

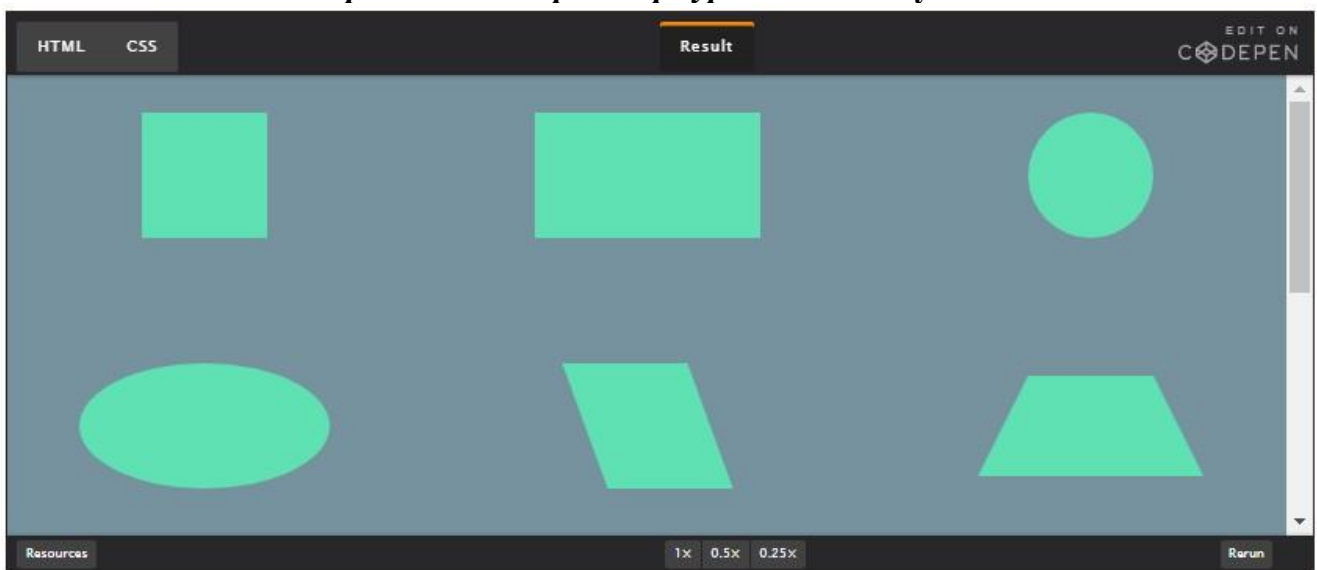
Змістовий модуль 3. Каскадні аркуші стилів CSS3

Тема 8: Анімації на CSS

За допомогою стилів можна задавати не лише статичний вигляд сторінки та її елементів, але й їх динамічну зміну.

Для початку, ознайомимось з можливостями побудови графічних фігур за допомогою CSS: <https://easycss.top/elements/geometriccheskie-figury-na-chistom-css/>.

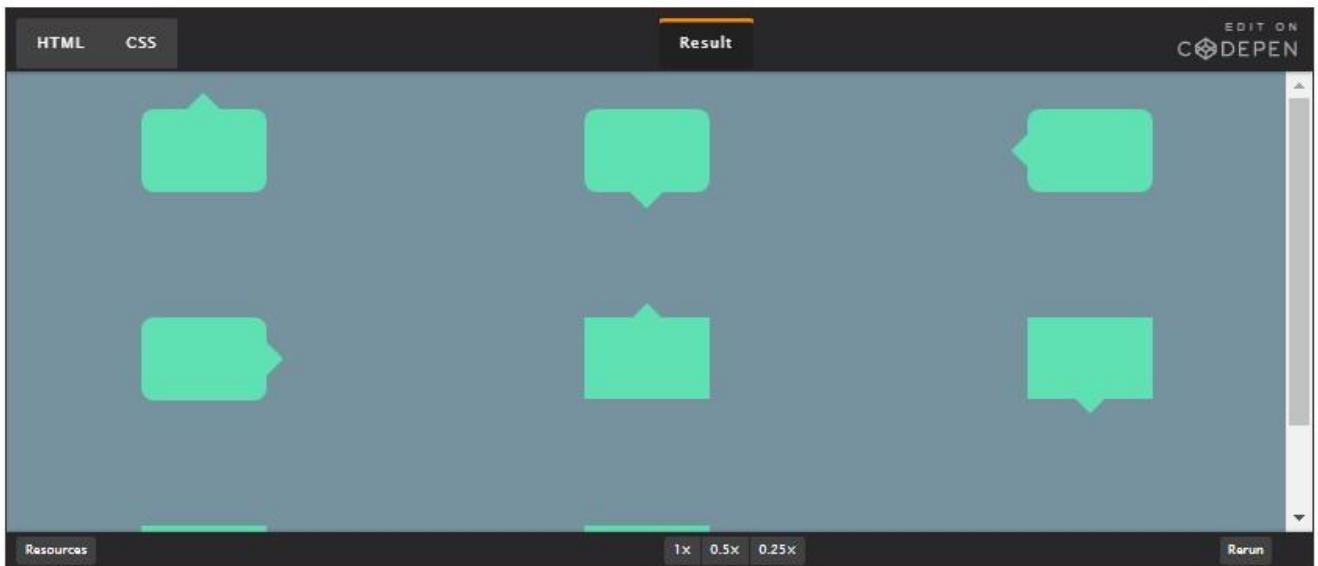
Прості геометричні фігури на чистому CSS



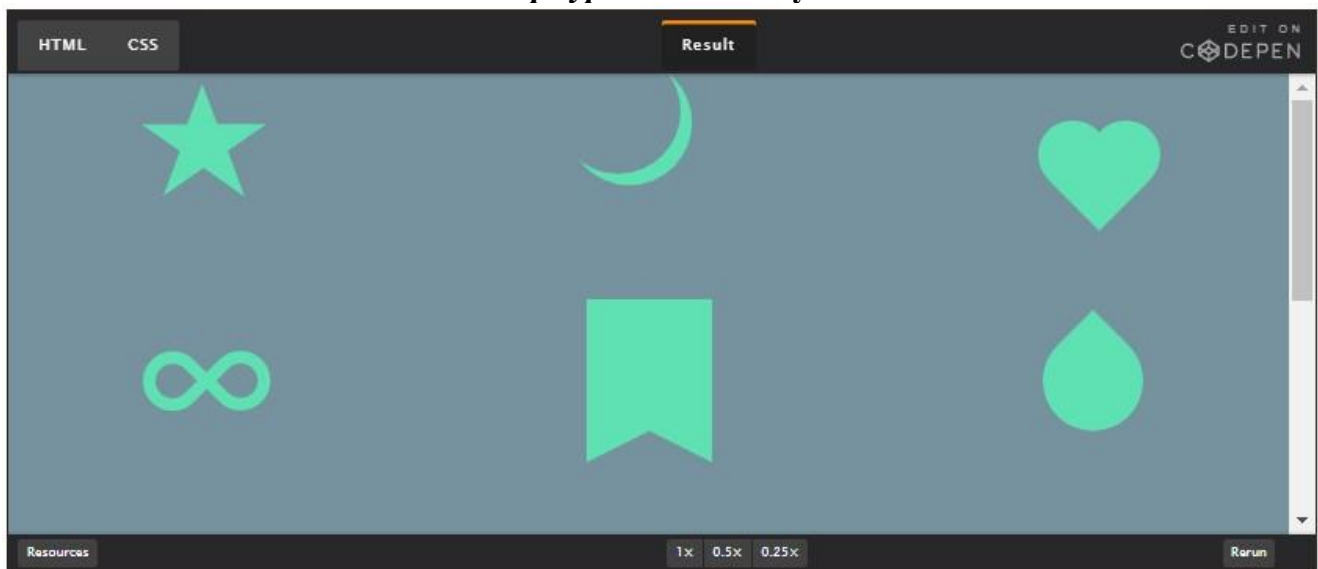
Стрілки на чистому CSS



Тултіп на чистому CSS



Інші фігури на чистому CSS



CSS надає широкі можливості для створення різних геометричних фігур, які можуть бути використані в дизайні веб-сайтів та додатків.

Наведені приклади можна легко адаптувати під свої потреби та використовувати у своїх проектах. Просто скопіюйте css код потрібної вам фігури та застосуйте його до порожнього блоку.

Використання CSS замість зображень може зменшити розмір сторінки та прискорити її завантаження, що важливо для користувачів із повільним інтернет-з'єднанням.

Далі, можемо задавати анімацію для цих фігур.

Так, можна задати стандартне оформлення для певного блоку (наприклад, div), а також його оформлення у момент наведення вказівника миші (hover) на цей блок.

Можна переглянути html-код, опис стилів та результат застосування такого ефекту: <https://jsfiddle.net/vleonteva/6ojrnaxq/>

HTML

```
1 <div>
2   Lorem ipsum dolor sit amet, consectetur adipiscing
   elit, sed diam nonummy nibh euismod tincidunt ut laoreet
   dolore magna aliquam erat volutpat. Ut wisis enim ad
   minim veniam, quis nostrud exerci tution ullamcorper
   suscipit lobortis nisl ut aliquip ex
3   ea commodo consequat.
4 </div>
```

CSS

```
1 div {
2   border: 4px double ● black;
3   background: ● #fc3;
4   padding: 10px;
5   width: 200px;
6   height: 200px;
7 }
8
9 div:hover {
10  width: 300px;
11  transition: 2s;
12 }
13
```

JavaScript

```
1
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

У цьому прикладі стандартно блок `div` має ширину та висоту 200 пікселів, а при наведенні мишки (подія `hover`), ширина змінюється на 300 пікселів протягом 2 секунд. Якщо додати властивість `transition: 2s` до початкового стилю, то повернення до 200 пікселів теж відбуватиметься не миттєво, а протягом вказаного часу.

Для гнучкого керування процесом анімації можна задавати визначення окремих кадрів, `keyframes`. Для такої анімації необхідно вказувати ім'я – для використання при виклику, а також стильові властивості у різні моменти часу:

- `<відсоток>` значення властивостей у заданий момент часу (у відсотках);
- `<властивість>` - анімована властивість;
- `<значення>` - значення в момент, описаний селектором кадру.

Так, у наступному прикладі (<https://jsfiddle.net/vleonteva/zptfab0g/>) блок `div` початково має червоний колір тла. Протягом 4 секунд відбувається анімація `example` з такими етапами: колір заливки змінюється з червоного до жовтого, до

синього, до зеленого. У кінці відбувається повернення до червоного кольору, визначеного для блоку div.

HTML

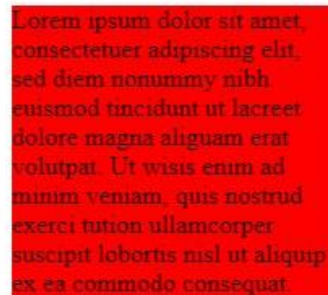
```
1 <div>
2   Lorem ipsum dolor sit amet, consectetur adipiscing
   elit, sed diam nonummy nibh euismod tincidunt ut laoreet
   dolore magna aliquam erat volutpat. Ut wisis enim ad
   minim veniam, quis nostrud exerci tution ullamcorper
   suscipit lobortis nisl ut aliquip ex
3   ea commodo consequat.
4 </div>
5
```

CSS

```
1 div {
2   width: 200px;
3   height: 200px;
4   background-color: red;
5   animation-name: example;
6   animation-duration: 4s;
7 }
8 @keyframes example {
9   0% {background-color: red;}
10  25% {background-color: yellow;}
11  50% {background-color: blue;}
12  100% {background-color: green;}
13 }
```

JavaScript

```
1
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Можна також задавати зміну позиції в анімації; для цього у початкових властивостях слід вказати відносне позиціювання, а у кадрах анімації - задати параметри розташування об'єкту відносно верху та лівої межі сторінки.

У наведеному прикладі (<https://jsfiddle.net/vleonteva/Lk49eph8/>) також змінений радіус фігури (**border-radius**) - таким чином можна отримати коло, чи еліпс. Цей атрибут може визначати форму фігури за параметрами скруглення окремих кутів, наприклад **border-radius:50% 0 0 0;** означає, що скруглений буде лише 1 верхній лівий кут.

HTML

```
1 <div>
2   boom
3   boom
4 </div>
```

CSS

```
1 div {
2   width: 50px;
3   height: 50px;
4   background-color: red;
5   animation-name: example;
6   animation-duration: 4s;
7   border-radius: 50%;
8   text-align: center;
9   position: relative;
10 }
11 @keyframes example {
12   0% { background-color: red; left: 0px; top: 0px; }
13   25% { background-color: yellow; left: 200px; top: 0px; border-radius: 50% 0 0 0; }
```

JavaScript

```
1
```



Серед анімованих властивостей можна виділити трансформацію **transform**, яка може набувати значень `matrix(n,n,n,n,n,n)` - перетворення, `translate(x,y)` - переміщення, `scale(x,y)` - масштаб, `rotate(кут)` - поворот, `skew(кут-x,кут-y)` - нахилання, `perspective` – перспектива.

Наприклад, <https://jsfiddle.net/vleonteva/4euyog75/>

HTML

```
1 <div>
2 boom
3 boom
4 boom
5 </div>
```

CSS

```
1 div {
2   width: 60px;
3   height: 60px;
4   background-color: red;
5   animation-name: example;
6   animation-duration: 4s;
7   border-radius: 50%;
8   text-align: center;
9 }
10 @keyframes example {
11   50% {transform: rotate(360deg)}
12 }
```

JavaScript

```
1
```



Серед додаткових параметрів анімації можна виділити наступні:

animation-delay: 2s; - затримка початку анімації

animation-iteration-count: 3; - кількість повторів анімації. можливе значення infinite; - безконечне повторення

animation-direction: reverse; - напрям анімації (повторити у зворотньому порядку). можливе значення alternate; - повторювати в прямому та зворотньому порядку

animation-timing-function: linear; - прискорення анімації (відсутнє). можливі значення: ease; ease-in; ease-out; ease-in-out;

Варто зазначити, що існує скорочений запис стильових властивостей.

```
div {
  animation-name: example;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-delay: 2s;
  animation-iteration-count: infinite;
```

```
    animation-direction: alternate;
}
```

можна записати:

```
div {
    animation: example 5s linear 2s infinite alternate;
}
```

Властивості анімації можуть специфічно підтримуватись різними браузерми, тож для коректного відтворення слід вживати так звані префікси - додавати до стильових властивостей уточнення того, для якого браузера вони призначені.

- Chrome & Safari: -webkit-
- Firefox: -moz-
- Opera: -o-
- Internet Explorer: -ms-

```
div {
    -webkit-animation: example 5s linear 2s infinite alternate;
    -moz-animation: example 5s linear 2s infinite alternate;
    -o-animation: example 5s linear 2s infinite alternate;
    -ms-animation: example 5s linear 2s infinite alternate;
    animation: example 5s linear 2s infinite alternate;
}
```

ВПРАВИ ТА ПРАКТИЧНІ ЗАВДАННЯ ДО ВИКОНАННЯ

1. Практичне завдання «Анімації»:

Створити анімаційний ефект для веб-сторінки.

Це може бути поява меню, завантажувач (крутиться на сторінці поки вона завантажується повністю), чи будь-який інший ефект на ваш розсуд.

Працюємо в онлайнному редакторі <https://jsfiddle.net/> . Після виконання завдання результат роботи збережіть і надішліть посилання (з рядка адреси) на виконану роботу.

Для натхнення: перегляньте відео, в коментарях наведені посилання на відео-уроки створення деяких з цих ефектів:
<https://www.youtube.com/watch?v=aOI0fpahGr8&t=4s>

Відомості та рекомендації щодо виконання даного завдання – див. Лекцію 8.