



**Generation Girl Summer Club 2020**

# **Build a Website I**

**MAIN ARCHIVE (DAY 1 TO 5)**

```
<!--CSS-->
<link type="text/css" rel="stylesheet" href="css/materialize.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="/css/animate.css">
<link rel="stylesheet" href="css/theme.css">
</head>
<body>
  <div class="banner">
    <div class="row">
      <div class="col s12 m12 l12" style="text-align: center;">
        <h1>Generation Girl Summer Club 2020</h1>
        <h2>Build a Website I</h2>
        <hr/>
        <div class="row" style="margin-top: 10px;">
          <div class="col s12 m12 l12" style="text-align: center;">
            
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
```

Generation Girl Summer Club 2020

# Build a Website I

DAY ONE



# Today's Outlines

## Introductions

HTML Basics

CSS Basics

Activity

# Hello! I'm Matahari

- IT major at NYU Tandon School of Engineering, minoring in Math & CS
- Graphic Designer for NYU Tandon Career Services & UX Engineer for Kamar Pelajar (@kamarpelajar)
- Traveler
- Musician?? Kind of lol





# Today's Outlines

Introductions

HTML Basics

CSS Basics

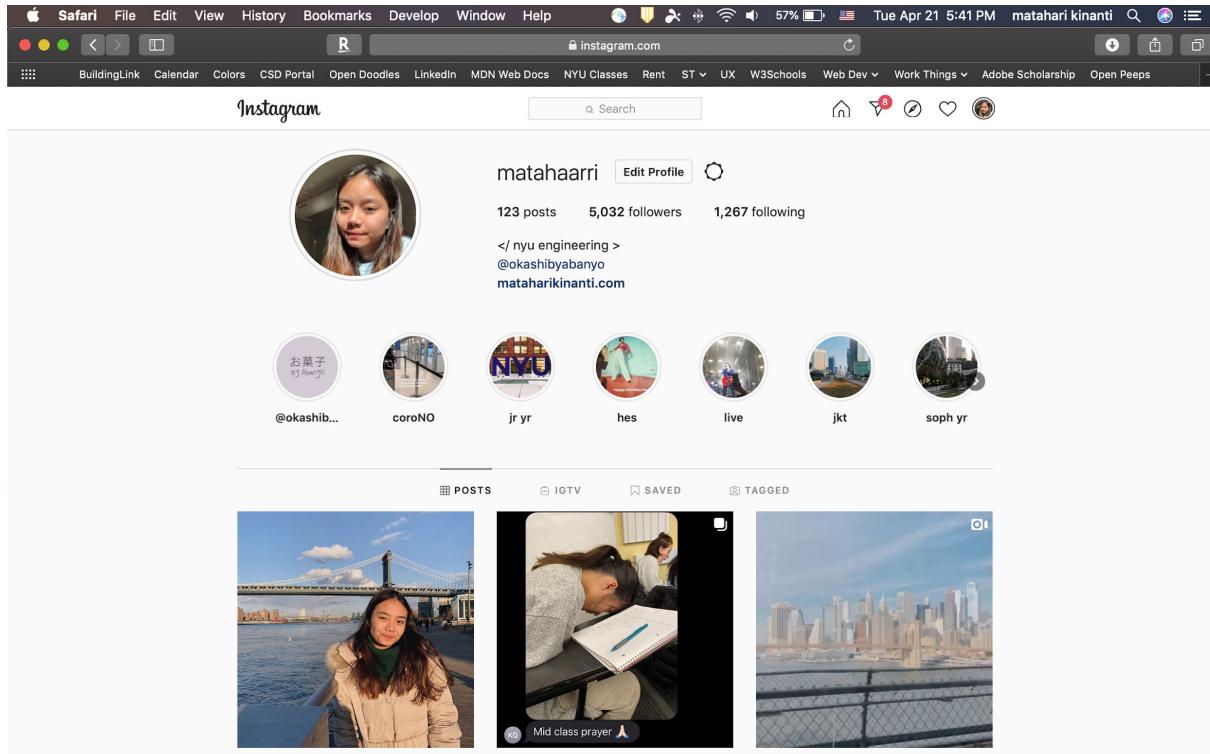
Activity

# What is HTML?

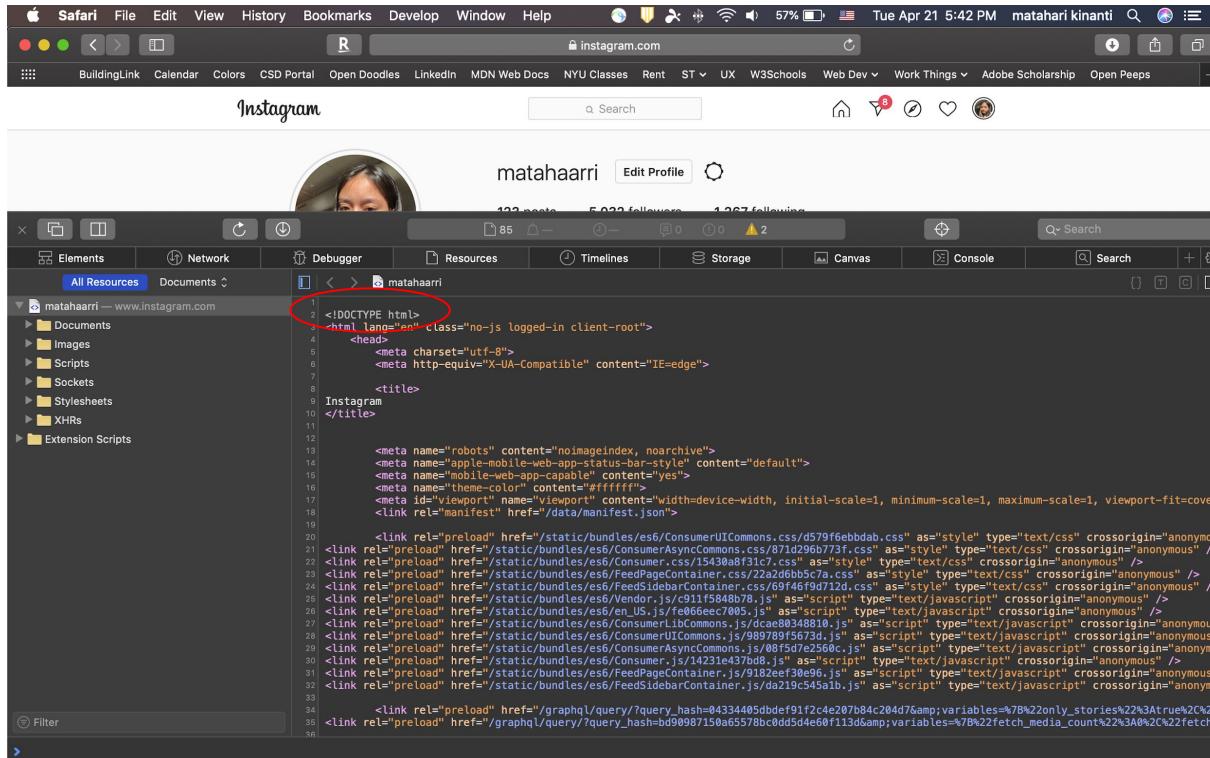
- Hyper Text Markup Language
  - Standard language for the web
  - HTML elements are building blocks of HTML pages
  - We use HTML to **display content** and **create structure (layout)**
  - We can use any text editor (including text edit!) to write html, as long as the extension of the file is .html



# Example: Instagram



# Example: Instagram



# HTML Document

- `<!DOCTYPE html>` is an instruction that tells the web browser what version of HTML is being used (in this case it's HTML5)
- The `<head>` element contains meta information of a document while the `<body>` contains web page elements
- The `<title>` tag declares the title of the page
- `<h1>` and `<p>` are content tags

```
<!DOCTYPE html>
<html>
  <head>
    <title> Page Title </title>
  </head>
  <body>
    <h1> Hello, this is a very simple HTML
        website! </h1>
    <h2> This is a smaller heading. </h2>
    <p> This is a paragraph! </p>
  </body>
</html>
```

# HTML Tags

- HTML tags are hidden *keywords* within a web page that defines how your web browser must **format** and **display** the content
- **Most** tags have two parts, a start and an end. For example, <html> is the start tag and </html> is the end tag
- Learn more about HTML tags:

<https://www.w3schools.com/tags/default.asp>

```
/* Common HTML Structure Tags */
<html></html>
<body></body>
<div></div>
<table>
    <tr>
        <td></td>
    </tr>
</table>
<nav></nav>
<section></section>
<header></header>
<footer></footer>

/* Common HTML Content Tags */
<title></title>
<img/>
<p></p>
<a></a>
<b></b>
<u></u>
<i></i>
<span></span>
<blockquote></blockquote>
<ul></ul>
<ol></ol>
<li></li>
<h1></h1>
<h2></h2>
<h3></h3>
<h4></h4>
<h5></h5>
<h6></h6>
```

```
<tagname attribute="value"> ... </tagname>
```

# HTML Attributes

- A description for the HTML tag
- They **MUST** always be placed in the opening tag
- We will have examples of attributes in the next few slides

# Headings

- Headings are defined with the `<h1>` to `<h6>` tags
- `<h1>` defines the most important heading, `<h6>` defines the least important heading

```
<h1> Heading 1 </h1>
<h2> Heading 2 </h2>
<h3> Heading 3 </h3>
<h4> Heading 4 </h4>
<h5> Heading 5 </h5>
<h6> Heading 6 </h6>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

# Paragraphs

```
<p>  
Hello, world! We use the p tag to create a paragraph. The difference between this and the h  
tag is that the default style of the p tag is smaller than the heading and is not bolded. But,  
we can use other tags like the b or em tag to <b> bold our words </b> within the p tag. We  
could also <i> italicize something </i> or <u> underline a phrase </u> using embedded tags!  

```

# Images

- To insert images to your website we use the `<img>` tag
- This is one of the HTML tags that doesn't require a closing tag
- The `src` attribute is the source of your image (where is it located within the folder)
- The `alt` attribute is an alternative text that will appear if the image is not found/source is wrong

```

```

i, matahari,  
want...



This is an image of a cat



## The code:

```
<div class="i-believe">
  <h3> i, matahari, want...</h3>

</div>
```

# Table

- <table> defines the grid system
- <tr> defines a table row
- <th> defines a table heading
- <td> defines a table cell or table data

First Name	Last Name	Age
Jane	Doe	20
John	Doe	20

P.S. I added styling here

```
<table>
  <tr>
    <th> First Name </th>
    <th> Last Name </th>
    <th> Age </th>
  </tr>
  <tr>
    <td> Jane </td>
    <td> Doe </td>
    <td> 20 </td>
  </tr>
  <tr>
    <td> John </td>
    <td> Doe </td>
    <td> 20 </td>
  </tr>
</table>
```

# Lists

- <ul> is an **unordered** list
- <ol> is an **ordered** list
- We have to use the <li> tags in between both the <ul> or <ol> tags

```
<h2> Drinks: </h2>
<ul>
    <li> Coffee </li>
    <li> Tea </li>
    <li> Water </li>
</ul>

<h2> US Cities Ranked by Me: </h2>
<ol>
    <li> New York </li>
    <li> Los Angeles </li>
    <li> Orlando </li>
</ol>
```

# Links

- We use the `<a>` tag to link a hyperlink to our website
- The common attributes for the `a` tag are `href` and `target`

```
<h3> Links </h3>
<p>
    <a href="https://www.google.com"> This
    will take you to Google </a>

    <br>

    <a href="https://www.nyu.edu">
        target="_blank"> This will open the NYU
        homepage on another tab </a>
</p>
```

Let's get started on building your website!



# Today's Outlines

Introductions

HTML Basics

CSS Basics

Activity

# What is CSS?

- Cascading Style Sheets
  - Used to “beautify” and style your webpages
  - There are three types of CSS:
    - Embedded: Located inside the `<head></head>` tag of your HTML document
    - External: A separate document with a `.css` extension
    - Inline: As an attribute of your tags



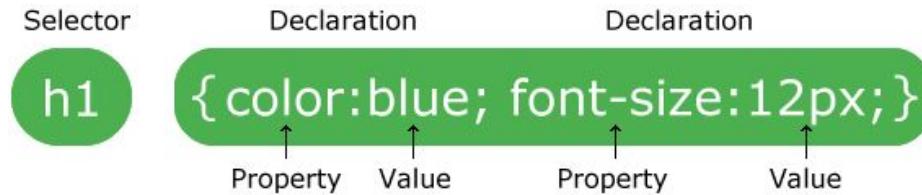
# Cascading Order

- If different styles are specified for HTML elements, the styles will cascade into new styles with the following priority:
    - Inline
    - External and internal (embedded)
    - Browser default



# Syntax

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

**Please open [css1.html](#) and [css1.css](#)**

Let's style your site!



# Today's Outlines

Introductions

HTML Basics

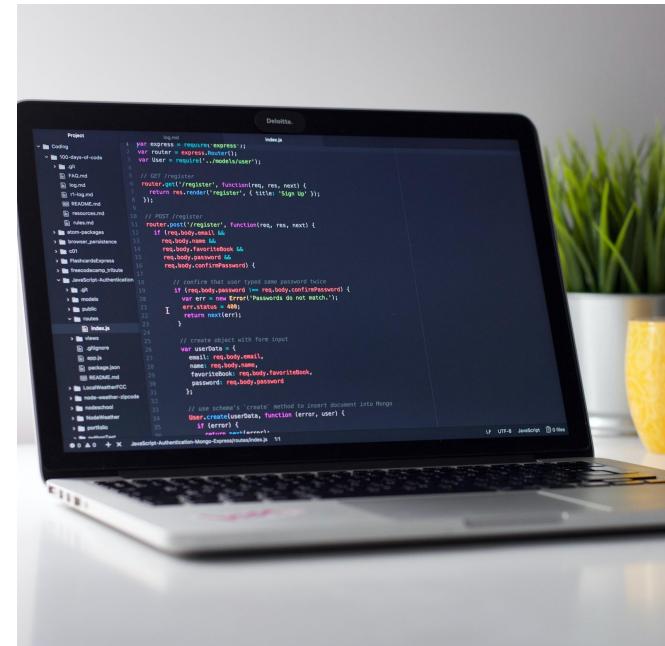
CSS Basics

Activity

# Activity/Homework

Make a new HTML document about yourself! Make sure to fulfill these criterias:

- Make sure your page has a <title>
  - Create a heading with the h1 tag
  - Change the color of the h1 tag
  - Change the background color
  - Embed min. 1 image
  - Link your favorite website to your page
  - Create an **unordered list** about things you like
  - Add anything you want to make the page interesting :)



# Helpful Resources:

<https://irahatam.github.io/baw-su-club-20/>

W3Schools

MDN Web Docs

HTML Color Codes

Please sign up for a Github account for  
tomorrow!

<https://github.com>

It's time for you to fill in the Happy Form

[bit.ly/happyggsummer](http://bit.ly/happyggsummer)

```
<!--CSS-->
<link type="text/css" rel="stylesheet" href="css/materialize.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="/css/animate.css">
<link rel="stylesheet" href="css/theme.css">
</head>
<body>
  <div class="banner">
    <div class="row">
      <div class="col s12 m12 l12" style="text-align: center;">
        <h1>Generation Girl</h1>
        <h2>Summer Club 2020</h2>
        <h3>Build a Website I</h3>
        <hr style="width: 100%; border: 0.5px solid black; margin-top: 10px;">
        <div style="display: flex; justify-content: space-around; width: 100%; margin-top: 10px;">
          <a href="#">Home</a>
          <a href="#">About</a>
          <a href="#">Contact</a>
        </div>
      </div>
    </div>
  </div>
</body>
```

Generation Girl Summer Club 2020

# Build a Website I

DAY TWO



# Today's Outlines

## Review

More on CSS

Version Control with Github

Activity

<https://app.sli.do/event/frnnn4uzc>



# Today's Outlines

Review

More on CSS

Version Control with Github

Activity

Let's learn something new today!

# A review on CSS

Selector

h1

Declaration

{ color:blue; font-size:12px; }

Declaration

Property



Value



Property



Value



# How do you apply CSS?

HTML

```
<p class="intro"> Title </p>
<div id="name"> Name </div>
```

CSS:

```
p {
  font-size: 40px;
  color: black;
}
.intro {
  font-weight: bold
}
#name {
  font-size: 12px;
  color: blue;
```

# Inline CSS

```
<p style="color: blue"> Hi Rookies! </p>
```

- Inline CSS is written as an attribute to an HTML element, so it doesn't need any selector.
- Only recommended if you want to override a global styling that you've set

# Internal CSS

- Internal CSS is written in the <head> part of an HTML document
- It's recommended to write internal CSS only if you don't have a lot of styling

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title> HTML CSS</title>
5      <style type="text/css">
6          body{
7              font-family: sans-serif;
8              background-color: black;
9          }
10         h1{
11             color: salmon;
12             text-align: center;
13         }
14
15         p{
16             color: white;
17         }
18     </style>
19 </head>
```

# External CSS

index.html

```
<head>
  <link
rel="stylesheet" type="text/css" href="style.css"
">
</head>
<body>
  <p>Hi, Rookies!</p>
</body>
```

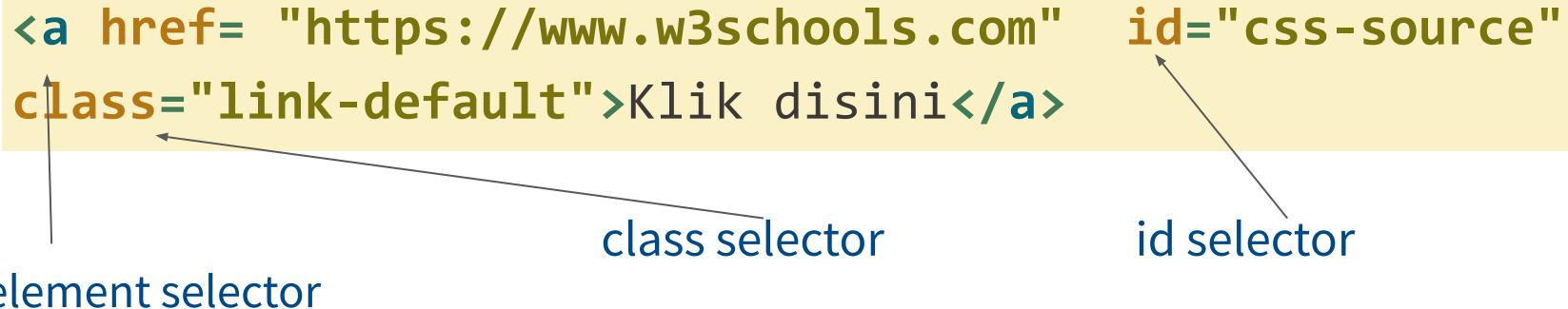
style.css

```
p {
  color: blue;
}
```

# CSS Selectors

- CSS selectors are used to select the content you want to style
- Selectors are part of the CSS rule set
- They select HTML elements according to its id, class, type, attribute, etc

```
<a href= "https://www.w3schools.com" id="css-source"  
class="link-default">Klik disini</a>
```



The diagram illustrates three types of CSS selectors applied to the provided HTML code:

- element selector**: Points to the opening tag <a>.
- class selector**: Points to the class attribute value "link-default".
- id selector**: Points to the id attribute value "css-source".

# Analogy



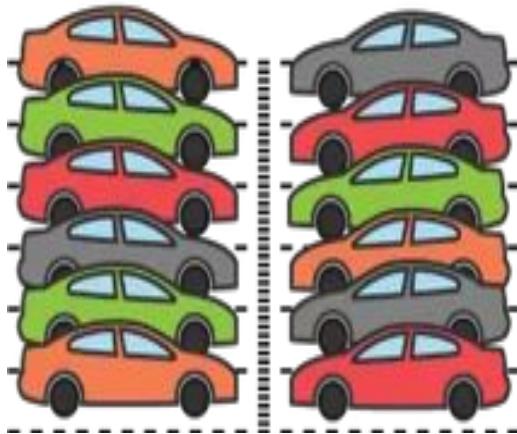
Imagine we want to buy a car...

# Analogy



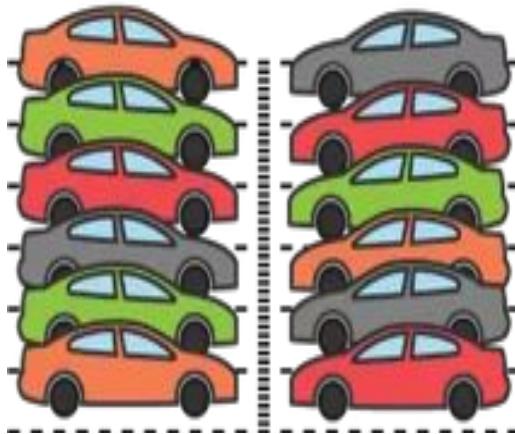
There are a lot of types of cars

# Analogy



Sedan

# Analogy

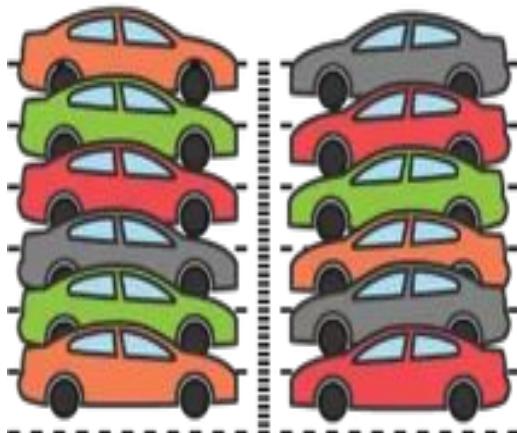


Sedan



Convertible

# Analogy



Sedan



Convertible



Truck

# Analogy

How can we use CSS to select a convertible?



Convertible

# Analogy

How can we use CSS to select a sedan?

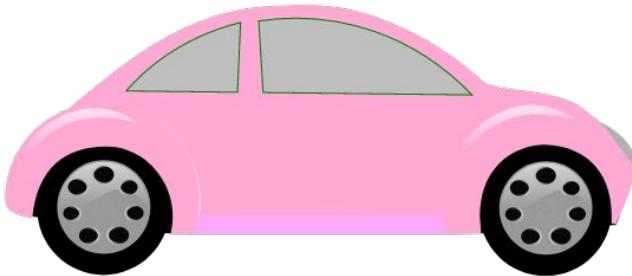


Convertible

```
convertible {  
    ...  
}
```

This is an example of an **element selector!**

# Analogy



What if I want to be more specific and choose a pink car?

```
.pink {  
    ...  
}
```

This is an example of a **class selector!**

# Analogy

How about **id** **selectors**?

# Analogy

How about **id** **selectors**?

What differentiates each car?

# Analogy

How about **id** **selectors**?

What differentiates each car?

Their license plate!



# Analogy

How can we select a car with a specific license plate?



```
#123xyz{  
    ...  
}
```

This is an example of an **id selector!**

# To recap

## Element Selectors

Element selectors will select all elements that have the same tags in the document.

```
a {
```

```
  font-size: 18px;
```

```
  text-align: justify;
```

```
}
```

## Class Selectors

Class selectors will select all elements that have a specific class attribute. A class selector can be used to identify more than one element, and an element can have more than one class.

```
.link-default {
```

```
  font-color: red;
```

```
}
```

## Id Selectors

An id selector is used to select an element with an id. One id can only be used to identify one specific element (cannot be reused).

```
#css-source {
```

```
  font-family: cursive;
```

```
}
```

Let's play a game!  
[flukeout.github.io](https://flukeout.github.io)

# CSS Attributes

## Font

Color: #RRGGBB, red, blue, green

Font-size: %, px, pt, em

Font-family: Arial, calibri

Font-style: Normal, italic

Font-weight: Normal, bold

## Background

Background-color: #RRGGBB, red, blue, green

Background-image: url("image url")

## Sizing-Element

Height : %, px, vh

Width : %, px, vw

## Border

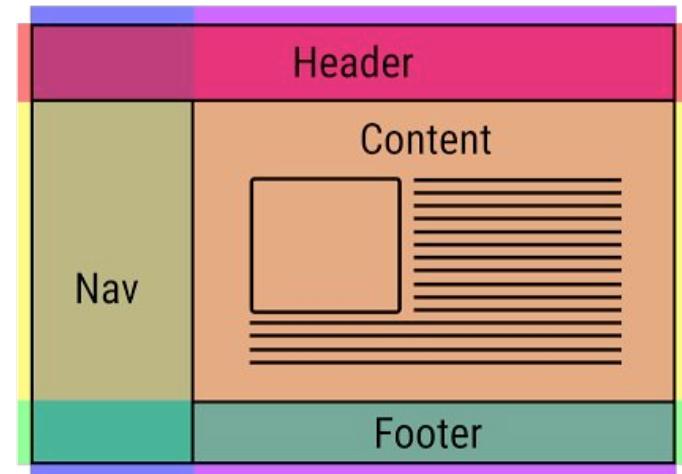
Border-color : red, #RRGGBB

Border-style : solid, dotted

Border-radius : 20px

# CSS Box

Everything that we see on a website is a **box/square!**

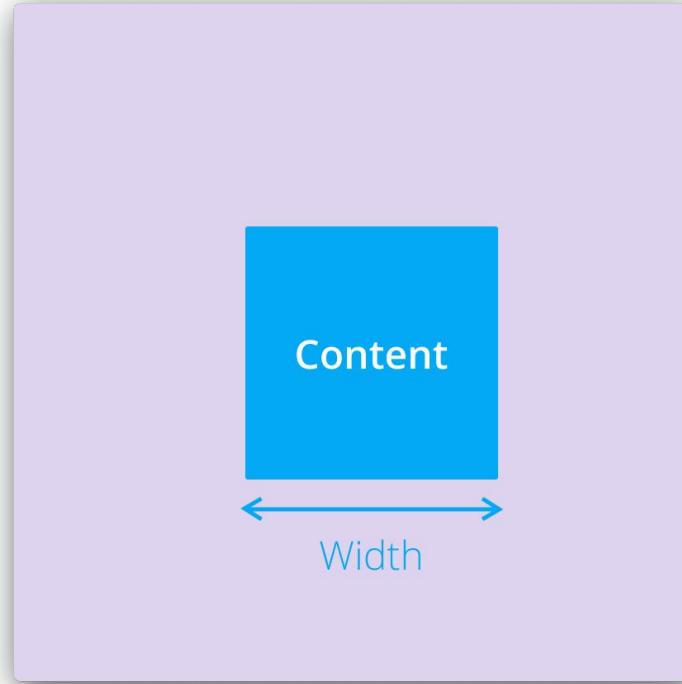


# CSS Box

## Width

```
#content {  
    width: 50px;  
}
```

[Let's try to understand more about width values!](#)

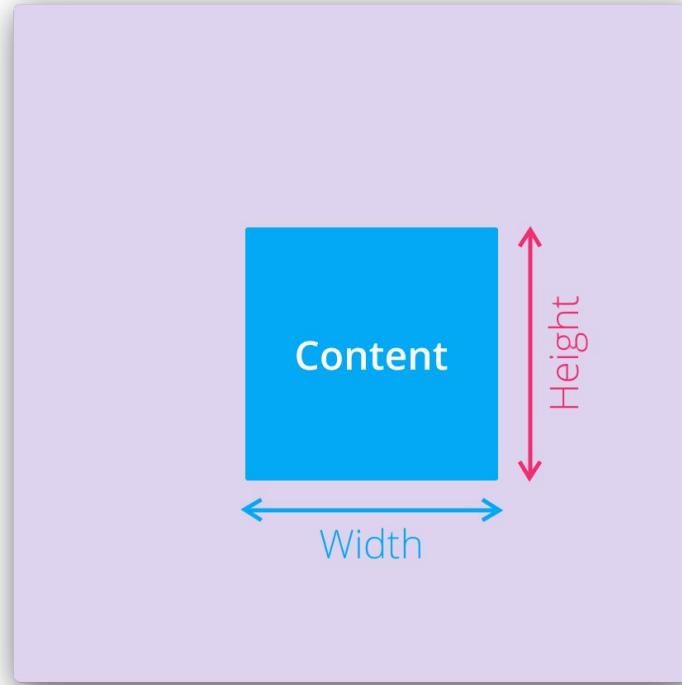


# CSS Box

## Height

```
#content {  
    width: 50px;  
    height: 50px;  
}
```

[Let's try to understand more about height values!](#)

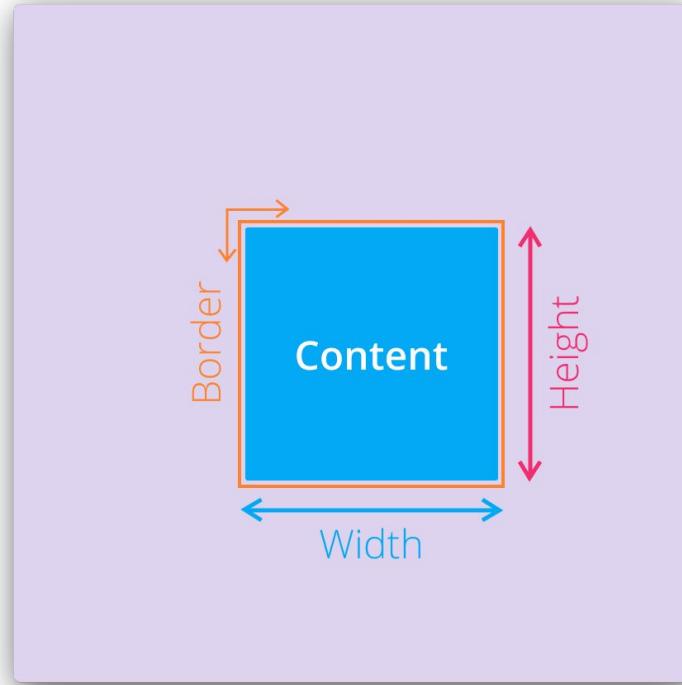


# CSS Box

## Border

```
#content {  
    width: 50px;  
    height: 50px;  
    border: 1px solid orange;  
}
```

[Let's try to understand more about borders!](#)

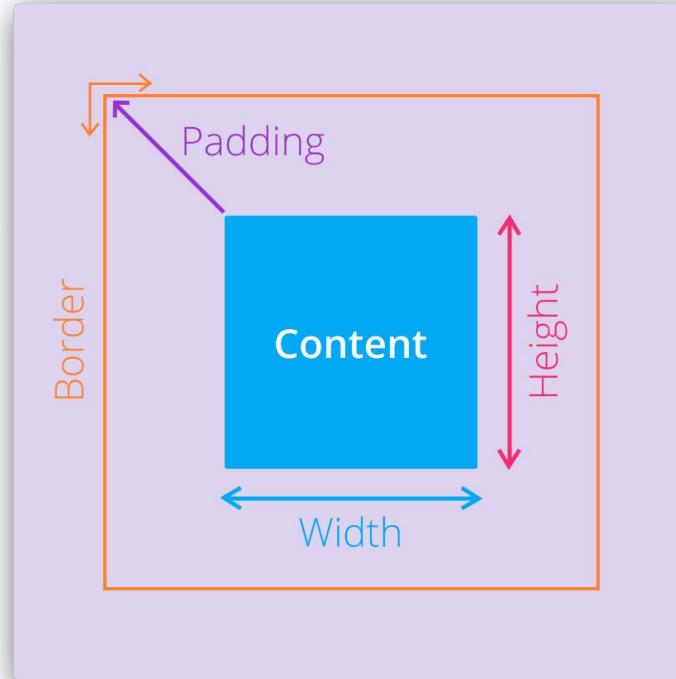


# CSS Box

## Padding

```
#content {  
    width: 50px;  
    height: 50px;  
    border: 1px solid orange;  
    padding: 35px;  
}
```

[Let's try to understand more about padding!](#)

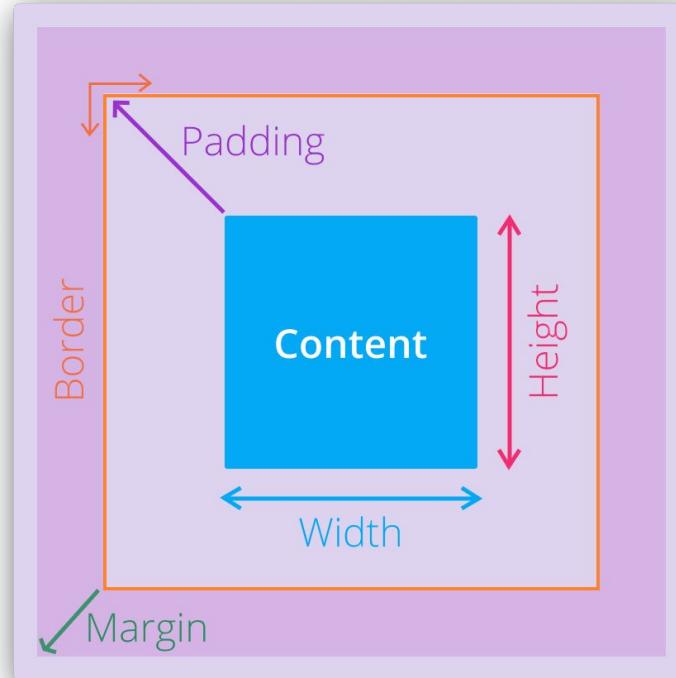


# CSS Box

## Margin

```
#content {  
    width: 50px;  
    height: 50px;  
    border: 1px solid orange;  
    padding: 35px;  
    Margin: 25px;  
}
```

[Let's try to understand more about padding!](#)



Let's do some exercises together!

[https://www.w3schools.com/css/exercise.asp  
?filename=exercise\\_boxmodell](https://www.w3schools.com/css/exercise.asp?filename=exercise_boxmodell)



# Today's Outlines

Review

More on CSS

Version Control with Github

Activity

# Version Control with Git

# What is Version Control?

- **Version control** is a system that stores our file updates from time to time
- It is usually used by a group of people that are working on the same project
- **Git** is one application that uses Version Control System (VCS)



# Examples of Git Services

Two commonly used Git services by developers are



&



# GitHub?

GitHub is one of the biggest online storehouse for collaboration work.

With GitHub, we can work with anyone, plan projects, track projects, and deploy our projects.



# GitHub

Let's set up our first GitHub repo!

# Today's Outlines



## Review

## More on CSS

## Intro to Bootstrap

### Activity

It's time for you to fill in the Happy Form

[bit.ly/happyggsummer](http://bit.ly/happyggsummer)



Generation Girl Summer Club 2020

# Build A Website

DAY 3



# Today's Outlines

## Review

Embedding Social Media/GMaps

Layouting with Flexbox

Activity

External is a method of applying CSS in a  
separate file  
true/false?

You can include an image in HTML as long  
as you have the name of the image.  
true/false?

You can not include GIFs in HTML.  
true/false?



# Today's Outlines

Review

**Embedding Social Media/GMaps**

Layouting with Flexbox

Activity

# Embedding Social Media/GMaps to Your Site (Live Code)



# Today's Outlines

Review

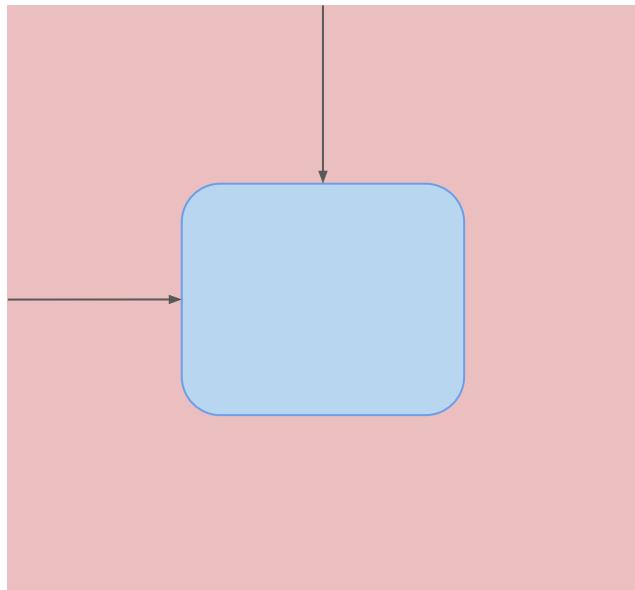
Embedding Social Media/GMaps

**Layouting with Flexbox**

Activity

# What if we want to put something in the middle of a container?

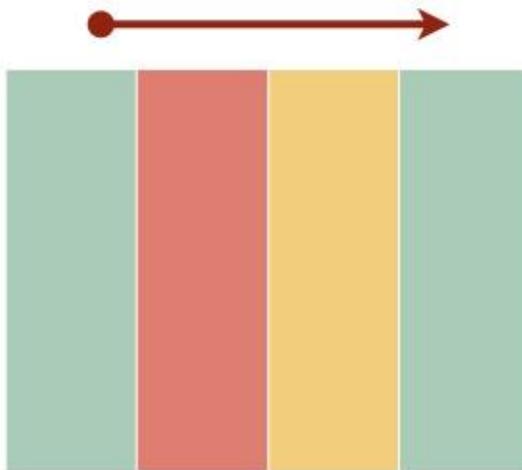
We can use either **padding** or  
**flexbox**!



# What's flexbox?

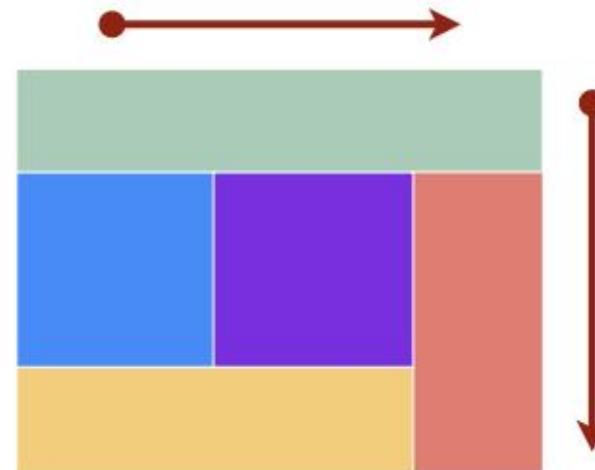
The **Flexible Box (Flexbox)** Module, was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an interface and powerful alignment capabilities.

# CSS Flexbox vs. Grid



**Flexbox**

One Dimensions



**CSS Grids**

Two Dimensions

# How do we use Flexbox?

Since flexbox is a **whole module** and **not a single property**, it involves a lot of things including its whole set of properties. Some of them are meant to be set on the container (**parent element**, known as “**flex container**”) whereas the others are meant to be set on the **children**, known as “**flex items**”.

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

# Let's Play!

**OPEN FLEXBOX FROGGY**  
**Finish the first 10 stages (or more if you can)**

# Today's Outlines



Review

Embedding Social Media/GMaps

Layouting with Flexbox

Activity

Let's layout our site with Flexbox  
and add more pages!

It's time for you to fill in the Happy Form

[bit.ly/happyggsummer](http://bit.ly/happyggsummer)



Generation Girl Summer Club 2020

# Build A Website

DAY 4



# Today's Outlines

**Responsive Website**

Intro to JavaScript

# What is a Responsive Website?

A **Responsive Website** is an approach to web design that makes web pages render well on a variety of devices and window or screen sizes.

Usually we would want to build our website “mobile first” for accessibility, but building our website and scaling it down to mobile size is also fine!



# What is a Responsive Website?

**Can you spot the differences?  
Which one do you think is better, and why?**



**Without Viewport**



**With Viewport**

# How Do You Make a Responsive Website?

## 1. Setting the Viewport

- A viewport is an area that users can see in our website.
- The size of a viewport varies based on the device
- We use the <meta> tag in our header to adjust the viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

# Relative CSS Units

**Relative CSS Units** is the most important concept in responsive web. Using relative units makes it easier for the code to adjust based on viewports, and we don't have to do things twice.

## Common CSS relative units:

- px
- %
- em
- rem
- vw
- vh



# Relative CSS Units

**px**

→ Best used for fixed values (not changing)

```
.rectangle {  
    width: 1000px;  
    height: 200px;  
}
```

**%**

→ Units that compares the element size with its parent.

→ Automatically adjusts to the viewport/device that the user is using.

```
.rectangle {  
    width: 50%;  
    height: 25%;  
}
```

# Relative CSS Units

## em

- A unit relative to an element's font size
- If there is no font size then **1em = 16px**

Styles Computed Event Listeners DOM Breakpoints Properties Accessibility Properties

```
.tenempadding {
  font-size: 18px;
  padding: ▶10em;
}
```

Styles Computed Event Listeners DOM Breakpoints Properties Accessibility Properties

```
display: block;
font-size: 18px;
height: 0px;
padding-bottom: 179.999984741211px;
padding-left: 179.999984741211px;
padding-right: 179.999984741211px;
padding-top: 179.999984741211px;
width: 1006.28472900391px;
```

Source: <https://webdesign.tutsplus.com/>

## rem

- Similar to em
- A unit relative to a **root element's** (html tag) font size
- If there is no font size then **1rem = 16px**

Styles Computed Event Listeners DOM Breakpoints Properties Accessibility Properties

```
html {
  font-size: 16px;
  padding: ▶10rem;
}
```

Styles Computed Event Listeners DOM Breakpoints Properties Accessibility Properties

```
display: block;
font-size: 16px;
height: 879.0625px;
padding-bottom: 160px;
padding-left: 160px;
padding-right: 160px;
padding-top: 160px;
width: 1382.22229003906px;
```

# Relative CSS Units

## vw

→ A unit relative to 1% of the viewport's width

→ E.g.:

Viewport width = 900px

Then, **1vw = 9px**

## vh

→ A unit relative to 1% of the viewport's height

→ E.g.:

Viewport height: 500px

Then, **1vh = 5px**

```
.rectangle {  
    width: 20vw;  
    height: 10vh;  
}
```

# Grid View

- Page is divided to several columns
- Makes it easier to insert elements on a website
- Usually divided into 12 columns with 100% width, which makes it responsive



# Grid View - Example



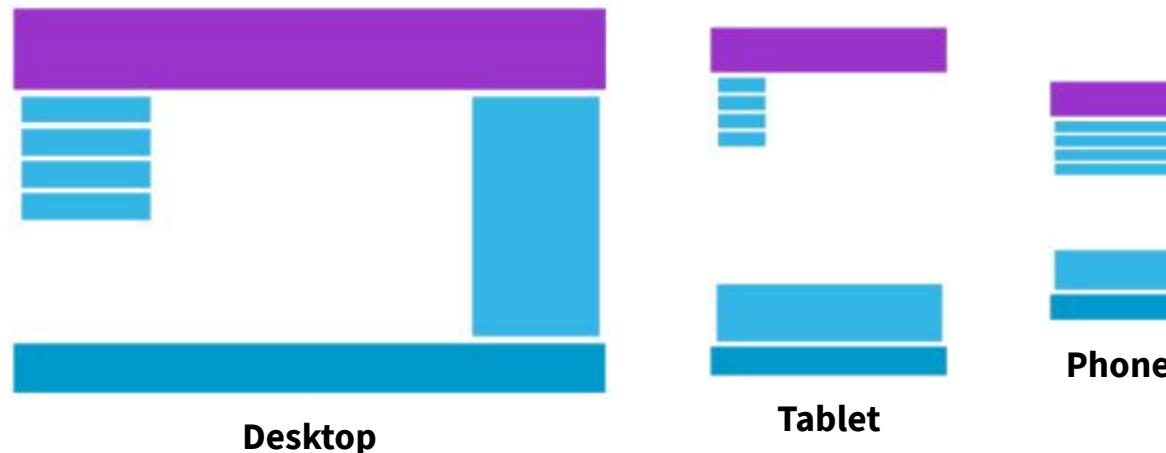
```
/* CSS files */
* {
    box-sizing: border-box;
}

.left {
    width: 25%;
    float: left;
    padding: 15px;
    border: 1px solid black;
}

.right {
    width: 75%;
    float: left;
    padding: 15px;
    border: 1px solid black;
}
```

# Media Queries

- Allows us to apply different styles on each display device
- Uses @media rule in CSS





# Media Queries - Example

→ E.g.:

Makes the font size different on screen with a maximum of 600 px, and more than 600px

---

## Hello Rookies!

Kalau lebar layarnya maksimal 600px, ukuran font pada ‘Hello Rookies’ menjadi 30px. Namun, bila lebar layarnya 600px ke atas, ukuran font tersebut menjadi 60px

**Screen ≤ 600px**

---

## Hello Rookies!

Kalau lebar layarnya maksimal 600px, ukuran font pada ‘Hello Rookies’ menjadi 30px. Namun, bila lebar layarnya 600px ke atas, ukuran font tersebut menjadi 60px

**Screen ≥ 600px**

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <style>
      @media screen and (min-width: 600px) {
        h1 {
          font-size: 60px;
        }
      }
      @media screen and (max-width: 600px) {
        h1 {
          font-size: 30px;
        }
      }
    </style>
  </head>
  <body>
    <h1>Hello Rookies!</h1>
    <p>Kalau lebar layarnya maksimal 600px, ukuran font pada ‘Hello Rookies’ menjadi 30px. Namun, bila lebar layarnya 600px ke atas, ukuran font tersebut menjadi 60px
    </body>
  </html>
```

# Media Queries

→ Can be set anyway you want it to

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {
    ...
}

/* Small devices (portrait tablets and large phones,
600px and up) */
@media only screen and (min-width: 600px) {
    ...
}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {
    ...
}
```

```
/* Large devices (laptops/desktops, 992px and up)
*/
@media only screen and (min-width: 992px) {
    ...
}

/* Extra Large devices (large laptops and
desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {
    ...
}
```



# Today's Outlines

Responsive Website

Intro to JavaScript

# What is JavaScript?

**Javascript** is a programming language that is useful for making websites more interactive with their users, such as for displaying effects and animations.



# What Can We Do with JS?

## Change the content of an HTML file

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo">JavaScript can change HTML content.</p>
    <button type="button"
      onclick="document.getElementById('demo').innerHTML
      = 'Hello JavaScript!'">Click Me!</button>
  </body>
</html>
```

JavaScript can change HTML content.

Click Me!

# What Can We Do with JS?

## Change the content of an HTML file

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo">JavaScript can change HTML content.</p>
    <button type="button"
      onclick="document.getElementById('demo').innerHTML
      = 'Hello JavaScript!'">Click Me!</button>
  </body>
</html>
```

Hello Javascript!

Click Me!

# What Can We Do with JS?

## Change the value of an HTML attribute

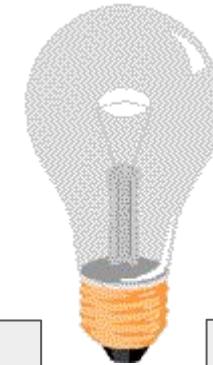
```
<!DOCTYPE html>
<html>
  <body>
    <p>In this case JavaScript changes the value<br>of the
      src (source) attribute of an image.</p>

    <button onclick="document.getElementById('myImage').src
      ='on.gif'">Turn On</button>

    <button onclick="document.getElementById('myImage').src
      ='off.gif'">Turn Off</button>
  </body>
</html>
```

In this case JavaScript changes the value of the src (source) attribute of an image.



Turn On

Turn Off

# What Can We Do with JS?

## Change the value of an HTML attribute

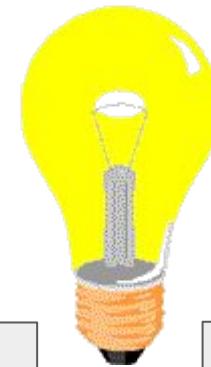
```
<!DOCTYPE html>
<html>
  <body>
    <p>In this case JavaScript changes the value<br>of the
      src (source) attribute of an image.</p>

    <button onclick="document.getElementById('myImage').src
      ='on.gif'">Turn On</button>

    <button onclick="document.getElementById('myImage').src
      ='off.gif'">Turn Off</button>
  </body>
</html>
```

In this case JavaScript changes the value of the src (source) attribute of an image.



Turn On

Turn Off

# What Can We Do with JS?

## Change HTML styling (CSS)

```
<!DOCTYPE html>
<html>
  <body id="bg" style="background-color:skyblue;">
    <h2>What Can JavaScript Do?</h2>
    <p>JavaScript can change the style of an
       HTML element.</p>
    <button type="button" onclick=
      "document.getElementById('bg').style.backgroundColor
      ='lightsalmon'">Click Me!</button>
  </body>
</html>
```

## What Can JavaScript Do?

JavaScript can change the style of an HTML element.

Click Me!

# What Can We Do with JS?

## Change HTML styling (CSS)

```
<!DOCTYPE html>
<html>
  <body id="bg" style="background-color:skyblue;">
    <h2>What Can JavaScript Do?</h2>
    <p>JavaScript can change the style of an
       HTML element.</p>
    <button type="button" onclick=
      "document.getElementById('bg').style.backgroundColor
      ='lightsalmon'">Click Me!</button>
  </body>
</html>
```

## What Can JavaScript Do?

JavaScript can change the style of an HTML element.

Click Me!

# What Can We Do with JS?

## Show/hide HTML elements

```
<!DOCTYPE html>
<html>
  <body>
    <h2>What Can JavaScript Do?</h2>
    <p id="demo">JavaScript can hide/show <br>
      HTML elements.</p>

    <button type="button" onclick=
      "document.getElementById('demo').style.display
      ='none'">Hide Text</button>

    <button type="button" onclick=
      "document.getElementById('demo').style.display
      ='block'">Show Text</button>
  </body>
</html>
```

## What Can JavaScript Do?

JavaScript can hide/show

HTML elements.

Hide Text

Show Text

# What Can We Do with JS?

## Show/hide HTML elements

```
<!DOCTYPE html>
<html>
  <body>
    <h2>What Can JavaScript Do?</h2>
    <p id="demo">JavaScript can hide/show <br>
      HTML elements.</p>

    <button type="button" onclick=
      "document.getElementById('demo').style.display
      ='none'">Hide Text</button>

    <button type="button" onclick=
      "document.getElementById('demo').style.display
      ='block'">Show Text</button>
  </body>
</html>
```

## What Can JavaScript Do?

Hide Text

Show Text

# Javascript Function

## What is a function?

Functions are sub-programs that can be reused both within the program itself and in other programs.

Because the function is part of a javascript code block, if we want to put it in our HTML file, we must write it in the `<script> ... </script>` tag which is placed inside the `<body> ... </body>` tag

We can also create a javascript function in a separate file with the `.js` format

```
// making a function
function sayHello(){
    console.log("Hello World!");
}

// calling the function
sayHello()
```

# Let's Create a Popup Box

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Confirm Box</h2>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
    var txt;
    if (confirm("Press a button!")) {
        txt = "You pressed OK!";
    } else {
        txt = "You pressed Cancel!";
    }
    document.getElementById('demo').innerHTML = txt;
}
</script>
</body>
</html>
```

## Javascript Confirm Box

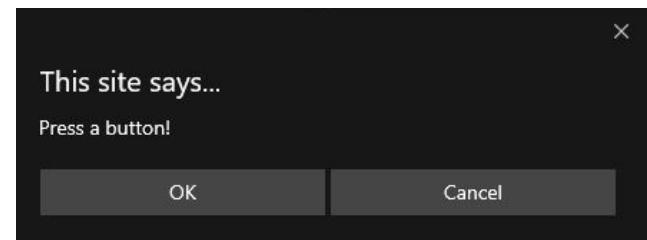
Try it

# Let's Create a Popup Box

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Confirm Box</h2>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
    var txt;
    if (confirm("Press a button!")) {
        txt = "You pressed OK!";
    } else {
        txt = "You pressed Cancel!";
    }
    document.getElementById('demo').innerHTML = txt;
}
</script>
</body>
</html>
```

## Javascript Confirm Box

Try it



# Let's Create a Popup Box

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Confirm Box</h2>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
    var txt;
    if (confirm("Press a button!")) {
        txt = "You pressed OK!";
    } else {
        txt = "You pressed Cancel!";
    }
    document.getElementById('demo').innerHTML = txt;
}
</script>
</body>
</html>
```

## Javascript Confirm Box

Try it

You pressed OK!

# Let's Create a Popup Box

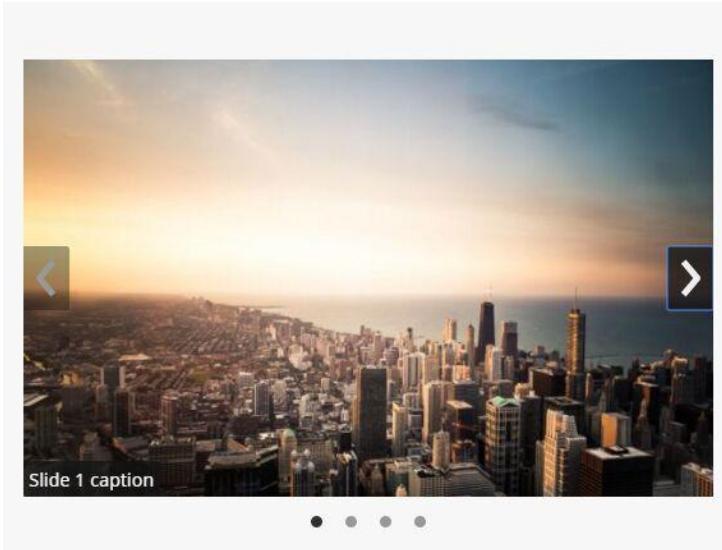
```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Confirm Box</h2>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
    var txt;
    if (confirm("Press a button!")) {
        txt = "You pressed OK!";
    } else {
        txt = "You pressed Cancel!";
    }
    document.getElementById('demo').innerHTML = txt;
}
</script>
</body>
</html>
```

## Javascript Confirm Box

Try it

You pressed Cancel!

# Some Other Cool Stuff to Make



Carousel (Slideshow)

1. Definition ^

---

2. Related Terms ▼

Given this definition, educational technology is an inclusive term for both the material tools and the theoretical foundations for supporting learning and teaching. Educational technology is not restricted to high technology.[8] Education technology is anything that enhances classroom learning in the utilization of blended, face to face, or online learning.[9]

---

3. Learning objects of Content ^

Accordion (Dropdowns)

Explore at:  
<https://www.w3schools.com/howto/default.asp>

## Challenge:

Change the following code so that the background color is changed using a confirmation box

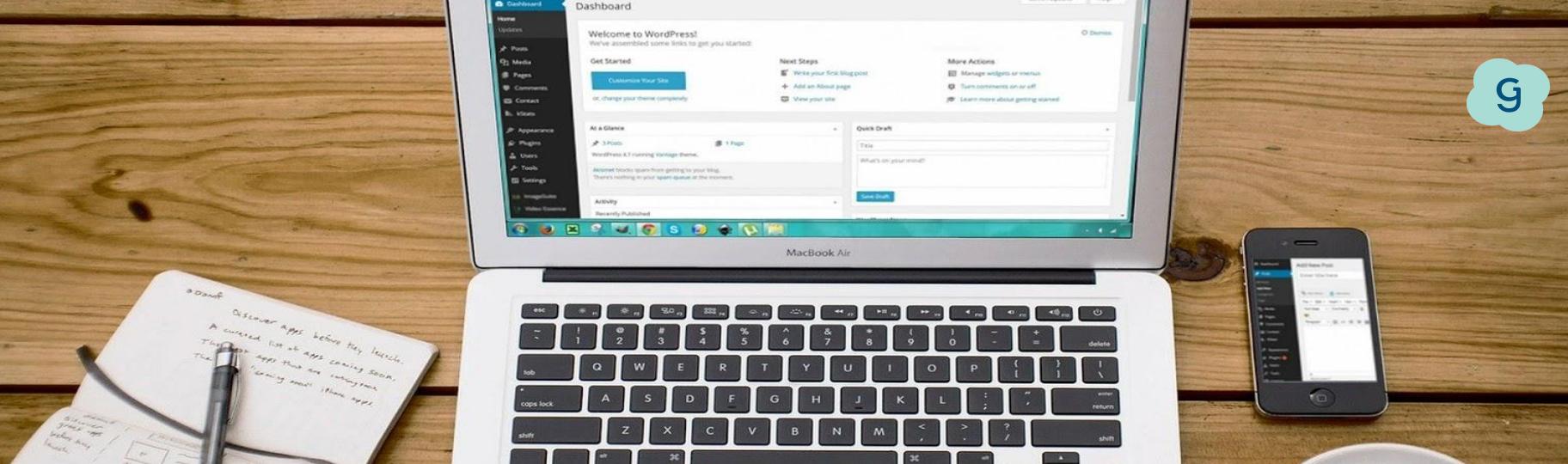
[tiny.cc/exercisejs](http://tiny.cc/exercisejs)

Let's make our website responsive and  
add JavaScript!

It's time for you to fill in the Happy Form

[bit.ly/happyggsummer](http://bit.ly/happyggsummer)

G



## Generation Girl Summer Club 2020

# Build A Website

DAY 5



# Today's Outlines

## Help with Final Project

Deployment Using Github Pages

Questions and Learning Resources

Anyone needs help with their site?



# Today's Outlines

Help with Final Project

**Deployment Using Github Pages**

Questions and Learning Resources

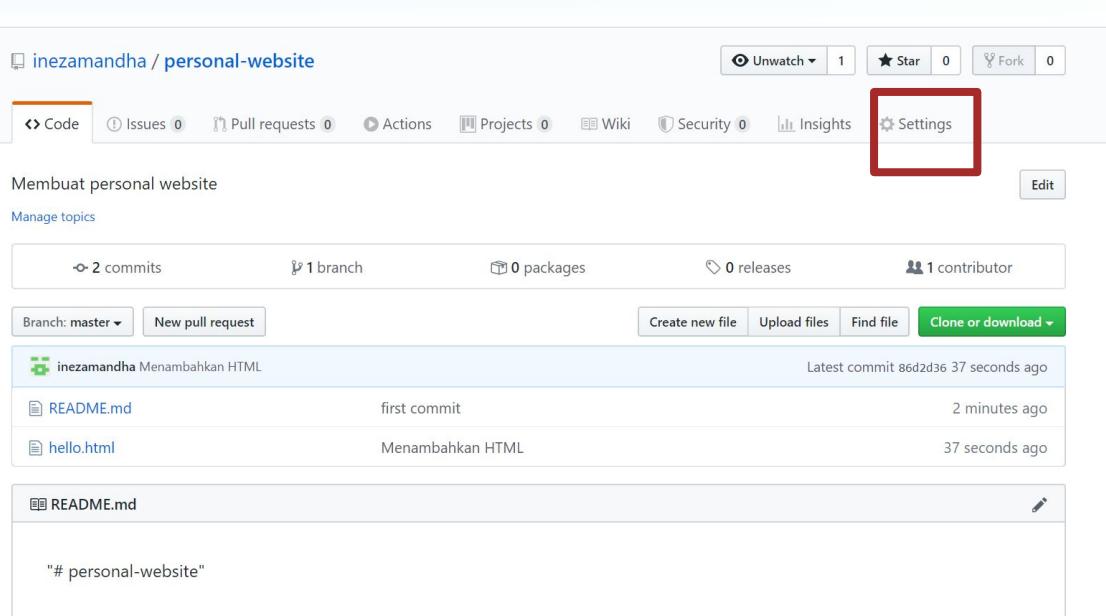
# Deployment using GitHub Pages!

The great thing about using GitHub is that you can instantly deploy your files using the domain `github.io` (GitHub Pages). But you have to make sure that your repo is **public**.



# Deployment using GitHub Pages!

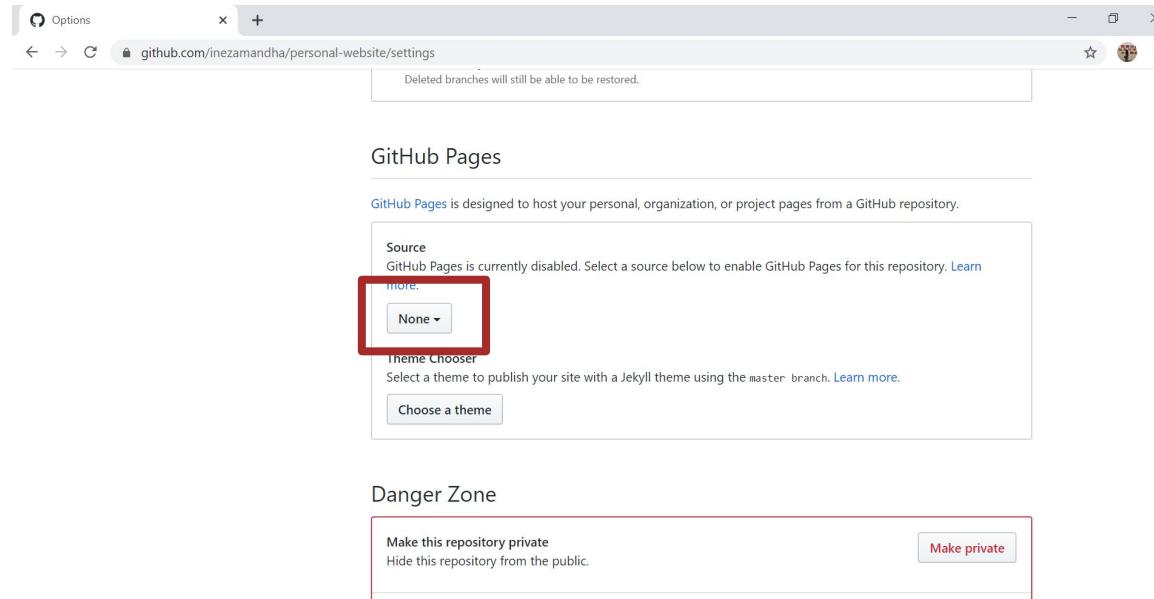
## 1. Click settings



The screenshot shows a GitHub repository page for 'inezamandha / personal-website'. At the top, there are buttons for 'Code', 'Issues 0', 'Pull requests 0', 'Actions', 'Projects 0', 'Wiki', 'Security 0', 'Insights', and 'Settings'. The 'Settings' button is highlighted with a red box. Below the header, there's a summary section with metrics: 2 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. A 'Clone or download' button is also visible. The main content area displays commit history for 'master' branch, showing two commits: 'README.md' (first commit, 2 minutes ago) and 'hello.html' (37 seconds ago). The 'hello.html' file was used to add HTML content. A preview of the 'README.md' file shows the text '# personal-website'.

# Deployment using GitHub Pages!

2. Scroll until you find the GitHub Pages section and change **None** to **master branch**

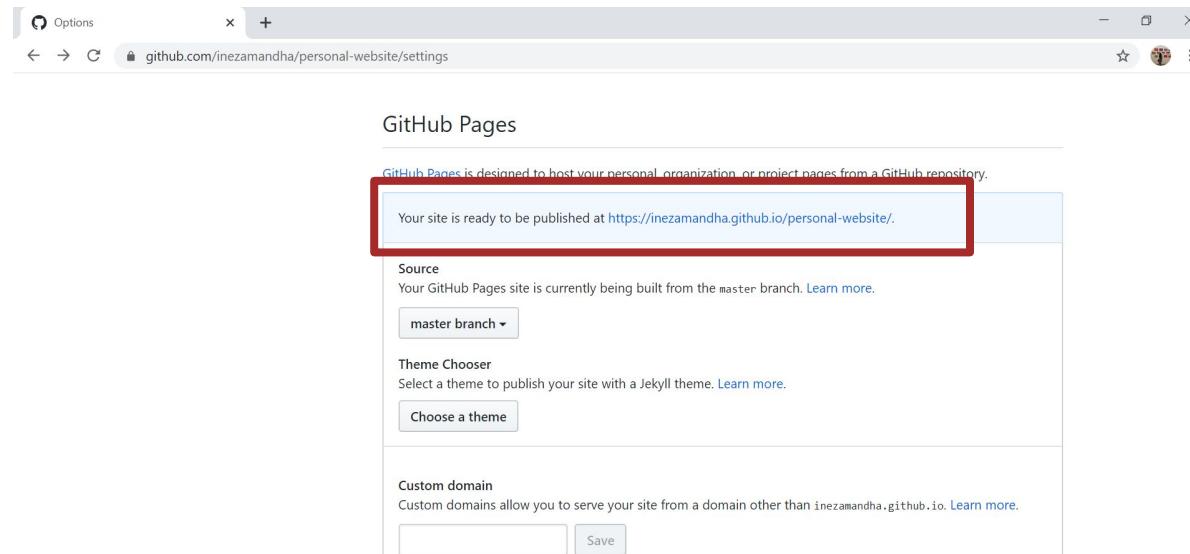


# Deployment using GitHub Pages!

3. Wait a few moments and your website is now accessible! :)

Sidenote:

For your website to be correctly deployed, the file name has to be **index.html**



# How to Access Your Site

If your html file inside your repo is named as **index.html**

`username.github.io/repo_name/`

**Else...**

`username.github.io/repo_name/file_name.html/`



# Today's Outlines

Help with Final Project

Deployment Using Github Pages

**Questions and Learning Resources**

Any questions about web  
dev/college/career/GenG/others?



# Recommended Learning Resources

## PODCASTS

Code Newbie  
Developer Tea  
Syntax  
Base CS  
freeCodeCamp

## WEBSITE & ONLINE COURSES

freeCodeCamp  
Codecademy  
W3Schools  
Udemy  
Linkedin Learning

## YOUTUBE CHANNELS

Kelas Terbuka  
Learn Code-Academy  
The New Boston  
Adam Khoury  
WebDev Mentors

It's time for you to fill in the Happy Form

[bit.ly/happyggsummer](http://bit.ly/happyggsummer)

GOOD LUCK  
for your presentations!

