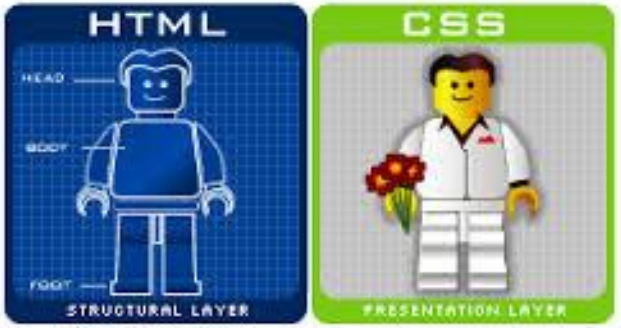
Курс “Основи web-програмування”
частина 1



Мета – здобуття теоретичних знань і практичних навичок з основ web-технологій.

Мета курсу

- В основі створення додатків для Інтернету лежить глибоке знання HTML, CSS та JavaScript. У цьому курсі ви дізнаєтесь як створювати програми для Інтернету.
- По-перше, ви навчитеся **відображати вміст** в Інтернеті за допомогою HTML.
- Далі ви вивчите **стилізацію** Інтернету за допомогою CSS.
- Нарешті, ви дізнаєтесь, як **зробити Інтернет інтерактивним** із JavaScript.
- Закінчивши цей курс, ви отримаєте основоположні знання HTML, CSS та JavaScript, які допоможуть вам рухатися вперед для створення додатків для Інтернету.

HTML, CSS, and JavaScript: The Big Picture

Чому вам слід дбати про те, як працює Інтернет

- **Історія Web**:
  - 60-ті роки. Дослідники намагаються створити мережі, які могли б пережити ядерну війну.
  - 1981: Опубліковано IP4.
  - 1989: Запропоновано WWW
  - 1990: Перший браузер
  - 1994: Поява CSS
  - 1995: JavaScript
  - 2000: 414 мільйонів користувачів Інтернету
  - 2008: Google Chrome
  - 2017: 49% переглядів через Mobile

- **Word Wide Web** - це інформаційний простір, де документи та інші веб-ресурси ідентифікуються за допомогою Uniform Resource Locators (URL-адрес), пов'язаних між собою гіпертекстовими посиланнями, і до них можна отримати доступ через браузер.
- **Інгредієнти для Web**:

1. Ресурси: HTML-документи, зображення ...
2. URL-адреси: Визначають ресурси.
3. HTTP: протокол передачі гіпертексту. Магія, яка може отримати ресурси та донести їх до вашого веб-браузера.

Як це працює:

- Ресурси розміщуються на веб-сервері, який має унікальну адресу в Інтернеті: http://server.com
- Для завантаження цього ресурсу ми можемо використовувати будь-який комп'ютер, на якому встановлений браузер
- Браузер використовує HTML, CSS та JavaScript для показу ресурсу на екрані
- Веб-переглядач отримує ресурси і надсилає їх також на сервер

Мова HTML

HTML - Hypertext Markup Language - мова розмітки гіпертексту - це стандартна мова для створення веб-сторінок та веб-додатків.

Стандарт HTML - це співпраця між **W3C** (World Wide Web Consortium) та **WHATWG** (Web Hypertext Application Technology Working Group).

Принципи роботи робочих груп:

- Стандарт – це HTML + CSS + DOM + JavaScript
- Ніяких плагінів (ні Java, ні Flash, ні Silverlight, нічого!)
- Більше розмітки, не будьте нав'язливими, менше прямих сценаріїв
- Пристрій-незалежність

Мова HTML

Історія:

- 1991: HTML
- 1995: HTML 2.0
- 1997: HTML 3.2 & HTML 4.0
- 1999: HTML 4.01
- 2001: XHTML1.1
- 2004: WHATWG
- 2006: WHATWG and W3C Cooperation
- 2014: HTML5
- 2016: HTML 5.1
- 2017: HTML 5.2

Styling The Web with CSS

- CSS - Cascading Style Sheets - Каскадні таблиці стилів - мова таблиць стилів, яка використовується для опису **подання** документа, написаного мовою розмітки. CSS ***відокремлює стиль від вмісту***.
- **Історія:**
- 1994: Пропозиція CSS
- 1996: CSS1
- 1998: CSS2
- 2014: CSS2.1
- 2011: CSS3...
- 2017: CSS Flexbox

Interacting with The Web with JavaScript

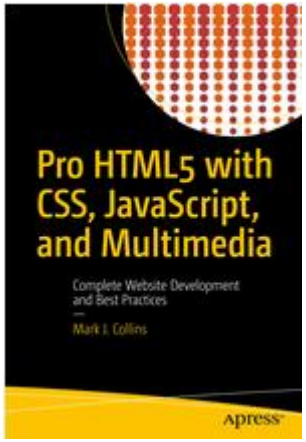
JavaScript -Клей Інтернету. Це інтерпретована мова програмування, яку розуміють браузери.

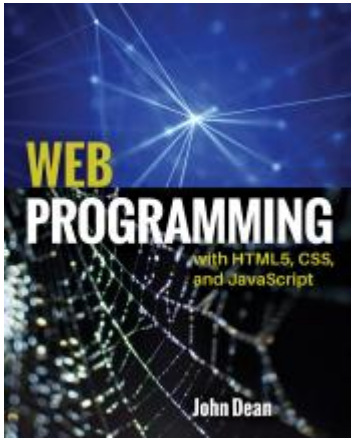
Історія:

- 1994: з'явився веб-браузер Netscape Navigator
- 1995: LiveScript в цьому ж році перейменовано в JavaScript
- 1995: Microsoft реалізує VBScript у Internet Explorer
- 1996: JavaScript подано до ECMA International
- 1997: ECMAScript 1
- 2004: ECMAScript 4
- 2009: ECMAScript 5
- 2015: ECMAScript 2015
- 2017: ECMAScript 2017

Література

1. Jörg Krause **Introducing Web Development** – Apress, 2016
2. Jennifer Robbins **Learning Web Design** Fifth Edition– O'Reilly Media, 2018. 790 p.
3. John Dean **Web Programming with HTML5, CSS and JavaScript** - Jones & Bartlett Learning, 2019. 678 p.
4. Mark J. Collins **Pro HTML5 with CSS, JavaScript, and Multimedia** - Apress, 2017. 560 p.
5. Creating a Website: The Missing Manual, Third Edition by Matthew MacDonald, 2011
6. Learning Web Design, 4th Edition, 2012



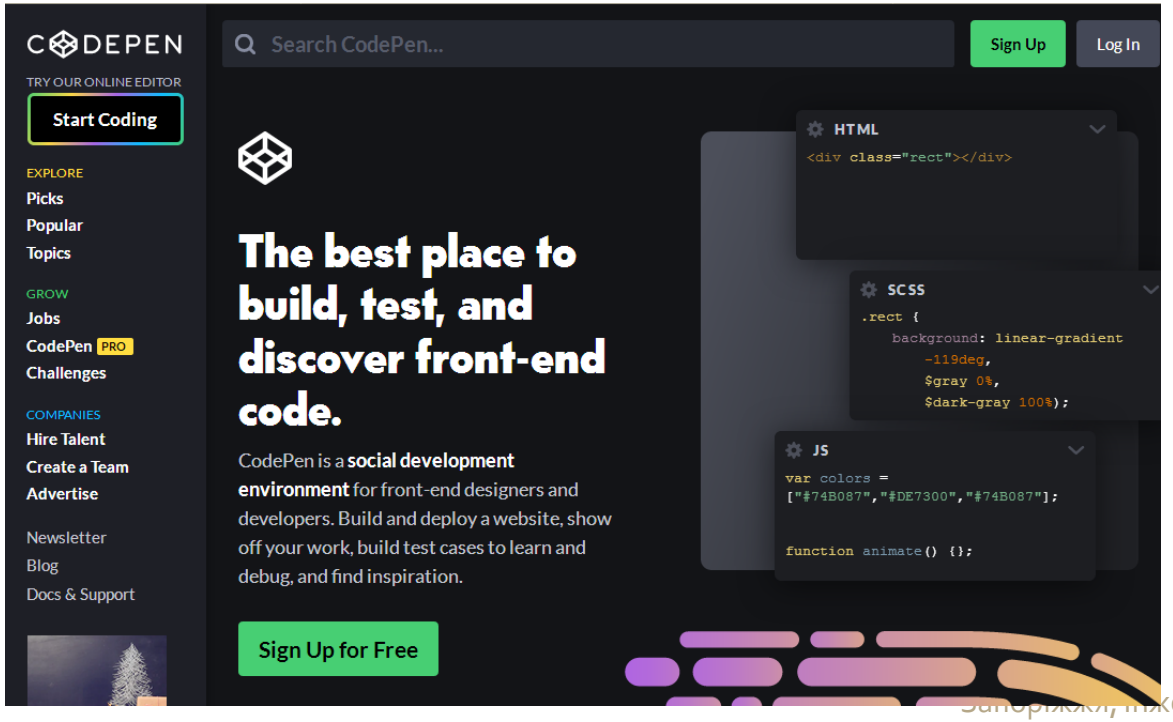


Coding tools [2]

- **Visual Studio Code** https://code.visualstudio.com
- **Sublime Text** (sublimetext.com)
- **Atom** (free from GitHub; atom.io)
- **Brackets** (free from Adobe; brackets.io)
- **CodeKit** (codekitapp.com; Mac only)
- **Adobe Dreamweaver** (www.adobe.com/products/dreamweaver.html)
- **Coda** (panic.com/coda/)

Online редактор CodePen

- **https://codepen.io/**
- Це щось типу індустріального стандарту
- Підтримка HTML, CSS, JavaScript



User interface and layout tools [2]

- **Sketch** (sketchapp.com, Mac only)
- **Affinity Designer** (affinity.serif.com/en-us/designer/)
- **Adobe XD** (www.adobe.com/products/xd.html)
- **Figma** (figma.com)
- **UXPin** (uxpin.com)

Web graphic creation tools [2]

- **Adobe Photoshop**
- **GIMP** (free, open source; gimp.org)
- **Corel PaintShop Pro** (for photo editing; paintshoppro.com; Windows only)
- **Corel Draw** (for vector drawing; coreldraw.com; Windows only)
- **Pixelmator** (pixelmator.com; Mac only)

Chrome DevTools (рекомендації)

- Drag-and-drop в панелі Елементи
- Використання панелі Console
- Додавання CSS і редагування стану елемента
- Збереження в файл зміненого CSS
- Перегляд виразу

Що таке HTML ?

HTML (англ. *HyperText Markup Language* — **Мова розмітки гіпертекстових документів**) — стандартна мова розмітки веб-сторінок в Інтернеті

HTML впроваджує засоби для:

- створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

Приклад

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Що таке CSS ?

CSS (англ. *Cascading Style Sheets, -* ***Каскадні таблиці стилей***) - спеціальна мова опису сторінок, написаних мовою розмітки HTML

CSS видає браузеру інструкції про те, як вивести певний елемент

Приклад

```
p { font-family: Verdana, sans-serif; }

h2 { font-size: 110%; color: red; background: white; }

.note { color: red; background: yellow; font-weight: bold; }
```

CSS надає можливість розділення змісту сторінки (даних) та його візуальної презентації.

Корисні посилання

- W3 Schools (http://www.w3schools.com/)
- **Ресурси для розробників** (https://developer.mozilla.org/uk/)
- Генератор шаблонів сайтів - csstemplater.com
- Інструмент вибору кольору https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Colors/Color_picker_tool
- GoogleFonts https://fonts.google.com/
- HTML5 BOOK https://html5book.ru/
- **ТАБЛИЦЯ БЕЗПЕЧНИХ КОЛЬОРІВ** http://getcolorcode.com/ua/colors/websafe
- Free Image Placeholder Service https://placehold.it
- Lorem ipsum generator online http://uk.lorem-ipsum.info/_latin

Нова школа проти Старої школи

Стара школа:

- Повністю побудувати сайт в HTML
- Використання таблиць для побудови макету сайту (заголовки, навігація, зображення, нижні колонтитули)
- Нерухома ширина таблиці не дуже добре переводиться на інші платформи (КПК, мобільні телефони, широкоформатних моніторів)
- Багато коду, часто брудного

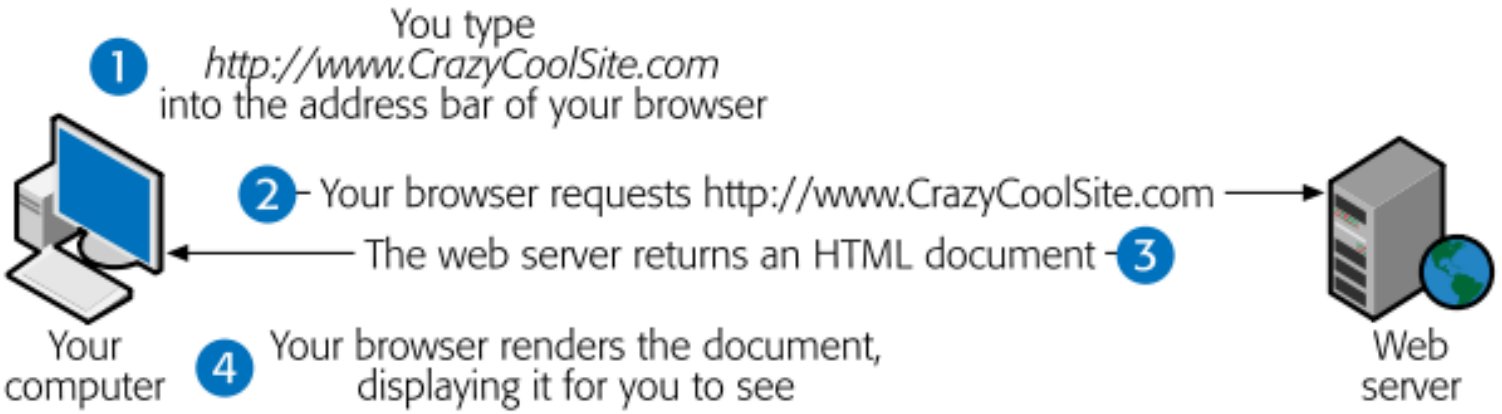
Нова школа:

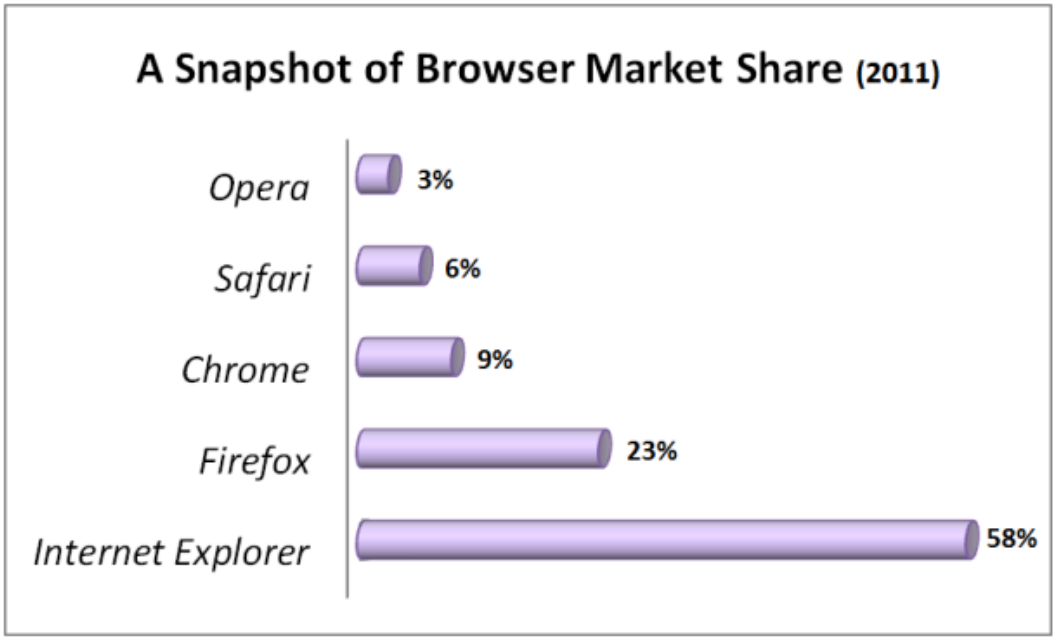
- Побудова сайту в HTML і CSS
- HTML, який визначає тільки "скелет" змісту.
- CSS, який забезпечує формат і макет.
- Код чистий і "худий"
- Використання таблиць тільки для табличних даних (формат рядків/стовпців: імена і телефонні номери, дати й частоти)

Чому?

- Розділення змісту від представлення – можливість змінити зовнішній вигляд всього сайту, змінивши один CSS файл.
- Більше гнучкості та контролю над сайтом, який буде розглядатися в різних браузерах і платформах, в тому числі КПК, мобільних телефонів, широкоформатних моніторів.
- Більш швидке завантаження.
- Простий HTML-контент є більш "прозорим" для браузерів.
- Менше коду і менше зображень у форматі HTML
- HTML не приймає участь в форматуванні коду та дизайну.

Introducing the World Wide Web





Types of Sites

- Personal sites
- Résumé sites
- Topical sites focus on a particular subject that interests you
- Event sites
- Promotion sites
- Small business (or e-commerce) sites
- . . . . . .

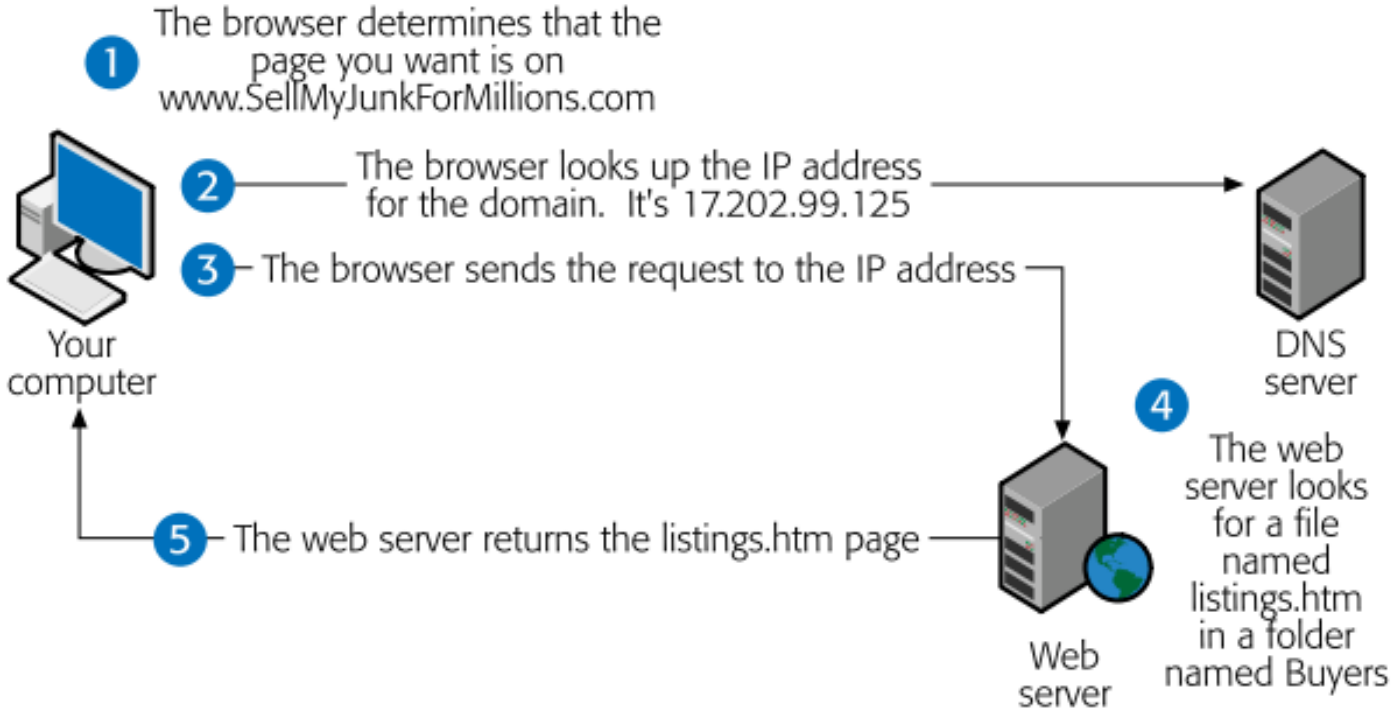
How Web Hosting Works

Understanding the URL (Uniform Resource Locator)

- The protocol indicates your chosen method of transmission—in other words, how your browser should communicate with the web server.
- Websites always use HTTP (HyperText Transfer Protocol)
- You may use other protocols (ftp:// (File Transfer Protocol) for uploading and downloading files and file:/// for retrieving a file directly from your own computer's hard drive)
- The domain name identifies the server that hosts the site you want to see.
- By convention, server names usually start with www to identify them as World Wide Web servers

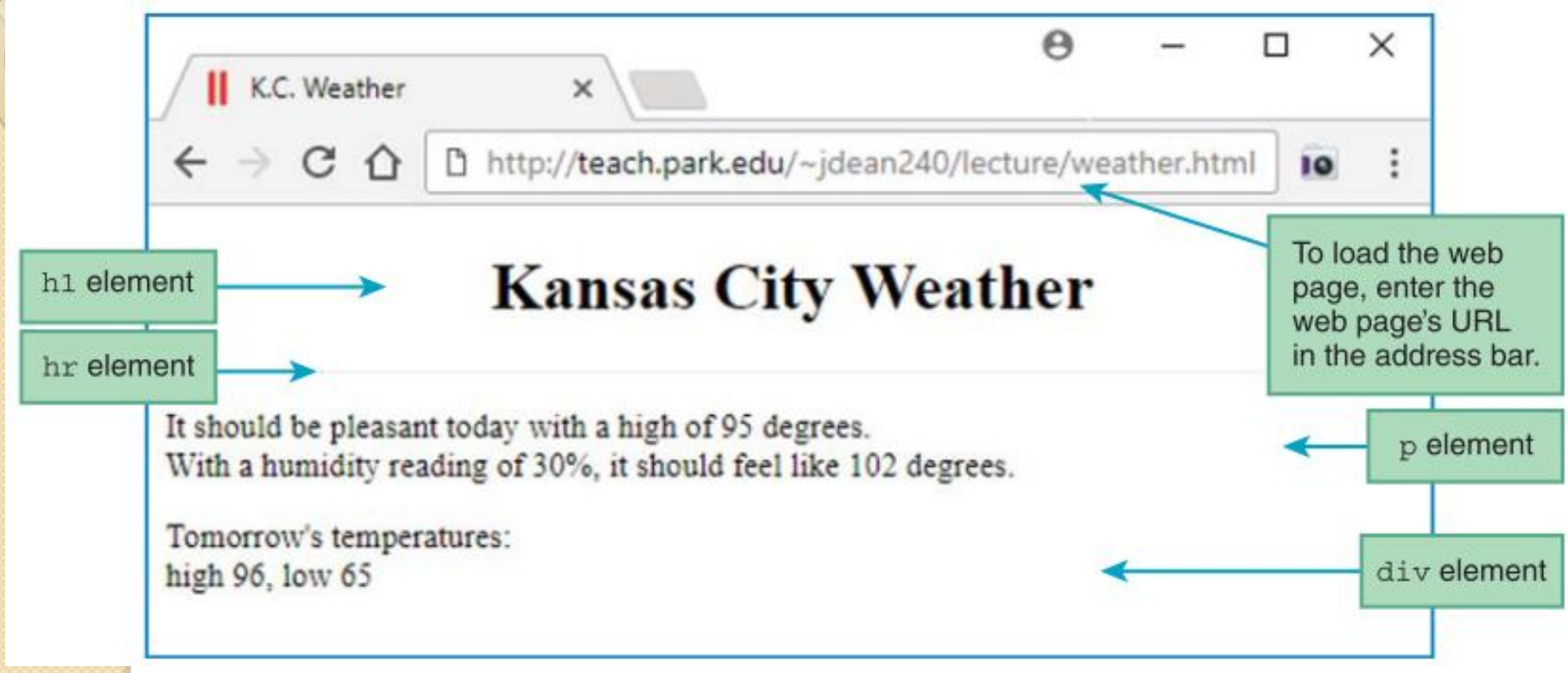
How Browsers Analyze a URL

For example, after you type
http://www.SellMyJunkForMillions.com/Buyers/listings.htm
into the address bar and press Enter, here's what happens:



DNS – Domain Name System

Web Page Example



http://teach.park.edu/~jdean240/lecture/weather.html

Source code for Kansas City Weather web page

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="author" content="John Dean">
<meta name="description" content="Kansas City weather
  conditions">
<title> K.C. Weather</title>
<style>
  h1 {text-align: center;}
  hr {width: 75%;}
</style>
</head>
```

```
<body>
<h1> Kansas City Weather</h1>
<hr>
<p>
  It should be pleasant today with a high of 95
   degrees.<br>
  With a humidity reading of 30%, it should feel like 102
   degrees.
</p>
<div>
  Tomorrow's temperatures:<br>
  high 96, low 65
</div>
</body>
</html>
```

HTML Tags

- Існує два типи елементів - контейнерні елементи та одиночні елементи (елементи void або Empty Elements).
- **Контейнер** має початковий тег і кінцевий тег, і він містить вміст між двома його тегами. Наприклад, елемент h1 є контейнером.
- **Елемент void** має лише один тег, і його вміст зберігається всередині тегу. Ми побачимо приклад цього, коли перейдемо до мета-елемента.

```
<head>
<meta charset="utf-8">
<meta name="author" content="John Dean">
<meta name="description" content="Kansas city weather conditions">
<title>K.C. Weather</title>
</head>
```

Empty Elements

Найвідоміші порожні елементи:

-
- <hr>
- <img>
- <input>
- <link>
- <meta>

HTML Attributes

- У наступному прикладі **charset** є атрибутом тегу meta:

```
<meta charset="utf-8">
```

- Атрибути частіше зустрічаються з void елементами, але їх можна використовувати і з елементами контейнерів. Ось приклад елемента контейнера, який використовує атрибут:

```
<html lang="uk">
•
•
•
</html>
```

Вам не обов'язково використовувати атрибут lang, але радимо включити його для елемента html.

Список стандартних HTML-атрибутів

Перелічимо стандартні атрибути, притаманні практично всім HTML-елементам.

Атрибут	Опис
accesskey	Визначає сукупність клавіш для доступу до елемента
class	Визначає ім'я класу для елемента
id	Визначає унікальний ідентифікатор для елемента
style	Визначає стиль елемента
title	Містить додаткову інформацію про елемент (значення цього атрибуту відображається у вигляді підказки при наведенні курсора миші на елемент)

Блокові та рядкові елементи HTML

- Історично HTML-елементи було прийнято ділити на блокові (block) і рядкові (inline). Блокові елементи займають всю ширину свого батька (контейнера), формально створюючи «блок» (звідси і назва).
- Браузери зазвичай відображають блокові елементи з переводом рядка до і після елементу.
- Існує кілька ключових відмінностей між блоковими і рядковими елементами:

1. **Блокові** елементи браузер виводить на екран в **прямокутні області**, що йдуть один за одним зверху вниз
2. **Займають весь доступний простір** по горизонталі.
3. Блокові елементи **починаються з нового рядка**;
4. Блоковим елементам можна задавати **ширину, висоту, внутрішні і зовнішні відступи**.

5. Блокові елементи можуть містити блокові і рядкові елементи. Однак **блоковий елемент P є винятком: всередині нього можуть знаходитись тільки рядкові елементи**.
6. Блокові елементи можуть міститися тільки в межах блокових елементів.
7. В блокові елементи можна вкласти як блокові, так і рядкові елементи.
8. Блокові елементи мають по замовчуванню властивість **display:block**, яка може не вказуватись в css

- Поділ елементів на блокові і рядкові використовувалося в специфікації HTML до версії 4.01.
- У HTML5 це протиставлення замінено більш складним набором **категорій контенту**.
- Категорія «рядкових» елементів приблизно відповідає категорії **текстового контенту**, а для «блокових» елементів прямого відповідності немає, але «блокові» і «рядкові» елементи разом приблизно відповідають категорії **потокового контенту** в HTML5 (тобто, грубо кажучи, «блокові» елементи - це потоковий контент мінус текстовий контент).
- Крім того, є й інші категорії, наприклад, інтерактивний контент.

Список блокових елементів

- <address>
- <article>
- <aside>
- <blockquote>
- <details>
- <dialog>
- <dd>
-
- <dt>
- <fieldset>
- <figcaption>
- <figure>
- <footer>
- <form>
- <header>
- <hgroup>
- <hr>
-
- <main>
- <nav>
-
- <p>
- <pre>
- <section>
- <table>
-

Список рядкових елементів

- b, big, i, small, tt
- abbr, acronym, cite, code, dfn, em, kbd, strong, samp, time, var
- a, bdo, br, img, map, object, q, script, span, sub, sup
- button, input, label, select, textarea

Теги для коментарів

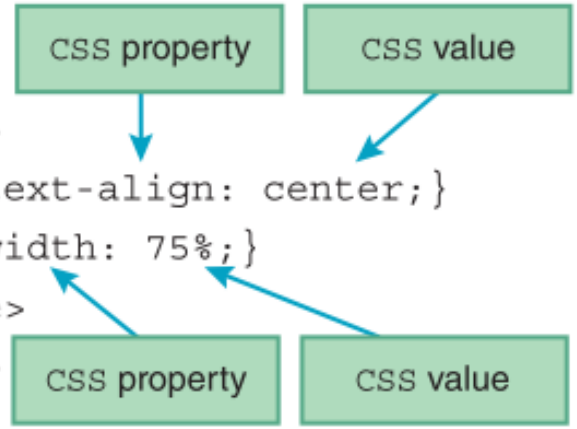
- Коментування в HTML необхідно для покращення читабельності коду.
- **Тег для коментаря починається з кутової дужки, знак оклику, два дефіси, текст коментаря, два дефіси, закінчується кутовою дужкою.**
- Коментарі в HTML не відображаються на сторінці в браузері користувача.

```
<!-- NAVIGATION -->
    <nav> … </nav>
<!-- END NAVIGATION -->
```

Cascading Style Sheets Preview

```
<head>
   •
   •
   •
<style>
  h1 {text-align: center;}
  hr {width: 75%;}
</style>
</head>

<body>
<h1>Kansas City Weather</h1>
<hr>
   •
   •
   •
```



Різниця між Old HTML і HTML5

- У реальному світі ви побачите багато старого HTML-коду. Старий код, який ви побачите найбільш ймовірно буде XHTML 1.0, оскільки він був найпопулярнішим попередником HTML5.
- Коли ви бачите цей код, не потрібно тривожитися; сьогоднішні браузери відмінно відтворюють код XHTML 1.0.
- Але в інтересах довгострокової стабільності та дотримання правил кодування вашої компанії іноді потрібно буде конвертувати старий HTML-код у HTML5.
- Для цього потрібно знати відмінності між старим HTML-кодом та HTML5.

Різниця між Old HTML і HTML5

Наступні відмінності показують, як HTML5 послабив деякі свої правила синтаксису порівняно з XHTML 1.0:

- У HTML5 більше **немає вимоги мати значення для кожного атрибута**. Отже, для деяких атрибутів HTML5 законно включати атрибут сам по собі, не додаючи до нього знак рівності чи значення. Однак стандарти кодування пропонують завжди включати лапки.
- У HTML5 більше **немає вимоги мати / для всіх void елементів**. Наприклад, специфікація XHTML вимагає написання елемента **br** з косою рисою,
. Стандарти кодування пропонують завжди пропускати /.
- У HTML5 більше **немає необхідності мати кінцевий тег для кожного контейнера**. Специфікація XHTML вимагає включення тегу </p> для кожного елемента контейнера p. Специфікація HTML5 говорить, що ви можете включити або пропустити кінцевий тег. Стандартні умови кодування пропонують завжди включати кінцевий тег.

Різниця між Old HTML і HTML5

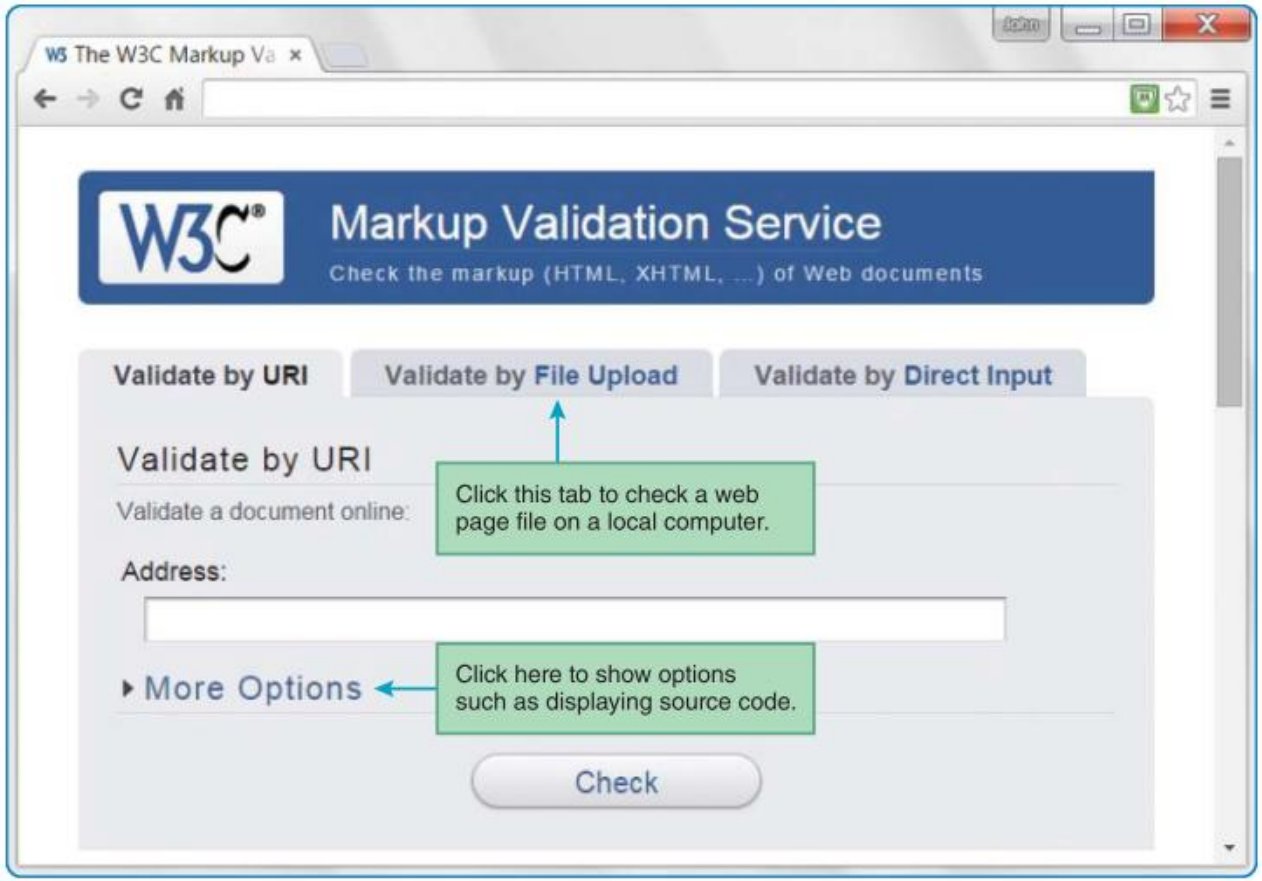
- Деякі теги в HTML5 вважаються застарілими (наприклад, теги <font> і)

З'явились нові елементи:

- Елементи структурної організації (приклади - header і footer).
- Елементи audio та video дозволяють користувачам відтворювати музику та відеофайли безпосередньо зі своїх браузерів без необхідності використання плагінів.
- Елемент canvas містить область малювання та набір команд, які веб-програміст може використовувати для малювання двовимірних фігур та анімації їх.
- Drag and drop functionality
- Web storage functionality
- Geolocation functionality

Як перевірити ваш HTML код

- Треба використати W3C's HTML validation service: https://validator.w3.org



Спеціальні символи

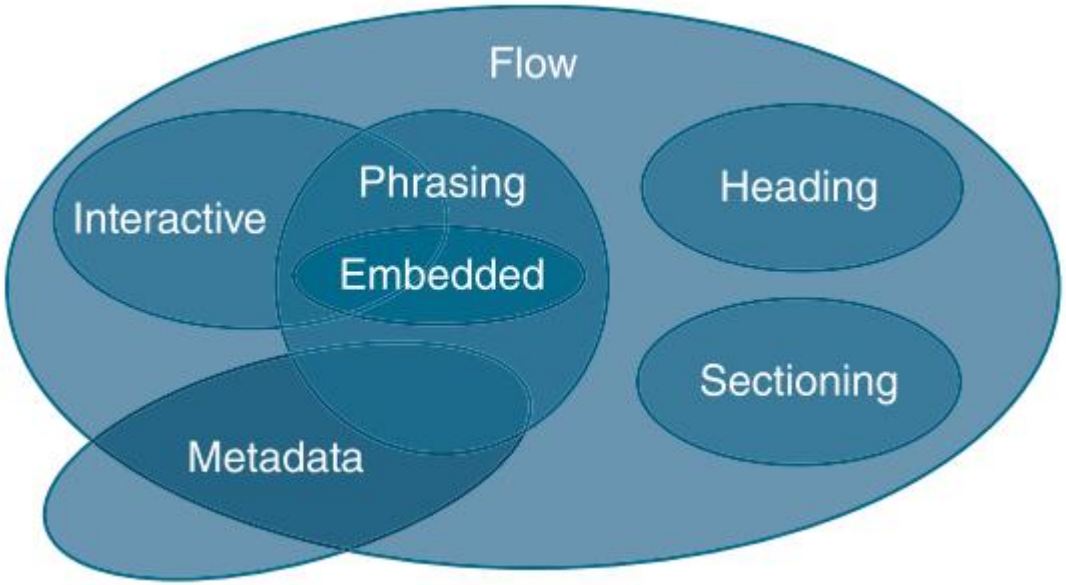
Character	Character Reference	Description
<	<	less than
>	>	greater than
≤	≤	less than or equal
½	½	one-half
¼	¼	one-fourth
&	&	ampersand
"	"	quote
'	'	apostrophe
space		nonbreaking space
←	← ←	left arrow
•	·	bullet
✓	✓	check mark
©	©	copyright

Стандарти кодування

- Незважаючи на недоліки сервісу перевірки HTML, він все ще є корисним інструментом.
- Добре виявляти синтаксичні помилки, як-от імена неправильно написаних тегів.
- Це також добре в правилах контейнерного вмісту (containership).
- Наприклад, контейнер **head** повинен містити заголовок title, а **p** контейнер не повинен містити контейнер **div**. Маючи безліч елементів (близько 115), існує безліч правил контейнерного вмісту (понад 11 000) [3].

Категорії елементів HTML5

- Елементи впорядковані в категорії вмісту (content categories) і на цих категоріях є загальні правила, які визначають, де елемент може бути використаний і що він може містити [3, 4].
- Елемент може бути в декількох категоріях, і є кілька елементів, які не належать до жодної категорії.



- **Metadata** - вони не мають реального вмісту, але надають метадані про HTML-документ та інформацію програмам, які обробляють документ. (base, link, meta, script, style, title)
- **Sectioning** - Ці елементи використовуються для організації сторінки в розділи, а також використовуються для побудови основи (outline) документа. (article, aside, command, nav, section)
- **Heading** - Ці елементи використовуються в розділах для визначення заголовка та підзаголовків розділу. (h1-h6)
- **Embedded** - Вбудовані елементи використовуються для вставки вмісту, що не містить HTML, наприклад зображень, у документ. (audio, canvas, embed, iframe, img, math, object, svg, textarea, video)
- **Interactive** - Ці елементи забезпечують взаємодію з користувачем. Кнопка або поле введення - поширені приклади цього. (a, audio, button, embed, fieldset,iframe, img, input, label, menu, object, output, progress, select,textarea, video, …)

- **Form** - Елементи форми використовуються для фіксації введення користувача та далі поділяються на кілька підкатегорій. (button, input, keygen, label, meter, object, output, progress, select, textarea)
- **Phrasing** - Ці елементи використовуються для позначення тексту, а точніше фраз, які поєднуються для формування абзаців. (abbr, audio, b, br, button, canvas, cite, code, datalist, dfn, em, embed, fieldset, i, iframe, img, input, kbd, keygen, label, legend, mark, math, …, object, output, progress, script, …)
- **Flow -** Переважна більшість елементів потраплятиме до цієї категорії. Це елементи, які мають фактичний вміст, наприклад, текст чи якийсь вбудований вміст, наприклад зображення чи відео. Термін "потік" використовується тому, що ці елементи займають простір і, як правило, переходять від одного елемента до іншого, поперек або вниз сторінки. (a, address, area, article, aside, b, blockquote, br, button, canvas, cite, code, command, data, datalist, del, div, figure, footer, form, h1, …)

Примітка

- Більшість елементів, які не входять до жодної категорії вмісту, використовуються лише як дочірні елементи для певного елемента. Наприклад, елемент таблиці (table) знаходиться в категорії потоку (Flow), але дочірнім елементам, таким як **tr** і **td**, не присвоюється категорія, оскільки вони використовуються лише в межах таблиці.

Sectioning Content

- Доцільно організувати свій HTML-документ **у логічні розділи**, особливо для великих документів. До HTML5 елемент поділу (division, **div**) використовувався для групування вмісту, і вони можуть бути вкладені так:

```
<div>
    <div>
        <div>
        </div>
    </div>
</div>
```

HTML5 представляє кілька нових елементів, які забезпечують **більш конкретні смислові групування**. Кожен з цих елементів використовується для групування вмісту у більшу одиницю. Однак кожен з них групує вміст **з іншої причини**.

1) Section

- Елемент section використовується для організації вмісту в логічні розділи.
- Подумайте про твір, який, можливо, ви написали для шкільного завдання, де ви мали б вступ, три пункти основної частини та висновок.
- Кожна з них буде представлена в HTML як розділ (section).
- Також елементи розділу можуть бути вкладені так само, як і елементи div.
- Якщо кожна ваша основна частина має підпункти, ви можете використовувати section для кожного підпункту.

2) Article

- Елемент статті (article) використовується для групування вмісту, який може стояти самостійно, тобто не залежить від решти сторінки.
- Одне з найпоширеніших застосувань елемента статті - **у блогах**. Кожна публікація в блозі, як правило, є окремим контентом і часто групується в елемент статті. Також коментарі, розміщені в блозі, також зазвичай розміщуються в елементі article.
- Ви можете використовувати елемент article будь-де, де ви б використовували елемент section, якщо вміст є незалежним та багаторазовим. Стаття, особливо велика, часто використовує елемент section.
- Також елемент article може включати в себе інші елементи article, як це було б у випадку з публікацією в блозі, включаючи коментарі.

3) Aside

- Елемент aside використовується для **групування вмісту, який не належить до нормального потоку** вашого документа.
- Це може бути допоміжна інформація, надана для довідок, або, можливо, інформація про автора. Це також може бути не пов'язана між собою інформація, наприклад рекламний простір або календар подій.
- Елемент aside часто представлений у вигляді **бічної панелі**, щоб не перешкоджати потоку іншого вмісту.
- В подальшому CSS визначить деталі; завдання автора вмісту - вказати, що група вмісту не є частиною нормального контенту. І це робиться за допомогою aside .

4) Nav

- Елемент nav використовується **для групування набору посилань**.
- Типовим прикладом є те, коли у вас є якесь меню на сторінці, щоб перейти до внутрішніх закладок або інших пов’язаних сторінок.
- Інший приклад - ви надаєте посилання для отримання додаткової інформації або пов'язаних матеріалів.
- Не потрібно вводити кожне посилання у елемент nav. Але якщо у вас є вміст, який складається в основному з посилань, введіть це в елемент nav. Це вкаже, що цей розділ вмісту забезпечує навігацію.

5) Address

- Елемент адреси не входить до вмісту розділів.
- Але його слушно згадати тут, оскільки він використовується для надання контактної інформації або для всього документа, або для певної статті.
- Щоб використовувати його для однієї статті, його слід розмістити десь усередині елемента статті.
- Якщо він застосовується до документа, він повинен знаходитися всередині елемента тіла; це часто розміщується в елементі колонтитулу.

Приклад використання address в елементі footer

```
<footer>
    <p>Closing content</p>
    <address>
        <p>Provided by
            <a href="mailto:mcollins@theCreativePeople.com">MarkJ.Collins</a>
        </p>
        <p>For more information
            <a href="www.theCreativePeople.com">visit his website</a>
        </p>
    </address>
</footer>
```

6) Outlines

- У специфікаціях HTML5 є поняття контуру (схеми, обрису, загальних рис, outline) документа.
- Елемент body, який є кореневим елементом для вашого вмісту, створює найвищий вузол в обрисі документа.
- Додавання будь-якого з елементів section, article, aside або nav, створить новий контур у контурі.
- Вставлення додаткових елементів секцій додасть більше контурів до контуру.

Listing 4-1. Creating a document outline using sections

```
<body>
    <h1> My Sample Page</h1>
    <nav>
        <h1>Navigation</h1>
    </nav>
```

```
<section>
    <h1>Top-level</h1>
    <section>
        <h1> Main content</h1>
        <section>
            <h1> Featured content</h1>
        </section>
        <section>
            <h1> Articles</h1>
            <article>
                <h1> Article 1</h1>
            </article>
        </section>
        </section>
        <aside>
            <h1> Related content</h1>
```

```
        <section>
            <h1> HTML Reference</h1>
        </section>
        <section>
            <h1> Book list</h1>
            <article>
                <h1> Book 1</h1>
            </article>
        </section>
    </aside>
</section>
</body>
```

Як виглядає в браузері

Сайт для надання Outlines

- Щоб додатково проілюструвати це, є зручна веб-сторінка, яка прочитає ваш HTML-документ і відобразить його контур.
- Ви можете знайти сайт за адресою https://gsnedders.html5.org/outliner
- Вставте свій HTML-документ і натисніть кнопку «Outline this». Контур, який відображатиметься, буде подібний до малюнка

Outline для файлу Listing 4-1

1. My Sample Page
    1. Navigation
    2. Top-level
        1. Main content
            1. Featured content
            2. Articles
                1. Article 1
        2. Related content
            1. HTML Reference
            2. Book list
                1. Book 1

7) Header and Footer

- Організовуючи свою сторінку, слід також розглянути можливість додавання header елемента вгорі та footer елемента внизу.
- Ці елементи дозволяють **групувати вступний або заключний зміст** для розділу документа.
- На відміну від секціонуючих елементів, таких як article чи section, елементи header та footer не створюють новий розділ у структурі документа а групують вмісту розділу, в якому вони розміщені.
- Їх зазвичай використовують в елементі body.
- Вони також можуть бути розміщені всередині дочірнього розділу, такого як елемент section. У цьому випадку вони згрупують вміст лише для розділу.

HTML5 Site Structure

```
1   <!doctype html>
2   <html>
3   <head>
4     <meta charset="utf-8">
5     <title>HTML5-Seite mit Grundstruktur</title>
6   </head>
7
8   <body>
9     <header>
10      <img src="logo.png" alt="logo">
11      <h1>My Company</h1>
12    </header>
13
14    <footer>
15       <a href="contact.html">Kontakt</a>
16       <p>&copy;   2016 by Joerg Krause</p>
17    </footer>
18   </body>
19  </html>
```

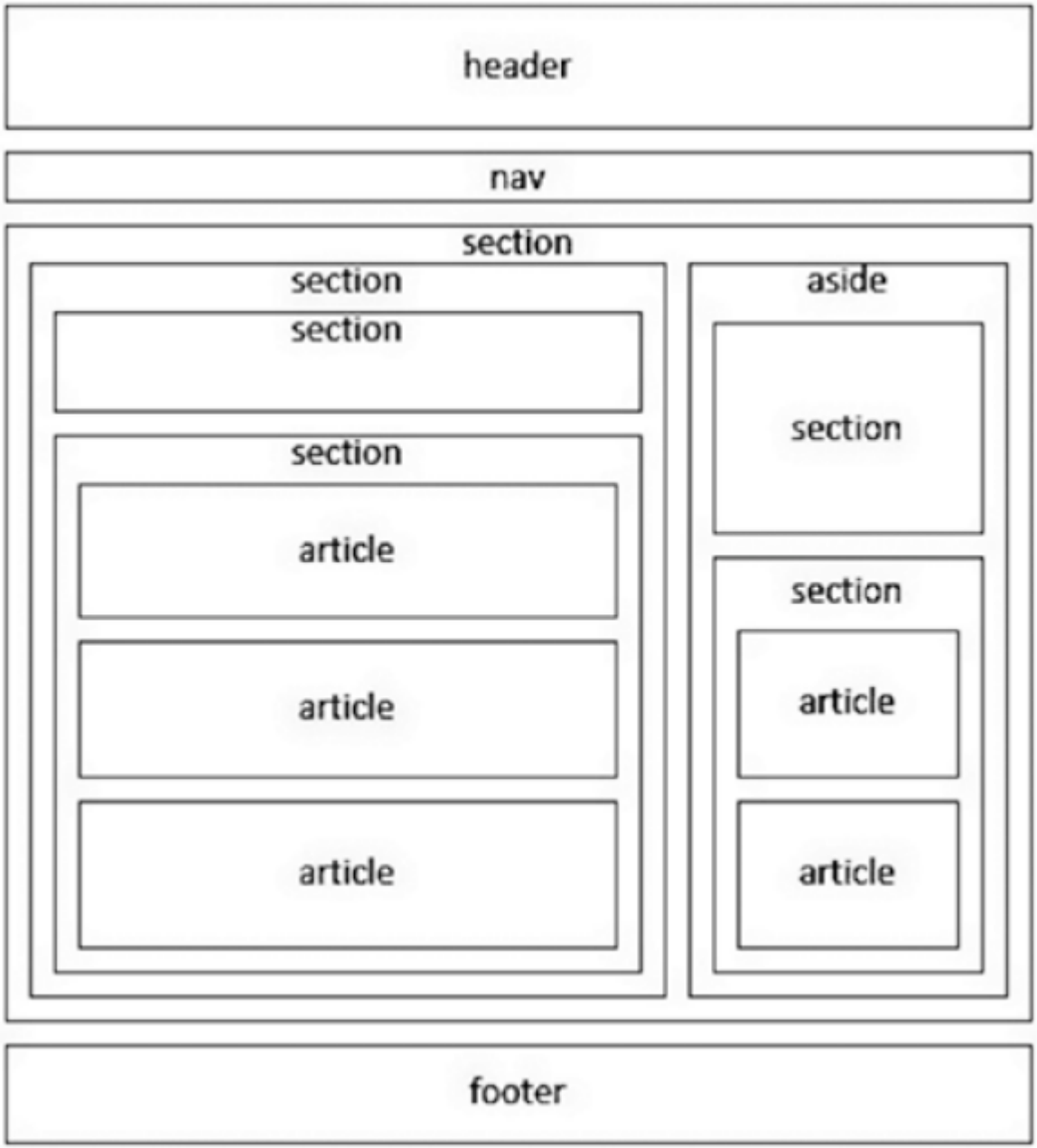
Structure of Navigation

```
1   <body>
2     <header>
3       <img src="logo.gif" alt="logo">
4       <h1>Heading</h1>
5       <nav>
6         <ul>
7           <li><a href="#link_1.html">Wiki</a></li>
8           <li><a href="#link_2.html">Blog</a></li>
9           <li><a href="#link_3.html">Forum</a></li>
10        </ul>
11      </nav>
12    </header>
```

8) Планування макета сторінки

(Planning the Page Layout)

- Перш ніж створити нову веб-сторінку, рекомендується накреслити основну структуру сторінки. Це допоможе вам візуалізувати загальний макет і побачити, як елементи вкладаються разом.
- Сторінка, яка продемонструється далі , буде використовувати header та nav елементи зверху, і footer елемент внизу.
- Основна область посередині використовуватиме елемент section та матиме дві бічні області, кожна з яких має низку тегів article. Більша площа буде укладена в інший елемент section та надаватиме основний вміст, який організований у елементи article. Менша область праворуч використовуватиме aside елемент і буде містити елемент section. Він буде містити ряд елементів article, в яких буде представлена відповідна інформація. Малюнок ілюструє макет сторінки.


header
nav
section
section
section
section
article
article
article
aside
section
section
article
article
footer

Sectioning Roots

- Є кілька елементів HTML, які називаються корінням секціонування (sectioning roots), які мають власний контур (схему), який не є частиною контуру решти документа.
- Елемент тіла є одним із цих елементів, хоча це, мабуть, особливий випадок; контур елемента тіла - це контур документа.
- Інші елементи цієї категорії включають блок-цитати (blockquote), details, fieldset, figure та td.

1) Blockquote

- Елемент blockquote використовується тоді, коли потрібно включити довгу цитату в документ.
- Вміст цитати розміщується всередині елемента blockquote і може складатися з декількох елементів, включаючи текст заголовка, абзаци та вкладений вміст. Проста цитата може виглядати приблизно так:

```
<blockquote cite="www.apress.com">
    <h1>Quotation</h1>
    <p>This is a quotation</p>
</blockquote>
```

2) Details

- Елемент details дозволяє створювати розділи вмісту, що згортаються.
- Всередині елемента details.ви можете включити необов'язковий елемент summary, який буде містити вміст, який відображається при згортанні елемента. Якщо не використовується елемент summary, згорнутим текстом буде "Details.".
- Решта вмісту елемента details буде прихована при згортанні. Початковий стан елемента деталей буде згорнуто. Якщо ви хочете, щоб вміст відображався під час завантаження сторінки, додайте open атрибут типу Boolean.

```
<details open>
    <summary>This is the collapsed text</summary>
    <h1>Details</h1>
    <p>These are collapsable details</p>
</details>
```

Знову ж таки, як і інші кореневі елементи секціонування, ви можете включити елементи h1 – h6 в елемент details, який визначатиме контур details. Однак ці розділи не будуть включені до структури документа.

3) Figure

- Елемент figure використовується для групування вмісту, який є автономним і може логічно переміщуватися в інше місце, не впливаючи на основний потік документа. Унікальною особливістю елемента figure є можливість включення підпису до вмісту.
- Елемент figure зазвичай використовується для **групування зображення** або іншого вбудованого вмісту разом із підписом. Його також можна використовувати для групування тексту, наприклад, **лістингу коду** разом із підписом.
- Щоб додати підпис, включіть елемент figcaption в елемент фігури. Елемент figcaption повинен бути першим або останнім дочірнім елементом у фігурі. Підпис буде вище вмісту, якщо елемент figcaption є першим дочірнім елементом. Інакше він буде нижче вмісту.

```
<figure>
        <h1>Figure</h1>
        <img src="HTML5Badge.png" alt="HTML5" />
        <figcaption>Official HTML5 Logo</figcaption>
</figure>
```

Sidebars (бічні смуги)

- Бічні смуги представлені в HTML5 елементом <aside>. Не має значення, де буде розміщена бічна панель (праворуч чи ліворуч або вниз збоку), оскільки ви це визначите пізніше, використовуючи конкретні функції CSS.
- Блок <aside> містить інформацію про вміст веб-сторінки; однак він автоматично не є частиною вмісту веб-сторінки.
- Це смисловий елемент (semantic element).

Contents Range of a Simple HTML Site

```
1   <!doctype html>
2   <html>
3   <head>
4     <meta charset="utf-8">
5     <title>HTML5 Page</title>
6   </head>
7
8   <body>
9     <header>
10      <img src="logo.gif" alt="logo">
11      <h1>Title</h1>
12    </header>
13
14    <article>
15      <h1>Heading</h1>
16      <p>This is my first HTML5 page</p>
17      ... more content
18    </article>
19
```

```
20      <aside>
21        <section>
22          <h2>Contact</h2>
23          <ul>
24            <li><a href="link_1.html">Wiki</a></li>
25            <li><a href="link_2.html">Blog</a></li>
26            <li><a href="link_3.html">Forum</a></li>
27          </ul>
28        </section>
29
30        <section>
31        <h2>More links</h2>
32        <ul>
33          <li><a href="link_1.html">Wiki</a></li>
34          <li><a href="link_2.html">Blog</a></li>
35          <li><a href="link_3.html">Forum</a></li>
36        </ul>
37        </section>
38      </aside>
39
40     <footer>
41     </footer>
42    </body>
43    </html>
```

Для розрізнення діапазону вмісту є три елементи:

- <main> : основний вміст сайту
- <article>: стаття, що містить заголовок (і, можливо, деякі) розділи, заголовок і колонтитул)
- <section>: Розділ, як глава, …

Використовуйте <article>, якщо вміст складається з одного абзаца.

Використовуйте <section>, якщо є кілька подібних блоків.

Власне зміст прикладу на попередніх слайдах насправді не має значення. Тут важлива структура.

Групування елементів

- У попередньому розділі описані елементи, що використовуються **для організації** документа HTML у більш дрібні розділи, та ті елементи, що стосуються контуру документа.
- Тепер ми розглянемо **решту елементів групування**. Ці елементи використовуються в основному для семантичних цілей і не впливають на контури.

1) Paragraph (<p>)
2) Horizontal Rule (<hr>)

```
<p>paragraph 1</p>
<hr />
<p>paragraph 2</p>
```

3) Preformatted (<pre>)

```
<pre>I heard the bells on Christmas Day
Their old, familiar carols play,
    And wild and sweet
    The words repeat
Of peace on earth, good-will to men!</pre>
<cite>Henry Wadsworth Longfellow - 1863, public domain</cite>
```

I heard the bells on Christmas Day
Their old, familiar carols play,
    And wild and sweet
    The words repeat
Of peace on earth, good-will to men!

Henry Wadsworth Longfellow - 1863, public domain

4) Main

- main елемент використовується для вказівки на те, що його вміст є основною метою або темою документа. Це визначає суть, центральну тему документа.
- Таким чином, main елемент не може знаходитись у елементах article, aside, footer, header або nav елемента, оскільки вони використовуються для допоміжного вмісту.

5) Division

Embedded HTML Elements

1) **Гіперпосилання** (hyperlink, Anchor)

```
<a href="http://www.apress.com">
   Apress
</a>
```

- Атрибут **href** також може містити посилання на певний елемент на поточній сторінці. Наприклад, встановлення **href = "#Chapter5"** буде прокручувати поточний документ до того елемента, у якого атрибут id встановлений у Chapter5.
- Також пов'язаний ресурс не обов'язково повинен бути веб-ресурсом.

- http: - web resource
- ftp: - file transfer
- mailto: - send an email
- tel: - dial a phone number
- file: - open a file

- Атрибут target використовується для визначення місця, де має відображатися пов'язаний ресурс. (_self – на тій же сторінці (значення за замовчуванням), _blank – на новій вкладці браузера)
- Якщо пов'язаний ресурс - це файл, який слід завантажувати, а не надавати браузером, використовуйте атрибут download.
- Є кілька інших атрибутів, які можуть бути включені, які дають суто семантичну інформацію про пов'язаний ресурс:
  - **hreflang** - indicates the language of the linked resource
  - **rel** - defines the relationship with the linked resource
  - **type** - indicates the MIME type of the resource

2) Зображення (Images)

Вставити зображення у HTML-сторінку можна за допомогою тега **<img>.**

```
<img src="Media/HTML5.jpg" alt="The HTML5 Badge logo“ />
```

- Атрибут src - URL of the image file (the top three formats, – JPEG, GIF, and PNG)
- Є ще атрибути height і width (If used, the units must be set in pixels)
- Multiple Sources

Елемент зображення включає атрибути **srcset** та **size**, які дозволяють задавати набір файлів зображень з інформацією про них, щоб браузер міг вибрати найбільш відповідний для завантаження та візуалізації.

Pixel Ratio Selection

- На звичайному мобільному або планшетному пристрої, як правило, ви будете мати набагато більше пікселів за однаковий простір.
- Наприклад, 24-дюймовий монітор з плоским екраном має роздільну здатність 1920 x 1200 пікселів.
- 5-дюймовий телефон Lumia має майже стільки ж, 1280 x 720. Монітор має щільність пікселів **93** пікселів / дюйм (PPI); телефон - **294** PPI, що приблизно втричі перевищує щільність пікселів.

```
<img src="Media/HTML5.jpg" alt="The HTML5 Badge logo"
srcset="Media/HTML5_2.jpg 2x, Media/HTML5_3.jpg 3x"/>
```

Більш детально дивись Mark J. Collins **Pro HTML5 with CSS, JavaScript, and Multimedia** - Apress, 2017 p. 118

Створення карт-зображень (Image Map)

- Ви можете легко перетворити зображення на гіперпосилання, розмістивши його всередині тега <a> таким чином:

```
<a href="https://html.spec.whatwg.org/multipage/embedded-content.html">
    <img src="Media/HTML5.jpg" alt="The HTML5 Badge logo" />
</a>
```

- В HTML можна створювати спеціальні карти-зображення, які містять список областей-посилань і прив'язуються до зображень. Карти в HTML створюються за допомогою тега **<map>,** а області-посилання на них з допомогою тега **<area>.** Форма і координати областей-посилань задаються з допомогою атрибутів **shape** і **coords** тега **<area>,** а місце, на яке вони посилаються, - за допомогою атрибута **href.**

```
<map name="shapeMap">
  <area shape="rect" coords="0,0,50,50"
      alt="square" title="Square"
      href="https://en.wikipedia.org/wiki/Square" />
  <area shape="circle" coords="75,25,25" alt="circle"
      title="Circle"
      href="https://en.wikipedia.org/wiki/Circle" />
  <area shape="poly" coords="101,50,126,0,150,50"
      alt="triangle" title="Triangle"
      href="https://en.wikipedia.org/wiki/Triangle" />
</map>
```

3) Audio 4) Video

```
<audio src="Media/Linus and Lucy.mp3“autoplay controls>
    <p>HTML5 audio is not supported on your browser</p>
</audio>
```


```
<video src="Media/BigBuckBunny.mp4" controls
  poster="Media/BBB_Poster.png" width="852" height="480">
  <p> HTML5 video is not supported on your browser</p>
</video>
```

Елементи списків

- HTML5 підтримує як упорядкований список за допомогою елемента **ol**, так і не упорядкований список за допомогою елемента **ul**.

```
<h2>Book Topics</h2>
<ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
</ul>
<h2>HTML Chapters</h2>
<ol start="4">
    <li>Structural Elements</li>
    <li>Text Elements</li>
    <li>Table Elements</li>
    <li>Embedded Elements</li>
    <li>Form Elements</li>
</ol>
```

Book Topics

- HTML
- CSS
- JavaScript

HTML Chapters

4. Structural Elements
5. Text Elements
6. Table Elements
7. Embedded Elements
8. Form Elements

Для тегу **<ol>** доступні наступні атрибути:

- **start** - задає початкове значення, від якого починається відлік нумерації
- **type** - задає тип нумерації для використання в списку (у вигляді букв або цифр). Прийняті значення:

  1 - значення за замовчуванням, десяткова нумерація.

  A - нумерація списку в алфавітному порядку, заголовні букви (A, B, C, D).

  a - нумерація списку в алфавітному порядку, малі літери (a, b, c, d).

  I - нумерація римськими великими цифрами (I, II, III, IV).

  i - нумерація римськими малими цифрами (i, ii, iii, iv).

Список визначень (Description List)

- Списки визначень описуються за допомогою тегу <dl> , який містить пари термін-визначення. Для додавання терміну застосовується блоковий тег <dt> , а для уставляння визначення - блочний парний тег <dd> .

```
<dl>
    <dt>Term1</dt>
    <dd>Definition</dd>
    <dt>Term2</dt>
    <dd>Definition</dd>
</dl>
```

Another use of the dl element is to list groups of things with a group header.

```
<h2>New York Yankees Starting
    Lineup</h2>
<dl>
<dt>Infielders</dt>
<dd>Teixeira, 1B</dd>
<dd>Castro, 2B</dd>
<dd>Gregorius, SS</dd>
<dd>Torreyes, 3B</dd>
<dd>McCann, C</dd>
<dt>Outfielders</dt>
<dd>Hicks, LF</dd>
<dd>Ellsbury, CF</dd>
<dd>Ackley, RF</dd>
<dt>Designated Hitter</dt>
<dd>Rodriguez, DH</dd>
<dt>Pitchers</dt>
<dd>Gausman, R</dd>
<dd>Wilson, R</dd>
<dd>Tillman, R</dd>
<dd>Price, L</dd>
<dd>Porcello, R</dd>
</dl>
```

New York Yankees Starting Lineup

Infielders
- Teixeira, 1B
- Castro, 2B
- Gregorius, SS
- Torreyes, 3B
- McCann, C

Outfielders
- Hicks, LF
- Ellsbury, CF
- Ackley, RF

Designated Hitter
- Rodriguez, DH

Pitchers
- Gausman, R
- Wilson, R
- Tillman, R
- Price, L
- Porcello, R

Вкладений список

```
<ul>
  <li>Пункт 1.</li>
  <li>Пункт 2.
  <ul>
    <li>Підпункт 2.1.</li>
    <li>Підпункт 2.2.
      <ul>
        <li>Підпункт 2.2.1.</li>
        <li>Підпункт 2.2.2.</li>
      </ul>
    </li>
    <li>Підпункт 2.3.</li>
  </ul>
  </li>
  <li>Пункт 3.</li>
</ul>
```

- Пункт 1.
- Пункт 2.
  - Підпункт 2.1.
  - Підпункт 2.2.
    - Підпункт 2.2.1.
    - Підпункт 2.2.2.
  - Підпункт 2.3.
- Пункт 3.

Вбудовані кадри (Inline Frames)

- Елемент **iframe** використовується для вбудовування іншої веб-сторінки в поточний документ. Ви можете включити iframe з розміткою так:

```
<iframe src="http://www.apress.com" width="100%"
  height="400">
<p>Your browser does not support iframes</p>
</iframe>
```

Демонстрація

- Chapter04 - файли до книги Mark J. Collins **Pro HTML5 with CSS, JavaScript, and Multimedia** - Apress, 2017

Таблиці HTML

- Таблиця - це структурований набір даних, що складається з рядків і стовпців (табличних даних). Таблиці дозволяють швидко і легко подивитися значення, що показують деяку взаємозв'язок між різними типами даних, наприклад - людина і його вік, або розклад в плавальному басейні.

Person	Age
Chris	38
Dennis	45
Sarah	29
Karen	47

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Public Swim 06:30 - 10:30	Public Swim 06:30 - 09:00	Public Swim 06:30 - 09:00	Public Swim 06:30 - 11:15	Public Swim 06:30 - 09:00	Lane Swim 08:00 - 09:00	Lane Swim 08:00 - 09:00
Aquacise 10:30 - 11:15	Aqua Jog 09:15 - 10:00	Education Swimming Lessons 09:00 - 12:00	Aquacise 11:15 - 12:00	Education Swimming Lessons 09:00 - 12:00	Oldham Active Kids Swimming Lessons 09:00 - 13:00	Public Swim 09:00 - 11:00
Lane Swim 11:30 - 13:00	Parent & Baby Class 09:30 - 10:15	Lane Swim 12:00 - 13:00	Lane Swim 12:00 - 13:00	Lane Swim 12:00 - 13:00	Parent and Baby 12:00 - 12:45	Aquacise 11:00 - 11:45
Education Swimming Lessons	Public Swim 10:00 - 11:45	Public Swim 13:00 - 16:00	Education Swimming Lessons	Oldham Active Kids Swimming	Public Swim 13:00 - 17:00	Public Swim 11:45 - 13:00

Коли не треба використовувати таблиці HTML?

- HTML-таблиці слід використовувати для табличних даних - це те, для чого вони призначені.
- На жаль, багато хто використовує таблиці HTML для оформлення веб-сторінок, наприклад, один рядок для заголовка і т.д.
- Це відбувалося через погану підтримки CSS в різних браузерах; в наш час таке зустрічається набагато рідше, але іноді все ж трапляється.

Таблиці HTML

Для побудови таблиці використовують теги-контейнери:

- тег таблиці **<table>,**
- тег комірки заголовка **<th>,**
- тег рядка **<tr>** та
- тег комірки даних **<td>.**

Таблиця має бути розміщена в тілі HTML-документа, будується вона рядками, причому кількість комірки в рядках повинна бути однакова.

Таким чином одній таблиці відповідає один тег **<table>** та стільки тегів **<tr>,** скільки рядків має таблиця. Якщо в таблиці немає об'єднаних комірок, то сумарна кількість тегів **<td>** та **<tr>** дорівнює добутку кількості рядків на кількість колонок таблиці.

Атрибути тега <table>

Атрибут	Призначення
border	Товщина межі таблиці в пікселях
bordercolor	Колір меж таблиці
cellspasing	Відстань між комірками таблиці в пікселях
cellpadding	Відстань від меж комірок до даних, що розміщені в цих комірках
width	Ширина таблиці в пікселях або відсотках від ширини HTML-документа
height	Висота таблиці в пікселях
align	Горизонтальне вирівнювання таблиці
bgcolor	Колір фону таблиці
background	Фоновий рисунок таблиці

Приклад 1. Проста таблиця

```
<table>
  <tr>
    <td>One</td>
    <td>Two</td>
    <td>Three</td>
  </tr>
  <tr>
     <td>Four</td>
     <td>Five</td>
     <td>Six</td>
  </tr>
  <tr>
      <td>Seven</td>
      <td>Eight</td>
      <td>Nine</td>
  </tr>
</table>
```

One	Two	Three
Four	Five	Six
Seven	Eight	Nine

Приклад 2. Таблиця з заголовками

```
<table>
  <caption>Squares and
   Cubes</caption>
  <tr>
     <td></td> <!-- empty cell -->
     <th scope="col">Number</th>
     <th scope="col">Squared</th>
     <th scope="col">Cubed</th>
  </tr>
  <tr>
     <th scope="row">Two</th>
     <td>2</td>
     <td>4</td>
     <td>8</td>
   </tr>
   <tr>
     <th scope="row">Three</th>
     <td>3</td>
     <td>9</td>
     <td>27</td>
   </tr>
   <tr>
     <th scope="row">Four</th>
     <td>4</td>
     <td>16</td>
     <td>64</td>
   </tr>
</table>
```

	Number	Squared	Cubed
Two	2	4	8
Three	3	9	27
Four	4	16	64

Приклад 3. Групування колонок

```
<table>
    <caption>Squares and Cubes</caption>
    <colgroup span="1"></colgroup>
    <colgroup span="1"></colgroup>
    <colgroup span="2" style="background-color:
                                #b6ff00"></colgroup>
    <tr> . . . .
```

	Number	Squared	Cubed
Two	2	4	8
Three	3	9	27
Four	4	16	64

Приклад 4. Заголовок таблиці та колонтитул

```
<table>
  <caption>Scoreboard</caption>
  <thead>
  <tr>
      <th>Inning</th>
     <th>Runs</th>
     <th>Hits</th>
     <th>Errors</th>
  </tr>
  </thead>
  <tbody>
      <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
      <tr><td>2</td><td>2</td><td>5</td><td>0</td></tr>
      <tr><td>3</td><td>0</td><td>1</td><td>1</td></tr>
      <tr><td>4</td><td>0</td><td>0</td><td>0</td></tr>
      <tr><td>5</td><td>1</td><td>2</td><td>0</td></tr>
      <tr><td>6</td><td>0</td><td>1</td><td>0</td></tr>
      <tr><td>7</td><td>0</td><td>0</td><td>0</td></tr>
```

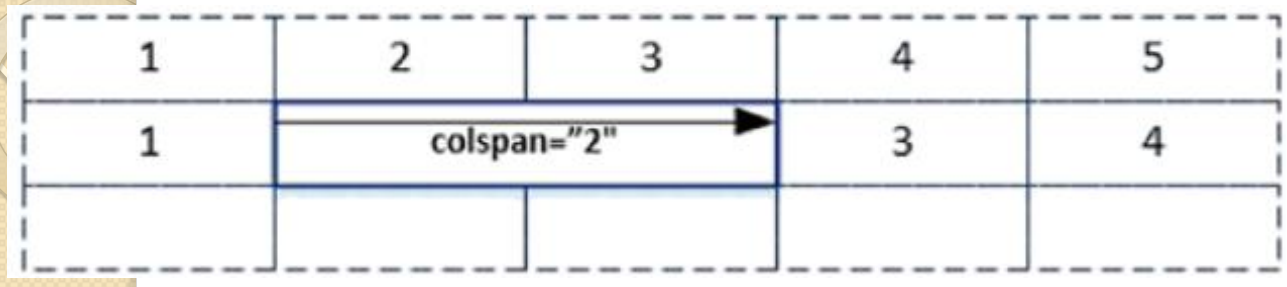
```
        <tr><td>8</td><td>1</td><td>3</td><td>1</td></tr>
        <tr><td>9</td><td>1</td><td>2</td><td>0</td></tr>
    </tbody>
    <tfoot>
        <tr><td>Final</td><td>5</td><td>15</td><td>2</td></tr>
    </tfoot>
</table>
```

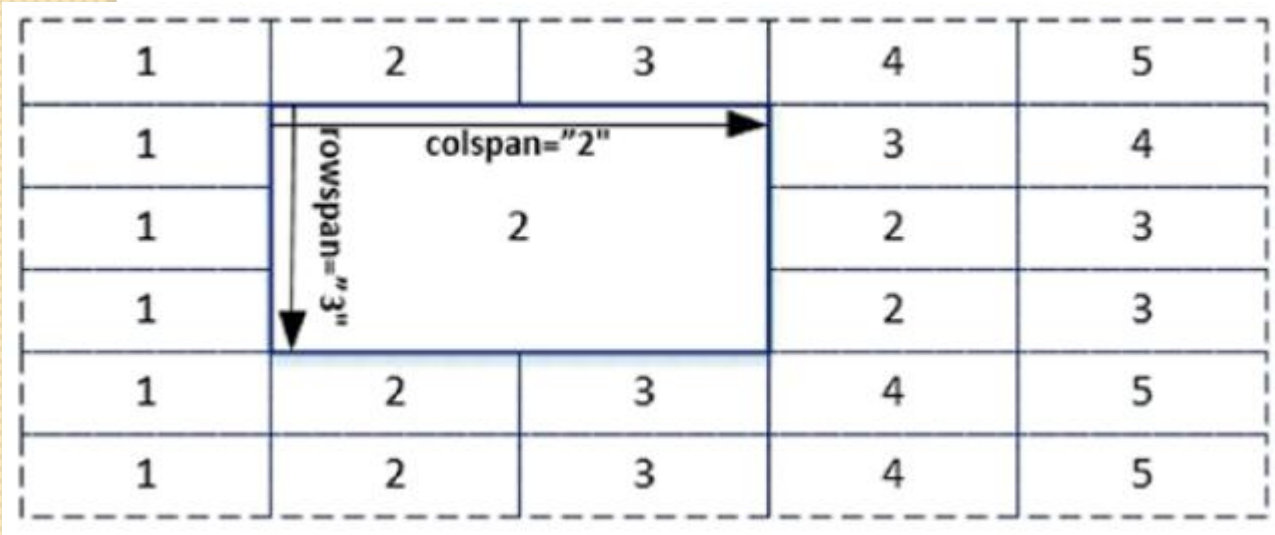
Inning	Runs	Hits	Errors
1	0	1	0
2	2	5	0
3	0	1	1
4	0	0	0
5	1	2	0
6	0	1	0
7	0	0	0
8	1	3	1
9	1	2	0
Final	5	15	2

Об’єднання комірок таблиці

- Працюючи з таблицями в HTML5, ви можете зробити одну клітинку з декількох сусідніх комірок.
- Це працює так само, як об’єднання комірок у електронній таблиці.
- Комірки таблиці HTML складаються з елементів <td> та <th>. Кожен з цих елементів займає одну клітинку в сітці таблиці.
- Однак обидва ці елементи підтримують атрибути colspan та rowspan. Значення за замовчуванням обох атрибутів дорівнює 1, тому кожен елемент займає рівно одну комірку.

Використання атрибутів colspan та rowspan



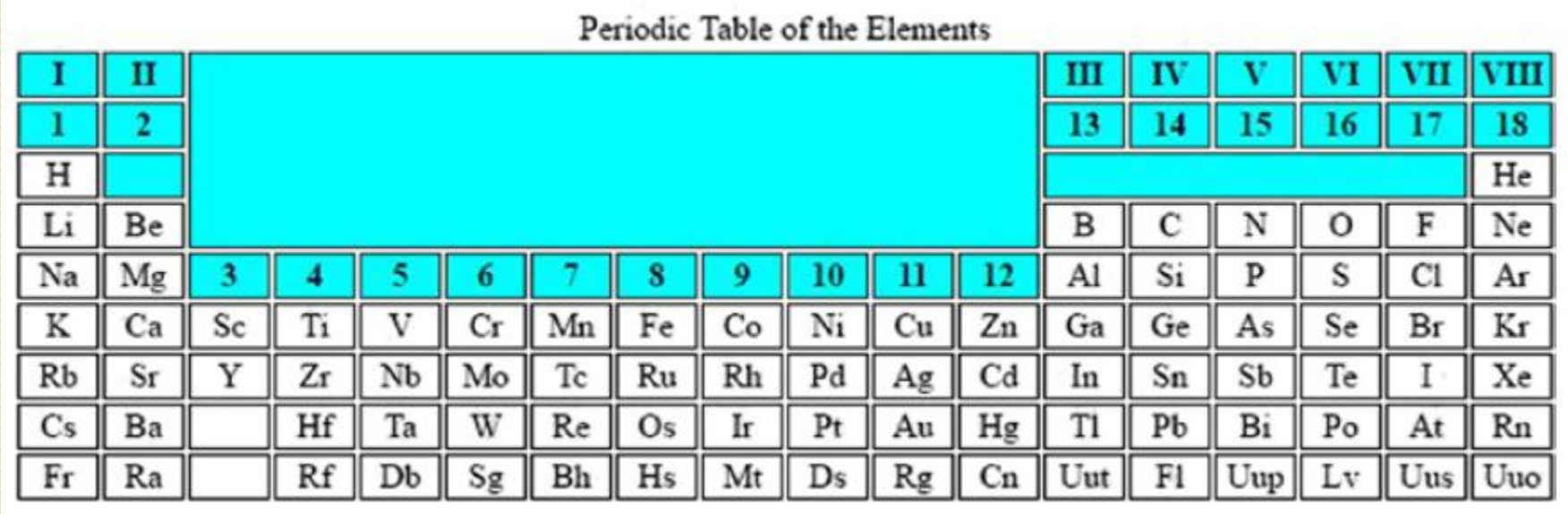


Ще дивись приклад Periodic Table of the Elements

Chapter04 - файли до книги Mark J. Collins **Pro HTML5 with CSS, JavaScript, and Multimedia** - Apress, 2017

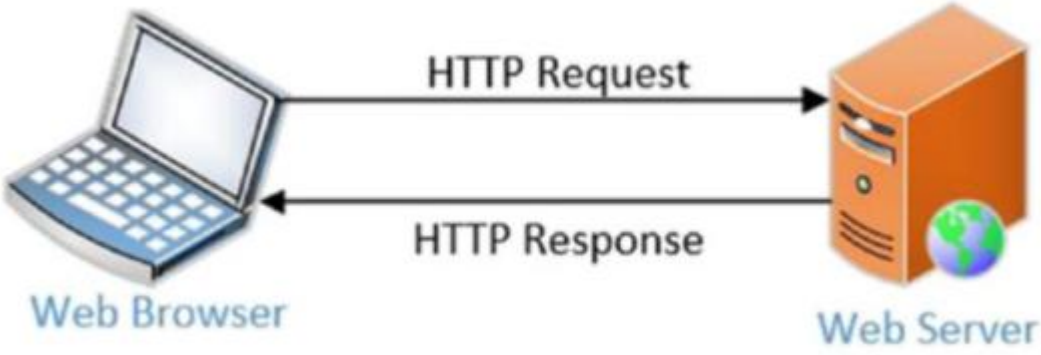
Приклад Periodic Table of the Elements

Chapter04 - файли до книги Mark J. Collins **Pro HTML5 with CSS, JavaScript, and Multimedia** - Apress, 2017



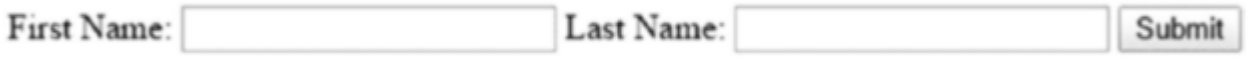
HTML Form Elements

- У традиційних веб-додатках форма - це веб-сторінка або її частина, яка **містить місця для введення користувачем інформації**.
- Після введення даних форма подається на сервер разом із вхідними даними. Це обробляється сервером, і нова сторінка повертається та надається клієнту.



Form Element

```
<form action="" method="get">
   <label for="iFirstName">First Name:</label>
   <input id="iFirstName" type="text" />
   <label for="iLastName">Last Name:</label>
   <input id="iLastName" type="text" />
   <input type="submit" value="Submit" />
</form>
```



Form Action

Якщо ви введете дані в ці поля і натисните кнопку “Submit", буде надісланий запит HTML, подібний до цього:

http://localhost:5266/?FirstName=**Mark**&LastName=**Collins**

Form Method

POST http://localhost:5266/ HTTP/1.1
Host: localhost:5266
Connection: keep-alive
Content-Length: 31
Cache-Control: max-age=0
Origin: http://localhost:5266
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://localhost:5266/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8

FirstName=Mark&LastName=Collins

Найбільш популярні елементи форми

`input` Element
button
text control
number
radio button
checkbox
password
date
color

`select` Element
pull-down menu
list box

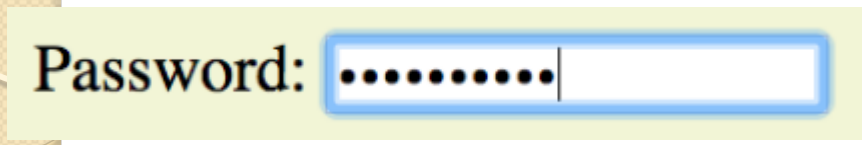
`textarea` Element
textarea control

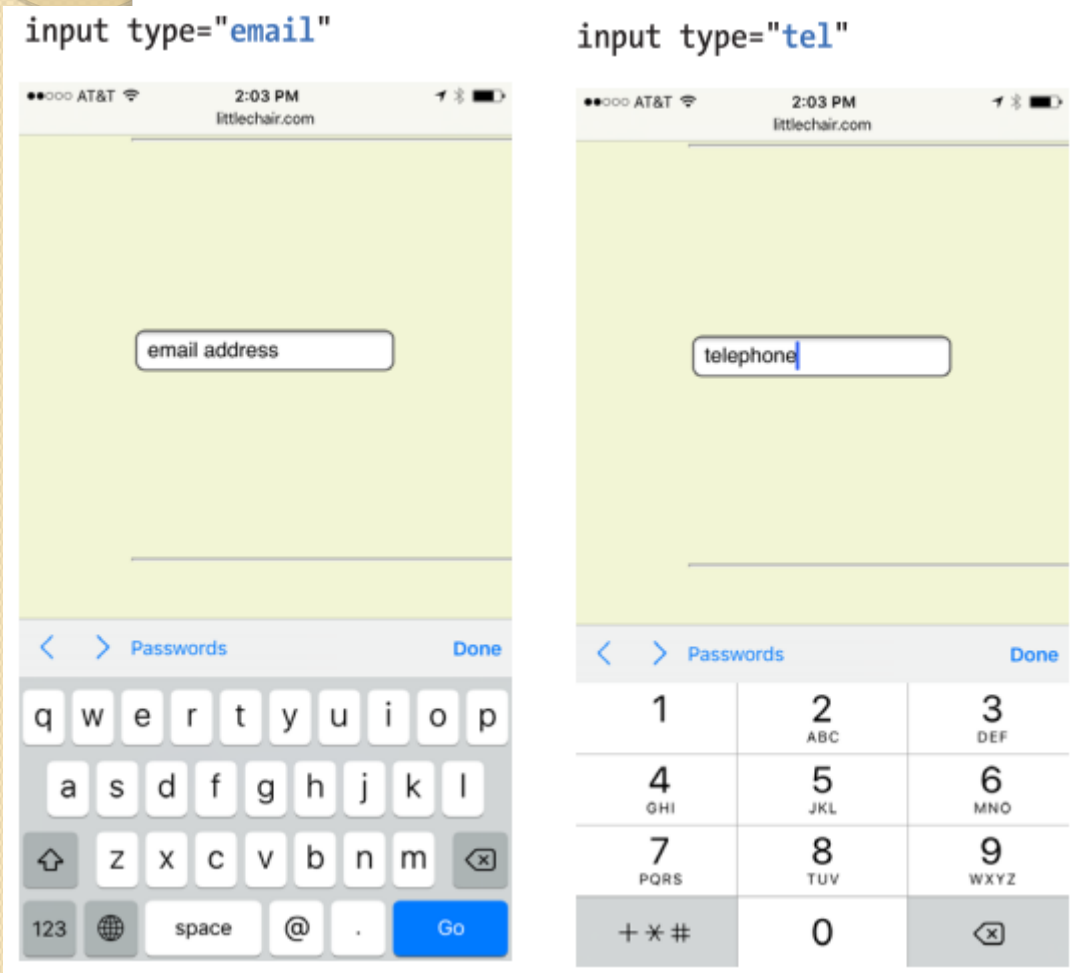
Text Values (для input елемента)

Існує кілька типів введення, які в основному є текстовими полями, які забезпечують перевірку даних на основі заданого атрибута type. Такі типи:

- **text** - це за замовчуванням, якщо атрибут type не вказаний
- **email** - це працює як звичайне текстове поле, за винятком вбудованої перевірки, щоб перевірити, чи формат тексту відповідає стандартній адресі електронної пошти.
- **password** - це просто відображає введені символи як зірочки або якийсь інший метод затемнення вводу
- **tel** - використовувати це для введення телефонних номерів;
- **url** – як для email type, перевіряється, що введений текст є добре сформованою URL-адресою.
- **search** – для пошуку

```
<li><label for="form-pswd">Password:</label><br>
 <input type="password" name="pswd" maxlength="12" id="form-pswd"></li>
```

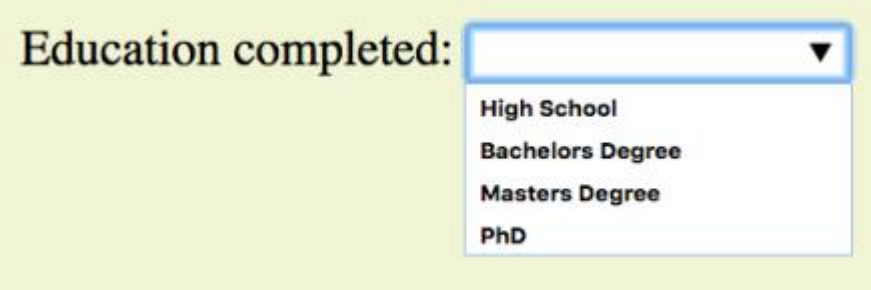




Safari on iOS provides custom keyboards based on the input type

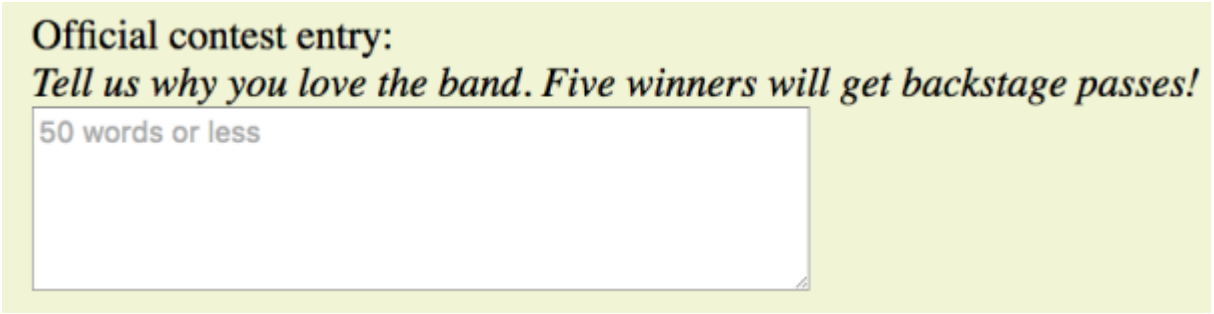
Drop-down menu input

```
<p>Education completed: <input type="text" list="edulevel" name="education">
<datalist id="edulevel">
  <option value="High School">
  <option value="Bachelors Degree">
  <option value="Masters Degree">
  <option value="PhD">
</datalist>
```



Multiline text-entry field (Textarea)

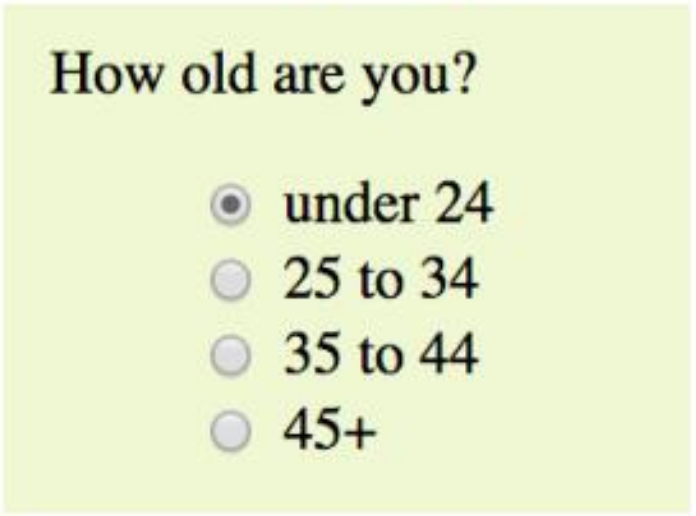
```
<p> Official contest entry: <br>
    <em>Tell us why you love the band. Five winners will get
          backstage passes!</em><br>
    <textarea name="contest_entry" placeholder="50 words or
                less" rows="5" cols="50">
    </textarea>
</p>
```



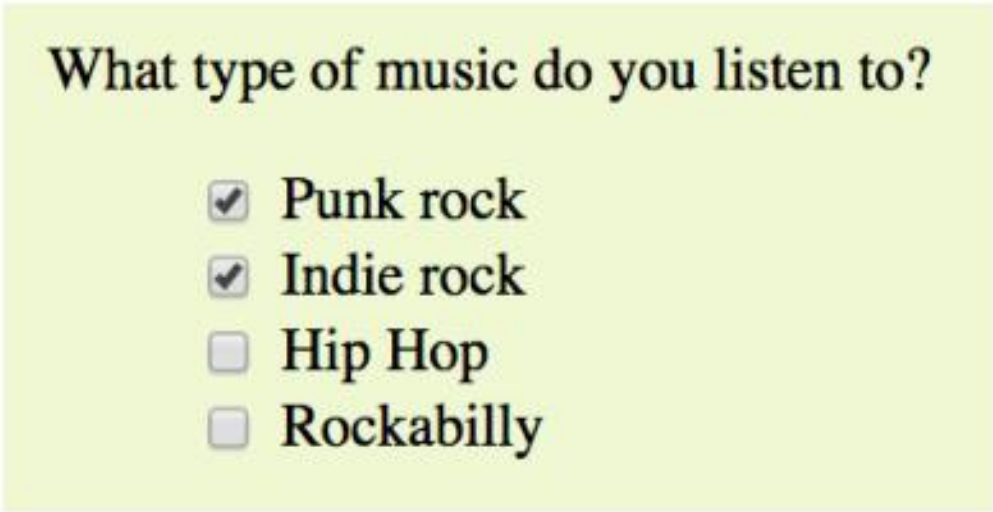
Radio and Checkbox Buttons

```
<p>How old are you?</p>
<ol>
  <li><input type="radio" name="age" value="under24" checked> under 24</li>
  <li><input type="radio" name="age" value="25-34"> 25 to 34</li>
  <li><input type="radio" name="age" value="35-44"> 35 to 44</li>
  <li><input type="radio" name="age" value="over45"> 45+</li>
</ol>
```

Radio buttons (`input type="radio"`)



```
<p>What type of music do you listen to?</p>
<ul>
  <li><input type="checkbox" name="genre" value="punk" checked> Punk rock</li>
  <li><input type="checkbox" name="genre" value="indie" checked> Indie rock</li>
  <li><input type="checkbox" name="genre" value="hiphop"> Hip Hop</li>
  <li><input type="checkbox" name="genre" value="rockabilly"> Rockabilly</li>
</ul>
```

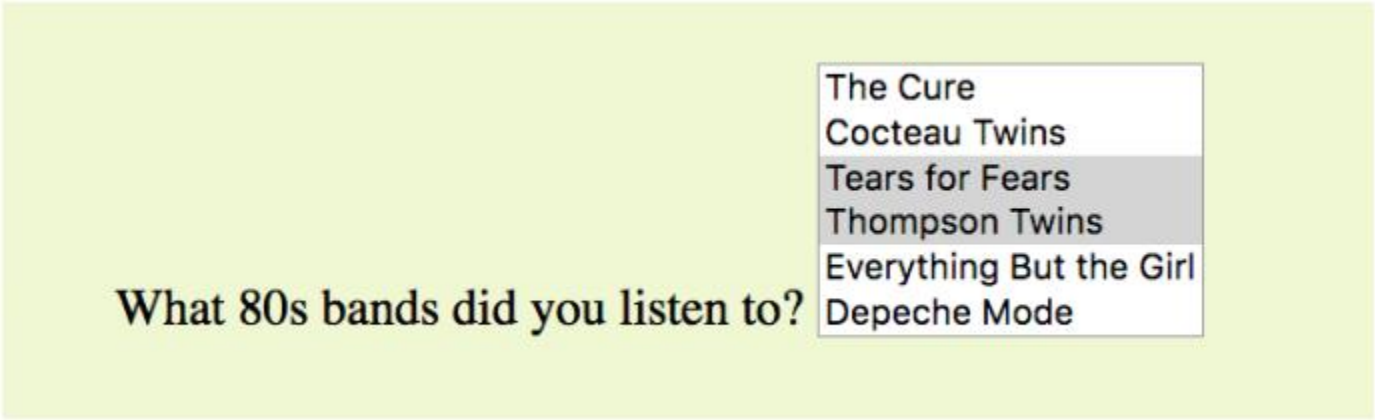


Drop-down menus

```
<p>What is your favorite 80s band?
<select name="EightiesFave">
   <option>The Cure</option>
   <option>Cocteau Twins</option>
   <option>Tears for Fears</option>
   <option>Thompson Twins</option>
   <option value="EBTG">Everything But the Girl</option>
   <option>Depeche Mode</option>
   <option>The Smiths</option>
   <option>New Order</option>
</select>
</p>
```

Scrolling menus

```
<p>What 80s bands did you listen to?
<select name="EightiesBands" size="6" multiple>
   <option>The Cure</option>
   <option>Cocteau Twins</option>
   <option selected>Tears for Fears</option>
   <option selected>Thompson Twins</option>
   <option value="EBTG">Everything But the Girl</option>
   <option>Depeche Mode</option>
   <option>The Smiths</option>
   <option>New Order</option>
</select>
</p>
```

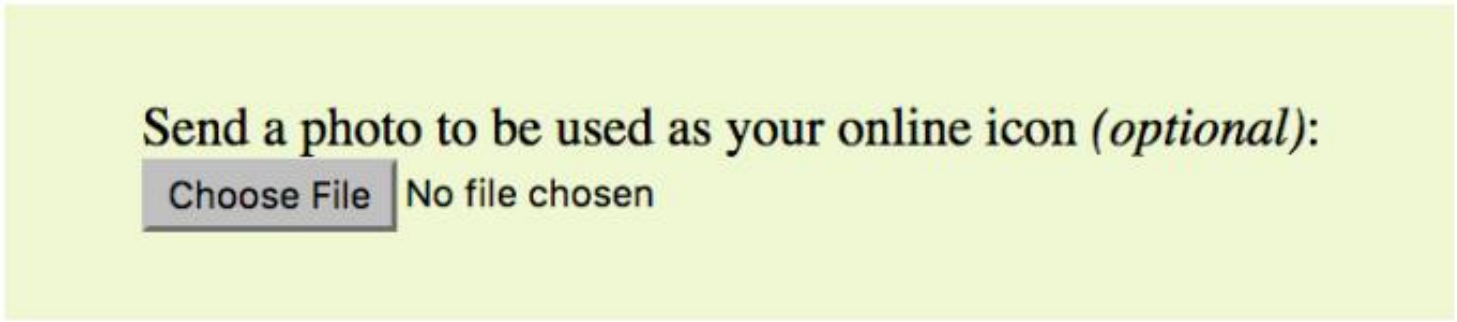


Grouping menu options

```
<select name="icecream" size="7" multiple>
  <optgroup label="traditional">
    <option>vanilla</option>
    <option>chocolate</option>
  </optgroup>
  <optgroup label="fancy">
    <option>Super praline</option>
    <option>Nut surprise</option>
    <option>Candy corn</option>
  </optgroup>
</select>
```

File Selection Control

```
<form action="/client.php" method="POST" enctype="multipart/form-data">
  <label>Send a photo to be used as your online icon <em>(optional)
</em><br>
<input type="file" name="photo"></label>
</form>
```



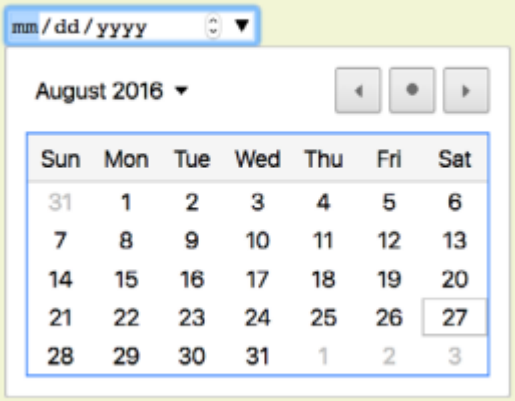
Hidden Controls

- Можливо, вам доведеться надсилати інформацію в додаток для обробки форми, яка не надходить від користувача. У цих випадках ви можете використовувати прихований елемент управління, який надсилає дані під час подання форми, але його не видно, коли форма відображається в браузері.

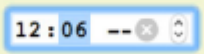
```
<input type="hidden" name="success-link"
    value="http://www.example.com/ thankyou.html">
```

Date and Time Controls

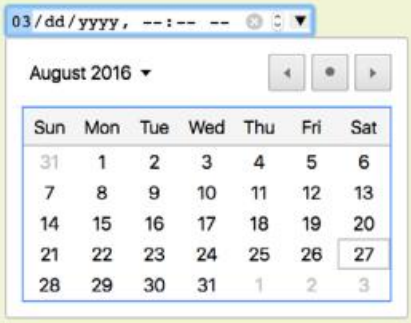
input type="date"



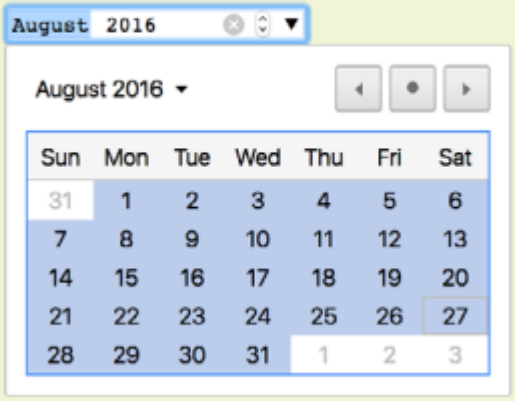
input type="time"



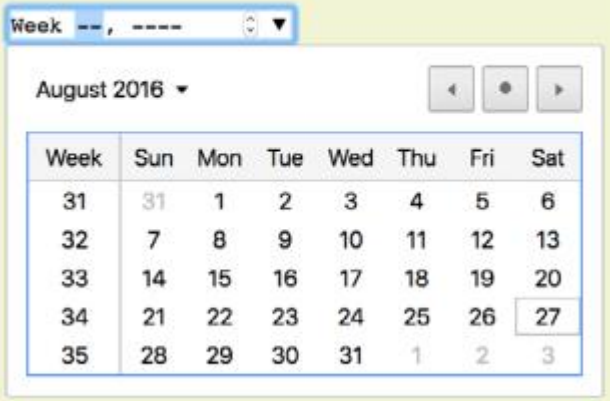
input type="datetime-local"



input type="month"

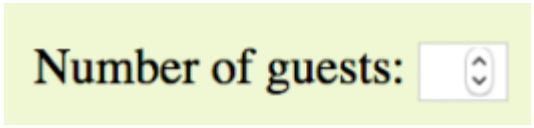


input type="week"

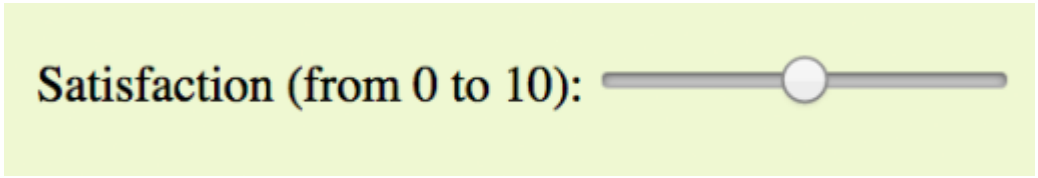


Numerical Inputs

```
<label>Number of guests <input type="number" name="guests"
   min="1"
max="6"></label>
```




```
<label>Satisfaction (0 to 10) <input type="range"
   name="satisfaction"
min="0" max="10" step="1"></label>
```



Color Selector

```
<label>Your favorite color: <input type="color"
   name="favorite">
</label>
```

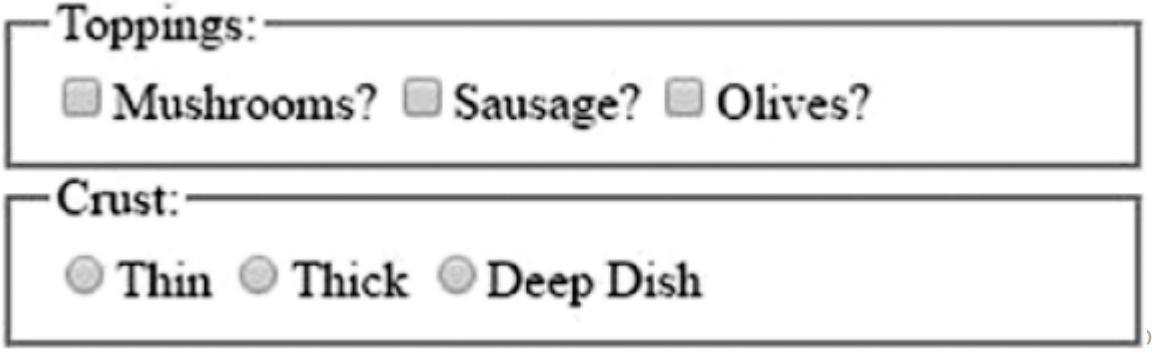
Progress Element

```
<p>Progress example:
<progress min="0" max="7" value="3">
Your browser does not support the progress element!
   Value is 3 of 7.
</progress>
</p>
```

Organizing a Form

```
<fieldset>
<legend>Toppings:</legend>
<input type="checkbox" name="Topping" value="Mushrooms"
   />Mushrooms?
<input type="checkbox" name="Topping" value="Sausage" />Sausage?
<input type="checkbox" name="Topping" value="Olives" />Olives?
</fieldset>

<fieldset>
<legend>Crust:</legend>
<input type="radio" name="Crust" value="Thin" required />Thin
<input type="radio" name="Crust" value="Thick" />Thick
<input type="radio" name="Crust" value="DeepDish" />Deep Dish
</fieldset>
```



Кольори безпечної веб-палітри

- Це такі кольори, які не підлягають змішуванню на 256-кольорових моніторах.
- Кольори безпечної веб-палітри можна виразити RGB-кодами 00, 33,66, 99, CC, FF
- **ТАБЛИЦЯ БЕЗПЕЧНИХ КОЛЬОРІВ**

http://getcolorcode.com/ua/colors/websafe