

## **Commands:**

Shift + Opt + F - format the Document in VS Code

Shift + Opt + ↑ / ↓ - duplicate line in up or down direction

fn/option - write text in multiple places

## Section 3: HTML: The Essentials

---

Essentials:

```
1 <b>...</b> - bold text
2 <p>...</p> - paragraph
3 <h1-h6>...</h1-h6> - headings
4 <a href="link.com or .html (for the file)"><!-- Name of a link --></a> -
  hyperlink
5 <!--...--> - comment
```

HTML

HTML Sceleton:

```
1 ! - create a HTML Sceleton
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <title> My page </title>
7 </head>
8 <body>
9   <!-- Content Goes Here -->
10 </body>
11 </html>
```

HTML

Lists:

```
1 <ol>...</ol> - Ordered List (Numbered List)
2 <ul>...</ul> - Unordered List (Bulletpoints)
3 <li>...</li> - List (Child of ul or ol)
```

HTML

Nested List:

```
1 <ul>
2   <li>Bantam
3     <ul>
4       <li>Silkie</li>
5       <li>Polish</li>
6     </ul>
7   </li>
8 </ul>
```

HTML

Image:

```
1  - image
```

HTML

```
2  - image with height and width (but  
better with CSS)  
3  - adding alt to the image (for the  
screenreader)
```

## Section 4: HTML: Next Steps & Semantics

- 1 `<div>...</div>` - container for elements (is a block element)
- 2 `<span>...</span>` - wrap an element, so it can be changed in CSS (mostly for a piece of text)
- 3 `<hr>` - divider
- 4 `<br>` - line break
- 5 `<sup>...</sup>` - create a superscript (f.e. 1 in 1/2)
- 6 `<sub>...</sub>` - create a subscript (f.e. 2 in 1/2)

HTML

HTML Entities:

- 1 `&lt;` - <
- 2 `&gt;` - >
- 3 `&copy;` - ©
- 4 ...

HTML

Semantik:

- 1 `<main>...</main>` - main content of the page
- 2 `<nav>...</nav>` - navigation links
- 3 `<section>...</section>` - section of the page / of the content
- 4 `<article>...</article>` - article (can be used on its own & independent)
- 5 `<aside>...</aside>` - side content (f.e. ads)
- 6 `<header>...</header>` - header of the page
- 7 `<footer>...</footer>` - footer of the page
- 8 `<time>...</time>` - time (f.e. date)
- 9 `<figure>...</figure>` - figure (f.e. image with caption)

HTML

Emmet:

- 1 `>` - child: `main>section>h1 =>`
- 2 `<main>`
- 3 `<section>`
- 4 `<h1>...</h1>`
- 5 `</section>`
- 6 `</main>`
- 7 `+` - sibling: `h1+h2+h2 =>`
- 8 `<h1>...</h1>`
- 9 `<h2>...</h2>`
- 10 `<h2>...</h2>`
- 11 `()` - grouping
- 12 `*` - multiplication: `h1*3 =>`
- 13 `<h1>...</h1>`
- 14 `<h1>...</h1>`

HTML

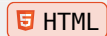
```
15     <h1>...</h1>
16 $ - item numbering: nav>ul>li.item$*3 =>
17     <nav>
18         <ul>
19             <li class="item1">...</li>
20             <li class="item2">...</li>
21             <li class="item3">...</li>
22         </ul>
23     </nav>
24 {} - text: a{Click me!} =>
25     <a href="...">Click me!</a>
```

## Section 5: HTML: Forms & Tables

---

### HTML Tables:

```
1 <table>...</table> - table
2 <td>...</td> - table data (cell)
3 <th>...</th> - table header (bold)
4 <tr>...</tr> - table row
5 <thead>...</thead> - table header (row)
6 <tbody>...</tbody> - table body (row)
7 <tfoot>...</tfoot> - table footer (row)
8 <th rowspan="2">...</th> - table header (row span)
9 <th colspan="2">...</th> - table header (column span)
```



### HTML: Forms

```
1 <form action="...">...</form> - form with an action / link ("/" + to
  where the form goes; f.e. /home)
2 <input type="..."> - input
3 <input type="..." placeholder="..."> - placeholder
4
5 Text input form with placeholder "name":
6 <form action="...">
7   <input type="text" placeholder="name">
8 </form>
9
10 <label for="...">...</label> - label for input
11
12 Using label (for = unique id of the input):
13 <label for="name">Name</label>
14 <input type="text" id="name" placeholder="name">
15
16 <input type="text" name="name"> - input with name (for the server; f.e. after
  submitting the form, name=John or https://www.google.com/search?q=John)
17
18 <button type="submit">...</button> - button (goes to the action of the form)
19 <button type="reset">...</button> - reset button (resets the form)
20 <button type="button">...</button> - button (does not go to the action of the
  form)
21
22 <input type="checkbox" name="..."> - checkbox with name
23 <input type="radio" name="..."> - radio button with name
24 <input type="..." name="..." value="..."> - value of the input (f.e. for
  checkbox or radio button)
25 <select>...</select> - select box
26 <option>...</option> - option in the select box
27
```



```
28 Drop down:
29 <select name="..." id="...">
30     <option value="1">...</option>
31     <option value="2">...</option>
32 </select>
33
34 <input type="range" id="..." min="..." max="..." step="..." value="..."
    name="..."> - range input (slider); value is the default(start) value
35 <input type="color" id="..." name="..."> - color input
36 ...
37
38 <textarea>...</textarea> - text area (multiline input)
39 <textarea id="..." rows="..." cols="...">...</textarea> - text area with rows
    and columns
```

Built in validations:

```
1 <input type="text" minlength="..."> - minimum length of the input
2 <input type="text" maxlength="..."> - maximum length of the input
3 <input type="text" required> - required input (must be filled out | must be not
    empty)
```



## Section 6: CSS: The Very Basics

---

Styles:

```
1  Style element: HTML
2  <h1 style="color: red;">...</h1> - inline style (not recommended)
3
4  External style (using CSS):
5  <head>
6      <link rel="stylesheet" href="style.css"> - link to the CSS file (in the head
        of the HTML file; works automatically)
7  </head>
8  <body>
9      <h1>...</h1>
10 </body>
```

External style: link to CSS (href="style.css")



```
1  style.css: CSS
2  h1 {
3      color: red;
4  }
```

Color changing in CSS:

```
1  ... {color: red;} CSS
2  /* CSS color*/
3  ... {background-color: red;}
4  /* CSS background color*/
5  ... {background: red;}
6  /* CSS background (color, image, position, repeat, size)*/
7  ... {color: rgb(255, 0, 0);}
8  /* CSS color (RGB)*/
9  ... {color: #FF0000;}
10 /* CSS color (hexadecimal)*/
```

Text properties:

```
1  ... {text-align: center;} CSS
2  /* CSS text alignment (can be left, right, center, justify)*/
3  ... {width: 100px;}
4  /* CSS width*/
5  ... {height: 100px;}
6  /* CSS height*/
7  ... {font-weight: bold;}
8  /* CSS font weight (can be normal, bold, bolder, lighter)*/
```



```
9 ... {font-size: 20px;}
10 /* CSS font size*/
11 ... {text-decoration: underline;}
12 /* CSS text decoration (can be none, underline, overline, linethrough)*/
13 ... {letter-spacing: 2px;}
14 /* CSS letter spacing*/
15 ... {font-family: Arial, sans-serif;}
16 /* CSS font family (can be serif, sans serif, monospace, cursive, fantasy) */
```

## Section 7: The World of CSS Selectors

```
1  * {color: red;}
2  /*CSS universal selector (selects all elements)*/
3  ... , ... {color: red;}
4  /*CSS group selector (selects all elements with the same style)*/
5  #id {color: red;}
6  /*CSS id selector (selects the element with the id)*/
7  .class {color: red;}
8  /*CSS class selector (selects all elements with the class; fe. <span
   class="class"> ... </span>)/
9  ... ... {color: red;}
10 /*CSS descendant selector (selects all elements inside the element, fe. all
   paragraphs p inside the div)*/
11 ... + ... {color: red;}
12 /*CSS adjacent sibling selector (selects the element that is next to the
   element; fe. h1 and p are siblings/are next to each other => h1 + p)*/
13 ... > ... {color: red;}
14 /*CSS child selector (selects the element that is a child of the element; fe. h1
   is child of the div => div > h1 => h1 will be changed)*/
15 ... [ ... = "..."] {color: red;}
   /* CSS attribute selector (selects the element with the attribute; fe. input
   with type text => input[type="text"] or section[class="class"] => section with
   class "class") */
17 ... [ ... *= "..."] {color: red;}
   /* CSS attribute selector (selects the element with the attribute that contains
   the value; fe. input with type text => input[type*="text"] or
   section[class*="class"] => section with class "class") */
```

Pseudo classes:

```
1  ... :hover {color: red;}
2  /* CSS hover selector (selects the element when the mouse is over it) */
3  ... :checked {color: red;}
4  /* CSS checked selector (selects the element when it is checked; f.e. checkbox or
   radio button) */
5  ... :nth-of-type(2n) {color: red;}
6  /* CSS nth type selector (selects the element that is the nth child of the
   element; f.e. every second child => 2n) */
```

Pseudo elements:

```
1  ... ::after {color: red;}
2  /* CSS after selector (selects the element after the element) */
3  ... ::before {color: red;}
4  /* CSS before selector (selects the element before the element) */
5  ... ::first-letter {color: red;}
```

```

6  /* CSS first letter selector (selects the first letter of the element) */
7  ... ::first-line {color: red;}
8  /* CSS first line selector (selects the first line of the element) */
9  ... ::selection {color: red;}
10 /* CSS selection selector (selects the selected text of the element) */

```

The CSS Cascade:

```

1  h1 {color:red}
2  h1 {color:blue}
3
4  => h1 is blue

```

CSS

CSS Specificity:

```

1  ID > Class > Element
2
3  section p {color:teal;} /* => 0 0 2 */
4  #submit {color: olive;} /* => 1 0 0 */
5  /* 1 0 0 is far more specific than 0 0 2 */
6  nav a.active {color: orange;} /* => 0 1 2 */
7
8  Also: Inline styles > ID
9
10 ... {color: red !important;} /* !important is the most important
11 => !important > Inline styles > ID > Class > Element */

```

CSS

CSS Inheritance:

```

1  /* certain things don't inherit at default: buttons, input, etc. ... */
2  button {color: inherit;} /* inherit color from the parent element */

```

CSS

## Section 8: The CSS Box Model

---

```
1 ... {width: 100px;} /* width of the element */
2 ... {height: 100px;} /* height of the element */
```

 CSS

Border:

```
1 ... {border: 1px solid red;} /* border (width, style, color) */
2 ... {border_width: 1px;} /* border width */
3 ... {border_style: solid;} /* border style (solid, dotted, dashed, double,
    groove, ridge, inset, outset) */
4 ... {border_color: red;} /* border color */
5 ... {box-sizing: border-box;} /* box sizing (content-box, border-box) */
6 ... {border-radius: 10px;} /* border radius (round corners) */
```

 CSS

Padding:

```
1 ... {padding: 10px;} /* padding (top, right, bottom, left) */
2 ... {padding: 10px 20px;} /* padding (top/bottom, right/left) */
```

 CSS

Margins:

```
1 ... {margin: 10px;} /* margin (top, right, bottom, left) */
2 ... {margin: 10px 20px;} /* margin (top/bottom, right/left) */
```

 CSS

Display properties:

```
1 ... {display: block;} /* display block (takes the whole width) */
2 ... {display: inline;} /* display inline (takes the width of the content) */
3 ... {display: inline-block;} /* display inline block (takes the width of the
    content, but can have width and height) */
4 ... {display: flex;} /* display flex (flexbox) */
5 ... {display: grid;} /* display grid (grid) */
```

 CSS

CSS Units:

```
1 /* Font size:
2 1em = the font-size of the parent
3 2em = 2 times the font-size of the parent
4
5 1rem = the font-size of the root element (html)
6 2rem = 2 times the font-size of the root element (html)
7
8 Other properties:
9 1em = computed from font-size of the element itself */
```

 CSS

## Section 9: Other Assorted Useful CSS Properties

Alpha:

```
1 ... {color: rgb(0, 0, 0, alpha);} /* color (RGB with alpha) */
2 ... {opacity: 0.5;} /* opacity (0 = transparent, 1 = opaque) */
```

CSS

Positions:

```
1 ... {position: static;} /* static (default) */
2 ... {position: relative;} /* relative (relative to the element itself) */
3 ... {position: absolute;} /* absolute (relative to the parent element) */
4 ... {position: fixed;} /* fixed (relative to the viewport(containbox)) */
5 ... {position: sticky;} /* sticky (relative to the viewport, but only when scrolling) */
```

CSS

Transitions:

```
1 ... {transition: 1s} /* transition (time) */
2 ... {transition: property name duration timingFunction delay} /* transition (property name, duration, timing function, delay) */
3 ... {transition-timing-function: ease;} /* transition timing function (ease, linear, ease-in, ease-out, ease-in-out) */
```

CSS

Transforming:

```
1 ... {transform: rotate(45deg);} /* transform (rotate) */
2 ... {transform-origin: top left;} /* transform origin (where the element is transformed from) */
3 ... {transform: scale(1.5);} /* transform (scale) */
4 ... {transform: translate(10px, 20px);} /* transform (translate X and Y) */
5 ... {transform: skew(10deg);} /* transform (skew) */
```

CSS

Background:

```
1 ... {background-image: url(image.png);} /* background image */
2 ... {background-size: cover;} /* background size (cover, contain) */
3 ... {background-position: center;} /* background position (top, bottom, left, right, center) */
4 ... {background-repeat: no-repeat;} /* background repeat (no-repeat, repeat, repeat-x, repeat-y) */
5 or
6 ... {background: url(image.png) no-repeat center / cover;} /* background (image, repeat, position, size) */
```

CSS

Google Fonts:

```
1 font-family: 'Roboto', sans-serif; /* font family (Google Fonts) */
```

CSS

Import Google font using link in HTML



```
1 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400&
  display=swap" rel="stylesheet"> /* link to Google Fonts */
```

