від прицілу в меню горизонтальних і вертикальних обмежувачів написано **Scale**. В цьому режимі фото пропорційно тягнеться з двох сторін.
- 6.



Міняємо розмір артборда на iPhone 8 і переконуємося, що фото розтягнулося.



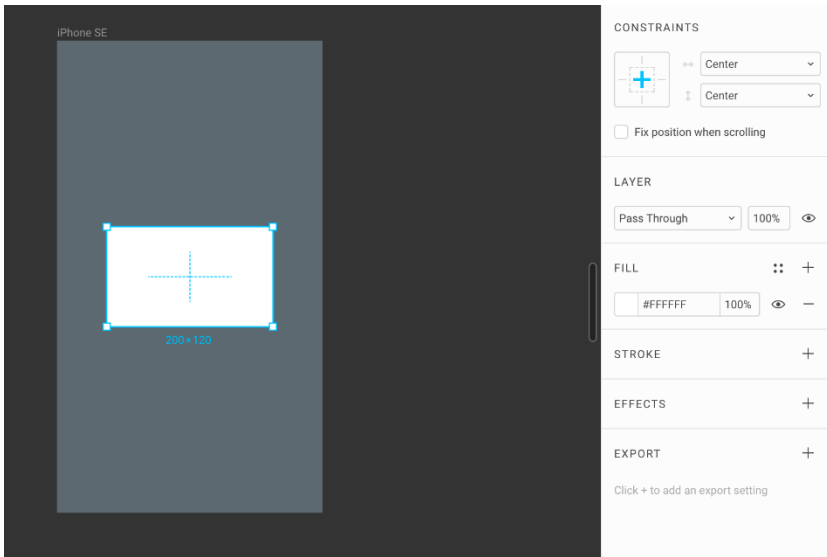
Як зафіксувати модальне вікно по центру екрана

Функції вирівнювання **Horizontal Center, Opt + H** і **Vertical Center, Opt + V** дозволяють рухати шар на центр його фрейму. Якщо той зміниться, шар залишиться на тих же координатах.

Якщо ми заздалегідь знаємо, що шар завжди повинен бути строго по вертикальному або горизонтальному центру, незалежно від розміру рамки, можна використовувати обмежувачі і прикріпити шар до центру.

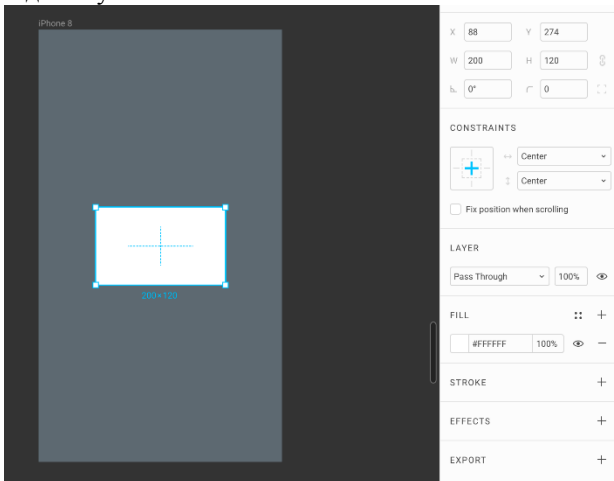
Задача: модальне вікно повинно бути по центру

1. Створюємо фрейм розміру **iPhone SE, F**.
2. Малюємо прямокутник розміром 200 x 120, **R**.
3. Ставимо його по центру фрейму, **Opt + H, V**. Натискаємо **V** після **H**, не відпускаючи **Opt**.
4. Коли затиснутий **Opt**, ми можемо дивитися відстані від прямокутника до країв фрейму.
5. Налаштовуємо приціл в блоці **Constraints** так щоб в ньому був блакитний плюс, клікнувши на кожен з перекладин. Або виберемо з меню значення **Center, Center**.

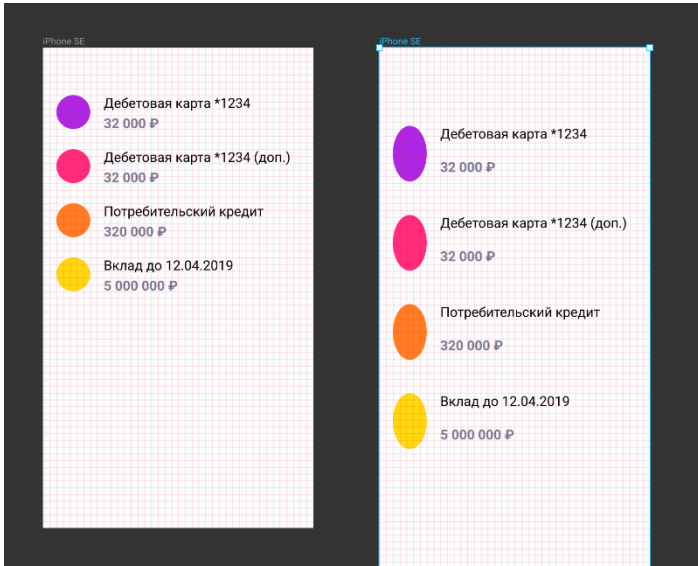


Звернемо увагу, що усередині прямокутника з'явився пунктирний хрест, ще одне підтвердження того, що ми в режимі **Center, Center**.

6. Змінимо розмір фрейму з **iPhone SE** на **iPhone 8**. Прямокутник підлаштує своє положення:



Закріплений шар можна зрушувати щодо центру. При зміні фрейма той збереже свою відстань зсуву в пікселях. Якщо шар зрушили наверх на 10 пікселів і змінили фрейм, в новому фреймі ця відстань не спотвориться.



Як зробити так, щоб при розтягуванні макета все не їхало

Перед початком розтягування затиснути **Cmd** на Mac або **Ctrl** на Windows.

Вимкнути всі обмежувачі галочкою як в Скетчі можна. Однак, за допомогою цієї клавіші можна тимчасово їх ігнорувати. Це зручно, якщо в макеті багато шарів, обмежувачі не налаштовані і прописувати правила поведінки занадто довго.

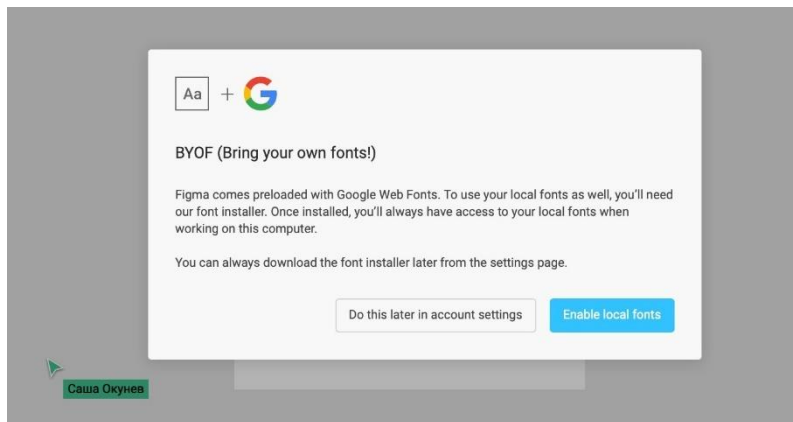


Проект в Фігмі →

80% дизайну інтерфейсів - це робота з типографікою.

У Фігмі є хороші можливості для роботи з текстом через спеціальний тип шарів.

Щоб створити текстовий шар, натискаємо **T** і натискаємо в потрібне місце, або розтягуємо блок як прямокутник або фрейм.



Шрифти із GoogleWebFonts

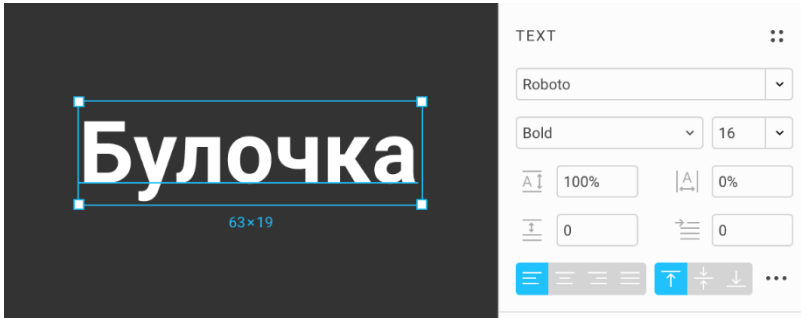
Хороша новина: в Фігмі спочатку доступна величезна бібліотека шрифтів Google Fonts, яку зручно використовувати в веб-розробці. Багато з доступних шрифтів підтримують кирилицю. Якщо обмежений бюджет проекту, рекомендую використовувати в дизайні ті шрифти, які доступні в цій платформі.

Повторю першу главу: щоб локальні шрифти були доступні в браузері, потрібно встановити десктопний клієнт, або запустити Font Helper.

У Скетчі текстовий шар може бути в двох режимах:

Auto і **Fixed**. У Фігмі таких режимів три:

- **Width**—повний аналог режиму **Auto**
- **Height**—немає в Скетчі
- **Fixed**—в Скетчі називається і працює так само.



Створимо текстовий шар. Якщо виділений текстовий шар, праворуч ми побачимо панель роботи з текстом.

Справа вгорі блоку іконка **Text Styles** - спливаюче меню стилів. Далі йдуть поля:

- **Typeface**
- **Weight**— жирний **Cmd + B** і курсив **Cmd + I**
- **Size**
- **Line Height**— висота рядків
- **Letter Spacing**— трекінг
- **Paragraph Spacing** — відступ між параграфами, розділеними обривом рядки, **Enter**
- **Paragraph Indentation**— червоний рядок

Далі йде настройка вирівнювання по вертикалі і горизонталі:



Opt + Cmd + L

Align Center **Opt + Cmd + R**

Align Justify

Text Align Left **Opt + Cmd + T**

Text Align Right **Opt + Cmd + J**

Text

Text

Через брак нормальних клавіш для вертикального вирівнювання, можемо знаходити команди через пошук:

Cmd + /, te to

Text Align Top **Cmd + /, te mi Text**

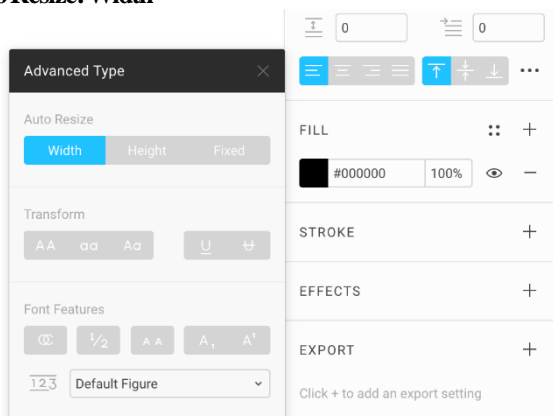
Align

Middle Cmd + /, **te bo**

Text Align **Bottom**

Навпроти вирівнювання - меню **Advanced Type** (три крапки).

Auto Resize: Width



Якщо текстовий шар був створений кліком, він має настройку **Auto Resize** в значенні **Width**, і його ширина підлаштовується під контент. Довгий текст розв'яжеться в один рядок.

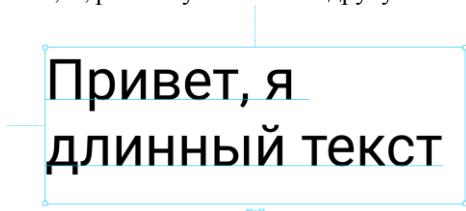
Якщо такий рядок потягнути за правий край, настройка **Auto Resize** переключиться в режим **Fixed**. Це означає, що текстовий блок буде мати фіксовану ширину.

Якщо шар спочатку створювався перетягуванням, текстовий блок відразу матиме фіксовану ширину. Такі шари зручно використовувати в компонентах, де ми заздалегідь не знаємо ширину контенту і задаємо максимально допустиму ширину.

Auto Resize: Height

Третій тип авторесайза, якого немає в Скетчі. Дозволяє підлаштовувати висоту текстового блоку під його контент.

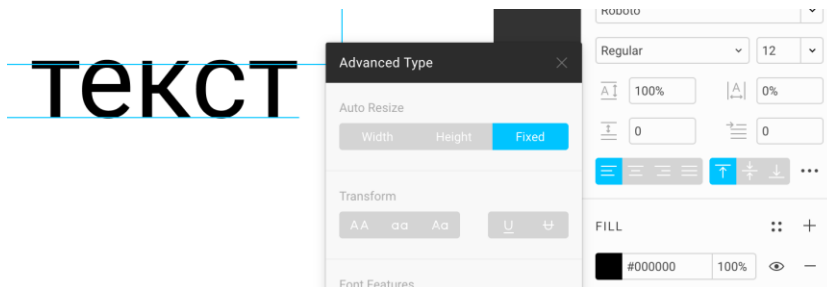
Припустимо, ми створили текст з фіксованою шириною і висотою. Вибрали **Text Tool**, **T**, розтягнули бокс і надрукували:



Привет, я
слишком
длинный текст

Все добре, поки рядки вміщаються в бокс. Але якщо текст стає занадто довгим, він перестає уміщатися в висоту блоку і некрасиво звисає:

Щоб цього не відбувалося, потрібно перемикнути режим текстового блоку в **Height**. Щоб зробити це, заходимо в меню **Advanced Type**, натиснувши на три точки в блоці **Text**.

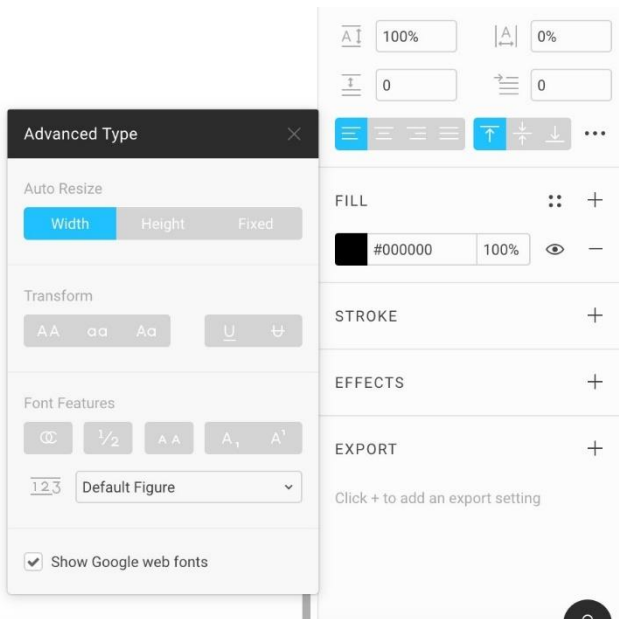


Після цього бокс буде підлаштовуватися під розмір тексту по висоті.

Також в панелі **Advanced Type** можна налаштувати додаткові властивості тексту:

Transform—трансформація тексту в заголовні або рядкові.

Underlineи**Strikethrough**[страйктуру]—підкреслений / закреслений.



Блок Font Features

Містить додаткові можливості шрифту. Іншими словами, ніколи не використовуються нетрі.

Discretionary and historical ligatures — використовувати чи дискретні і історичні лігатури. Про можливості OpenType шрифтів є жирна стаття на ilovetypography.com (англ). Мало які шрифти підтримують цю функцію.

Fractions—дроби. З'єднувати такі значення як 1/2 в один гліф з дробом.


21. Вирівнювання та розподіл

Проект в Фігмі →

Ця глава про базові способи вирівнювання шарів в макетах і про розумний розподіл карток в просторі.

Align: Вирівнювання


Щоб вирівнювати по кадру, виділяємо квадрат кліком.

 **Opt + A** **Align Left.** По лівому краю.

Також працює **Ctrl + Cmd + ←**.

 **Opt + H** **Align Horizontal Centers.**

По горизонтальному центру.

 **Opt + D** **Align Right.** По правому.

Також працює **Ctrl + Cmd + →**.



Opt + W

Align Top. По верхньому.

Також працює **Ctrl + Cmd + ↑**.



Opt + V

Align Vertical Centers.

По вертикальному центру.



Opt + S

Align Bottom. По нижньому.

Також працює **Ctrl + Cmd + ↓**.

Корисно запам'ятати комбіновану клавішу **Opt + H, + V**, яка дозволяє вирівнювати шар по центру фрейму.

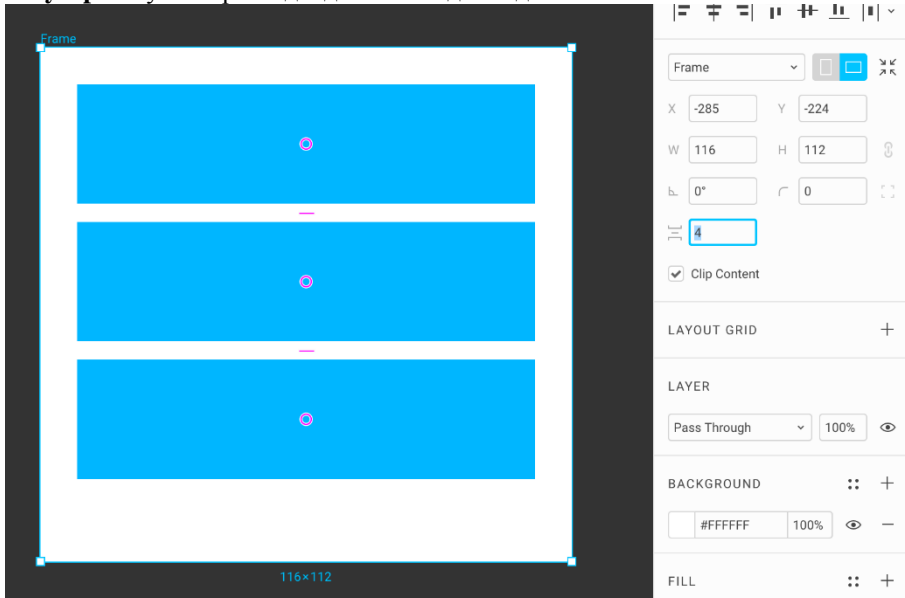
Distribute: Розподіл

Фігма вводить новий стандарт роботи зі списками карток і таблиць. У Скетчі немає нічого подібного. Там вирівнювати картки можна плагінами, але на них покладатися ризиковано, оскільки кожну нову версію Скетча вони ламаються. Щоб розподіляти, виділяємо кілька фігур, мінімум 3.



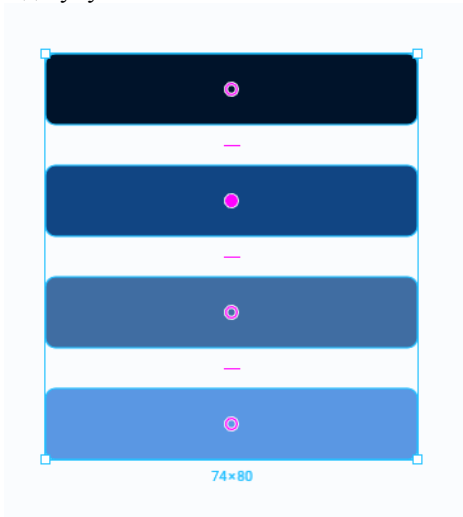
Ctrl + Opt + T

Tidy Up. Розумний розподіл дозволяє задати один і той же



відстань між фігурами. Наприклад, якщо в дизайні використовуються картки в стовпчик, їм можна задати однаковий відступ, який зручно змінювати в поле

відступу.



Ctrl + Opt + V

Distribute Vertical Spacing

Вирівнює по верхньому об'єкту, задає рівні вертикальні відступи в стопці шарів. Якщо застосувати його, у вкладці **Design** з'являється додаткове поле, куди можна вписати бажаний відступ.

Shift + ↑ / ↓ якщо виділено поле відступу, дозволяє змінювати розмір відступу по всій виділеній групі об'єктів зі збільшеним кроком.

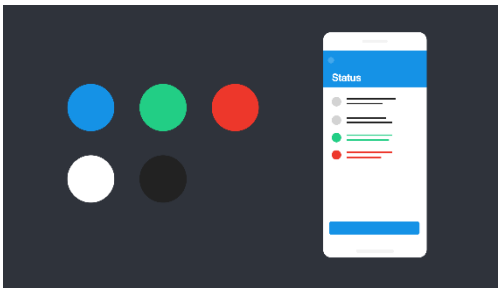
Фіолетові ручки в центрі дозволяють сортувати послідовність блоків.



Ctrl + Opt + H

Distribute Horizontal Spacing

Працює аналогічно **Vertical Spacing**.



22. Стилі кольорів

Проект в Фігмі →

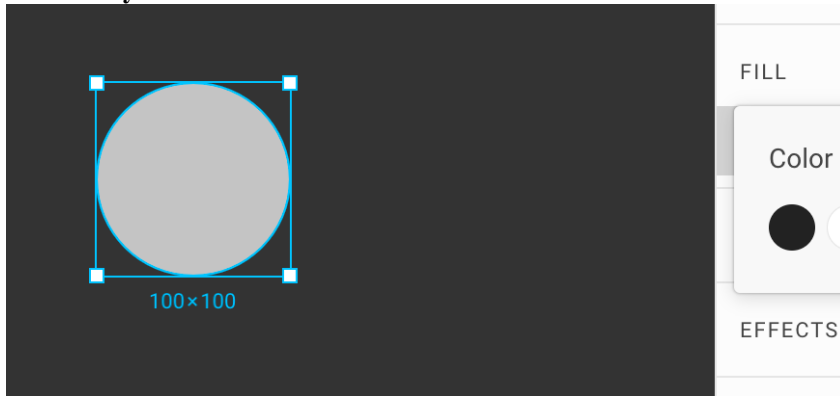
Ми використовуємо функцію **Color Styles**, коли потрібно домогтися єдності в

кольорах по всьому проекту. Вони дозволяють задавати основну палітру проекту і призначати іменовані кольори, які можна використовувати для заливки, в обведеннях і тексті.

Згодом колір в стилі можна замінювати по всьому проекту, редагуючи стиль.

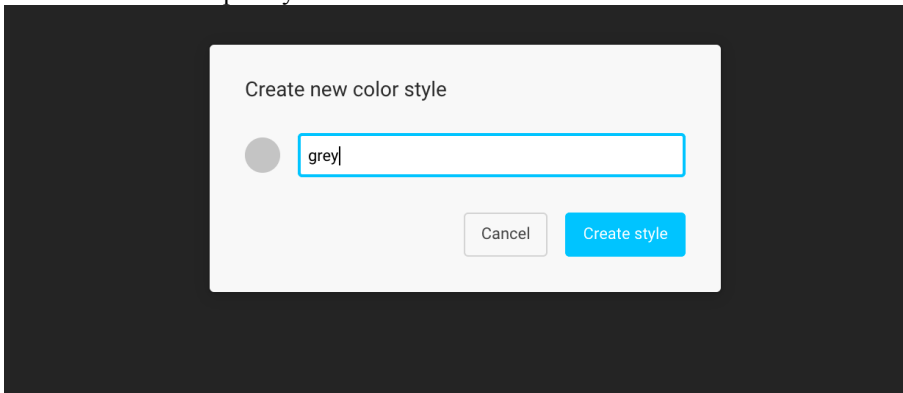
Задача: створимо просту палітру проекту і підв'яжемо стилі до шарів.

1. Створюємо коло, **O**. Він буде пробником, в якому будемо підбирати колір.
2. Вибираємо потрібний колір або залишаємо сірий. Він буде тим кольором, який ми пропишемо в стиль.
3. У блоці **Fill** знаходимо іконку с чотирма точками. Клік по ній розкриває панель **Color Styles**.

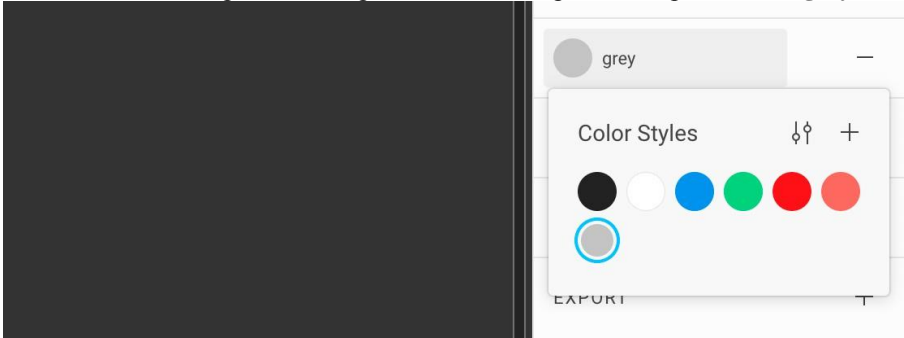


4. Нажимаємо кнопку +, **Create Style**.

5. Заповнюємо ім'я кольору, бажано латиницею і без пробілів. Замість них можна використовувати - чи camelCase. Це потрібно, тому що згодом твої розробники будуть так називати змінні і було б круто, якби стилі кольорів і назви змінних синхронізувалися.

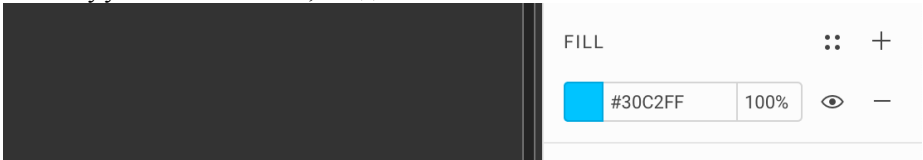


6. Стиль створений. Тепер коло залитий сірим кольором стилю **grey**.



Копіюємо стилі з одного об'єкта на інший

7. Створюємо ще один круг, **O**. Міняємо йому колір, щоб не був сірий. Спочатку у нього немає стилю, а є однотонна заливка.



8. Виділяємо перше коло, копіюємо його стиль командою **Copy Properties, Opt + Cmd + C**.

9. Виділяємо друге коло, вставляємо скопійований стиль другого командою **Paste Properties, Opt + Cmd + V**. Він стає сірим, а блок **Fill** в панелі властивостей пропадає. Тепер до другого шару підв'язані стиль **grey**.



Подібності та відмінності стилів в Фігмі і Скетчі

Реалізація стилів в Фігмі відрізняється від Скетча в кращу сторону. Порівняємо їх.

Скетч: стиль зберігає всі настройки шейпа

У Скетчі палітру можна контролювати через стилі шейпів - **Layer Styles**. У стиль потрапляють всі його налаштування:

- Колір і товщина обведення
- Колір заливки
- Налаштування тіней
- Опасіті

Якщо обводок або заливок кілька, вони теж потраплять в стиль.

Коли це незручно: Наприклад, виникає потреба зробити певний шейп

напівпрозорим. Зв'язок між вихідним стилем і видозміненим розривається. Налаштування шару більше не відповідають вихідному стилю, про що сигналізує зірочка після його назви в блоці **Appearance**.

При зміні вихідного стилю налаштована напівпрозорість злетить і це буде важко відстежити.

Щоб не наступити на ці граблі, доводиться робити два незалежних стиля, навіть якщо обидва шейпи повинні бути одного кольору і різниця тільки в значенні опасіті.

З одного боку, підхід Скетча дозволяє тонко контролювати всі властивості шейпів в проекті. З іншого, це нікому не потрібно.

Ціна порядку занадто висока. Така строгість призводить до того, що стилі множаться з гнітючою швидкістю, примхливі в налаштуванні, їх складно утримувати в порядку, тому що вікно роботи зі стилями **Organize Layer Styles** не дає ніякої інформації про їх налаштування. До версії Скетча 53 я взагалі не використовував стилі шейпів, тому що підтримувати вид шарів в актуальному стані було дешевше через символи з прямокутниками-кольорами.

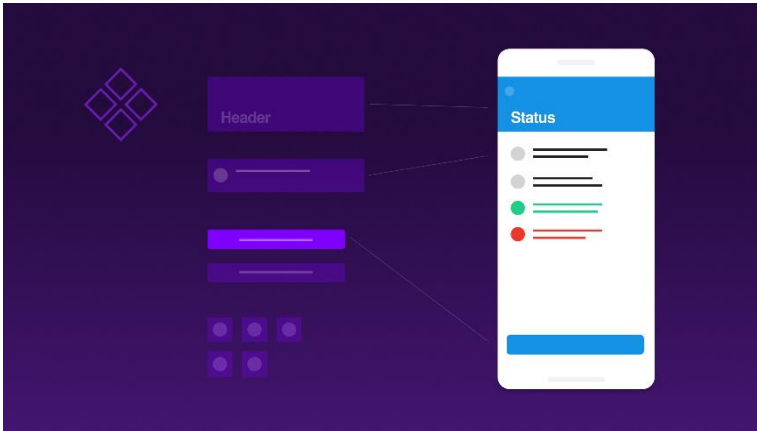
Фігма: стиль - це універсальна настройка кольору, застосовна до будь-яких типів шарів: шейпам, текстам, векторам і фреймам.

У стилі зберігається тільки колір. Це означає, його можна застосувати до заливки, обводки і на текстових шарах. Такий підхід дозволяє контролювати колір в проекті незалежно від того, де він використовувався: в заливці блоку, лініях або в текстах.

Якщо в Фігмі потрібно зробити шейп напівпрозорим, це не порушить його зв'язок зі стилем.

Кольори можна називати і коментувати

Навіть якщо два стилі містять дуже схожі кольори, можна уникнути плутанини, даючи їм унікальні назви. Прийнято використовувати латиницю без пробілів: **green, blue, grey-dark**. Такі назви розробникам буде простіше заносити в змінні. При наведенні на зразок кольору його назва спливає в контекстній підказці.



23. Компоненти

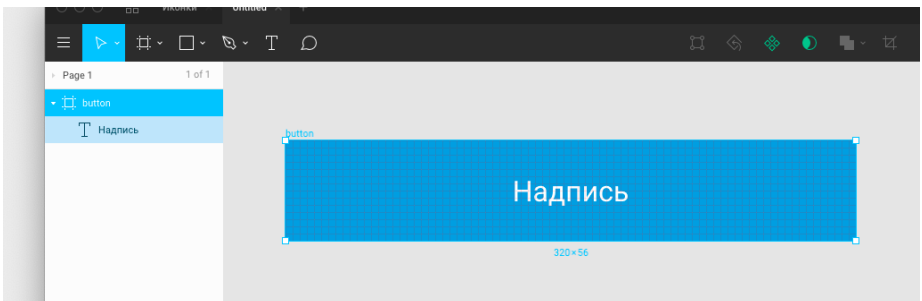
Проект в Фігмі →

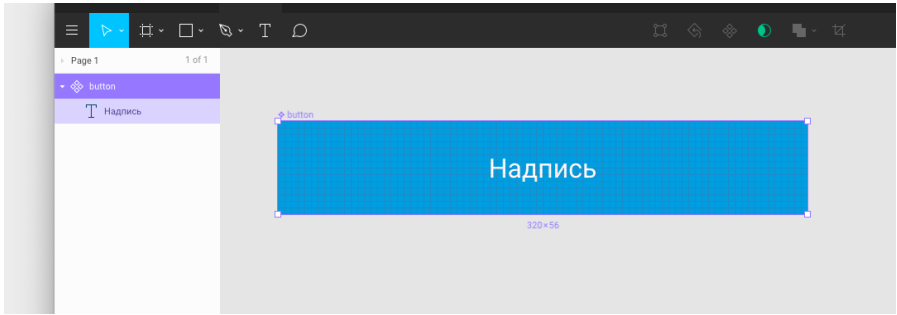
Найважливіша тема, яка лежить в основі дизайн-систематизації і дозволяє робити складні інтерфейси, які реально підтримувати в актуальному стані. Компоненти в Фігмі - аналог символів в Скетчі і класів в програмуванні.

Компоненти можна створити в одному місці, а потім використовувати в інших. В основі будь-якого компонента є фрейм. Його вміст служить основою подальших копій, називається **майстром** або батьком. Всі копії з цього шару - **екземплярами** або дітьми.

Зробимо компонент кнопки

1. Зробимо прямокутник і текстовий шар всередині. Щоб створити компонент, виділяємо об'єкт і натискаємо **Opt + Cmd + K**.





Мнемоніка: Компонент. Фрейм стає фіолетовим і перетворюється в майстер-компонент:

Запам'ятаємо: фіолетовий в Фігмі - колір компонентів.

Трохи про клавіші

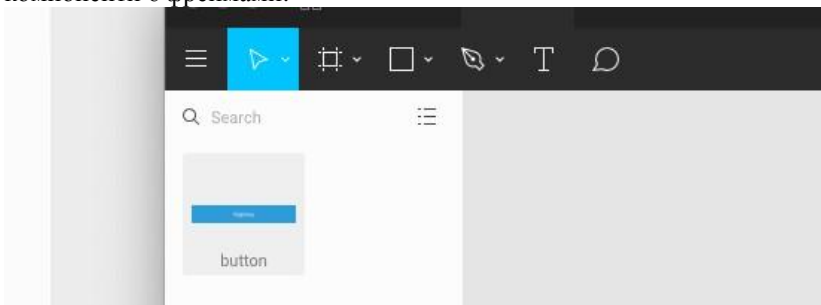
Створення компонентів - одна з команд, яку зазвичай дають з клавіатури, тому **Opt + Cmd + K** важливо запам'ятати.

У скетчі для створення символу я призначав клавішу **Cmd + Y**. Мнемоніка: **Symbol**. Але в Фігмі на цю клавішу вже призначено іншу корисну дію і ускладнювати не хочеться. Які поєднання використовувати - особиста справа кожного. Буває, що дизайнер змушений працювати в обох редакторах, постійно перемикаючись між ними.

В цьому випадку я б призначив на одну дію одну і ту ж клавішу. Тоді вийде працювати з редактором на моторному рівні, не замислюючись, який з них відкрито в даний момент.

Порівняння символів Скетча і компонентів Фігми

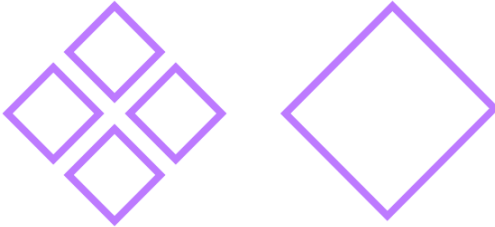
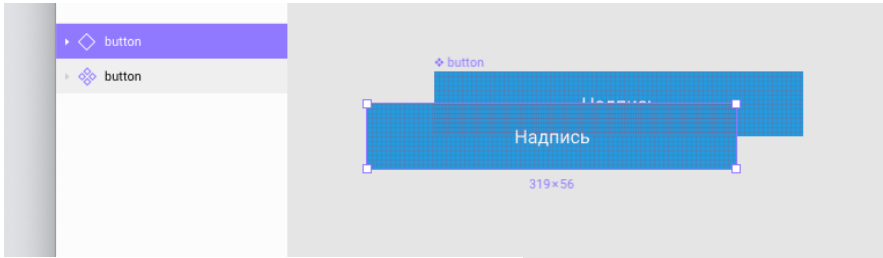
У Скетчі майстер-символи є артбордами. За аналогією, в Фігмі майстер-компоненти є фреймами.



Тепер створений раніше компонент **button** доступний у вкладці **Components**, **Opt + 2**. З неї екземпляри компонентів можна переносити на полотно.

Також можна виділяти екземпляри з майстра, затиснувши **Opt** і перетягуючи

ЛІВОЮ.



Майстер Екземпляр

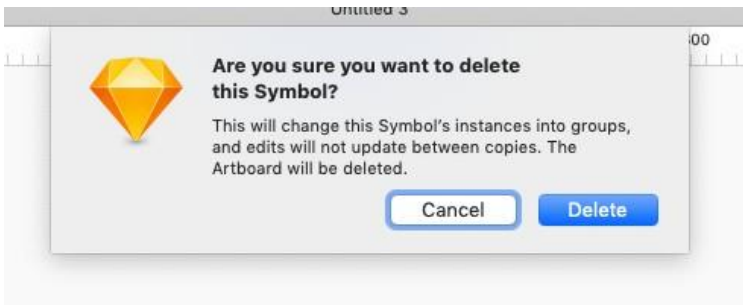
Майстри: схожість

Майстер-компонент і майстер-артборд - прототип об'єкта, з якого відбуваються екземпляри. Їм можна перезаписувати властивості, стилі і контент.

Майстри: відмінності

Скетч: коли об'єкт стає символом, майстер-артборд створюється на сторінці **Symbols**. Якщо в процесі створення символу зняли галочку **Send to Symbols page**, символ з'являється в неконтрольованому місці тієї ж сторінки.

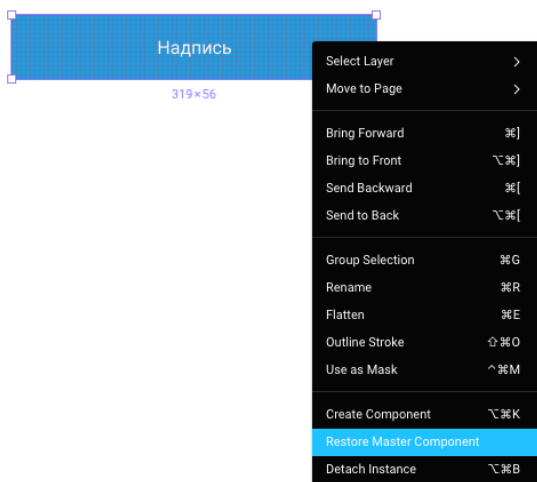
Фігма: коли об'єкт стає майстер-компонентом, він залишається на тому ж місці де був. Зовні він відрізняється від екземпляра тільки іконкою. Нічого не заважає використовувати майстер-компонент як екземпляр, але цього не варто робити.



Різниця в логіці дублювання

Скетч: Якщо дублювати артборд майстра, **Cmd + D**, ми отримаємо ще один майстер з іншим внутрішнім ідентифікатором. Вставити екземпляр можна тільки через меню **Insert** або через плагін **Runner**.

Фігма: Якщо дублювати фрейм майстра, **Cmd + D**, ми отримаємо екземпляр цього майстра. Він встане поверх майстра.



Різниця в логіці детача

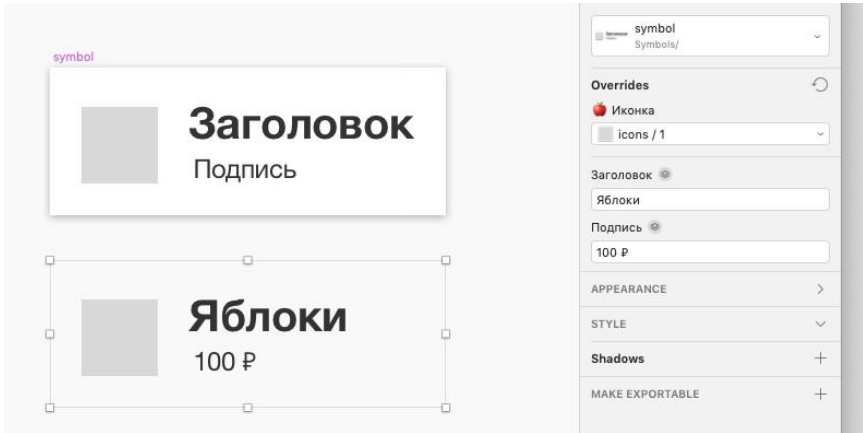
Скетч: коли ми намагаємося видалити майстер-артборд, відбувається перевірка, чи є у нього живі екземпляри, які використовують його як батька. Якщо є, ми бачимо меню **Are you sure you want to delete this Symbol**. Якщо натискаємо **Delete**, всі екземпляри втрачуть зв'язок з майстром і залишаються групами.

Фігма: Якщо видалити майстер, ніякого вікна про підтвердження не буде. Екземпляри після видалення зберігають інформацію про майстра. Якщо вирізати або видалити майстер, екземпляр залишиться екземпляром без явного майстра. У будь-який момент його можна відновити в контекстному меню: **Restore Master Component, Cmd + /, resto**.

Майстер відновиться на тих же координатах, де він був.

Різниця в реалізації оверрайдов

Скетч: оверрайди можна перезаписувати тільки в панелі **Inspector** справа. Контент відділений від шарів в робочій області.

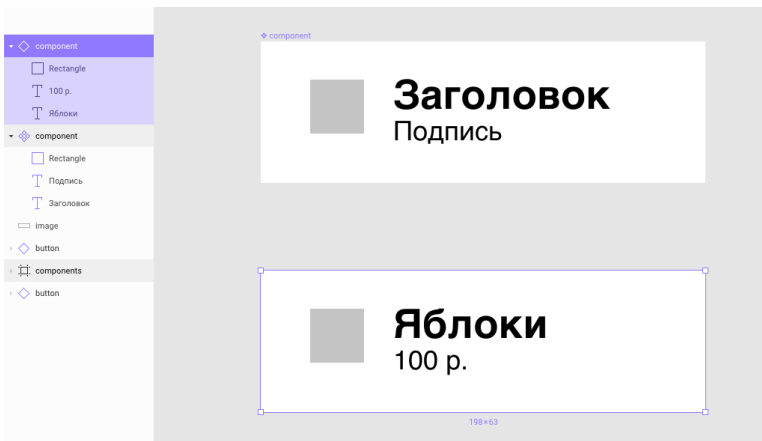


Обмеження

Весь контент, будь то текст, растрові зображення або стилі можна перезаписувати тільки у відповідних полях і тільки якщо послідовність шарів всередині майстер-артборда зрозуміла, а самі шари мають читаемі назви. Це рішення не візуально і ускладнює адаптацію дизайнерів до використання символів. Не можна за один клік обнулити контент примірника.

Можливості

Однак, в ньому є і плюс: можна виділити необмежену кількість споріднених екземплярів і перезаписати їм один і той же загальний оверрайд однією дією.



Фігма: екземпляр компонента можна розкрити як групу і перезаписувати безпосередньо в робочій області.

Обмеження

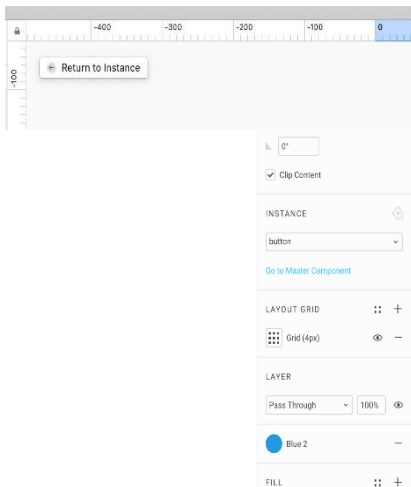
- Не вдається перезаписати одним текстом відразу кілька примірників.
- У структуру примірника не можна впровадити сторонні шари, наприклад, растрові картинки, однак можна перезаписувати заливку.

Можливості

Реалізація дає широкі можливості по роботі зі стилем і видимістю елементів компонента. Якщо в компоненті є інший компонент, наприклад, іконка, його можна перемикаати на інший в правій частині інтерфейсу в поле **Instance**. Клавішею **Delete** можна приховувати елементи в примірниках, що дуже зручно. Також можна обнулити оверрайди командою **Reset Instance**, яка є в контекстному меню. Вона приведе екземпляр в початковий стан. Через пошук: **Cmd + /, rese**.

Хороша практика: Не застосовувати майстри в якості екземплярів. Якщо вирішуємо зробити об'єкт компонентом, відтягуємо його з макету, а краще відправляємо в сторінку «Компоненти». Інакше він може змішатися з екземплярами і загубитися в рядовому екрані.

У Скетчі такої проблеми немає, оскільки майстер-артборд і екземпляр не сплутаєш: екземпляр можна редагувати безпосередньо.



Перехід до майстра

Як в Скетчі, так і в Фігмі, перебуваючи в екземплярі, ми можемо перейти до майстра.

Скетч: виділяємо екземпляр, натискаємо **Enter**. Щоб повернутися до примірника після редагування майстра, натискаємо команду **Return to Instance, Cmd + Esc**.

Фігма: **Enter** розгортає групу примірника і не веде на майстер. Щоб перейти на нього, потрібно використовувати команду **Go To Master Component** в блоці **Instance** або в контекстному меню.

Через пошук: **Cmd + /, go**

Команди повернення до примірника, аналогічної **Return to Instance** в Фігмі немає, тому, перед стрибком добре б запам'ятати місце в проєкті, звідки ми стрибаємо до редагування майстра.

Кращі практики роботи з компонентами

1. Будь-який компонент в дизайн-системі повинен лягати в сітку, бажано 4 або 8рх.
2. Небажано, щоб в компонентах був якийсь осмислений контент. Їх потрібно робити максимально знеособленими.
3. Не слід використовувати майстер-компонент як екземпляри в макеті, його потрібно виносити окремо.
4. Не слід робити з компонентів все, оскільки інтерфейс буде дуже жорстким. Деякі об'єкти в дизайні все-таки краще залишати групами.
5. В першу чергу потрібно налаштувати стилі тексту і кольорів. Точно не варто робити компоненти для зберігання кольорів.
6. Потрібно двічі подумати, перш ніж укласти в компонент заголовок з текстом, що містить конкретний контент. Чи буде він використовуватися достатню кількістю разів, щоб це рішення було виправдано, або лише створить когнітивне навантаження на дизайнерів, які будуть працювати з документом? Тема організації компонентів заслуговує окремої книги, яка називалася б на кшталт «Як створити дизайн-систему».

Практика: компоненти

1. Створити поле введення висотою 5брх в стані порожнього і заповненого. Повинен бути підпис поля і введений в нього текст.
2. Створити екземпляр поля, перемикає в ньому стан між порожнім і заповненим.
3. За допомогою обмежувачів зробити так щоб поле коректно тяглося на різній ширині. Особливу увагу приділити довгим текстам в екземплярі.
4. Зробити дві іконки розміром 24 x 24, зробити з них компоненти.
5. Вставити екземпляр іконки в обидва майстри поля.
6. В екземплярі поля перемикає іконки через оверрайд.