HTML & CSS

Ben Gavan

June 18, 2019

Contents

1 Terminal Essentials

g

1.1 Basic Commands

- pwd: Print working directory
- ls: list
- ls -la : list list all - shows more detail
- clear: clear current terminal
- cat: Prints out the contents of a file into the termial

- mkdir dir-name: makes a new directory with the name as the string supplied after 'mkdir' within the current dirrectory
- rm file-name: remove directory
- rm -rf dir-name: remove directory recursively (until complete)
- touch txt-file-name: creates new blank txt file with with file name specified
- open file-name: opens the file specified

1.2 Git

- git init: creates an emtpty git repo within current dir
- git status: Gives info on what files haven't been added
- git add
  - dir-name: adds only that dir to the staging index
  - . : add everything below current dir that's changed
  - - -all : adds all files below and above the current dir
- git commit
  - -m "commit-message": commits everything in the staging index along with a commit message.
- git remote add origin : connects the local repo to the cloud
- git push -u origin master : : pushes to the master
- git push: pushes to remote git 'origin/master' (on github)

2 Key Board Shortcuts

- 'cmnd + / ' comment current line
- 'p{TEXT_HERE} + tab' creates p tag with text here as text
- 'lorem' creates some lorem text
- 'control + space' when in element "" to bring up options of possible selections
- 'option + click' to select curser on multiple lines

3 HTML Essentials

3.1 File Naming conventions

- all html files end with the extension ’.html’
- all css style sheets end with the extension ’.css’
- home/default/main page should be called ’index.html’
- style sheet for main page should be called ’main.css’
- only use lower case alphanumeric characters (a-z, 0-9)
- no spaces

3.2 Folder Naming Conventions

- only use lower case alphanumeric characters (a-z, 0-9)
- no spaces
- follow seperation of cencerns
- for folder holding css files, use ’css’

3.3 Emmet.io

- ’cmnd + / ’ comment current line
- ’p{TEXT_HERE} + tab’ creates p tag with text here as text
- ’lorem’ creates some lorem text

3.4 Tag Attributes

To link a CSS Style sheet to html page:

```
<link rel="stylesheet" href="main.css">
```

Add a picture:

```
<img src="puppy.jpg" alt="An image of cute puppy">
```

Link to other page

```
<a href="http://www.google.com" target="_blank">go to google</a>
```

3.5 Absolute vs Relative URLs

Absolute

- Full URL
- Examples:

  ’¡img src=”https://pbs.twimg.com/media/DLUDtXlX0AEiCnG.jpg”¿’

Relative

- Within one domain, sshows where one resource is relative to another
- Examples:

  pic/anatomy-of-an-html-element.png

  ../pic/anatomy-of-an-html-element.png (goes up one level first by using ’../’))

4 Block vs Inline

4.1 Block

- Will take up the width of the parrent (so will stack verticlly)

4.2 Inline

- Will take up only the size/space required - so will stack horizontally; hence inline.

5 HTML Tags

5.1 div

- ”block” level element = takes up the width of the parrent (so will stack verticlly)

5.2 a

Attributes

- ’href’ urlm to open when clicked
- ’target’ how to do it when clicked

  ’_blank’ opens url in new tab

6 CSS Attributes

6.0.1 background-size

- 'cover': covers all of the background - avoids repeates

6.0.2 background-repeat

- 'no-repeat': prevents repeates

6.0.3 display

- 'inline' - Will take up only the size/space required - so will stack horizontally; hence inline.
- 'block' - Will take up the width of the parrent (so will stack verticlly)
- 'inline-block' - displays items inline, but leaves enogh room for each element
- 'none' - will not be displayed

6.0.4 Dimensions

- 'width: ....' - sets the width
- 'height: ....' - sets the height

```
div {
  width: 100px;
  height: 100px;
}
```

6.0.5 text-align

Sets how text should be aligned within the element.

6.0.6 Content - Padding - Border - Margin

- Content - the content itself
- Padding - The Amount of padding around the content before the border
- Border - Around the padding of the content
- Margin - The margin/padding around the border before another element can be displayed.

6.0.7 border

-

- ”border-radius: ...” - sets the radius of border

  ”...%” - sets the radius as a percentage of the width/height

```
border: 1px solid red;
border-width: 1px;
border-style: solid;
border-color: red;
```

6.0.8 margin

- TRBL
- TB RL
- T R B L
- ”margin: 0 auto” - no padding on top or bottom, then auto set left/right so that element is on the centre.

6.0.9 box-sizing

- ”box-sizing: border-box” - sets the max size of the box/div (up-to the outside of the border) to the set dimensions via width/height. So the border outside will stay as it is, but the content will shrink if padding is added. If the border is width increased - it will increase inwards, shrinking the content. The Element will not grow outwardly (grows inwards)

6.1 CSS Reset

Brings the CSS formatting down to a uniform base line.

6.1.1 Motivation

Each browser has its own base CSS (’user agent style sheet’ for chrome). So if we want to have our website uniform across all browsers, we need to apply our own.

6.1.2 Aim

To provide uniformity across browsers.

6.1.3 Requirements

- Light weight
- is applied to all used elements

6.1.4 Solution

Create our own style sheet to override what the browser (agent) applies.

6.1.5 Alternatives

There are some standards, e.g. meyer css reset (oldest), normalize.css (newer), sanitize.css (newest).
However, some of these don't bring the formatting down to zero. Instead they provide their own baseline formatting - overriding what the browser's baseline is.

7 CSS Selectors

Different ways to select HTML elements.

7.1 element

Formats that HTML element.

7.2 class

Adds the 'class' attribute' to a HTML element.
Can add them wherever and as many times as we want.
A period before the selector is used to denote that the selector is for a class. ".example {}"

7.3 id

- Can only be used once.
- Notation to denote id selector is "#example {}"

```
<p id="example"></p>
```

```
#example {
}
```

7.4 Attributes

- Only applies that selector to HTML tags which have that attribute

```
<p example="something"></p>
```

```
[example] {
}
```

- Only applies that selector to HTML tags that have the exact attribute and set to that specific value

```
<p example="something"></p>
```

```
[example=something] {
}
```

7.5 Pseudo Class

Denoted by ":" (single :)

7.5.1 Link

```
  /\* order matters */
  /\* LVHA */
  /\* Link Visited Hover Active */
  a:link {
    color: green;
  }

  a:visited {
    color: blue;
  }

  a:hover {
    color: red;
  }

  a:active {
    color: yellow;
  }
```

```
  div:hover {
    background-color: green;
    cursor: pointer;
  }
```

Use 'active' for links and focus for other elements such as forms.

7.5.2 Focus

Used for other elements such as forms.

7.5.3 n^{th} child

Used to select and format the nth child of a parent in HTML.
First-Child:

```
li:first-child {
}
```

Last Child:

```
li:last-child {
}
```

To select every even child of the parent:

```
li:nth-child(even) {
}
```

To select every odd child of the parent:

```
li:nth-child(odd) {
}
```

Zebra-striping:

```
table, tr, td {
  width: 100%;
  border: 1px solid black;
}

tr:nth-child(even) {
  background-color: rgba(128, 128, 128, 0.49);
}

tr:nth-child(odd) {
  background-color: rgba(128, 128, 128, 0.19)
}
```

To select a specific child:

```
li:nth-child(3) {
}
```

To select the nth child from the bottom:

```
li:nth-last-child(10) {
}
```

To Select more complex pattern, use a linear line (mx+c)

```
li:nth-child(3n+2) {
  color: red;
}

/\*
Try changing the selector to select each of the following:
8, 18, 28, 38, ... 10n+8
9, 12, 15, ..., 39, ... 3n+9
3, 12, 21, 30, 39, ... 9n+3
\*/
```

To Select the a child of a parent where it is the only child:
(To Select the a HTML element of a parent where the element is the only child:)

```
article:only-child {
}
```

7.6 Pseudo Element

Denoted by ”::” (Double :)
Typography:

7.6.1 First-letter

Selects the first letter

```
p::first-letter {
  font-family: cursive;
  font-size: 36px;
  line-height: .5;
}
```

7.6.2 First-line

Dynamically selects the first line (dynamically adapts/reforms depending on width of view port)

```
p::first-line {
  color: red;
  font-weight: 900;
}
```

7.7 Nested Selectors

7.7.1 all-children

Selects all children of a parent = ”parent child {}”. No matter how nested

```
parent child {
}
```

To select all children of a ’div’ which are ’p’ elements

```
div p {
}
```

will select all ’p’ tags in the following case

```
<!--div>p*2+section>p^^p-->
<div>
  <p>first p</p>   <- selected
  <p>second p</p>  <- selected
  <section>
    <p>third p</p> <- selected
  </section>
</div>
<p>fourth p</p>
<p>fifth p</p>
<p>sixth p</p>
```

7.7.2 immediate

To select all the immediate children of a 'div' which are 'p' elements

```
div > p {
}
```

will select all 'p' tags except the ones that are not immediate descendants of a 'div' so only 'first p' and 'second p' will be selected in the following case

```
<!--div>p*2+section>p^^p-->
<div>
  <p>first p</p>      <- selected
  <p>second p</p>     <- selected
  <section>
    <p>third p</p>
  </section>
</div>
<p>fourth p</p>
<p>fifth p</p>
<p>sixth p</p>
```

7.7.3 all-siblings

All 'p' tags that are siblings of a 'div'

```
div ~ p {
}
```

will select .... 'p' tags in the following case

```
<!--div>p*2+section>p^^p-->
<div>
  <p>first p</p>
  <p>second p</p>
  <section>
    <p>third p</p>
  </section>
</div>
<p>fourth p</p> <- selected
<p>fifth p</p>  <- selected
<p>sixth p</p>  <- selected
```

7.7.4 Immediate Siblings

will select the tag that is immediately after the tag - only the immediate sibling.

```
div + p {
}
```

will select only the 'fourth p' 'p' tag in the following case

```
<!--div>p*2+section>p^^p-->
<div>
  <p>first p</p>
  <p>second p</p>
  <section>
    <p>third p</p>
  </section>
</div>
<p>fourth p</p> <- selected
<p>fifth p</p>
<p>sixth p</p>
```

7.8 Compound Selectors

```
ul#some-id li.some-class {
}
```

Will select an 'li' with the class 'some-class' which is a child of a 'ul' with an id 'some-id'.

7.9 CSS Specificity

- Element selector = 1
- Class = 10
- ID = 100
- Inline = 1000

If same specificity, go to the order that rule sets are declared in the CSS. Last = More powerful.

Selector Type	Value	Place
Element	1	0-0-1
Class/attribute	10	0-1-0
ID	100	1-0-0
Inline	1000	1-0-0-0

7.9.1 Example

```
ul#some-id li.some-class {
}
```

The first section

```
ul#some-id
```

has a specificity of 101 (1 from ul (element selector) 100 from '#some-id' (id selector))
The Second section

```
li.some-class
```

has a specificity of 11
So the total specificity is 112.

8 Formatting Text

8.1 Font property

A short-hand property.
Best not to use the short hand property version when starting off. Ensure you understand the individual properties, then use the short hand property.

8.1.1 Font-family

Sans-serif is as the best font to use on the web. (the de-facto standard). It's the most readable on the web.

9 Forms

```
<input type="text" id="fname" name="first-name">
```

- 'id' is what is used on the website, e.g. for targeting that specific element with CSS or JavaScript.
- 'name' is the name that is given to the variable that contains the data from that element which is sent off to the server.