Learn how to
How to create web pages with HTML and CSS
Create interactive web content with JavaScript
Basics of coding JavaScript
Bring your web pages to life with Code
Modern Responsive WebSites
Interactive and Dynamic Web Pages

Are you curious about creating websites, this course is the perfect place to start.
Learn all the core fundamentals of modern web design, covering all the commonly used syntax for HTML and CSS. This course is designed to get you started quickly and easily with creating web pages.
Course covers all the essentials so that you can begin quickly on your journey to create amazing looking websites. Add JavaScript to make your web pages come to life.
Covering the core concepts of JavaScript so that you can try the code and get familiar with what it does. Using JavaScript you can access the HTML DOM in the browser which allows you to interact and manipulate web page content. JavaScript creates the interactive and dynamic content seen today in all major modern web pages.
Explore how you can start coding quickly and focus on designing your webpages, loaded with source code and examples.
Each section comes with a PDF guide that includes useful section resources, and source code from the lessons so that you can try the code for yourself.

HTML

Learn HTML - how to create an HTML file and structure your HTML code in a modern format ready to be styled. Lessons of this section cover how to get started with coding and creating web pages.

- Setup your Editor and create HTML files
- Debug your code
- HTML tags for page structure
- What makes up an HTML element
- What are Self closing Tags
- How to use Element Attributes
- Linking pages together with Hyperlinks
- Adding Images to your web page
- Lists and Tables for readable content
- Semantic Page elements
- How to create a simple webpage
- Go live with a Github page and your HTML site

Getting started with Code

Editor and setup to write HTML - create an html file

Tools needed - browser to run the html code.
Chrome Browser comes with Developer tools that are a powerful way to interact with your code, including debugging, inspecting and viewing changes.
https://developer.chrome.com/docs/devtools/

Online Code Editor - great to practice code without a need for download of applications
https://codepen.io/pen/

Help and Code Samples - MDN developer of Firefox has an excellent resource with code samples and browser compatibility/
https://developer.mozilla.org/en-US/docs/Web/HTML

Code Editor - write code with a code editor as it can help structure and suggest tags for code.
https://code.visualstudio.com/

Create Your First html Page

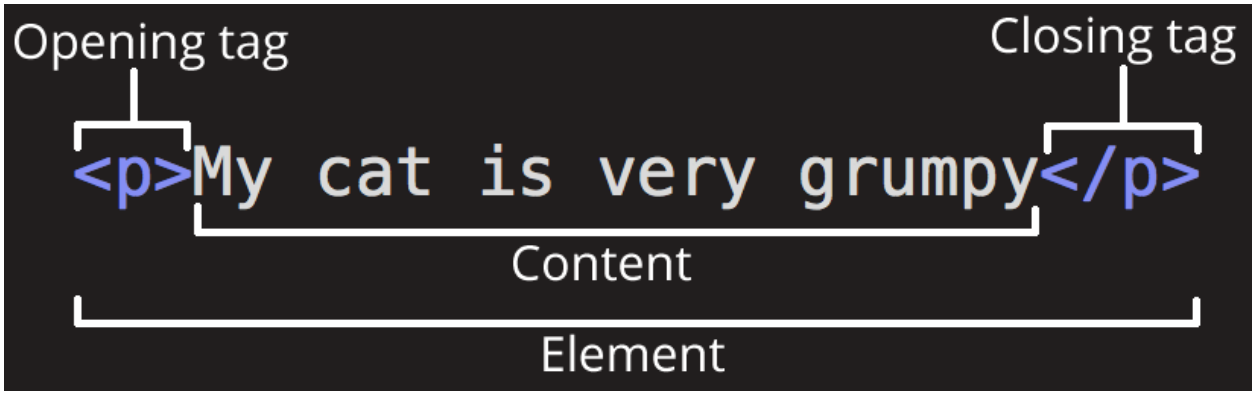
4 must add tags for HTML pages html, head, title, body

```
<!DOCTYPE html>
```

```
<html>
 <head>
     <title>My Page</title>
 </head>
 <body>
     Hello
 </body>
</html>
```

HTML Element

Anatomy of an HTML element - opening tag, content, element, closing tag




```
<!DOCTYPE html>
<html>
 <head>
     <title>My Page</title>
 </head>
 <body>
     <div>
       <p>Hello World!</p>
       <div>Welcome to my site.</div>
       <div>Welcome to my site.</div>
       <p>My name is Laurence.</p>
     </div>
```

```
 </body>
</html>
```

Headings HTML

Comments in Code <!-- -->
Create a basic template
Introduction to common HTML tags <h1><p>

```
<!DOCTYPE html>
<html>
 <head>
     <title>My Page</title>
 </head>
 <body>
     <div>
       <h1>Hello World!</h1>
       <div>Welcome to my site.</div>
       <div>Welcome to my site.</div>
       <p>My name is <span>Laurence</span>
<span></span>Svekis</span>.</p>
     </div>
     <h2>Heading 2</h2>
     <h4>Heading 4</h4>
     <h6>Heading 6</h6>
     <!-- Add another div here -->
     <!--
     <div>
         Content
     </div>
     -->
 </body>
```

```
</html>
```

Self Closing Tags

```
<!DOCTYPE html>
<html>
 <head>
     <title>My Page</title>
 </head>
 <body>
     <p>
         Hello<br>World
     </p>
     <div>
       <h1>Hello World!</h1>
       <div>Welcome to my site.</div>
       <div>Welcome to my site.</div>
       <hr>
       <p>My name is <span>Laurence</span>
<span></span>Svekis</span>.</p>
     </div>
     <hr>
     <h2>Heading 2</h2>
     <h4>Heading 4</h4>
     <h6>Heading 6</h6>
     <!-- Add another div here -->
     <!--
     <div>
         Content
     </div>
     -->
 </body>
```

```
</html>
```

Element Attributes

```
<!DOCTYPE html>
<html>
 <head>
     <meta charset="utf-8">
     <title>My Page</title>
     <style>
         .blue{
             color:blue;
         }
         .red{
             color:red;
         }
     </style>
 </head>
 <body>
     <p id="myID">
         Hello<br>World
     </p>
     <div>
       <h1>Hello World!</h1>
       <div style="color:blue;">Welcome to my site.</div>
       <div style="color:red">Welcome to my site.</div>
       <hr>
       <p title="My Name">My name is <span>Laurence</span>
<span></span>Svekis</span>.</p>
     </div>
     <hr>
     <h2 class="blue">Heading 2</h2>
```

```
    <h4 class="red">Heading 4</h4>
    <h6 class="green">Heading 6</h6>
    <!-- Add another div here -->
    <!--
    <div>
        Content
    </div>
    -->
</body>
</html>
```

HTML Links anchor Tag

Linking to new pages
Relative links vs Absolute
Relative is based on the current file location and absolute paths will be the full path that includes the domain and file.
Target of hyperlink is by default self which is to open the current window to the new file, target _blank opens a new window and focuses on that, but keeps the original source page for the link open as well.
Anchor tags can also be used to link to page elements by using their id added a # within the href path of the link.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>My Page</title>
</head>
<body>
  <div>
    <a href="#bottom">Go to bottom</a>
  </div>
    <p id="myID">
```

```
        Hello<br><a href="http://www.google.com"
target="_blank">World</a>
    </p>
    <div>
      <h1>Hello World!</h1>
      <div style="color:blue;">Welcome to my site.</div>
      <div style="color:red">Welcome to my site.</div>
      <hr>
      <p title="My Name">My name is <span>Laurence</span>
<span></span>Svekis</span>.</p>
    </div>
    <hr>
    <h2 class="blue">Heading 2</h2>
    <h4 class="red">Heading 4</h4>
    <h6 class="green">Heading 6</h6>
    <p>Got to <a href="new.html">New Page</a></p>

    <p>end Email <a
href="mailto:myemail@myemail.com">myemail@myemail.com</a></p>
    <div id="bottom">
      hello world blah blah blah.  <a href="#myID">Go to the
top element with ID of myID.</a>

    </div>
    <!-- Add another div here -->
    <!--
    <div>
        Content
    </div>
    -->
 </body>
</html>
```

Images in HTML code IMG tag

```
  <div>
    <a href="#bottom">Go to bottom</a>
    <img src="download.png" alt="HTML Logo">
    <img width="200px"
src="https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_
basics/grumpy-cat-small.png">
  </div>
```

HTML Lists UL OL DL Examples

```
    <ol type="i" start="20">
      <li>One</li>
      <li>New List
        <ol>
          <li>One</li>
          <li>Two</li>
        </ol>
      </li>
      <li>Three</li>
    </ol>
    <ul style="list-style-type:disc">
      <li>One</li>
      <li>Two</li>
      <li>Three</li>
    </ul>
    <dl>
      <dt>Main 1</dt>
      <dd>-One</dd>
      <dd>-Two</dd>
      <dt>Main 2</dt>
      <dd>-One</dd>
```

```
      <dd>-Two</dd>
    </dl>
  </div>
```

HTML Table

```
<!DOCTYPE html>
<html>
 <head>
    <meta charset="utf-8">
    <title>My Page</title>
    <style>
      table, th, td{
        border:1px solid black;
        border-collapse:collapse;
        padding:5px;
      }
    </style>
 </head>
 <body>
   <table>
    <colgroup>
      <col span="2">
      <col span="1" style="background-color: red;">
    </colgroup>
    <tr>
      <th>Name</th>
      <th>Chapter 1</th>
      <th>Chapter 2</th>
      <th>Chapter 3</th>
      <th>Chapter 4</th>
```

```
    </tr>
    <tr>
      <td>Laurence</td>
      <td>x</td>
      <td> </td>
      <td>x</td>
      <td> </td>
    </tr>
    <tr>
      <td>John</td>
      <td>x</td>
      <td>x</td>
      <td> </td>
      <td> </td>
    </tr>
    <tr>
      <td>Anne</td>
      <td></td>
      <td>x</td>
      <td> </td>
      <td> </td>
    </tr>
  </table>
```

Semantic Elements

```
<!DOCTYPE html>
<html>
 <head><title>My Website</title></head>
 <body>
   <header class="header">Header</header>
```

```
  <nav class="navbar">Nav</nav>
  <section class="content">
    <section>Main</section>
    <aside>Side Bar Content</aside>
    <article>More details</article>
  </section>
  <footer class="footer">Footer</footer>
 </body>
</html>
```

WebPage HTML

```
<!DOCTYPE html>
<html>
 <head><title>Laurence Svekis Resume</title>
 <style>
  .footer{
    text-align:center;
    padding:10px;
    background-color:#ddd;
  }
  table{
    width:100%;
    border:1px solid black;
  }
  dt{
    font-weight:bold;
    font-size:1.2em;
  }
  section{
    padding:10px;
    border:1px solid black;
```

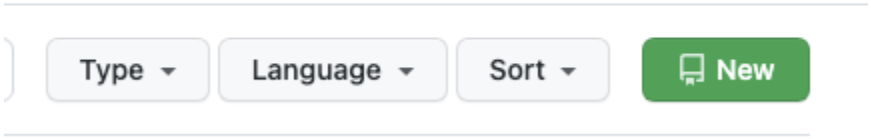
```
    margin:10px;
  }
 </style>
 </head>
 <body>
  <header class="header"><h1>Laurence Svekis
Resume</h1></header>
  <nav class="navbar">
    <ul>
      <li><a href="#Exp">Experience</a></li>
      <li><a href="#Edu">Education</a></li>
      <li><a href="#Int">Interests</a></li>
      <li><a href="#Ski">Skills</a></li>
      <li><a href="#Con">Contact</a></li>
    </ul>
  </nav>
  <article><p>
    Welcome to my website, hope you enjoy it.
  </p>
    <hr>
  <p>My name is Laurence Svekis</p>
  <img
src="https://media-exp1.licdn.com/dms/image/C4E03AQEAtQ9VQ0Z7rA/
profile-displayphoto-shrink_200_200/0/1632867219913?e=1639612800
&v=beta&t=_B4leYocoUhtPWxzpUx5rcrhOTU4suzGe5CuVXR-IE8">
  </article>
  <section id="Exp">
    <h2>Expereince</h2>
    <table>
      <colgroup>
        <col span="1" style="background-color: #ddd;">
      </colgroup>
```

```
      <tr><th>Company</th>
        <th>Details</th>
        <th>Years</th>
      </tr>
      <tr>
        <td>Discoveryvip</td>
        <td>Created Courses</td>
        <td>2002 - 2021</td>
      </tr>
      <tr>
        <td>Basescripts</td>
        <td>Online Elearning</td>
        <td>2010 - 2015</td>
      </tr>
      <tr>
        <td>Discoveryvip</td>
        <td>Created Courses</td>
        <td>2002 - 2021</td>
      </tr>
    </table>
  </section>
  <section id="Edu">
    <h2>Education</h2>
    <dl>
      <dt>My University</dt>
      <dd>MBA</dd>
      <dt>Degree</dt>
      <dd>Computer Science</dd>
    </dl>
  </section>
  <section id="Int">
    <h2>Interests</h2>
```

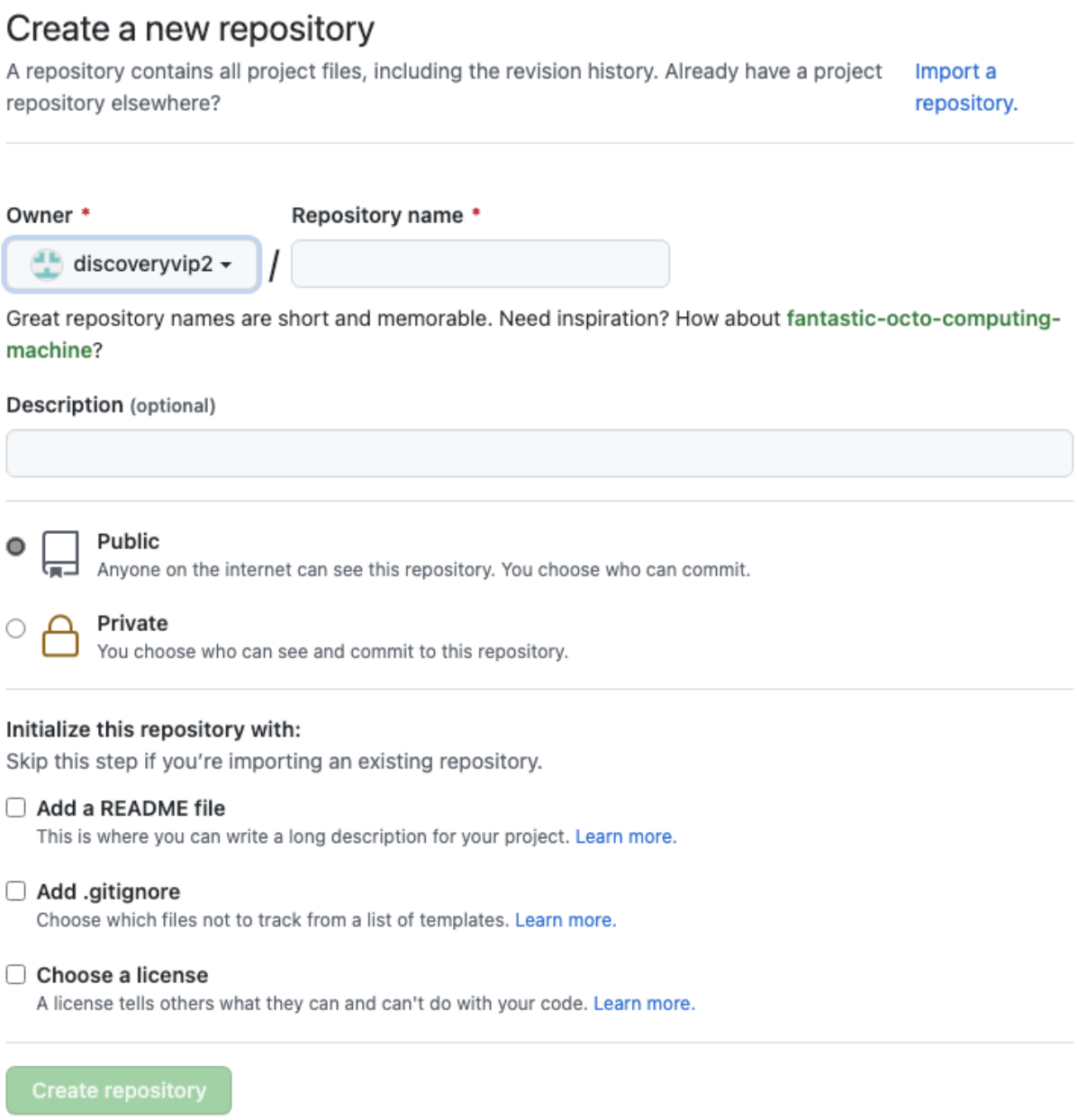
```
    <ul>
      <li>Computers</li>
      <li>Skiing</li>
      <li>Swimming</li>
    </ul>
  </section>
  <section id="Ski">
    <h2>Skills</h2>
    <ol>
      <li>CSS</li>
      <li>HTML</li>
      <li>JavaScript
        <ul>
          <li>DOM</li>
          <li>Node</li>
          <li>Google Apps Script</li>
        </ul>
      </li>
    </ol>
  </section>
  <section id="Con">
    <h2>Contact Me</h2>
    <p>
      Contact me at my email <a
href="mailto:laurenceSvekis@myemail.com">My Email Address</a>
    </p>
  </section>
  <footer class="footer">Copyright (c) 2022 Laurence Svekis
Content please contact me to find out more.</footer>
 </body>
</html>
```

HTML online Create a GitHub Page

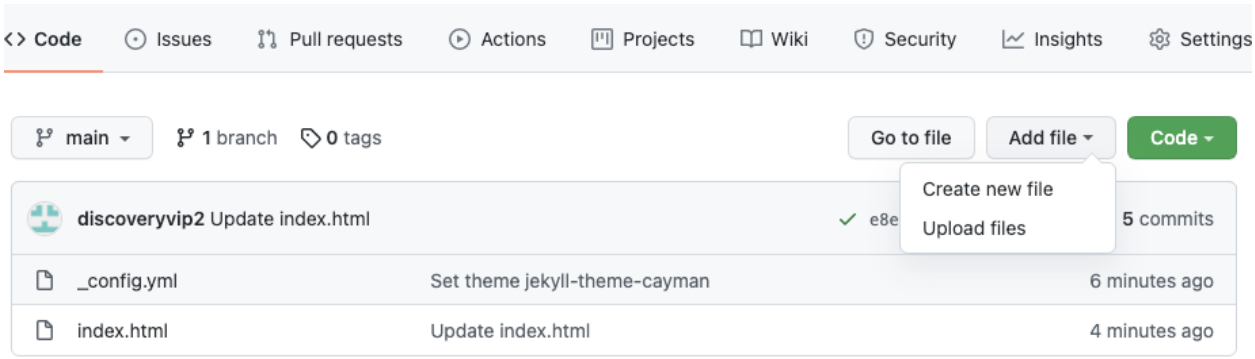
1 - Sign into your Github account https://github.com/
2 - View your repositories
3 - Select the NEW button in the top right

1. 

4 - Fill Out the REPO details

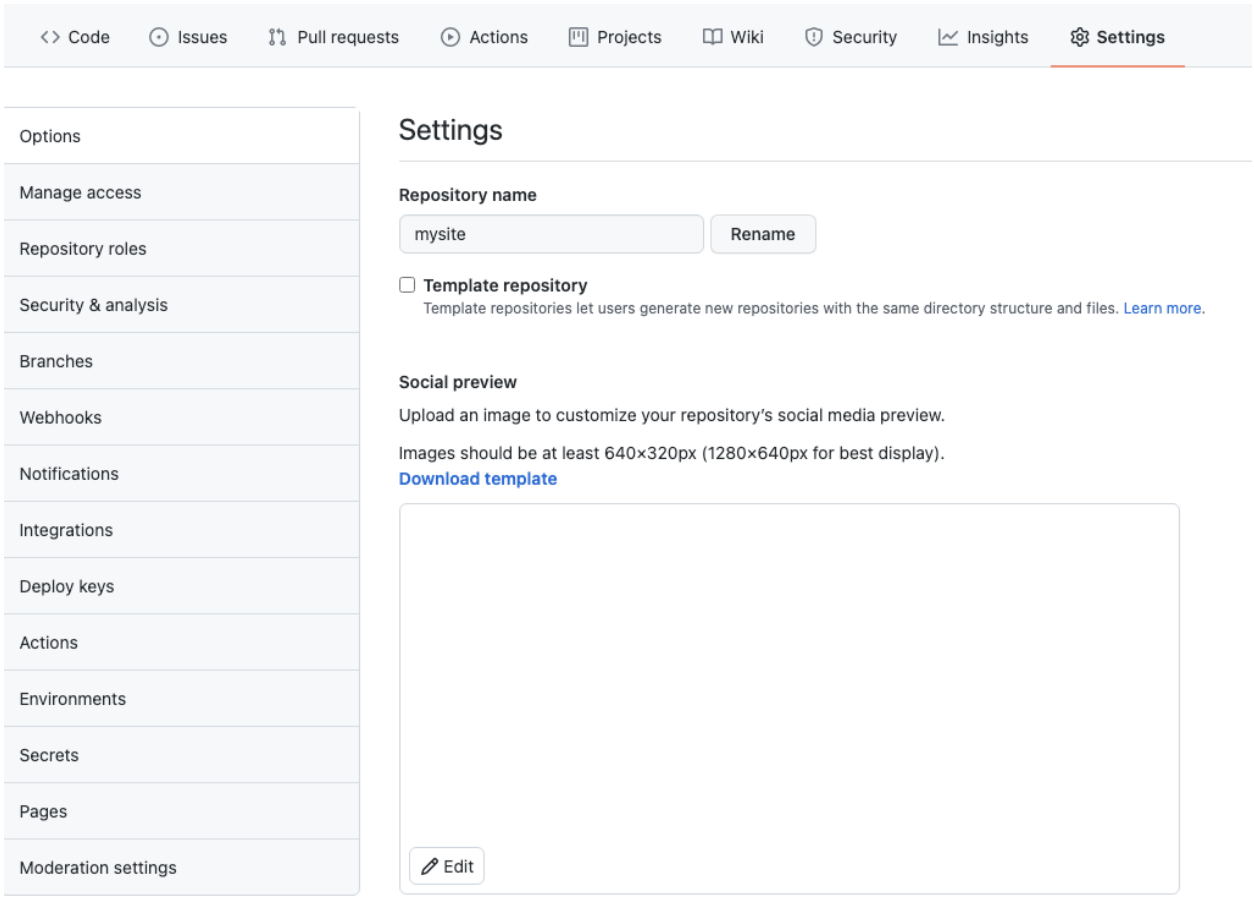


5 - Create the repository
6 - Open the repository and select the create new file



7 - Copy and Paste your HTMl code into the editor and new the file index.html

8 - Under the settings select Pages



Features

9 - enable the github page and view the page at the web URL

CSS

What CSS is and how you can style your web pages with Cascading Style Sheets. HTML provides structure for your webpage, CSS allows you to style your page. Design the page with your style, setup page layouts, add colors, fonts, and more. Present your web pages as you want them to look, independent of the HTML you can make your web content look and style as you imagine it should.

- Explore how to add CSS to your HTML page
- How to add colors to page element backgrounds and text
- What the box model is and how you can apply borders, margins, and padding to any page element.
- Style the text, update the font and customize your text output.
- Update you links, adding Pseudo classes to your page elements
- How to apply display properties, position and floats to set up your page layout.
- Really useful CSS properties explored
- CSS combinators for selection of elements.
- How to build a responsive webpage with CSS float, CSS grid and CSS flexbox.

Getting started with Code

Editor and setup to write CSS - create an html file add CSS

Tools needed - browser to run the html code.
Chrome Browser comes with Developer tools that are a powerful way to interact with your code, including debugging, inspecting and viewing changes.
https://developer.chrome.com/docs/devtools/

Online Code Editor - great to practice code without a need for download of applications
https://codepen.io/pen/

Help and Code Samples - MDN developer of Firefox has an excellent resource with code samples and browser compatibility/
https://developer.mozilla.org/en-US/docs/Web/HTML

Code Editor - write code with a code editor as it can help structure and suggest tags for code.
https://code.visualstudio.com/

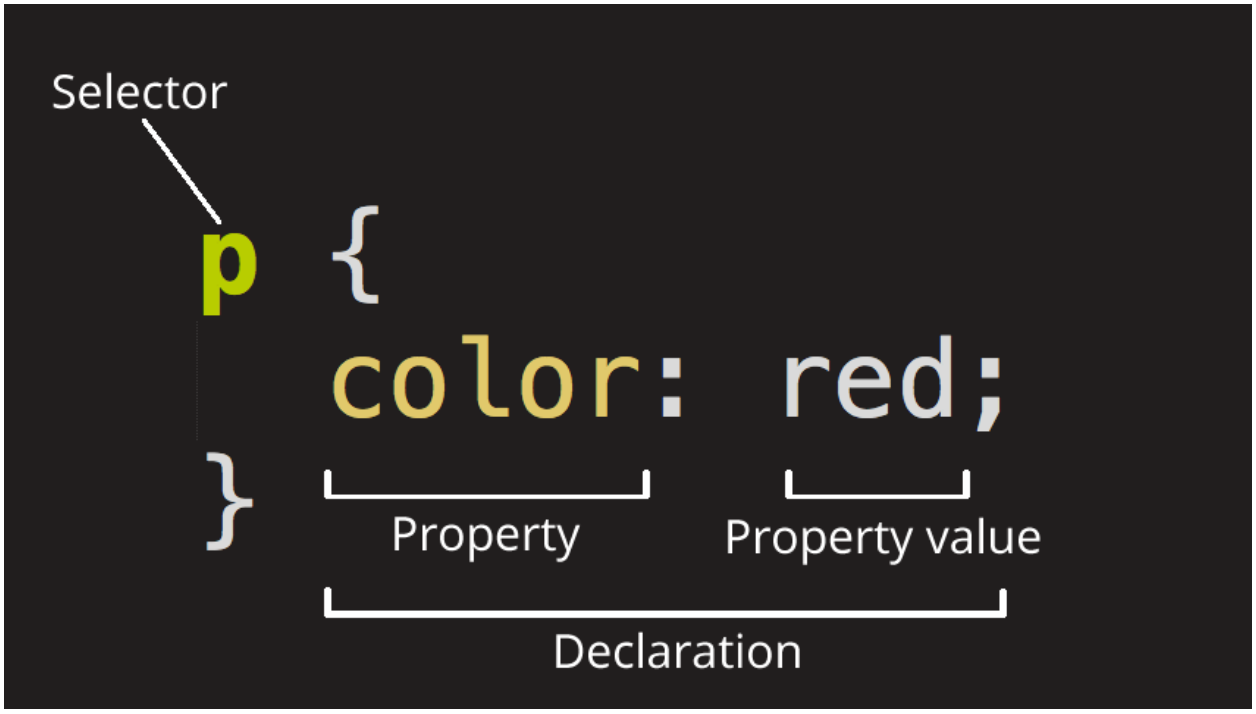
Adding Styling to your HTML

- Google Chrome Dev Tools

- Adding Styling to HTML tags style <style> link to CSS file
- CSS rule CSS Syntax (Selector) (Declaration) {Property:value}
- Element Selection by tag name, id, class

Styling Overview

- Comments in Code /* */
- find help with CSS
- Google Fonts
- https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics
- https://fonts.google.com/




```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <link rel="stylesheet" href="style1.css">
 <title>My Page</title>
</head>
<body>
```

```
 <div>
   <h1 class="blue">Hello World!</h1>
   <div style="font-size:2em">Welcome to my site.</div>
   <hr>
   <p title="My Name">My name is <span
class="blue">Laurence</span> <span></span>Svekis</span>.</p>
 </div>
 <hr>
 <h2 class="blue" >Heading 2</h2>
 <h4 class="red">Heading 4</h4>
 <h6 >Heading 6</h6>
 <span class="h6">Heading 6 #2</span>
 <div id="bottom">
   hello world blah blah blah. <a href="#myID">Go to the top
element with ID of myID.</a>
   <div class="green" >TEST</div>
 </div>
</body>
</html>
```

```
/*
Comments
*/
@import
url('https://fonts.googleapis.com/css2?family=ZCOOL+KuaiLe&displ
ay=swap');
body {
   font-family: 'ZCOOL KuaiLe', cursive;
}
h1, h6, h4{
   background-color: purple;
}
```

Colors Background and Font Color

- Colors - Color types named colors HEX RGB RGBA
- Background - color - image - repeat - attachment - position - background shorthand

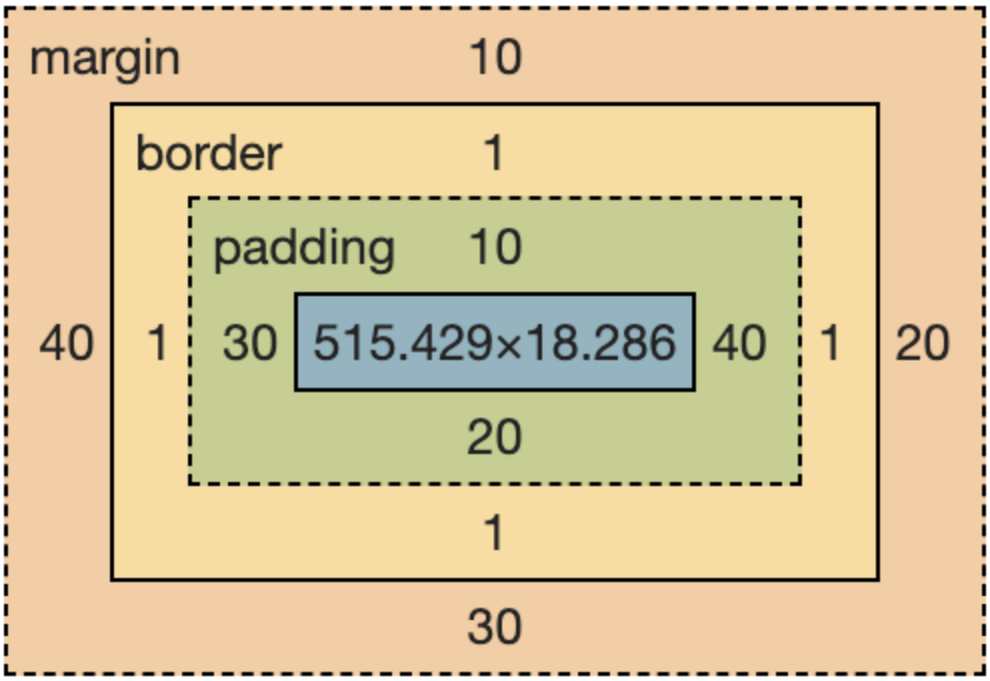
https://developer.mozilla.org/en-US/docs/Web/CSS/color_value
https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units

```
.blue {
   background-color: rgba(255,0,255,0.5);
   color:white;
}
.red{
   background-color:#ff00ff80;
   color:rgb(255,255,255);
}
.h6{
   background-color:hsl(120,50%,50%);
   color:hsla(0,100%,50%,0.5);
}
body{
   background-color:#000000;
   color:violet;
}
```

Box Model Border Padding Margin

- Box Model - Borders - Margins - Padding - Height and Width

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model

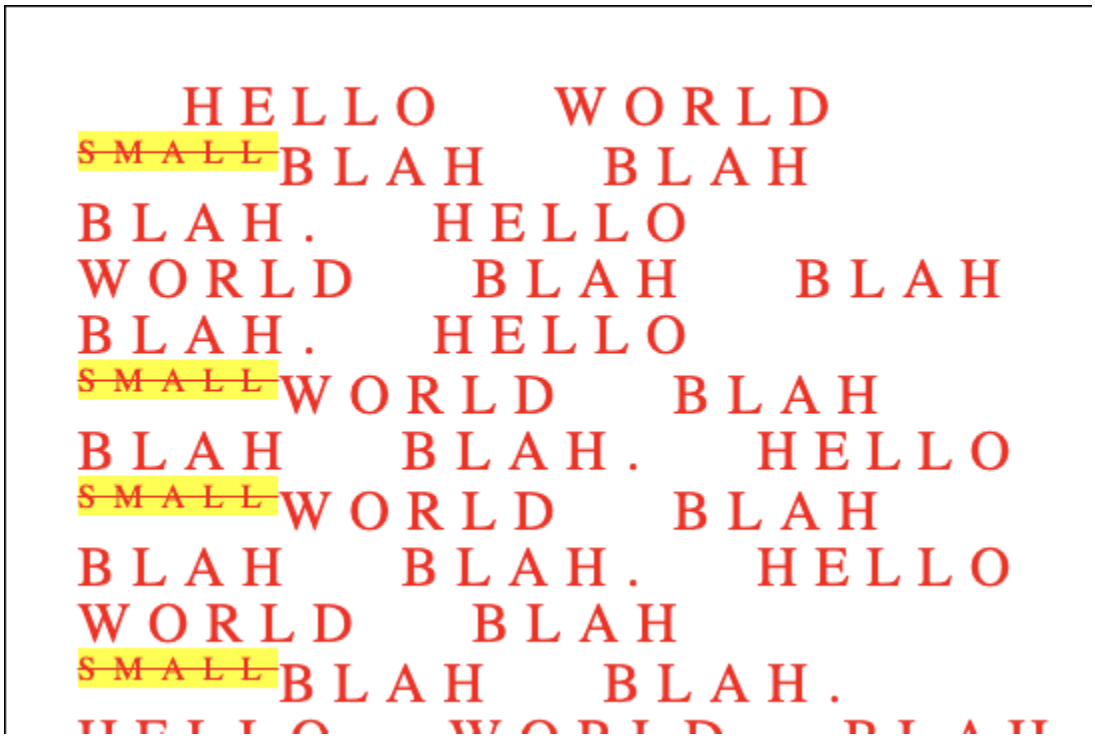



```
div {
   border-style: solid;
   border-width: 1px;
   border-color:red;
}
h2{
   border:solid 1px blue;
   background-color:aqua;
   width:100px;
   height:20px;
   overflow:hidden;
   margin-top:10px;
   margin-bottom:30px;
   margin-right:20px;
   margin-left:40px;
}
.red{
   background-color:red;
```

```
   border:1px dotted black;
   padding-top: 10px;
   padding-bottom: 20px;
   padding-left: 30px;
   padding-right: 40px;
   margin:10px 20px 30px 40px;
}
#bottom{
   border:1px dotted black;
   padding:10px 20px 30px 40px;
   background-color:yellow;
}
```

Text options

- Text - text-align text-decoration text-transform



```
#bottom {
   color:red;
   border:1px solid black;
   padding:20px;
   margin-top:50px;
   line-height: 1;
   text-align: left;
   text-transform: uppercase;
   text-indent: 30px;
   letter-spacing:5px;
   word-spacing: 20px;
   white-space: normal;
}
.small{
    font-size:0.6em;
   background-color:yellow;
   vertical-align:text-top;
   text-decoration:line-through;
}
a{
   text-decoration: none;
   text-shadow: 5px 2px 5px black;
}
```

Fonts and Font Styles

- Fonts - font-style Google Fonts

Hello World!

Welcome to my site.

My name is Laurence Svekis.

CSS Heading 2

CSS Heading 4

Heading 6

Cascading Stylesheets

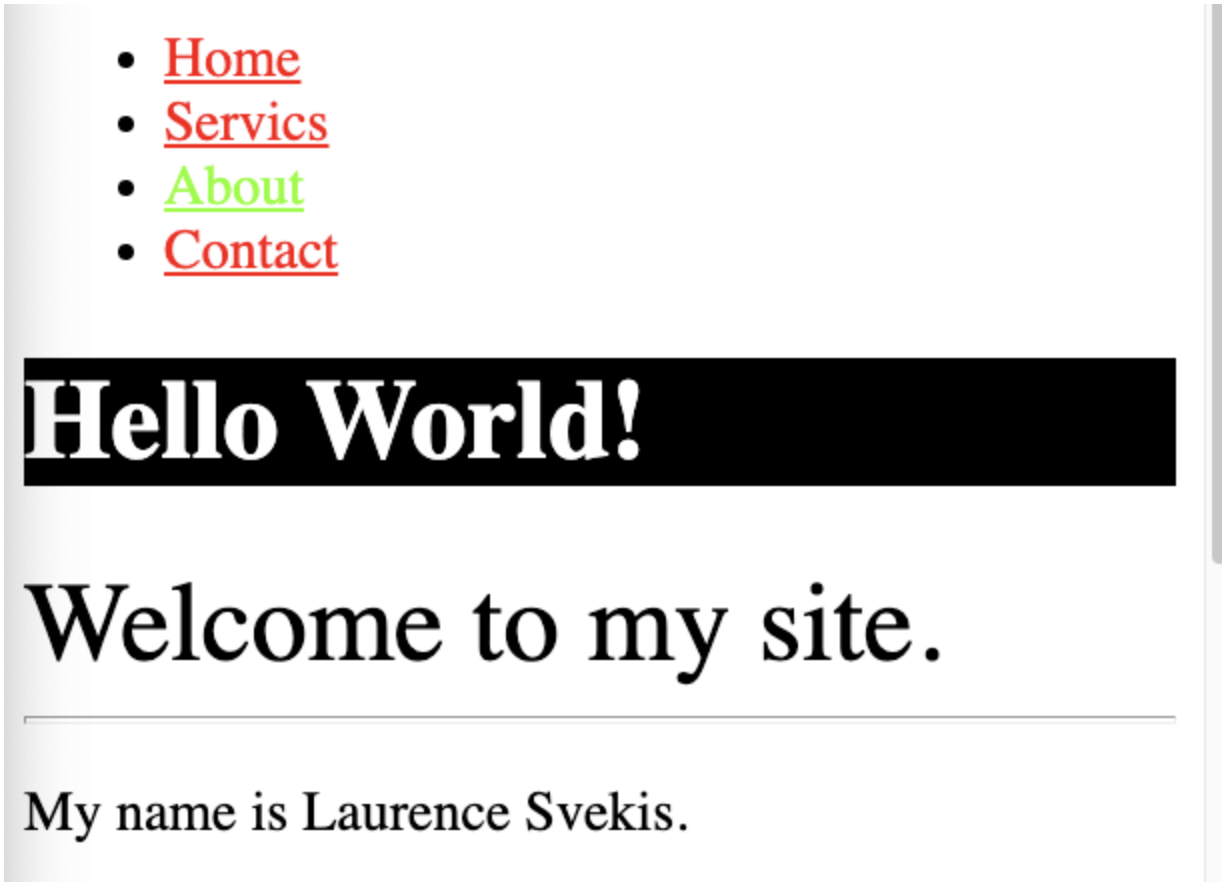
Cascading Stylesheets — or CSS — is the

https://fonts.google.com/specimen/Fira+Sans?category=Sans+Serif#standard-styles
https://developer.mozilla.org/en-US/docs/Web/CSS/font-size

```
@import
url('https://fonts.googleapis.com/css2?family=Fira+Sans&display=
swap');

body{
   font-family: 'Fira Sans', sans-serif;
   font-style: italic;
   font-weight: bold;
   font-variant: small-caps;
   font-size:100%;
}
h1{
   font-size: 12vw;
   color:blue;
   text-align: center;
}
.h6{
   color:aquamarine;
   font: italic bold 1.5em Arial;
   background-color:black;
}

#bottom{

   font-size: 1.5em;
}

.bigger{
   color:red;
   font-size:16px;
}
```

Link States Pseudo-Classes

- Link states - a:link a:visited a:hover a:active




```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Servics</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>


a:link{
  color:red;
```

```
}

a:visited{
   color:chartreuse;
}

a:hover{
   color:darkblue;
   text-decoration: none;
}

a:active{
   color:darkmagenta;
}
h1{
   background-color:black;
   color:white;
}
h1:hover{
   background-color:darkorange;
}
h1:active{
   background-color:red;
}

#bottom span:first-child {
   color:red;
}
```

Display Properties

- Display Properties - inline - none - block

https://developer.mozilla.org/en-US/docs/Web/CSS/display

```
li {
   background-color:black;
   display:inline-block;
   padding:10px;
   list-style: none;
}
ul {
   margin:0;
   padding:0;
}
a {
   color:white;
}

#bottom {
   display:block;
}

span{
   visibility:visible;
   display:none;
   border:1px solid black;
   font-size:1em;
}

 <nav>
   <ul>
     <li><a href="#1">Home</a></li>
     <li><a href="#2">Servics</a></li>
     <li><a href="#3">About</a></li>
```

```
    <li><a href="#4">Contact</a></li>
  </ul>
</nav>
```

CSS Position

- Position : static relative fixed absolute sticky

```
header{
  background-color:rgba(255,0,0,0.5);
  padding:10px;
  position:static;
}

header{
  position:relative;
}

header{
  position:fixed;
}

header{
  position:absolute;
  top:10px;
  left:10px;
  width:200px;
  height:200px;
}
header{
  position:sticky;
}
```

CSS Float

- Float and clear left right and none

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <link rel="stylesheet" href="style9.css">
 <title>My Page</title>
</head>
<body>
 <header>
   <h1>Welcome to my Website</h1>
 </header>
 <nav>
   <ul>
     <li><a href="#1">Home</a></li>
     <li><a href="#2">Servics</a></li>
     <li><a href="#3">About</a></li>
     <li><a href="#4">Contact</a></li>
   </ul>
 </nav>
 <section>
   <article class="left">
     <h1 class="blue">Hello World!</h1>
     <div style="font-size:2em">Welcome to my site.</div>
     <hr>
     <p title="My Name">My name is <span
class="blue">Laurence</span> <span></span>Svekis</span>.</p>
     <span class="bigger">Cascading Stylesheets </span>— or CSS
is used to style it and lay it out. <img src="download.png"> For
example, you can use CSS to alter the font, color, size, and
spacing of your content, <span>HTML</span>split it into
<span>HTML</span> multiple columns, or add animations and other
decorative features.<br><span>HTML</span>
```

```
    </article>
    <aside>
      side memu
    </aside>
</section>
 <footer id="bottom">
    <div>Thank you for visiting Laurence Svekis</div>
     <a href="#myID">Go to the top element with ID of myID.</a>
    <div class="green" >TEST</div>
 </footer>
</body>
</html>
```

```
li {
    float:right;
}
li {
    float:left;
}
ul{
    border:1px solid black;
    padding:30px;
}

article{
    width:80%;
    background:lightpink;
    float:left;
}
aside{
    width:20%;
    background:lightblue;
```

```
   float:left;
}
img{
   float:right;
}
footer{
   clear:both;
   background:lightslategray;
}
```

Useful CSS Properties

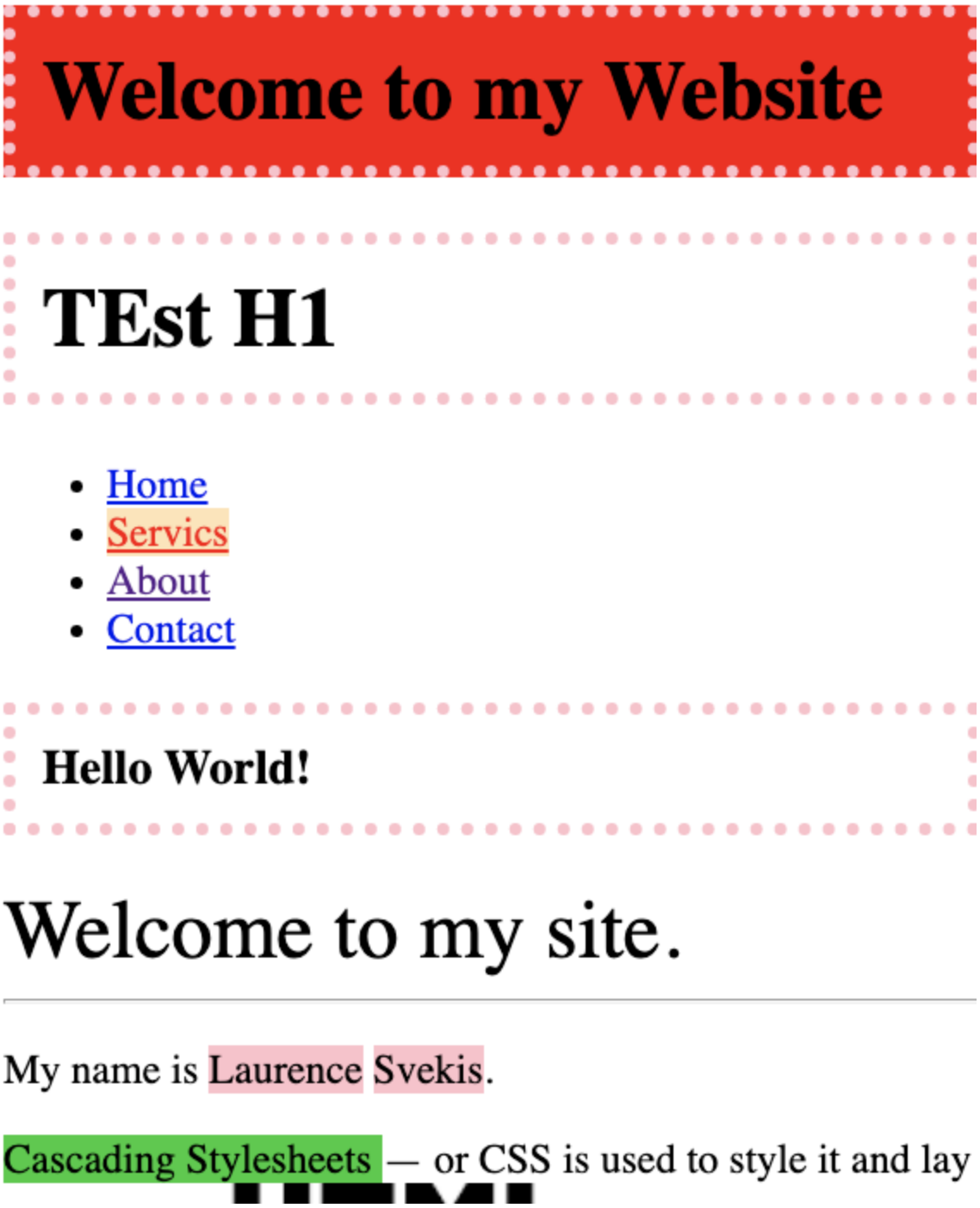
- Max Width
- Overflow
- Z-index
- outline

```
header {
   outline-color: magenta;
   outline-width: 2px;
   outline-style: solid;
   border: 5px solid yellow;
   max-width:800px;
   margin:auto;
}

ul{
   z-index: -1;
   position:fixed;
   top:0px;
   left:0px;
```

```
   outline: solid 2px green;
   background-color:midnightblue;
}
article, aside{
   position:absolute;
   left:0px;
   top:0px;

}
article{
   background:rgba(255,0,0,0.5);
   z-index: 5;
}
aside{
   background:blue;
   z-index: 3;
}
footer{
   font-size: 2em;
   height:20px;
   width:70%;
   background:black;
   color:white;
   overflow:auto;
   margin-bottom:100px;
}
```

CSS Combinators

- CSS combinators for selection - space child selector > https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Combinators



```
header h1{
   background-color: red;
}
article span{
   background-color:pink;
}
article > span{
   background-color:yellow;
}

article > p ~ span{
   background-color:purple;
}

article > p + span{
   background-color:rgb(27, 204, 56);
}

a[href="#2"] {
   color:red;
   background-color: moccasin;
}

h1,h2,h4,h6{
   padding:10px;
   border:5px dotted pink;
}
```

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
```

```
 <link rel="stylesheet" href="style11.css">
 <title>My Page</title>
</head>
<body>
 <header>
   <h1>Welcome to my Website</h1>
 </header>
 <h1>TEst H1</h1>
 <nav>
   <ul>
     <li><a href="#1">Home</a></li>
     <li><a href="#2">Servics</a></li>
     <li><a href="#3">About</a></li>
     <li><a href="#4">Contact</a></li>
   </ul>
 </nav>
 <section>
   <article class="left">
     <h1 class="blue">Hello World!</h1>
     <div style="font-size:2em">Welcome to my site.</div>
     <hr>
     <p title="My Name">My name is <span
class="blue">Laurence</span> <span>Svekis</span>.</p>
     <span class="bigger">Cascading Stylesheets </span>— or CSS
is used to style it and lay it out. <img src="download.png"> For
example, you can use CSS to alter the font, color, size, and
spacing of your content, <span>HTML</span>split it into
<span>HTML</span> multiple columns, or add animations and other
decorative features.<br><span>HTML</span>
   </article>
   <aside>
     side memu
```

```
   </aside>
</section>
 <footer id="bottom">
   <div>Thank you for visiting Laurence Svekis</div>
    <a href="#myID">Go to the top element with ID of myID.</a>
   <div class="green" >TEST</div>
 </footer>
</body>
</html>
```

CSS Pseudo Elements

- Pseudo-Elements - ::first-line , ::first-letter, ::after

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements

```
article::first-line{
   color:navy;
}
article::first-letter{
   color:red;
   font-size:3em;
}

article::before{
   content:'NEW content';
}
article::after{
   content:'After';
}
```

Responsive Website CSS Float


```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
```

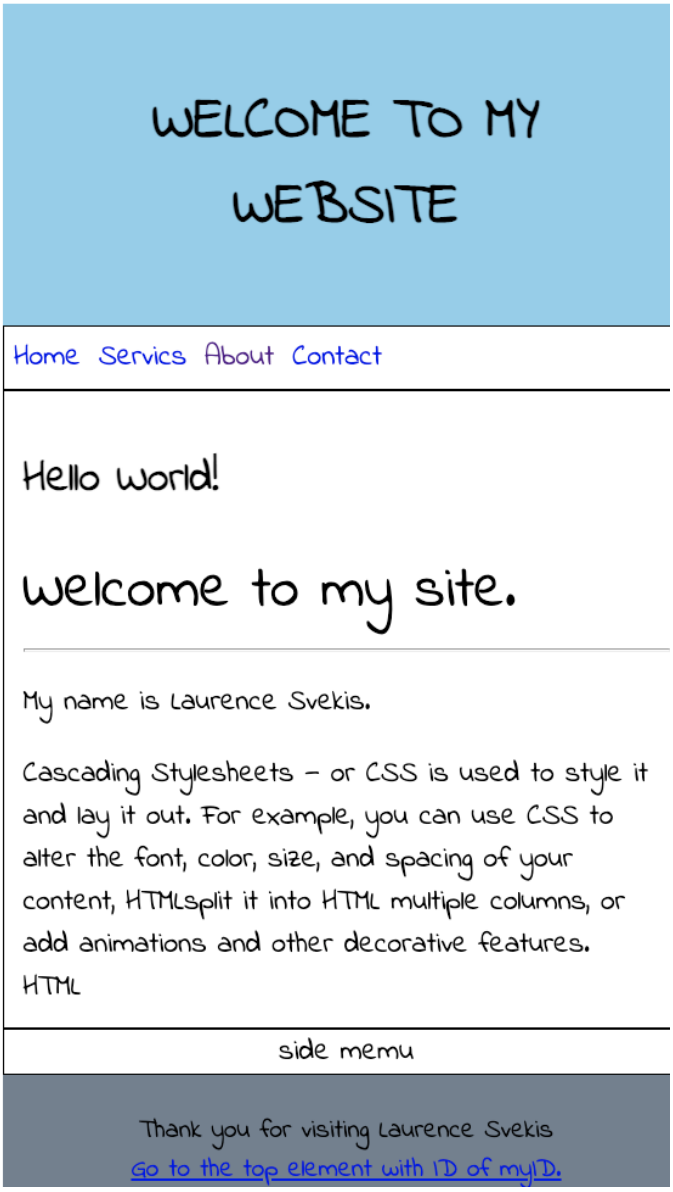
```
 <link rel="stylesheet" href="site1.css">
 <title>My Page</title>
</head>
<body>
 <header class="header">
   <h1>Welcome to my Website</h1>
 </header>
 <div class="main">
   <aside class="col">
     <ul>
       <li><a href="#1">Home</a></li>
       <li><a href="#2">Servics</a></li>
       <li><a href="#3">About</a></li>
       <li><a href="#4">Contact</a></li>
     </ul>
   </aside>
   <article class="left col">
     <h1 class="blue">Hello World!</h1>
     <div style="font-size:2em">Welcome to my site.</div>
     <hr>
     <p title="My Name">My name is <span
class="blue">Laurence</span> <span>Svekis</span>.</p>
     <span class="bigger">Cascading Stylesheets </span>— or CSS
is used to style it and lay it out.  For example, you can use
CSS to alter the font, color, size, and spacing of your content,
     <span>HTML</span>split it into <span>HTML</span> multiple
columns, or add animations and other decorative
     features.<br><span>HTML</span>
   </article>
   <aside class="col">
     side memu
   </aside>
```

```
 </div>
 <footer class="footer">
   <div>Thank you for visiting Laurence Svekis</div>
   <a href="#myID">Go to the top element with ID of myID.</a>
   <div class="green">TEST</div>
 </footer>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <link rel="stylesheet" href="site1.css">
 <title>My Page</title>
</head>
<body>
 <header class="header">
   <h1>Welcome to my Website</h1>
 </header>
 <div class="main">
   <aside class="col">
     <ul>
       <li><a href="#1">Home</a></li>
       <li><a href="#2">Servics</a></li>
       <li><a href="#3">About</a></li>
       <li><a href="#4">Contact</a></li>
     </ul>
   </aside>
   <article class="left col">
     <h1 class="blue">Hello World!</h1>
     <div style="font-size:2em">Welcome to my site.</div>
```

```
    <hr>
    <p title="My Name">My name is <span
class="blue">Laurence</span> <span>Svekis</span>.</p>
    <span class="bigger">Cascading Stylesheets </span>— or CSS
is used to style it and lay it out.  For example, you can use
CSS to alter the font, color, size, and spacing of your content,
    <span>HTML</span>split it into <span>HTML</span> multiple
columns, or add animations and other decorative
    features.<br><span>HTML</span>
  </article>
  <aside class="col">
    side memu
  </aside>
 </div>
 <footer class="footer">
  <div>Thank you for visiting Laurence Svekis</div>
  <a href="#myID">Go to the top element with ID of myID.</a>
  <div class="green">TEST</div>
 </footer>
</body>
</html>
```

Responsive Website Flexbox

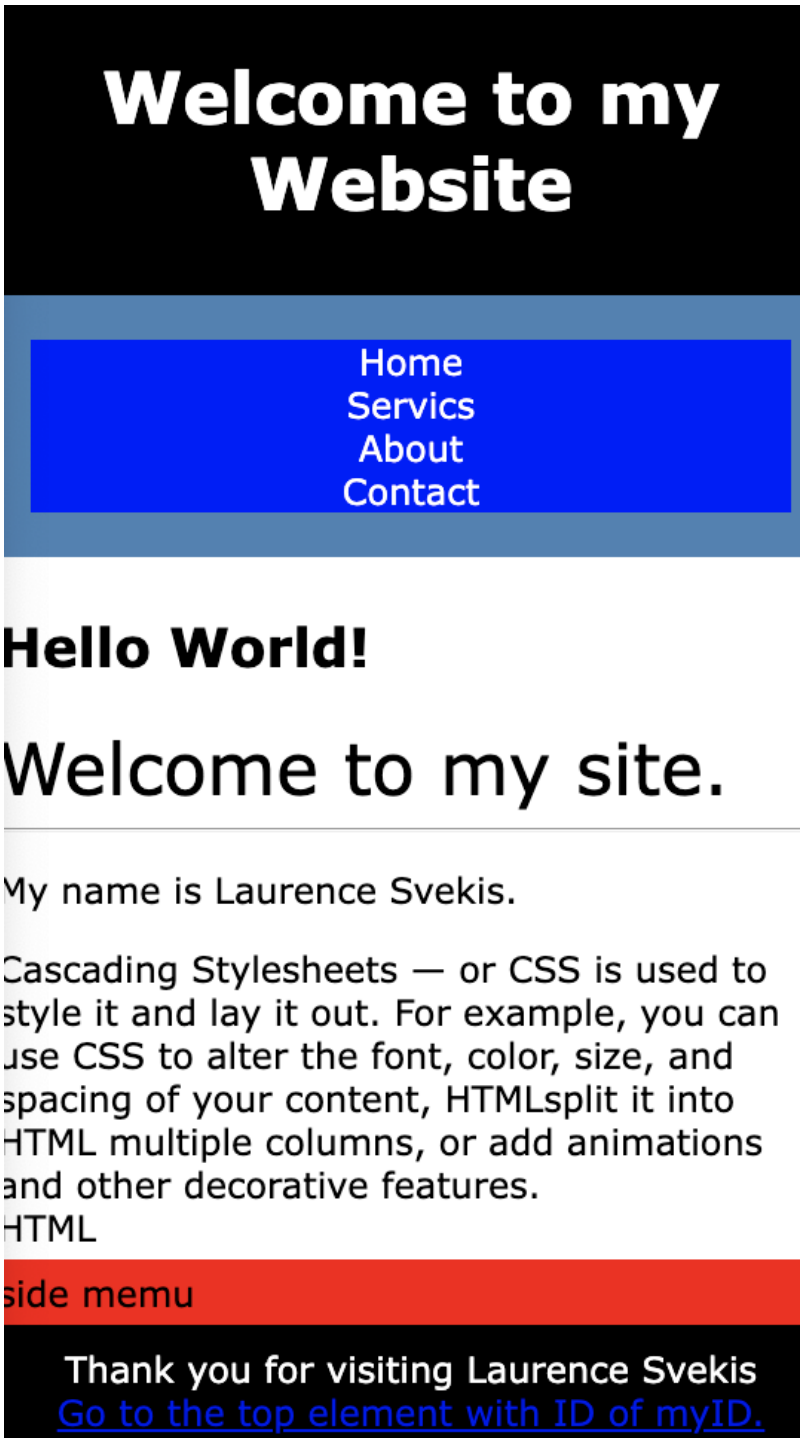



```
@import url('https://fonts.googleapis.com/css2?family=Indie+Flower&display=swap');

* {
   box-sizing: border-box;
}

body {
   font-family: 'Indie Flower', cursive;
}
```

```css
.header {
   background-color: skyblue;
   text-align: center;
   padding: 20px;
   text-transform: uppercase;
}

ul {
   display: flex;
   flex-direction: row;
   margin: 0;
   padding: 0;
}

li {
   list-style-type: none;
   padding: 5px;

}

li:hover {
   background-color: steelblue;
}

li:hover a {
   color: white;
}

li a {
   text-decoration: none;

}

.main {
   display: flex;
   flex-direction: column;
}

.main article {
   flex: 3;
   padding: 10px;
}
```

```css
.main aside {
   flex: 1;
   text-align: center;
}

.col {
   border: 1px solid black;
}

.footer {
   background-color: slategrey;
   padding: 20px;
   text-align: center;
   font-size: 0.9em;
}

@media (min-width:640px) {
   body {
       background-color: #ddd;
   }

   .main {
       flex-direction: row;
   }

   ul {
       flex-direction: column;
   }

   li {
       padding: 10px;
   }
}
```

Responsive Website CSS Grid




```
* {
   box-sizing: border-box;
}
body{
   background-color:#ddd;
```

```
   font-family: Verdana, Geneva, Tahoma, sans-serif;
}

.header , .footer{
   background-color: black;
   color:white;
   text-align: center;
   padding:10px;
}
.main{
   display:grid;
   grid-template-areas:
   'cola colb  colb colc';
}

ul {
   margin:0px;
   padding:0px;
}
li{
   list-style-type: none;
}
li a{
   text-decoration: none;
   display:block;
   background-color:blue;
   color:white;
   text-align: center;
}
li a:hover{
   background-color:red;
}
.col{padding:5px;}
.main aside:first-child{
   grid-area: cola;
   background-color:steelblue;
   padding:20px;
}
.main article{
   grid-area: colb;
   background-color:white;
}
```

```
.main aside:last-child{
   grid-area: colc;
   background-color:red;
}
.footer{}

@media (max-width:640px){
   .main{
       grid-template-areas:
       'cola'
       'colb'
       'colc';
   }
}
```

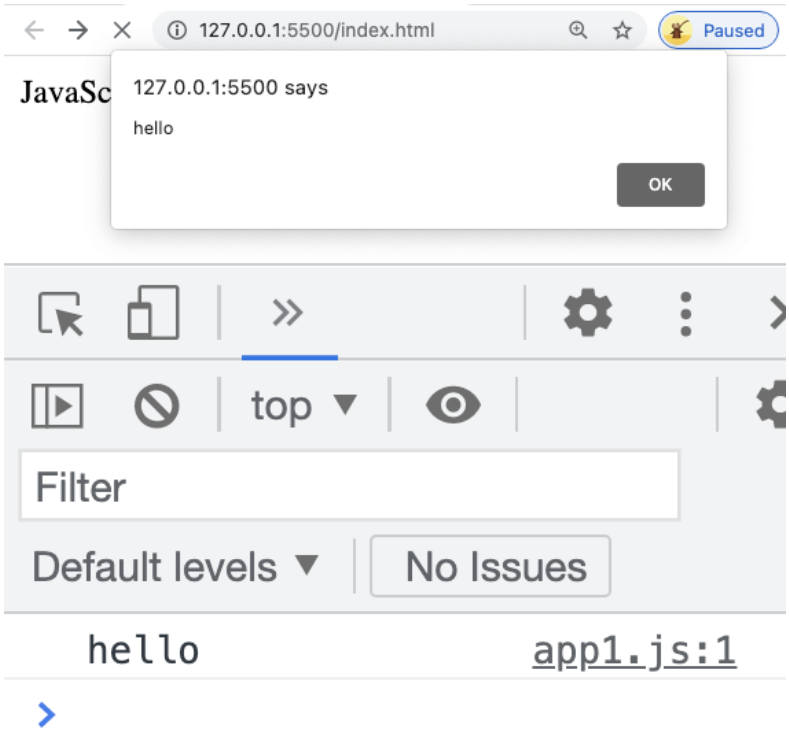
JavaScript Code

JavaScript code runs your browser as your html page loads. Adding JavaScript to your code can improve the user experience of the web page. Go from a static webpage to an interactive one with JavaScript. This section will cover the basics of getting started with JavaScript and the code syntax to write JavaScript code. JavaScript is a set of instructions you can add in your code to let the browser know what to do next.

- Variables are at the heart of coding
- The different data types of JavaScript and how JavaScript can set the data type.
- The power of Objects and Arrays and how you can use them to store multiple values in one variable.
- Operators to provide calculations within your code
- Functions to run predefined blocks of code as needed.
- Conditions for logic within your code
- Loops to save time and iterate over blocks of code

Getting Started with JavaScript

- JavaScript Introduction alert prompt




```
<!doctype html>
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <div class="output" onclick="alert('hello')">JavaScript</div>
 <script src="app1.js"></script>
</body>
</html>
```

```
console.log('hello');
alert('hello');
alert('world');
console.log('world');
```

JavaScript Variables

- Variables Let and Const

```
console.log('ready');
// No space in the variable name
// $_0-9a-zA-Z
// Case sensitive
// Can't begin with 0-9
// can't use reserved JavaScript Keyword
/// var used prior to let and const introduction
let myName = 'Laurence Svekis';
console.log(myName);
myName = 'Laurence Smith';
console.log(myName);
const myName1 = 'Laurence 1';
///myName1 = 'Linda';
console.log(myName1);



if(true){
   const myName1 = 'Laurence 2';
   console.log(myName1);
   console.log(myName);
}
```

JavaScript reserved keywords
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar

JavaScript DataTypes

- Data Types

```
const myName = 'Laurence\'s "Svekis"'; //String
let val = "String's";
val = 100;
val = "100" + 100 + 100 + "100";
val = undefined;

let val1;
val = null;

let a,b,c,d;
b =100; // Number
val = true; //boolean
val = false; //boolean

console.log(myName);
console.log(val);
console.log(typeof(b));
```

JavaScript Objects and Arrays

- Arrays and Objects

```
const myArr = ['string',100,true];
//console.log(myArr);
//console.log(myArr[0]);

myArr[0] = 'New Value';
```

```
//console.log(typeof myArr);

const myObj = {
   first:'string',
   val : 100,
   boo : true
};

//console.log(myObj);
//console.log(typeof myObj);

//console.log(myObj['boo']);
myObj['boo'] = 'New Value';

//console.log(myObj.boo);

const myName = {
   first : 'Laurence',
   last : 'Svekis',
   arr : [1,2,3,4],
   myObj : {
       one : 'one',
       two : 'two'
   }
}
console.log(myName.myObj.one);
console.log(myName['myObj']['one']);

const myArr2 = myArr;
myArr2[2] = 'wow';
console.log(myArr);
console.log(myArr2);
```

```
const myObj2 = myObj;
myObj.first = 'Laurence';
console.log(myObj2);
```

JavaScript Operators

- Operators

```
let val = 1;
val = val * 5;
val = val - 3;
val = val / 2;
val = val + 10 + 10 +  30;
val = 51 % 10;
val++;
val--;

val -= 3;
val += 10;
val *= 5;

let val1 = "Laurence";
let val2 = "Svekis";

val = val1 + " " +  val2;
val += " Course Instructor";
//console.log(val);
```

```
let val3 = 10 + 10 + "10";
//console.log(val3);

let output = (10 == 10);
output = (10 != 10);
output = (10 !== "10");
output = (5 <= 10);
console.log(output);
```

JavaScript Functions

- Functions
- function expressions vs function declarations

function declarations

- global scope and make it available throughout your code

function expression

- function expression can be used as an IIFE

```
//console.log(myFun2());
//myFun2();
///myFun2();
const val2 = myFun2();
//console.log(val);

const myFun1 = function(){
   //console.log('Fun 1');
   return '1';
}

const val1 = myFun1();
//console.log(val1);
```

```
//myFun1();
//myFun1();

const myFun3 = function(){
   //console.log('Fun 3');
   return '3';
}();
//console.log(myFun3);



function myFun2(){
   //console.log('Fun 2');
   return '2';
}
let val = 100;
val = adder(5,10);
console.log(val);
console.log(adder(7,80));
console.log(adder(117,80));

let a = 50;
let b = 94;
let test = a + ' + ' + b + ' = '+ adder(a,b);
console.log(test);
console.log(a + ' + ' + b + ' = '+ adder(a,b));

function adder(a,b){
   //let val = a + b;
   console.log(val);
   return a + b * 1;
   console.log('message');
}
```

JavaScript Conditions

- Conditions
- if, else if, and else

```
let boo = true; //null 0 undefined
if (boo ) {
   console.log('boo is true');
} else if(boo == false) {
   console.log('boo is false');
} else {
   console.log('boo is something else');
}



let a = 40;
let b = 10;
checker(50,100);
checker(70,10);
checker(80,100);

function checker(a,b){
   let res;
   if ( a > b) {
      res = (a + ' is bigger than ' + b);
   } else {
      res = (a + ' was not bigger than ' + b);
   }
   console.log(res);
```

```
}
```

JavaScript Loops

- Loops Do While, For, While, for each

```
for (let x=0;x<10;x++) {
   //console.log(x);
}
let x = 100;
while ( x < 10) {
   console.log(x);
   x++;
}
do {
   //console.log(x);
   x++;
}
while( x < 10 )
//console.log(x);

const test = [10,34,54,32,32234,3234];

//console.log(test.length);
for(let x=0 ; x < test.length ; x++){
   //console.log(test[x]);
}

for(let item of test){
   //console.log(item);
```

```
}

for (let x in test) {
   //console.log(x + " = " + test[x]);
}

test.forEach(function(val,index,array){
   console.log(val + ' ' + index);
});

const myObj = {
   first : 'Laurence',
   last : 'Svekis',
   one : 1,
   two : 2,
   three : 3
}

for (let x in myObj){
   //console.log(x + ':' + myObj[x]);
}
```

JavaScript and the DOM

Document Object Model (DOM) is a programming interface for HTML documents, that is the logical structure of a page and how the page content can be accessed and manipulated. Bring your web pages to life with JavaScript and connect to the web page elements. Create fully interactive content that responds to the user. Create Dynamic web page content that can change without page refresh and present new elements and updated content to the user. Improve your web users experience with JavaScript and the DOM.

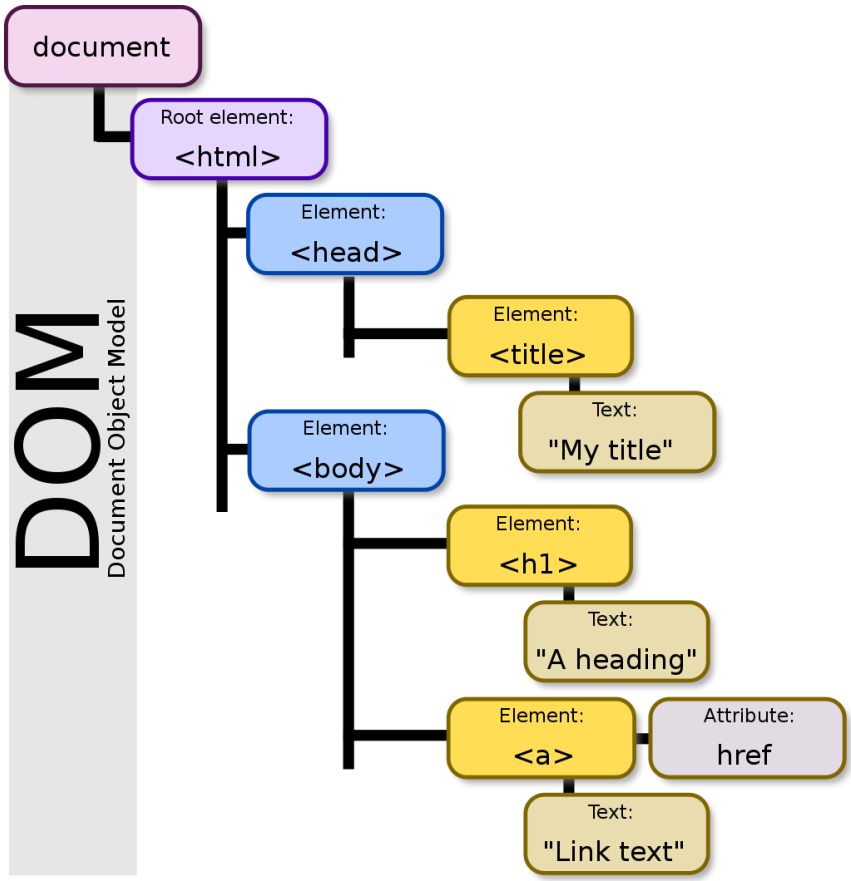
- What is the DOM Document Object Model
- How to select elements from your webpage with JavaScript
- Manipulate and change your page elements with JavaScript
- How to set styling attributes for elements
- Make them interactive with Event listeners
- DOM events and Page events with JavaScript
- How to create elements with code and add them to your webpage
- Moving elements and animation of elements.

Introduction to the DOM

Document Object Model (DOM)

- DOM tree

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model




```
console.dir(document);
let val = document.URL;
console.log(val);
```

```
let ele = document.querySelector('.output');
console.dir(ele.textContent);
ele.textContent = "Hello World";

let myDOM = {
   URL : "My URL",
   children : [
       {html : {
           "body" : "Content"
       }}
   ]
}
```

Element Selection

- Element Selection
- Multiple Elements selection

```
const ele1 = document.getElementById('myEle');
console.log(ele1);

const ele2 = document.querySelector('div.output');
console.log(ele2);

const val1 = 'h1';
const ele3 = document.querySelector(val1);
console.log(ele3);

const ele4 = document.querySelector('#myEle');
console.log(ele4);

console.log(ele1.textContent);
```

```
ele1.textContent = 'Laurence Svekis';
console.log(ele4.textContent);

const myObj1 = {
   first : "Laurence"
}
const my1 = myObj1;
const my2 = myObj1;
my1.first = "Svekis";

console.log(my2);

const eles1 = document.querySelectorAll('.output');
console.log(eles1);
const eles2 = document.querySelectorAll('h1');
console.log(eles2);
console.clear();
eles1.forEach(function(ele,ind){
   console.log(ele.textContent);
   ele.textContent = 'My Element ' + ind;
})
```

```
<!doctype html>
<html>
<head>
 <title>JavaScript</title>
 <style>
   div.output{
     color:red;
   }
 </style>
</head>
```

```
<body>
 <h1 class="output">Hello DOM</h1>
 <div class="output">JavaScript</div>
 <div id="myEle">Hello World</div>
 <script src="dom2.js"></script>
</body>
</html>
```

Element Manipulation DOM

- Element Manipulation

```
const outputs = document.querySelectorAll('.output');
let html = '<div style="color:red">Laurence <br> Svekis</div>';
console.dir(outputs[0].tagName);
if(outputs[0].tagName != 'H1'){
   outputs[0].textContent = html;
}
outputs[1].innerHTML = html;

outputs.forEach(function(el){
   if(el.tagName != 'H1'){
       el.innerHTML = html;
   }
})

outputs.forEach((el,ind)=>{
   if(el.tagName != 'H1'){
       el.innerHTML = html;
   }
   el.innerHTML += ' ' + ind;
```

```
})

document.write(html);
```

Element styling attributes

- Element Manipulation

```
const output = document.querySelector('.output');
console.log(output.style);
output.style.color = 'red';
console.log(output.style.color);

const ele1 = document.querySelector('div.output');
ele1.style.background = '#0000ff';
ele1.style.color = 'rgb(255,255,255)';
ele1.style.padding = '10px';
ele1.style.border = '2px solid black';
ele1.style.borderColor = 'green';
ele1.style.fontSize = '30px';

ele1.setAttribute('style','');

const eles = document.querySelectorAll('div');
eles.forEach((el,index)=>{
   console.log(el);
   el.style.textTransform = 'uppercase';
   if(el.classList.contains('output')){
       el.innerHTML += '***OUTPUT***';
   }
   el.classList.add('box');
```

```
   el.classList.remove('output');
   el.classList.toggle('active');
   el.setAttribute('id','id'+(index+1));
})
```

```
<!doctype html>
<html>
<head>
 <title>JavaScript</title>
 <style>
   .box{
     background-color:blueviolet;
     color:white;
     padding:10px;
     margin:10px;
   }
 </style>
</head>
<body>
 <h1 class="output">Hello DOM</h1>
 <div class="output">JavaScript</div>
 <div id="myEle">Hello World</div>
 <div>Hello World 2</div>
 <div class="output">JavaScript</div>
 <script src="dom4.js"></script>
</body>
</html>
```

DOM and Form Elements

- Input Values

```
const first = document.querySelector('input[name="first"]');
const last = document.querySelector('input[name="last"]');
const chk = document.querySelector('input[type="checkbox"]');
first.setAttribute('placeholder','first name');
last.setAttribute('placeholder','last name');

first.value = 'Laurence';
last.value = 'Svekis';

chk.checked = true;
```

DOM Element Event Listeners

- Adding event listeners
- onclick vs addeventlistener

```
const btn = document.querySelector('button');
let counter = 0;

btn.addEventListener('click',btnClicked);

function btnClicked(e){
   console.log(e.target.id);
   e.target.id = 'NEW' + counter;
   counter++;
   console.log('clicked '+counter);
```

```
    e.target.textContent = 'Clicked ('+counter+')';
    e.target.classList.toggle('box');
}

const h1 = document.querySelector('h1');
h1.onclick = btnClicked;

const holder = document.querySelector('.holder');
const divs = holder.querySelectorAll('div');
console.log(divs);
console.clear();
divs.forEach((el,ind)=>{
    const val = el.className;
    el.onclick = function(){
        console.log('onclick ' + val);
    }
    el.addEventListener('click',(e)=>{
        console.log('event False ' + val);
    },false);
    el.addEventListener('click',(e)=>{
        console.log('event True ' + val);
    },true);
})
```

```
<!doctype html>
<html>
<head>
 <title>JavaScript</title>
 <style>
```

```
   .box{
     background-color:blueviolet;
     color:white;
     padding:10px;
     margin:10px;
   }
 </style>
</head>
<body>
 <h1 id="h1" class="output" >Hello DOM</h1>
 <div class="output">JavaScript</div>
 <div id="myEle">Hello World</div>
 <input name="first">
 <input name="last">
 <input type="checkbox">
 <button>Click Me</button>
 <div>Hello World 2</div>
 <div class="output">JavaScript</div>

 <div class="holder">
   <div class="one">One
     <div class="two">Two
       <div class="three">
           Three
       </div>
     </div>
   </div>
 </div>

 <script src="dom6.js"></script>
</body>
</html>
```

JavaScript DOM Events

```
const btn = document.querySelector('button');
const holder = document.querySelector('.holder');
const inputs = document.querySelectorAll('input');
const h1 = document.querySelector('h1');
btn.addEventListener('click',adder);
btn.addEventListener('click',(e)=>{
   console.log('two');
})

holder.addEventListener('mouseover',(e)=>{
   holder.style.background = 'red';
   //holder.classList.toggle('box');
})
holder.addEventListener('mouseout',(e)=>{
   holder.style.background = 'white';
   //holder.classList.toggle('box');
})
holder.addEventListener('click',(e)=>{
   holder.style.background = 'blue';
})
holder.addEventListener('click',(e)=>{
   output('test',e);
});

function output(a,e){
   console.log(a);
}
```

```
function adder(e){
   const el = e.target;
   el.removeEventListener('click',adder);
   el.style.color = '#ddd';
   console.log('clicked');
}

inputs.forEach((el)=>{
   el.addEventListener('change',updater);
   el.addEventListener('focus',(e)=>{
       console.log(el.textContent);
       el.style.background ='red';
       el.style.color = 'white';
   })
   el.addEventListener('blur',(e)=>{
       console.log(el.textContent);
       el.style.background ='white';
       el.style.color = 'black';
   })
})

function updater(e){
   const val = e.target.value;
   h1.textContent = val;
}
```

```
h1.addEventListener('click',(temp,{once:true})

function temp()
   console.log('test');
   h1.style.color = 'red';
```

```
}
```

Page Events DOM

```
const h1 = document.querySelector('h1');

document.body.onload = ()=>{
   console.log('body loaded');
}



document.addEventListener('DOMContentLoaded',(e)=>{
   console.log('ready');
   const output = document.querySelector('.output');
   console.log(output);
})

document.addEventListener('keydown',(e)=>{
   console.log(e.key);
   h1.textContent += e.key;
})
document.addEventListener('keyup',(e)=>{
   console.log(e.key);


})

console.dir(window);
//window.alert('hello');
window.onresize = ()=>{
   h1.innerText = window.innerHeight + ' ' + window.innerWidth;
}
```

Create Elements JavaScript

- Create Elements

```
const first = document.querySelector('input');
const btn = document.querySelector('button');
const holder = document.querySelector('.holder');
const div = document.createElement('div');
const output =  document.querySelector('div.output');
output.style.border = '1px solid black';
let counter = 0;
output.addEventListener('click',adder);

first.value = "Laurence";
btn.addEventListener('click',(e)=>{
   //holder.innerHTML = "";
   console.log(first.value);
   div.textContent = first.value;
   console.log(div);
   const div1 = document.createElement('div');
   holder.append(div1);
   div1.textContent = "HELLO";
   holder.append(div);
   holder.prepend(div);
   div.style.color = 'red';
})
```

```
function adder(){
   counter++;
   console.log('clicked');
   const newEle = createEle('div',output,'hi '+counter);
   console.log(newEle);
   newEle.style.background = 'blue';
}

function createEle(elType,parent,html){
   const ele = document.createElement(elType);
   ele.innerHTML = html;
   return parent.appendChild(ele);
}
```

Element Movement

```
const holder = document.querySelector('.holder');
holder.style.position = 'absolute';
const ele = {x:holder.offsetLeft,y:holder.offsetTop}
holder.addEventListener('click',(e)=>{
   ele.x += 50;
   ele.y -= 5;
   holder.style.left = ele.x + 'px';
   holder.style.top = ele.y + 'px';
})

document.addEventListener('keydown',(e)=>{
   console.log(e.key);
   if(e.key == 'ArrowLeft'){
       ele.x -= 50;
```

```
    }
    if(e.key == 'ArrowRight'){
        ele.x += 50;
    }
    if(e.key == 'ArrowUp'){
        ele.y -= 50;
    }
    if(e.key == 'ArrowDown'){
        ele.y += 50;
    }
    holder.style.left = ele.x + 'px';
    holder.style.top = ele.y + 'px';
})
```

Element Animation

https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame

```
<!doctype html>
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <script src="dom11.js"></script>
</body>
</html>
```

```
const output = document.createElement('div');
```

```
document.body.prepend(output);
output.textContent = 'JavaScript';
output.style.width = '100px';
output.style.height = '100px';
output.style.background = 'red';
output.style.textAlign = 'center';
output.style.lineHeight = '100px';
output.style.color = 'white';
output.style.position = 'absolute';

function ran(){
   return Math.floor(Math.random()*255)
}

output.addEventListener('click',(e)=>{
   const col = 'rgb('+ran()+','+ran()+','+ran()+')';
   output.style.background = col;
   console.log(col);
})
const player = {
   x : 0,
   y : 0,
   speed : 10,
   ani : window.requestAnimationFrame(moveMe)
}

const keyz = {
   ArrowDown : false,
   ArrowUp : false,
   ArrowRight : false,
   ArrowLeft : false
}
```

```
window.addEventListener('keydown',(e)=>{
   keyz[e.code] = true;
})
window.addEventListener('keyup',(e)=>{
   keyz[e.code] = false;
})

function moveMe(){
   if(keyz.ArrowRight){player.x += player.speed}
   if(keyz.ArrowLeft){player.x -= player.speed}
   if(keyz.ArrowUp){player.y -= player.speed}
   if(keyz.ArrowDown){player.y += player.speed}
   output.style.left = player.x + 'px';
   output.style.top = player.y + 'px';
   player.ani = window.requestAnimationFrame(moveMe);
}
```