

# Webpages

1

---

---

---

---

---

---

---

## Basic HTML Page

- HTML tags typically come in pairs
  - `<tagname> ... </tagname>`
- Nesting tags
  - Proper nesting `<tag1>... <tag2>... </tag2> </tag1>`



```
1 <!doctype html>
2 <html>
3   <head>
4     <title> My first webpage! </title>
5   </head>
6   <body>
7     Hello world!!
8   </body>
9 </html>
```

2

---

---

---

---

---

---

---

## Core HTML tags

- `<!doctype html>`
  - Tell the browser that this file is standard HTML 5
- `<html></html>`
  - Container for all html tags
- `<head></head>`
  - Header information (e.g. Title, Script files, etc, etc).
- `<title></title>`
  - Sets the title of the webpage
- `<body></body>`
  - Main content of the webpage

3

---

---

---

---

---

---

---

## HTML Terminology

- Tags
  - `<body>`, `<title>`, `<font>`
- Elements
  - `<title>This is a title</title>`
- Attributes
  - `<p title="I'm a tooltip!">`
    - Attribute: `name="value"`

4

---

---

---

---

---

---

---

## Text Formatting

- Simple formatting tags
  - `<b> Bold </b>`
  - `<i> Italics </i>`
  - `<u> Underline </u>`
  - `<s> Strikethrough </s>`
  - `<sup> Superscript </sup>`
  - `<sub> Subscript </sub>`
  - `<big> Big Text </big>`
  - `<small> Small text </small>`
  - `<mark> Highlighted </mark>`

5

---

---

---

---

---

---

---

## Text Layout

- Header Tags
  - `<h1>Header</h1>`
    - `<h2>Sub header</h2>`
    - `<h3>Sub sub header</h3>`
    - `<h4> ... </h4>`
    - `<h5> ... </h5>`
    - `<h6> ... </h6>`
- The paragraph tag
  - `<p>This is a paragraph of text</p>`

6

---

---

---

---

---

---

---

## Text Layout (2)

- Line break tag
  - `<br/>`
  - Line breaks (enter key) in the html code are ignored, must use the break tag
  - Note: "Empty element" (not a paired tag)
- Horizontal Rule
  - `<hr/>`
  - `<hr width="50%" />`

7

---

---

---

---

---

---

---

## Pre-format Tag

- HTML will only display one space between words
  - All other whitespace is ignored
- `<pre>` tag
  - Maintains all of the whitespaces as written in the source code.
  - `<pre>` This text      contains  
                         many spaces      in the  
                         source.  
`</pre>`

8

---

---

---

---

---

---

---

## HTML entities

- `&nbsp;`;
  - Non-breaking space
- `&lt;` and `&gt;`;
  - `<` and `>`
- `&amp;`;
  - Ampersand (&)
- `&copy;`;
  - ©

9

---

---

---

---

---

---

---

## Lists

- `<ul>` Unordered lists
  - `<li>` List item `</li>`
  - `<li>` Another list item `</li>``</ul>`
- `<ol>` Ordered lists
  1. `<li>` List item `</li>`
  2. `<li>` Another list item `</li>``</ol>`

10

---

---

---

---

---

---

---

## List Attributes

- `<ol type="a" start="3">`
  - c. 'type' attribute specifies the type of marker.
  - d. Options include: 1, a, A, i, I
  - e. 'start' attribute specifies the first value of the list
  - f. Values must be a number.

11

---

---

---

---

---

---

---

## Tables

- `<table> ... </table>`
  - Encloses the definition of a table
- `<tr> ... </tr>`
  - Encloses a single row in a table
- `<th> ... </th>`
  - Encloses a single table heading cell
- `<td> ... </td>`
  - Encloses a single table data cell

12

---

---

---

---

---

---

---

## Simple Table Example

- `<table>`
  - `<tr><th>Title</th></tr>`
  - `<tr><td>Row 1</td></tr>`
  - `<tr><td>Row 2</td></tr>`
  - `<tr><td>Row 3</td></tr>`
- `</table>`



Title  
Row 1  
Row 2  
Row 3

13

---

---

---

---

---

---

---

## Table Cell Attributes

- **colspan**
  - Number of columns a cell should span
- **rowspan**
  - Number of rows a cell should span

th colspan="2"		th	
td	td	td rowspan="2"	
td	td		

14

---

---

---

---

---

---

---

## Comment tags

- `<!-- Your comment here -->`
- Does not affect the webpage at all
  - Used for explaining your html code
  - Good for labelling parts of large files
  - Temporarily removing code without deleting it

15

---

---

---

---

---

---

---

## Hyperlinks

- `<a href="url">Sample link</a>`
  - The “anchor” (`<a>...</a>`) tag creates hyperlinks
  - The href attribute specifies where that link points
  - The element body is what’s shown.
  - E.g.: [Sample link](#)
  - URL can be a local file, e.g., “page2.html”
    - Use `../` to go up a level, e.g., “../otherfolder/page3.html”
  - Or a web address, e.g., “http://www.google.ca”

16

## Images!

- ``
  - The img tag is an empty tag (no closing tag)
  - src attribute specifies what the image is
- Other attributes
  - title – specifies the mouseover text
  - height/width – in pixels or in %

17

## Invisible Elements

- `<div> ... </div>`
  - Invisible *block-level* element
  - Can be used for grouping areas of a page
- `<span> ... </span>`
  - Invisible *inline* element
  - Can be use for grouping sections of text

18

## Element IDs

- All tags support the **id** attribute
  - Gives a unique name to the element
  - `<h1 id="myHeadline"> ... </h1>`
  - `<p id="introParagraph"> .... </p>`
  - No spaces, must contain at least one character
- Can be referred to in hyperlinks
  - `<a href="aPage.html#myHeadline"> Wow! </a>`
  - `<a href="#introParagraph"> Start here </a>`

19

---

---

---

---

---

---

---

## Element Classes

- Global attribute **class**
  - `<p class="warning">...</p>`
  - Like IDs but not unique
  - Specify multiple elements to be treated the same
  - Used for css styling and javascript selection

20

---

---

---

---

---

---

---

## Style Attribute

- `style="property:value;"`
  - Global attribute
  - Specifies the style for a specific element
- E.g.: `<p style="color:red"> ... </p>`

It was the best of times it was the worst of times

21

---

---

---

---

---

---

---

## Color values

- Constants
  - E.g.: Red, Blue, Cyan, Grey, Magenta, ForestGreen, Salmon, LightSalmon, LightGoldenRodYellow,...
- Hexadecimal (0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f)
  - #RRGGBB
  - E.g.: #ff0000, #00ff00, #0000ff, #008a8a, #A0522D
- rgba(255,255,128,0.6)

22

## Color Styles

- Background-color: colorValue;
  - Changes the background colour of the element
- Color: colorValue;
  - Changes the colour of the text of the element
- <table border="1" style="color:lightgreen; background-color: #000"> ... </table>

heading1	heading2	heading3
data	data	data
data	data	data

23

## Background Image Styles

- Background-image: url('Image.gif');
  - Add a background image to an element
  - Can use the body element to set the whole page
- Background-repeat: repeat;
  - Determine if the background image should tile
  - Other options are repeat-x, repeat-y, no-repeat
- Background-attachment: fixed;
  - Background stays static while text scrolls

24



## Border Styles

- **Border-style: solid;**
  - Specifies the type of border
  - Other options: none, dotted, dashed, double
    - Groove, ridge, inset, outset
- **Border-width: 5px;**
  - Specifies the width of the border in pixels
- **Border-color: blue;**
  - Specifies the colour of the border.

25

---

---

---

---

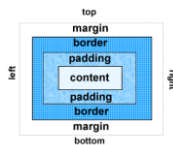
---

---

---

## Margins and Padding

- **Margin: top right bottom left;**
- **Padding: top right bottom left;**
  - Where each value is a number (in px, pt, cm, or %)
  - Can also be specified individually
    - **Margin-top: 10cm;**



26

---

---

---

---

---

---

---

## Sizing & Placement

- **Width: 50%;**
- **Height: 200px;**
  - Sets the size of the content of an element.
  - Before paddings and margins are considered
- **Float: left;**
  - "floats" a block element in its container
  - inline elements will wrap around it
- **clear: both;**
  - Element cannot be next to a float

27

---

---

---

---

---

---

---

## Positioning

- Specify an element outside of the normal flow
  - `position:static;` -- Normal flow (default)
  - `position:relative;` -- Relative shift (from normal)
  - `position:absolute;` -- Shift from top left corner of container element
  - `position:fixed;` -- relative to browser viewport, does not scroll
- Set shifts, e.g.:  
`left: 10px; top:10%;`

28

---

---

---

---

---

---

---

## Text Styling

- Color: `#55cc99;`
  - Color of the text
- Text-align: `left/right/center;`
  - Aligns the text within its containing element
- Font-family: `'Times New Roman';`
  - Specifies the font to use
    - Can specify a comma separated list of fallbacks
    - Font-family: `'Times New Roman', Times, serif;`
- Font-size: `30px;`

29

---

---

---

---

---

---

---

## CSS Files

- HTML is for the content of a webpage.
  - Not intended for style information.
- Create a single CSS file.
  - Instead of repeating style attributes all over
  - Link to it by adding a link tag in the `<head>` section
  - `<link rel="stylesheet" type="text/css" href="myStyle.css" />`

30

---

---

---

---

---

---

---

## CSS Styles

- Styles are defined per tag type. E.g.:

```
body {  
  background-color: #222;  
}  
h1 {  
  color: red;  
}  
p {  
  background-color: black;  
  color: white;  
}
```

31

---

---

---

---

---

---

---

## CSS Selectors

- Selectors determine where to apply the style

- Element selectors

- Select all elements of a given tag to style.
- E.g.

```
p {  
  text-align: center;  
  color: red;  
}
```

32

---

---

---

---

---

---

---

## ID Selectors

- Recall the ID attribute

- `<div id="aboutMe"> ... </div>`

- Can be selected individually

- Styles a single element

- E.g.

```
#aboutMe {  
  background-color: blue;  
  color: gold;  
}
```

33

---

---

---

---

---

---

---

## Class Selectors

- HTML class attribute
  - Specifies a class label for an element
  - `<p class="abstract"> ...</p>`
- Select all elements of a class
  - E.g.,

```
.abstract{
    font-size: 20px;
    font-style: italic;
}
```

34

---

---

---

---

---

---

---

## Class Selectors (2)

- Elements can be of multiple classes.
- `<h1 class="warning headline"> ... </h1>`
- ```
.warning{
    color: red;
    font-style: bold;
}
```
- ```
.headline{
    text-align: center;
    font-size: 60px;
}
```

35

---

---

---

---

---

---

---

## Compound selectors

- Can target more specifically by combining selectors.
  - `tag.className { styles }`
- E.g. Change the warning class when it appears as a link?
  - `a.warning { color: #ff0055 }`
  - `<a class="warning" href="help.html"> Help! </a>`

36

---

---

---

---

---

---

---

## Specificity

- Consider the following scenario:
  - `<p id="aboutMe"> ... </p>`
  - `p{ color: white; }`
  - `#aboutMe{ color: gold; }`
- What color will the text be?
- Specificity rules resolve conflicting styles
  - ID > class > element
  - Inline styles > external css files
  - !important > everything

37

---

---

---

---

---

---

---

## Grouped Selectors

- Multiple selectors with the same styles?
  - `h1 { text-align : center; }`
  - `h2 { text-align : center; }`
  - `h3 { text-align : center; }`
- Group them to save typing (errors/time).
  - `h1, h2, h3 { text-align : center; }`
  - Separate independent selectors with commas.

38

---

---

---

---

---

---

---

## Descendant Selectors

- Style only paragraphs within div elements?
  - `div p { text-shadow: 2px 2px #553322; }`
  - Descendant selectors separated by spaces
- Change the style of the `<em>` tag in abstracts?
  - `.abstract em{ text-decoration: underline; }`

39

---

---

---

---

---

---

---

## Special Selectors: Link Pseudo-classes

- Links can be styled based on their state...
  - Normal unvisited link
    - `a:link{ color: green; }`
  - Previously visited link
    - `a:visited{ color: red; }`
  - Currently active (being clicked) link
    - `a:active{ color: blue; }`

40

---

---

---

---

---

---

---

## Special Selectors: Hover Pseudo-class

- Style content differently when the mouse is over it!
  - Add the selector `:hover`
  - For links, `:link -- :visited -- :hover -- :active`
- Examples:
  - `div:hover{ border: solid 2px lightBlue; }`
  - `a:hover{ text-shadow: 2px 2px green; }`

41

---

---

---

---

---

---

---

## Special Selectors

- Universal selector
  - `*{text-align:center}`
- Attribute selectors
  - `.myClass[myAttr]{font-weight:bold}`
  - `.myClass[myAttr="value"]{`  
    `text-decoration:underline;`  
    `}`

42

---

---

---

---

---

---

---

## Scripts

- `<script > </script>`
  - Run javascript in an html browser
  - Either as file or as raw code
- Attributes:
  - `type="application/javascript"` (default)
  - `src="./lib/javascript.js"`
  - `defer, async`

43

---

---

---

---

---

---

---

## Document Object Model

- Tree-based API structure for HTML documents
  - Elements in HTML become nodes (objects)
  - Attributes and styles become properties
  - Methods for getting and setting content
- Accessible through javascript
  - Empowers dynamic content!
  - Editing HTML and CSS on the client-side
  - React to and trigger events

44

---

---

---

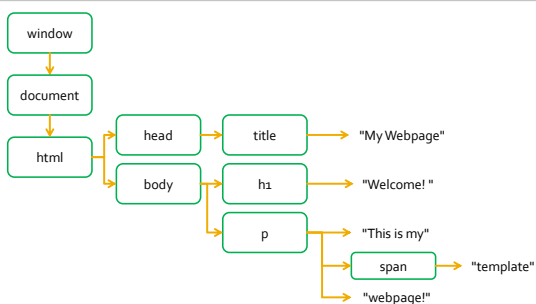
---

---

---

---

## DOM Structure



45

---

---

---

---

---

---

---

## Accessing Elements

- `document.getElementById("someID")`
  - Returns the only matching element
- `document.querySelector("cssStyleSelector")`
  - Returns the first matching element
- `document.querySelectorAll("aSelector")`
  - Returns an array-like list of all matching elements

46

---

---

---

---

---

---

---

## DOM Node Content

- HTML element nodes contain page content
- Useful properties:
  - `ele.innerHTML`
    - Get/set html string inside the element
    - `ele.innerHTML="<p id='greeting'>Hello!</p>"`
  - `ele.style`
    - Get/set style attributes of the element
    - `ele.style.color="blue"`
  - `ele.setAttribute("key","value")`
    - Create/change attributes

47

---

---

---

---

---

---

---

## Creating New Nodes

- Create new elements:
  - `var p = document.createElement("p");`
  - `var text = document.createTextNode("Hello!");`
- Add elements into the DOM:
  - `var target = document.getElementById("target");`
  - `p.appendChild(text);`
  - `target.insertBefore(p, target.firstChild);`

48

---

---

---

---

---

---

---