Day 4

Project Heartcode

Hands-on Workshop:
Javascript & Github

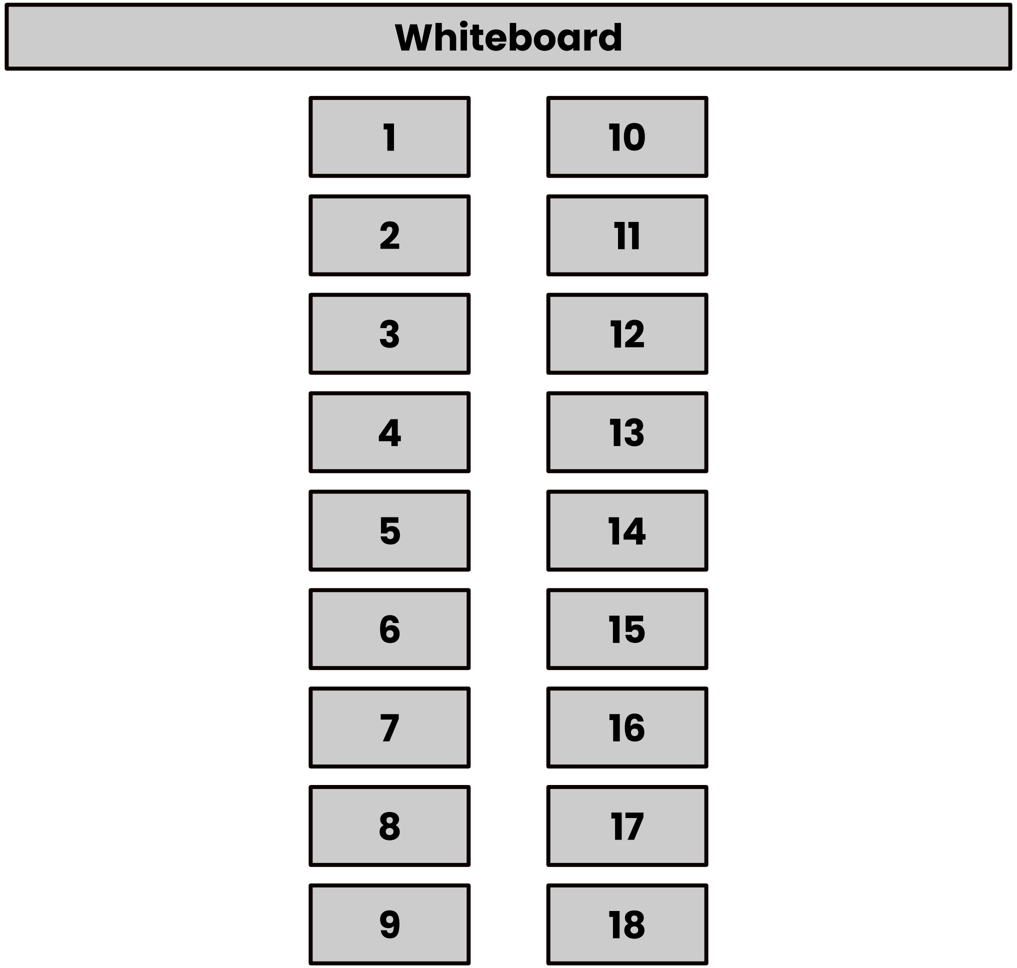

Whiteboard
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Table of contents

Workshop Materials

https://tinyurl.com/heartcodews

Wooclap

https://app.wooclap.com/heartcode22

01

RECAP

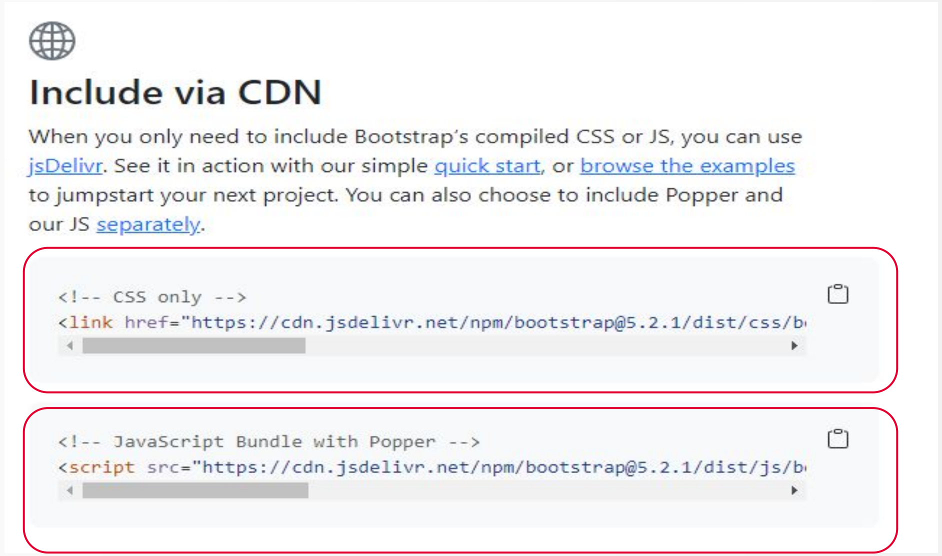
Bootstrap

- It is a front-end development framework that enables developers to quickly build fully responsive websites

https://getbootstrap.com/

Using Bootstrap – CDN

Using BootstrapCDN - Include the following in the **<head>** and **<body>** of the webpage


Include via CDN

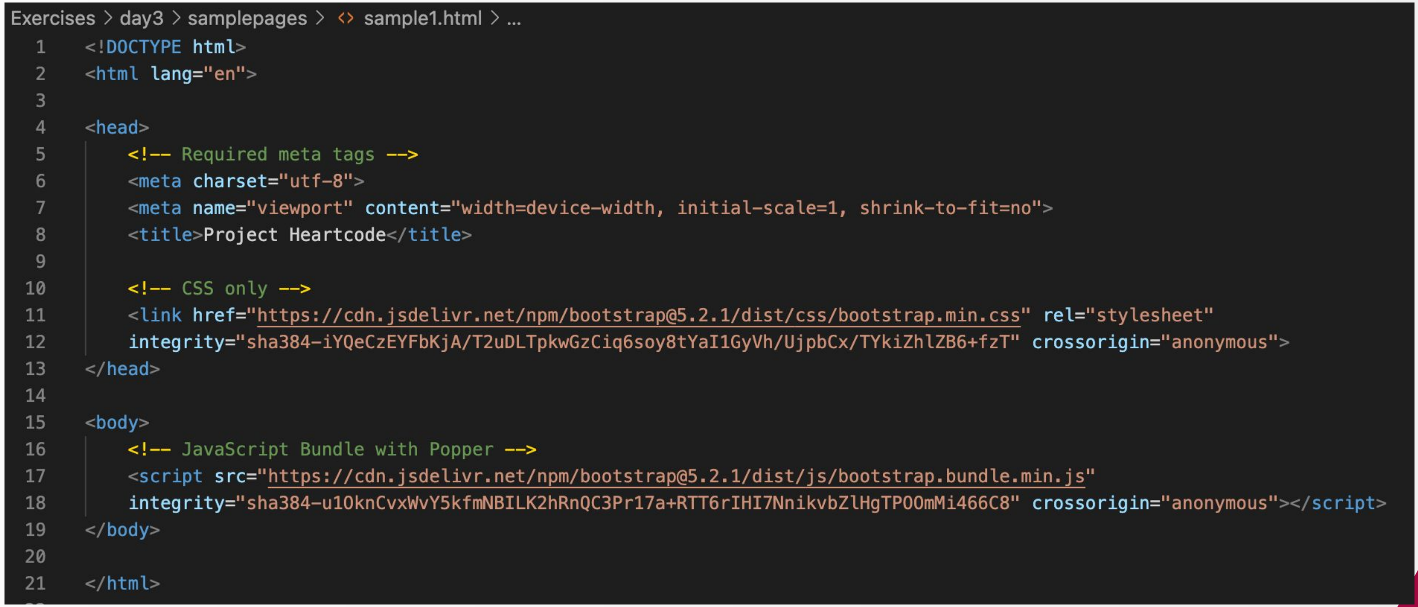
When you only need to include Bootstrap's compiled CSS or JS, you can use jsDelivr. See it in action with our simple quick start, or browse the examples to jumpstart your next project. You can also choose to include Popper and our JS separately.

```
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/b
```

```
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/b
```

NOTE: This will load the libraries through the CDN – not from your server

Bootstrap Template

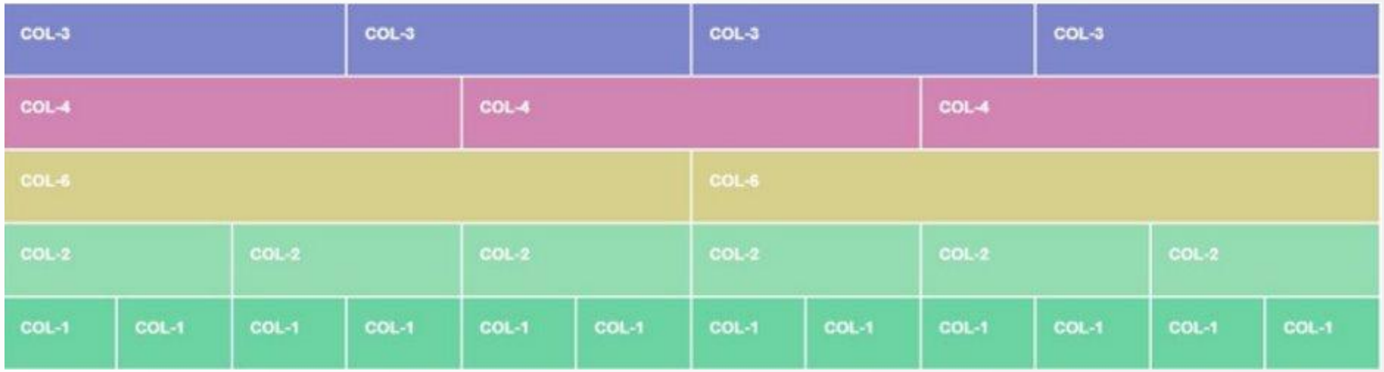


Exercises > day3 > samplepages > <> sample1.html > ...

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Project Heartcode</title>

    <!-- CSS only -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">
</head>

<body>
    <!-- JavaScript Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8" crossorigin="anonymous"></script>
</body>

</html>
```

Bootstrap Features: Grid System

- Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a **twelve column system**, six default responsive tiers, Sass variables and mixins, and dozens of predefined classes



Bootstrap Features: Grid System

Container: most basic layout element in bootstrap; required for using bootstrap's grid system

Responsive breakpoints: scale up / down elements as the viewpoints (browser width) changes

Six Responsive Breakpoints

- Extra small (.xs)
- Small (.sm)
- Medium (.md)
- Large (.lg)
- Extra large (.xl)
- Extra extra large (.xxl)

https://getbootstrap.com/docs/5.2/layout/breakpoints/

<div class="container">

- Responsive, fixed-width container, meaning its max-width changes at each breakpoint

<div class="container-fluid">

- Full width container, spanning the entire width of the viewport

Grid System – Breakpoints

Breakpoint	Class infix	Dimensions
X-Small	*None*	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Bootstrap Templates

- **Nav** - https://getbootstrap.com/docs/5.2/components/navs-tabs/
- **NavBars** - https://getbootstrap.com/docs/5.2/components/navbar/
- **Jumbotron** - https://getbootstrap.com/docs/4.5/components/jumbotron/
- **Carousel** - https://getbootstrap.com/docs/5.2/components/carousel/
- **Cards** - https://getbootstrap.com/docs/5.2/components/card/
- **Forms** - https://getbootstrap.com/docs/5.2/forms/overview/
- **Modal** - https://getbootstrap.com/docs/5.1/components/modal/

LottieFiles

LottieFiles consists of two elements for it to work:

1. **Script** for Lottiefiles to run
2. **Lottiefiles tag** to designate where to place the Lottiefile and how you want to run it

```
# Lottiefiles Script - do not change
<script src="https://unpkg.com/@lottiefiles/lottie-player@latest/dist/lottie-player.js"></script>

# Lottiefiles Tag - change src to your own animation link
<lottie-player src="<<LINK TO ANIMATION>>" background="transparent"  speed="1"  style="width: 300px; height: 300px;" hover loop controls autoplay></lottie-player>
```

02

JAVASCRIPT

Javascript

Javascript allows for:

- Operations
- Logic
- Loops
- Interactiveness

How to use Javascript

```
<html>
        <head>
                <title>Hello World</title>
        </head>
        <body>
                <script type="text/javascript">document.write("Hello World!")</script>
        </body>
</html>
```

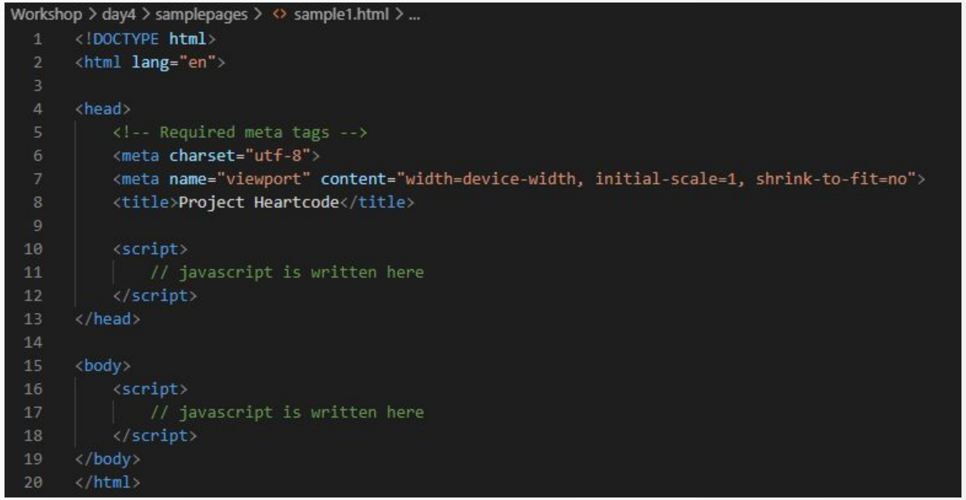
Note:

- 'type="text/javascript"' is optional

https://www.w3schools.com/js/default.asp

How to use Javascript

- Javascript code can be placed in both HTML <head> and <body> sections
- Ideally, placing javascript code in <head> section only if it needs to be executed before the page is rendered
- Best practice is to put it at the bottom of <body>


Workshop > day4 > samplepages > <> sample1.html > ...

```
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Project Heartcode</title>

    <script>
        // javascript is written here
    </script>
</head>

<body>
    <script>
        // javascript is written here
    </script>
</body>
</html>
```

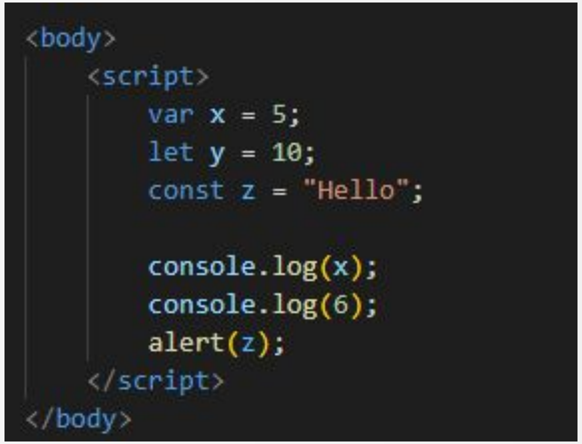
sample1.html

Variables

● **var**	[var x = 5]
● **let**	[let y = 10]
● **const**	[const z = “hello”]

Notice:

- console.log()
- alert()



```
<body>
    <script>
        var x = 5;
        let y = 10;
        const z = "Hello";

        console.log(x);
        console.log(6);
        alert(z);
    </script>
</body>
```

sample2.html

Variables – Naming Conventions

- **Underscore**

  E.g first_name, last_name, heart_code

- **Upper Camel Case (Pascal Case)**

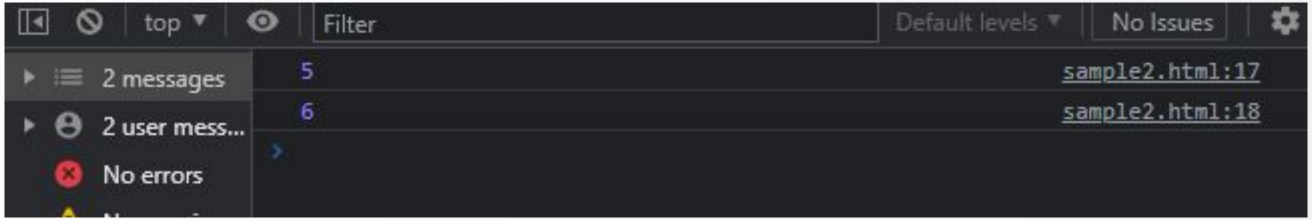
  FirstName, LastName, HeartCode

- **Lower Camel Case**

  firstName, lastName, heartCode

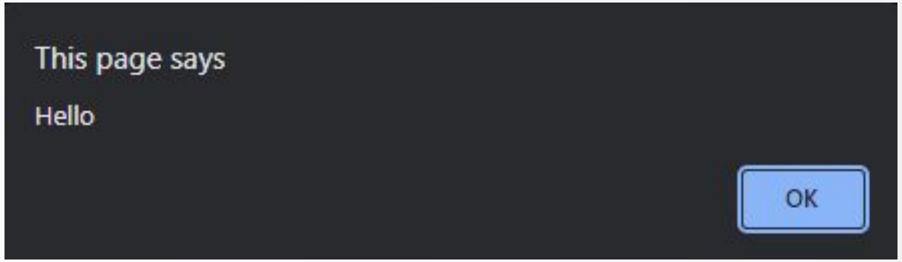
Variables

- console.log()



```
5                                   sample2.html:17
6                                   sample2.html:18
```

- alert()



This page says

Hello

Data Types

```
// Number

let length = 16;

// String

let lastName = "Johnson";

// Object

let x = {firstName:"John", lastName:"Doe"};
```

String Methods

1. Length
2. Slice / Substring / Substr
3. Replace
4. Uppercase and Lowercase
5. Concat
6. Trim

Exercise 1 – String Methods

Given **var a = “Project Heartcode is the best CSP in SMU!”,**

Determine the log output without running the codes in console:

a.length

a.slice(25, 29)

a.replace(‘SMU’, ‘The World’)

a.toUpperCase()

Number Methods

1. toString
2. toExponential
3. toFixed
4. Number
5. parseInt / parseFloat

Javascript Dates

```
var d = new Date();
```

7 numbers specify year, month, day, hour, minute, second, and millisecond (in that order)

```
var d = new Date(2022, 10, 6, 10, 30, 0 ,0)
```

Javascript Dates

6 5 4 3 2 Date Formats

6 - year, month, day, hour, minute, second

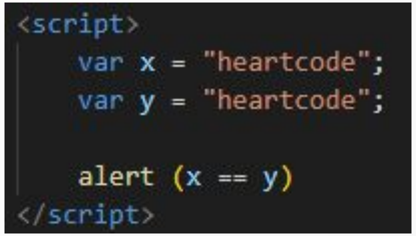
5 - year, month, day, hour, minute

4 - year, month, day, hour

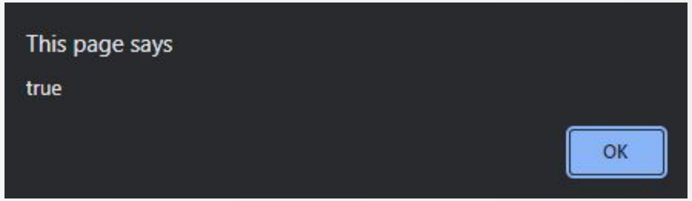
… and so on

Javascript – Logic

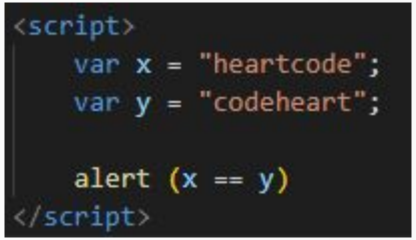
- == [Checks whether its two operands are equal]



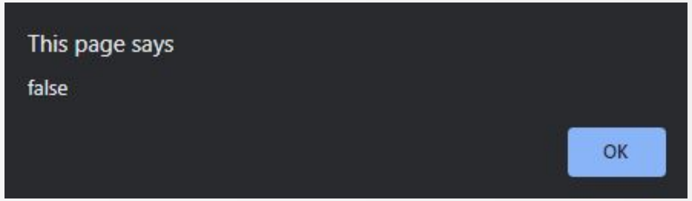
```
<script>
    var x = "heartcode";
    var y = "heartcode";

    alert (x == y)
</script>
```



```
This page says
true
```



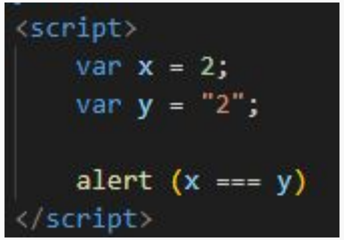
```
<script>
    var x = "heartcode";
    var y = "codeheart";

    alert (x == y)
</script>
```



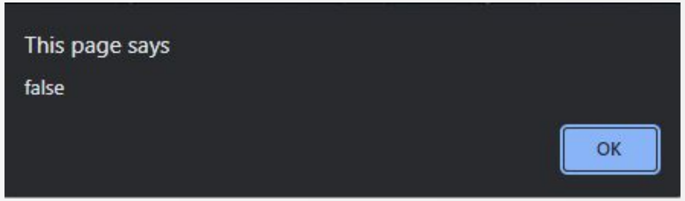
```
This page says
false
```

Javascript – Logic

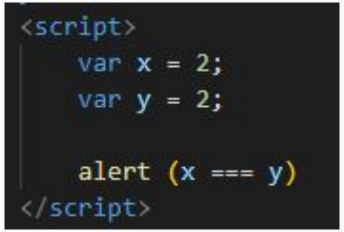
- **===** [Checks if the type is the same and then if the value is the same]



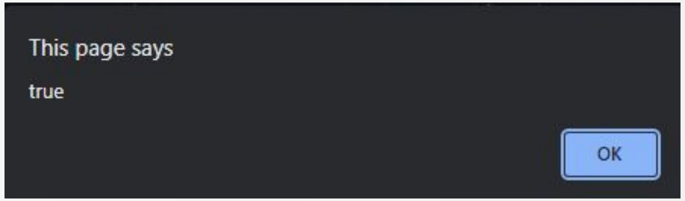
```
<script>
    var x = 2;
    var y = "2";

    alert (x === y)
</script>
```



```
This page says
false
```

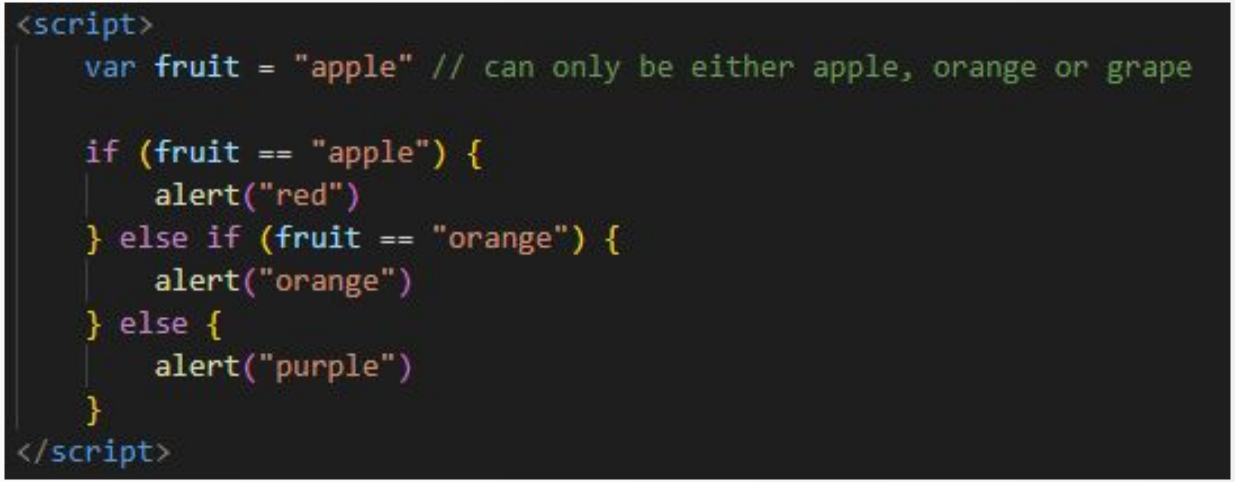


```
<script>
    var x = 2;
    var y = 2;

    alert (x === y)
</script>
```



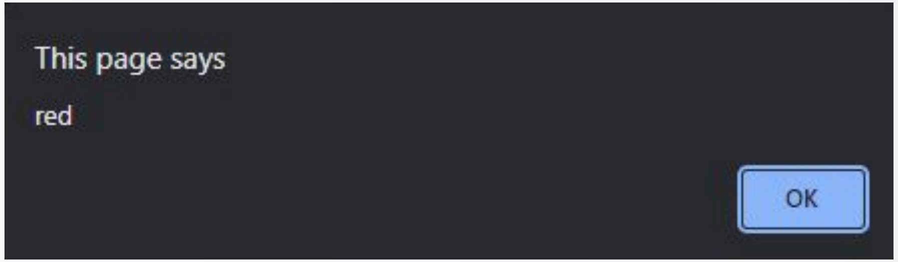
```
This page says
true
```

Javascript – Logic



```
<script>
    var fruit = "apple" // can only be either apple, orange or grape

    if (fruit == "apple") {
        alert("red")
    } else if (fruit == "orange") {
        alert("orange")
    } else {
        alert("purple")
    }
</script>
```

sample3.html



```
This page says

red
```

Prompt

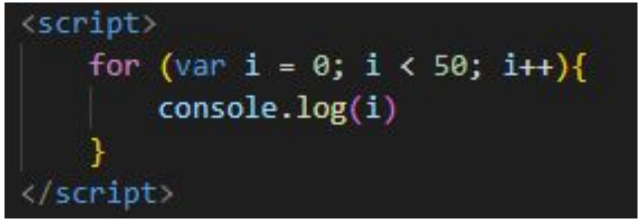
The **prompt(question, predefined answer)** method displays a dialog box that prompts the user for input.

var name = prompt('What is your age?', 24)

Exercise 2 – Date and Prompt

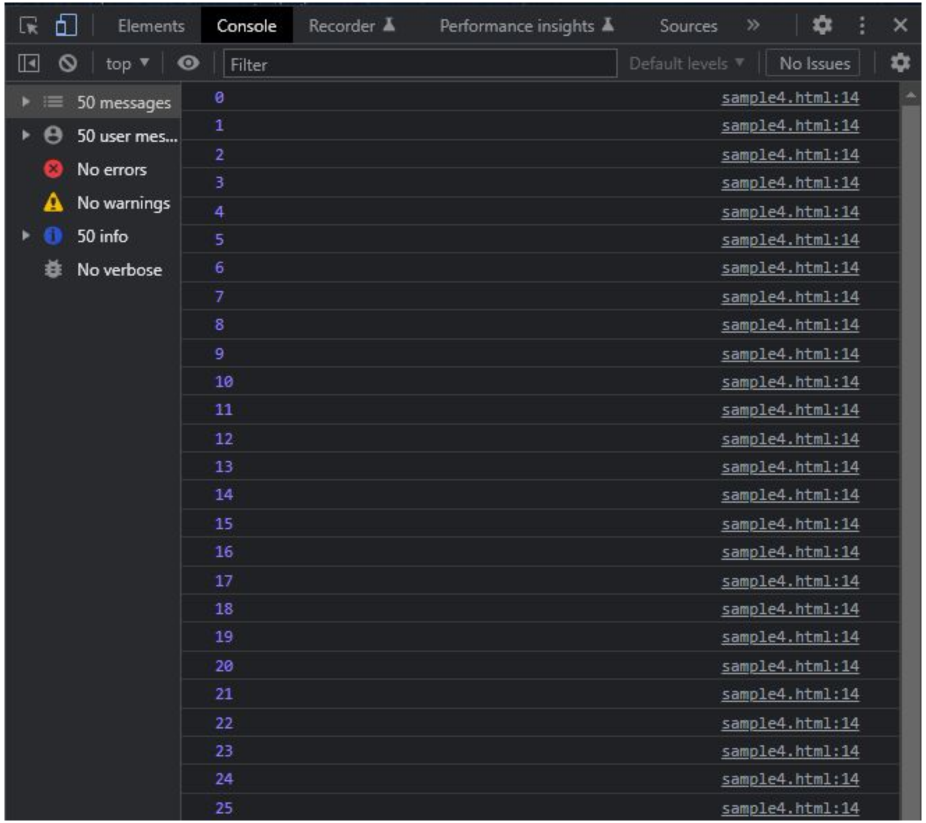
Prompt a user for their **name, age, birthday** in Javascript’s date format, and display them in the webpage

Javascript - Loops



```
<script>
    for (var i = 0; i < 50; i++){
        console.log(i)
    }
</script>
```

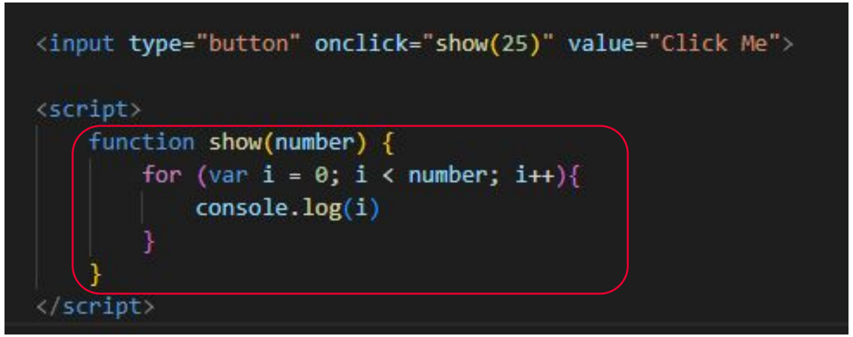
sample4.html



```
0     sample4.html:14
1     sample4.html:14
2     sample4.html:14
3     sample4.html:14
4     sample4.html:14
5     sample4.html:14
6     sample4.html:14
7     sample4.html:14
8     sample4.html:14
9     sample4.html:14
10    sample4.html:14
11    sample4.html:14
12    sample4.html:14
13    sample4.html:14
14    sample4.html:14
15    sample4.html:14
16    sample4.html:14
17    sample4.html:14
18    sample4.html:14
19    sample4.html:14
20    sample4.html:14
21    sample4.html:14
22    sample4.html:14
23    sample4.html:14
24    sample4.html:14
25    sample4.html:14
```

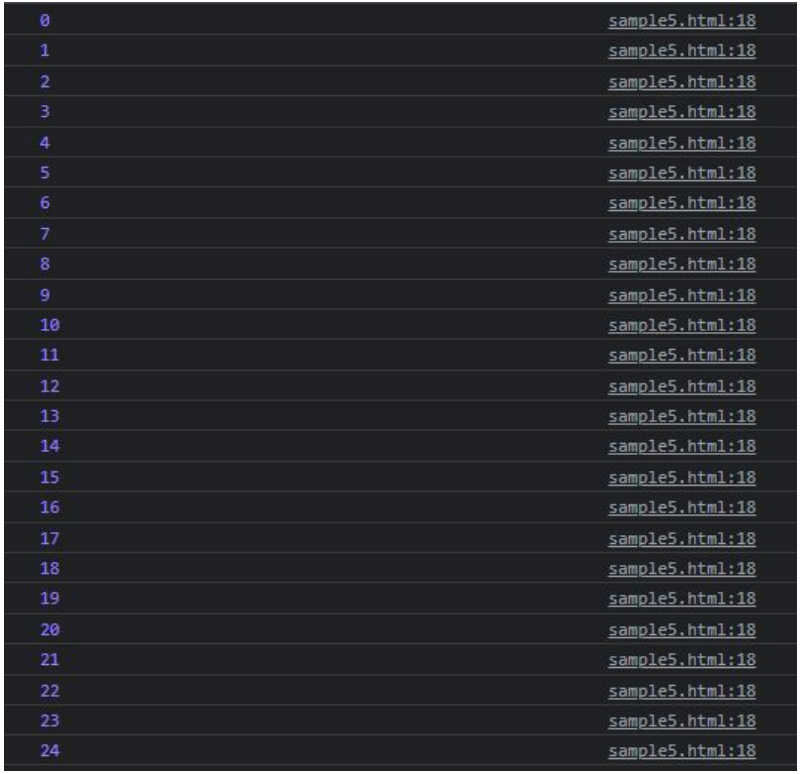
Javascript – Functions

Function allows you to define a block of code, giving it a name and then executing it as many times as you want



```
<input type="button" onclick="show(25)" value="Click Me">

<script>
    function show(number) {
        for (var i = 0; i < number; i++){
            console.log(i)
        }
    }
</script>
```

sample5.html



```
0                                                  sample5.html:18
1                                                  sample5.html:18
2                                                  sample5.html:18
3                                                  sample5.html:18
4                                                  sample5.html:18
5                                                  sample5.html:18
6                                                  sample5.html:18
7                                                  sample5.html:18
8                                                  sample5.html:18
9                                                  sample5.html:18
10                                                 sample5.html:18
11                                                 sample5.html:18
12                                                 sample5.html:18
13                                                 sample5.html:18
14                                                 sample5.html:18
15                                                 sample5.html:18
16                                                 sample5.html:18
17                                                 sample5.html:18
18                                                 sample5.html:18
19                                                 sample5.html:18
20                                                 sample5.html:18
21                                                 sample5.html:18
22                                                 sample5.html:18
23                                                 sample5.html:18
24                                                 sample5.html:18
```

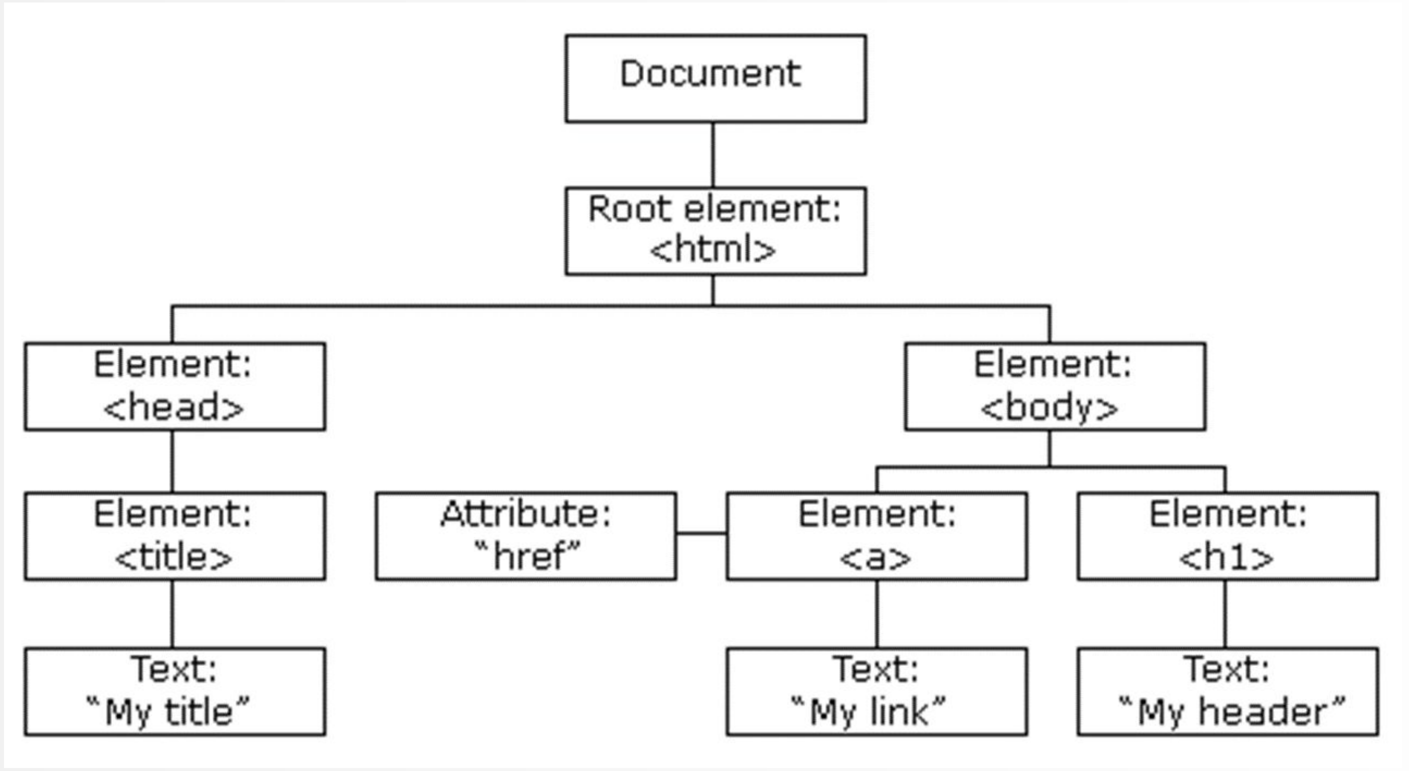
Exercise 3 – Odd Numbers

Code a function that will print the **ODD NUMBERS from 0 to 50** in the console and a **button** that will run the function

DOM – Document Object Model

- Browser represents an HTML document as DOM
- Represents the page so that program can change the document structure, style and content
- HTML entities are represented as nodes (object) in a hierarchical data structure
- All these objects are accessible using Javascript

DOM – Document Object Model



Event Handling

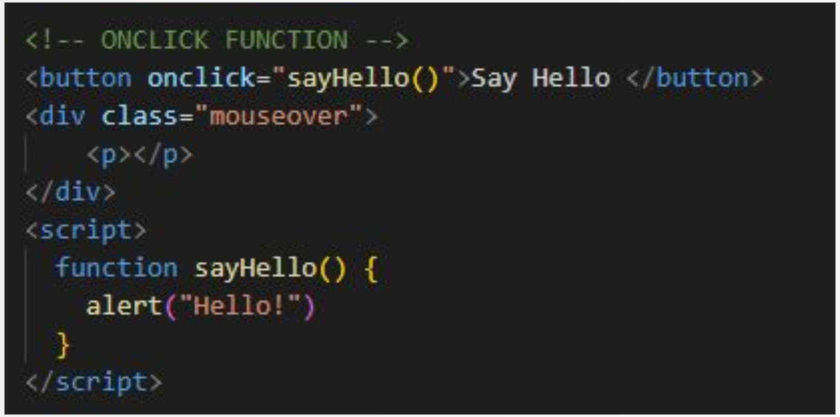
- Javascript can be used to listen to events and react based on the user’s action
- Examples of events
    - Clicking a mouse
    - Hovering over an image
    - Typing something in a search bar

Events

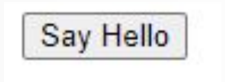
Event	Description
onclick	The user clicks an HTML element
onchange	An HTML element has been changed
oninput	The user inputs something in an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Events – onclick

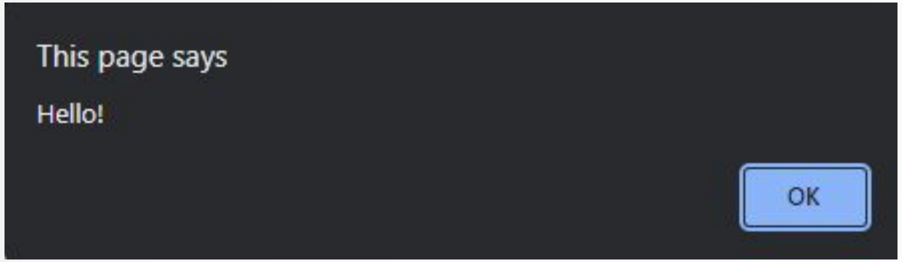
- onclick() - The user clicks on an HTML element



```
<!-- ONCLICK FUNCTION -->
<button onclick="sayHello()">Say Hello </button>
<div class="mouseover">
    <p></p>
</div>
<script>
  function sayHello() {
    alert("Hello!")
  }
</script>
```



events.html

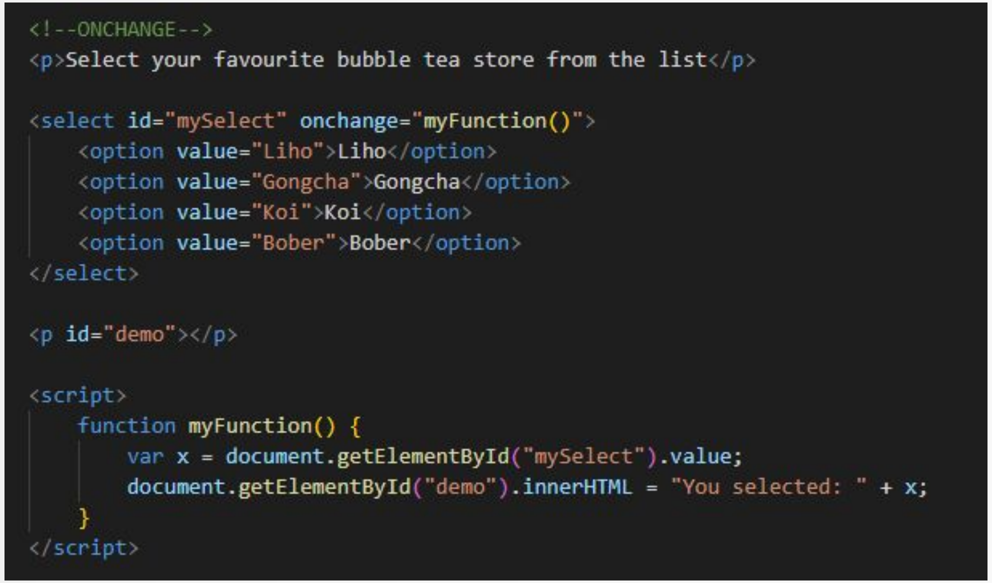


This page says

Hello!

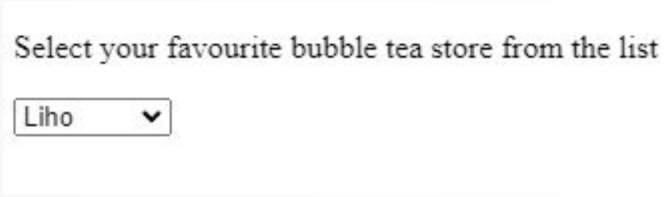
Events – onchange

- onchange - An HTML element has been changed

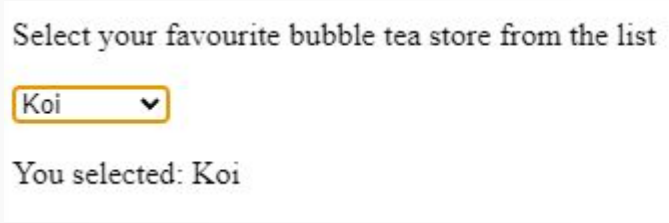


```
<!--ONCHANGE-->
<p>Select your favourite bubble tea store from the list</p>

<select id="mySelect" onchange="myFunction()">
    <option value="Liho">Liho</option>
    <option value="Gongcha">Gongcha</option>
    <option value="Koi">Koi</option>
    <option value="Bober">Bober</option>
</select>

<p id="demo"></p>

<script>
    function myFunction() {
        var x = document.getElementById("mySelect").value;
        document.getElementById("demo").innerHTML = "You selected: " + x;
    }
</script>
```

events.html



Select your favourite bubble tea store from the list

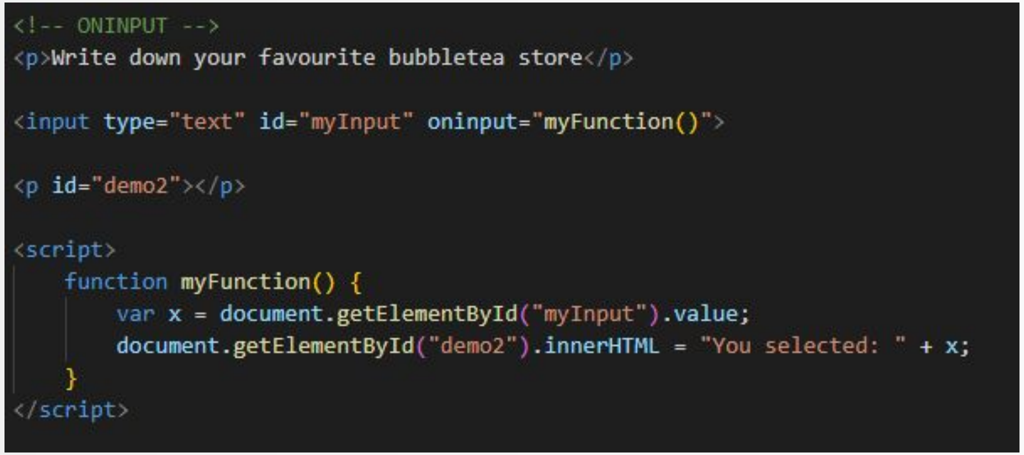


Select your favourite bubble tea store from the list

You selected: Koi

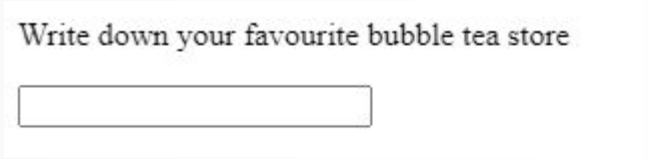
Events – oninput

- oninput - The user inputs something into an HTML element

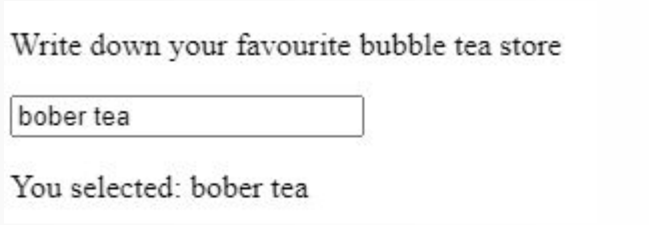


```
<!-- ONINPUT -->
<p>Write down your favourite bubbletea store</p>

<input type="text" id="myInput" oninput="myFunction()">

<p id="demo2"></p>

<script>
    function myFunction() {
        var x = document.getElementById("myInput").value;
        document.getElementById("demo2").innerHTML = "You selected: " + x;
    }
</script>
```

events.html



Write down your favourite bubble tea store



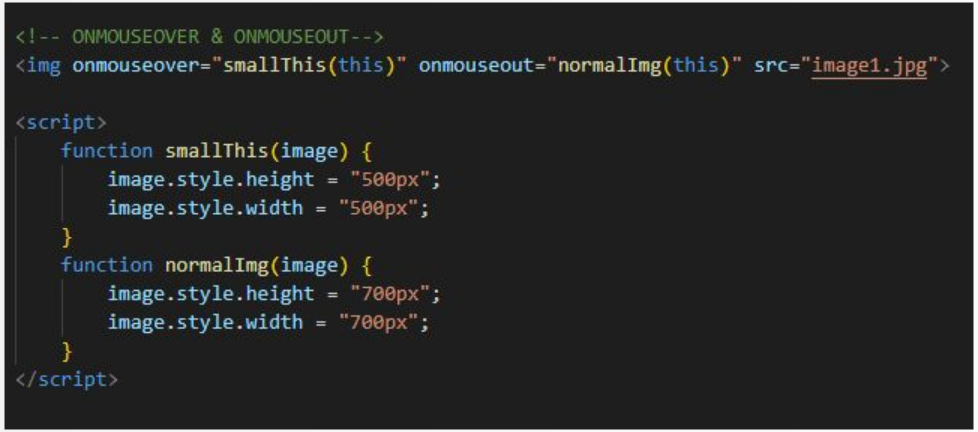
Write down your favourite bubble tea store

bober tea

You selected: bober tea

Events – onmouseover / onmouseout

- onmouseover - The user moves the mouse over an HTML element
- onmouseout - The user moves the mouse away from an HTML element

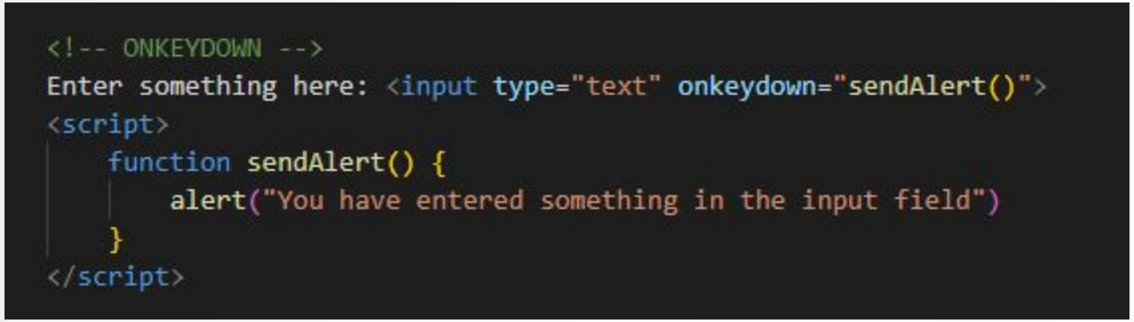


```html
<!-- ONMOUSEOVER & ONMOUSEOUT-->
<img onmouseover="smallThis(this)" onmouseout="normalImg(this)" src="image1.jpg">

<script>
    function smallThis(image) {
        image.style.height = "500px";
        image.style.width = "500px";
    }
    function normalImg(image) {
        image.style.height = "700px";
        image.style.width = "700px";
    }
</script>
```

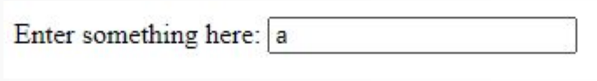
events.html

Events – onkeydown

- onkeydown - The user presses a key on the keyboard

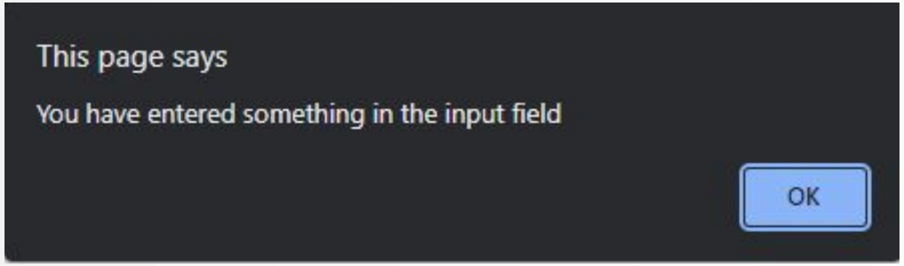


```
<!-- ONKEYDOWN -->
Enter something here: <input type="text" onkeydown="sendAlert()">
<script>
    function sendAlert() {
        alert("You have entered something in the input field")
    }
</script>
```



Enter something here: a

events.html

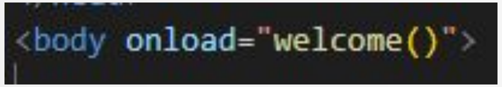


This page says

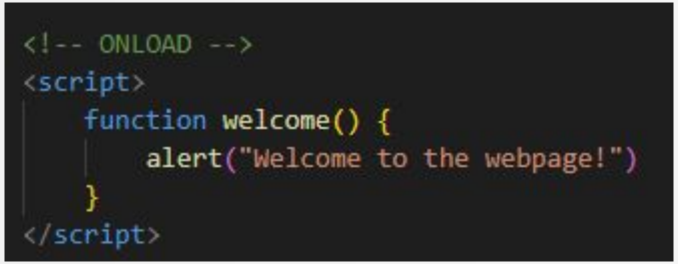
You have entered something in the input field

Events – onload

- onload - The browser has finished loading the page

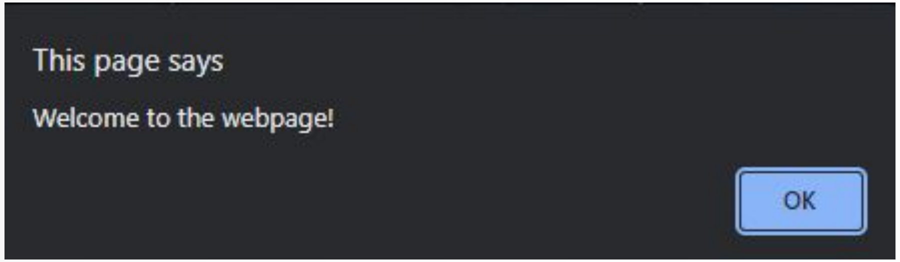


```
<body onload="welcome()">
```



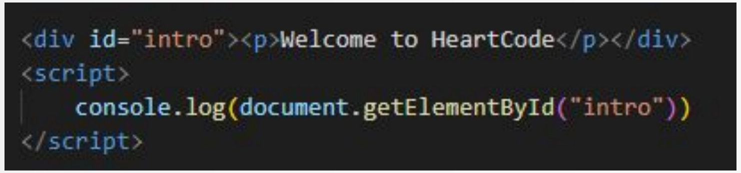
```
<!-- ONLOAD -->
<script>
    function welcome() {
        alert("Welcome to the webpage!")
    }
</script>
```

events.html

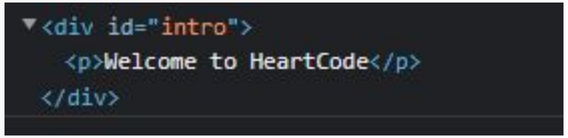


```
This page says
Welcome to the webpage!
```

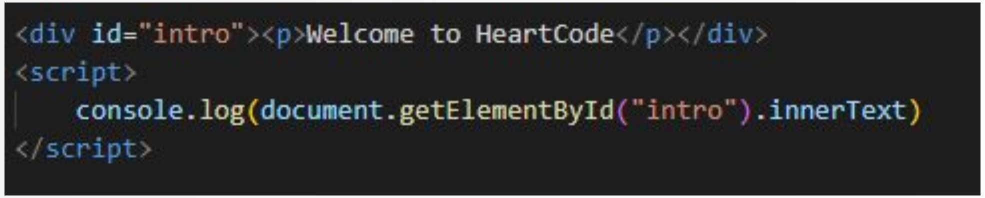
DOM – getElementByID



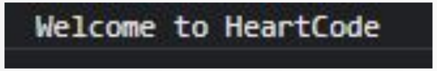
```
<div id="intro"><p>Welcome to HeartCode</p></div>
<script>
    console.log(document.getElementById("intro"))
</script>
```



```
▼<div id="intro">
   <p>Welcome to HeartCode</p>
 </div>
```



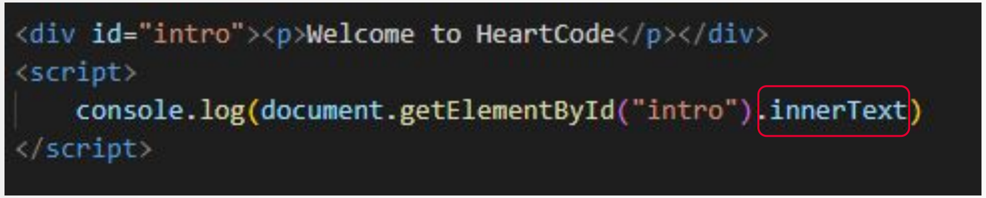
```
<div id="intro"><p>Welcome to HeartCode</p></div>
<script>
    console.log(document.getElementById("intro").innerText)
</script>
```



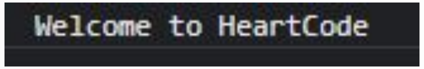
```
Welcome to HeartCode
```

InnerText vs InnerHTML

InnerText - Retrieve and sets the content of the tag as plain text

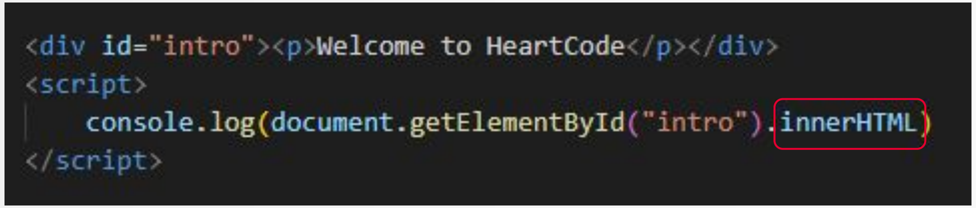


```
<div id="intro"><p>Welcome to HeartCode</p></div>
<script>
    console.log(document.getElementById("intro").innerText)
</script>
```

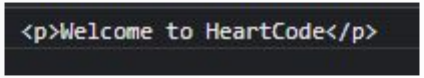


```
Welcome to HeartCode
```

InnerHTML - Retrieve all the content within the element



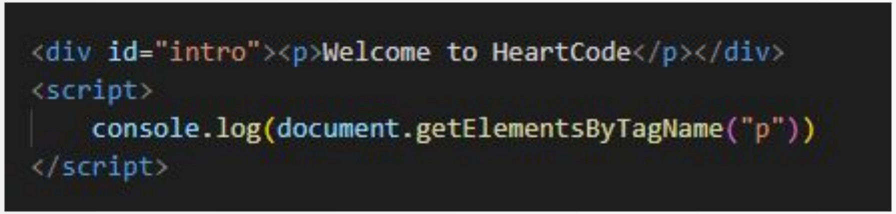
```
<div id="intro"><p>Welcome to HeartCode</p></div>
<script>
    console.log(document.getElementById("intro").innerHTML)
</script>
```



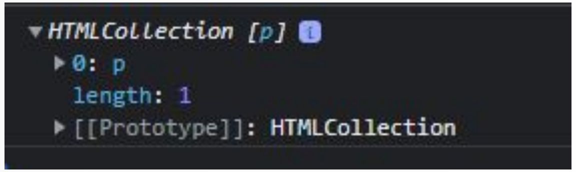
```
<p>Welcome to HeartCode</p>
```

DOM – getElementsByTagName

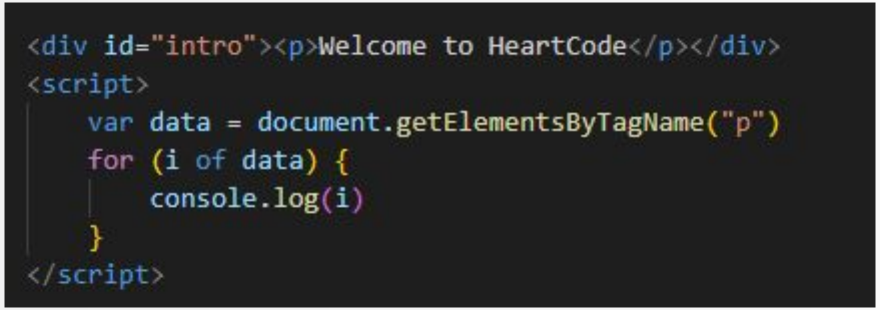
- Find elements by tag name [Returns HTML collection Obj]



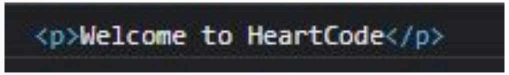
```
<div id="intro"><p>Welcome to HeartCode</p></div>
<script>
    console.log(document.getElementsByTagName("p"))
</script>
```



```
▼HTMLCollection [p]
  ▶0: p
   length: 1
  ▶[[Prototype]]: HTMLCollection
```



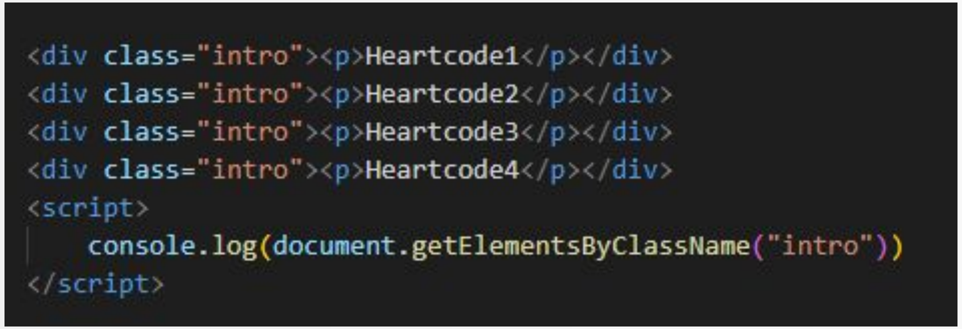
```
<div id="intro"><p>Welcome to HeartCode</p></div>
<script>
    var data = document.getElementsByTagName("p")
    for (i of data) {
        console.log(i)
    }
</script>
```



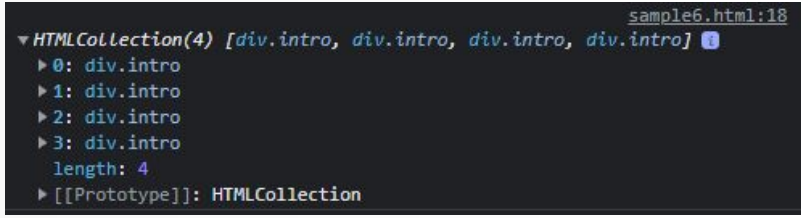
```
<p>Welcome to HeartCode</p>
```

DOM – getElementsByClassName

- Find elements by class name [Returns HTML collection Obj]

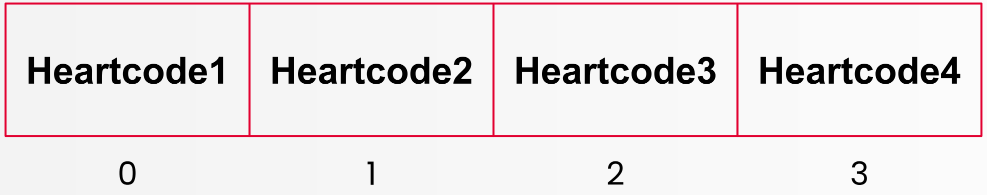


```html
<div class="intro"><p>Heartcode1</p></div>
<div class="intro"><p>Heartcode2</p></div>
<div class="intro"><p>Heartcode3</p></div>
<div class="intro"><p>Heartcode4</p></div>
<script>
    console.log(document.getElementsByClassName("intro"))
</script>
```

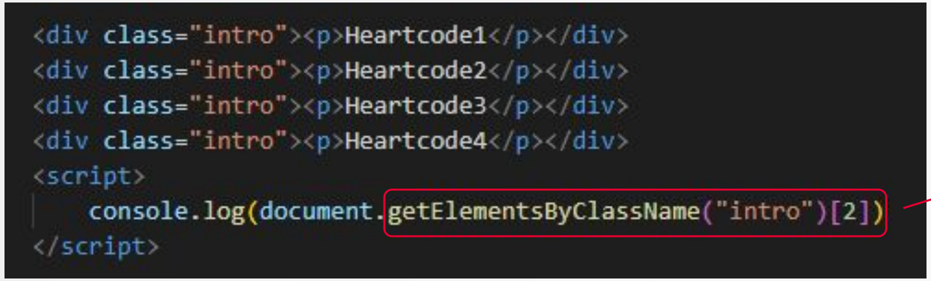


```
                                                        sample6.html:18
▼HTMLCollection(4) [div.intro, div.intro, div.intro, div.intro]
  ▶0: div.intro
  ▶1: div.intro
  ▶2: div.intro
  ▶3: div.intro
   length: 4
  ▶[[Prototype]]: HTMLCollection
```

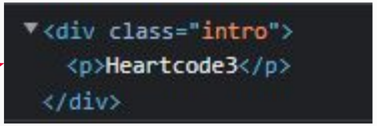
Collection



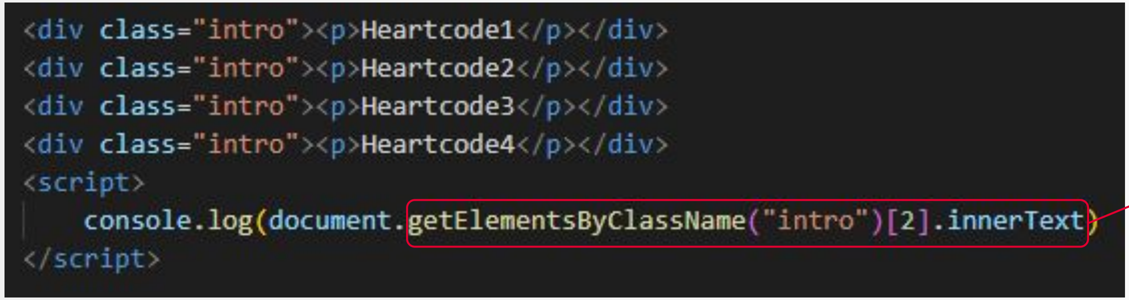
DOM – getElementsByClassName



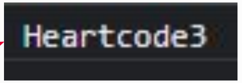
```
<div class="intro"><p>Heartcode1</p></div>
<div class="intro"><p>Heartcode2</p></div>
<div class="intro"><p>Heartcode3</p></div>
<div class="intro"><p>Heartcode4</p></div>
<script>
    console.log(document.getElementsByClassName("intro")[2])
</script>
```



```
▼<div class="intro">
   <p>Heartcode3</p>
 </div>
```



```
<div class="intro"><p>Heartcode1</p></div>
<div class="intro"><p>Heartcode2</p></div>
<div class="intro"><p>Heartcode3</p></div>
<div class="intro"><p>Heartcode4</p></div>
<script>
    console.log(document.getElementsByClassName("intro")[2].innerText)
</script>
```



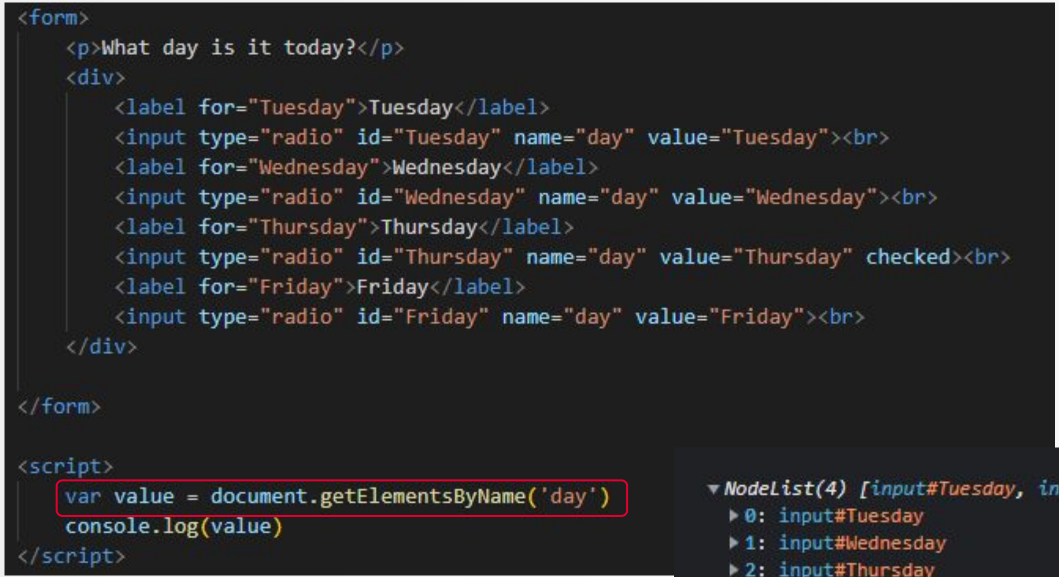
```
Heartcode3
```

Collection

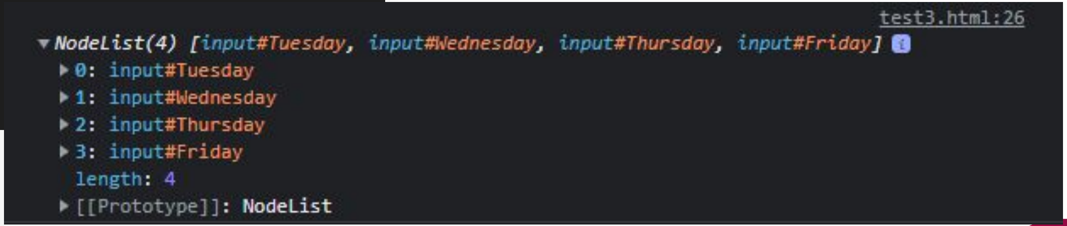
Heartcode1	Heartcode2	Heartcode3	Heartcode4
0	1	2	3

DOM – getElementsByName

- Find elements by name [Returns Collection of Elements]



```html
<form>
    <p>What day is it today?</p>
    <div>
        <label for="Tuesday">Tuesday</label>
        <input type="radio" id="Tuesday" name="day" value="Tuesday"><br>
        <label for="Wednesday">Wednesday</label>
        <input type="radio" id="Wednesday" name="day" value="Wednesday"><br>
        <label for="Thursday">Thursday</label>
        <input type="radio" id="Thursday" name="day" value="Thursday" checked><br>
        <label for="Friday">Friday</label>
        <input type="radio" id="Friday" name="day" value="Friday"><br>
    </div>

</form>

<script>
    var value = document.getElementsByName('day')
    console.log(value)
</script>
```

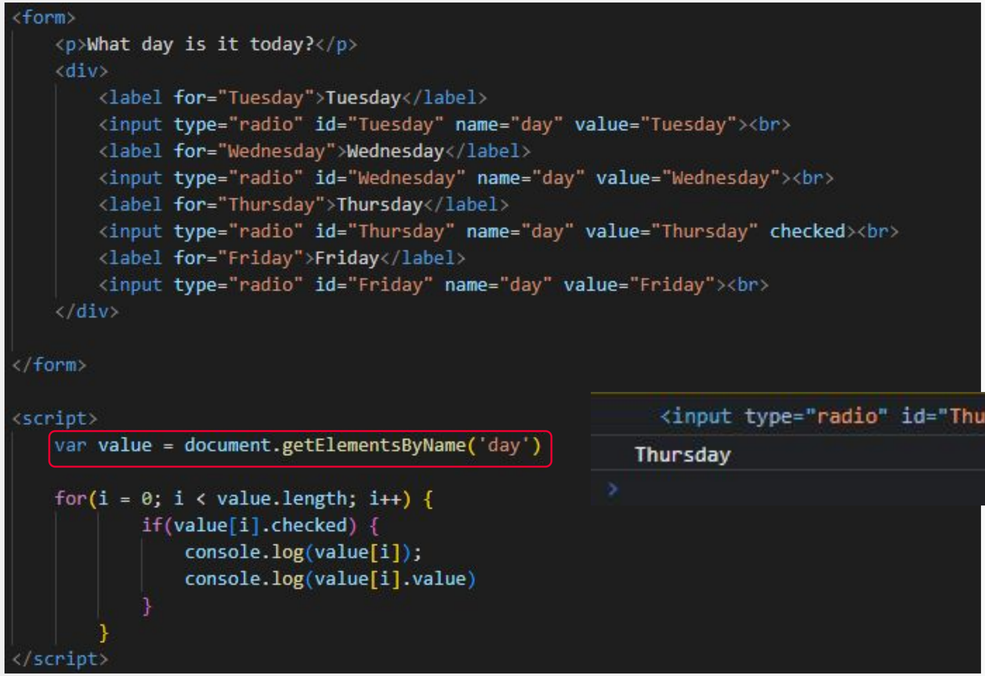


```
                                                                              test3.html:26
▼NodeList(4) [input#Tuesday, input#Wednesday, input#Thursday, input#Friday]
  ▶0: input#Tuesday
  ▶1: input#Wednesday
  ▶2: input#Thursday
  ▶3: input#Friday
   length: 4
  ▶[[Prototype]]: NodeList
```

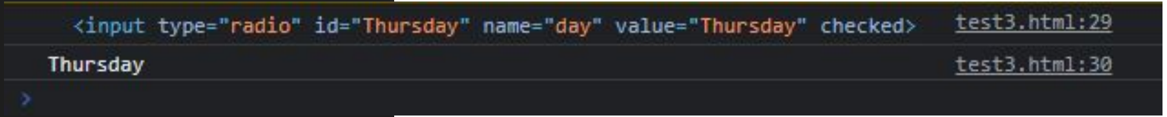
NodeList

- Array-like collection (list) of nodes
- Can be assessed by index

DOM – getElementsByName



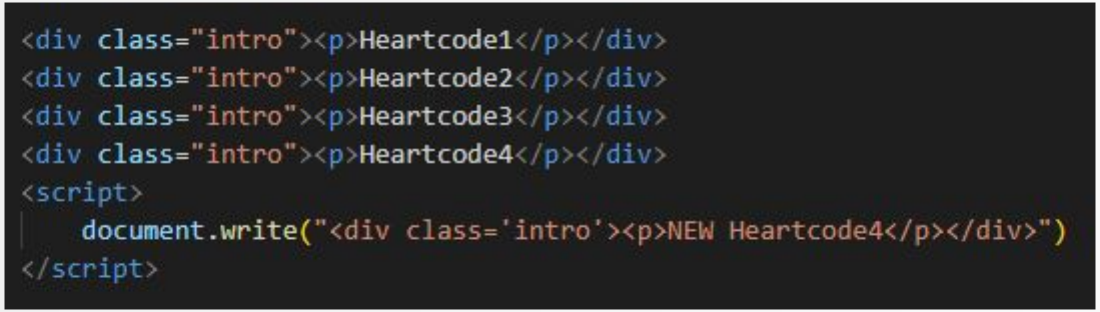
```
<form>
    <p>What day is it today?</p>
    <div>
        <label for="Tuesday">Tuesday</label>
        <input type="radio" id="Tuesday" name="day" value="Tuesday"><br>
        <label for="Wednesday">Wednesday</label>
        <input type="radio" id="Wednesday" name="day" value="Wednesday"><br>
        <label for="Thursday">Thursday</label>
        <input type="radio" id="Thursday" name="day" value="Thursday" checked><br>
        <label for="Friday">Friday</label>
        <input type="radio" id="Friday" name="day" value="Friday"><br>
    </div>

</form>

<script>
    var value = document.getElementsByName('day')

    for(i = 0; i < value.length; i++) {
            if(value[i].checked) {
                console.log(value[i]);
                console.log(value[i].value)
            }
        }
</script>
```



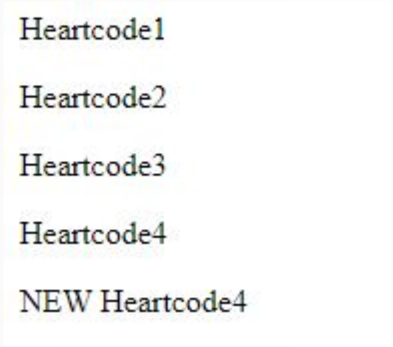
```
    <input type="radio" id="Thursday" name="day" value="Thursday" checked>   test3.html:29
  Thursday                                                                   test3.html:30
>
```

DOM – Adding Element

- document.write() can be used to write directly to the HTML output stream



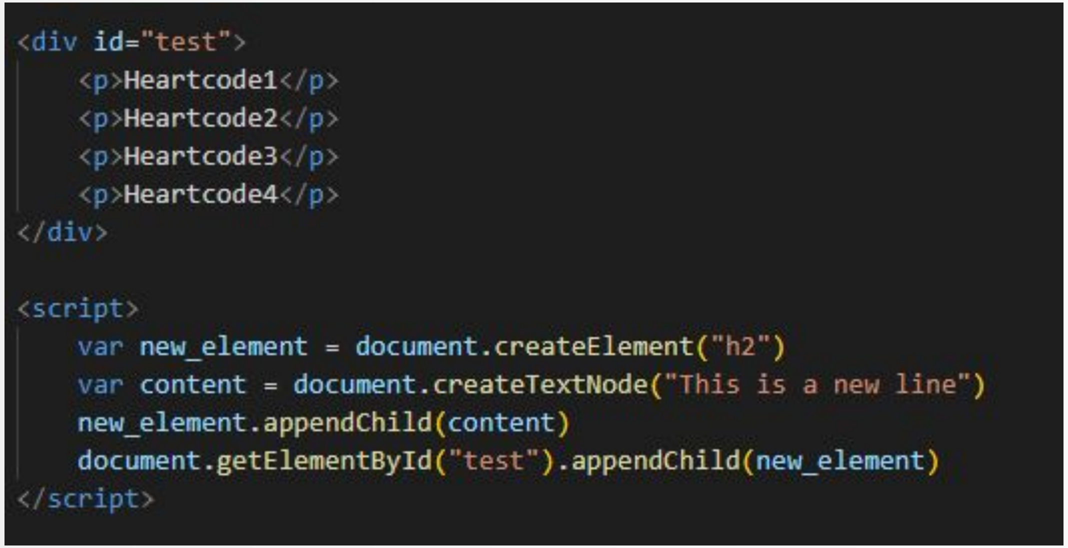
```
<div class="intro"><p>Heartcode1</p></div>
<div class="intro"><p>Heartcode2</p></div>
<div class="intro"><p>Heartcode3</p></div>
<div class="intro"><p>Heartcode4</p></div>
<script>
    document.write("<div class='intro'><p>NEW Heartcode4</p></div>")
</script>
```



```
Heartcode1

Heartcode2

Heartcode3

Heartcode4

NEW Heartcode4
```

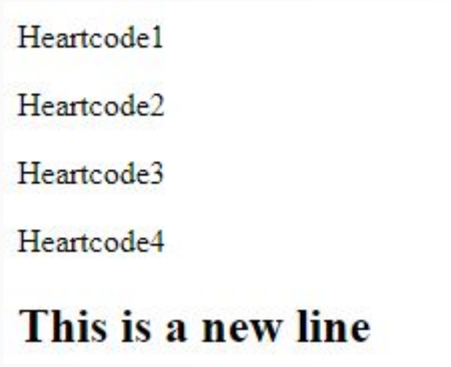
DOM - Create New HTML Elements

- Create a header element and append it to an existing element.
- This can be used to dynamically modify your HTML element



```
<div id="test">
    <p>Heartcode1</p>
    <p>Heartcode2</p>
    <p>Heartcode3</p>
    <p>Heartcode4</p>
</div>

<script>
    var new_element = document.createElement("h2")
    var content = document.createTextNode("This is a new line")
    new_element.appendChild(content)
    document.getElementById("test").appendChild(new_element)
</script>
```

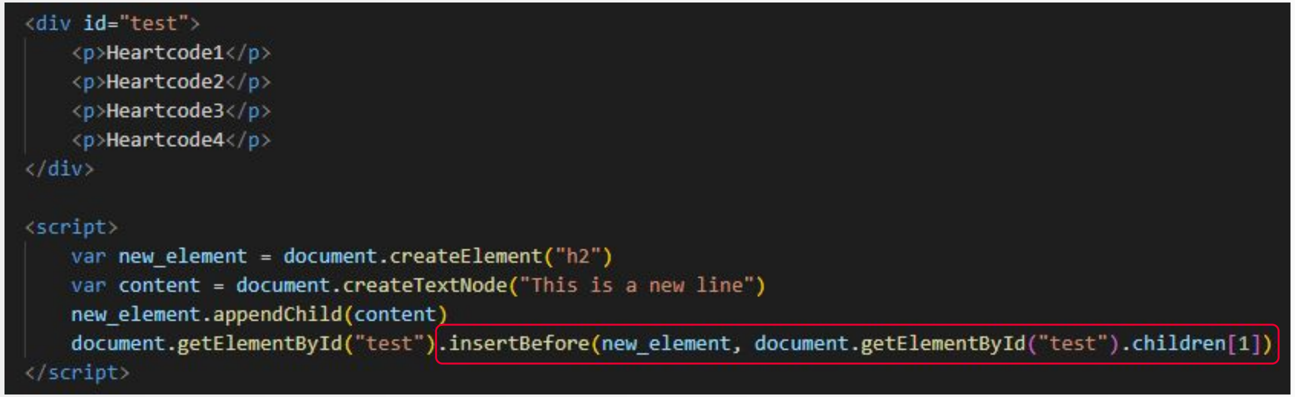
sample6.html



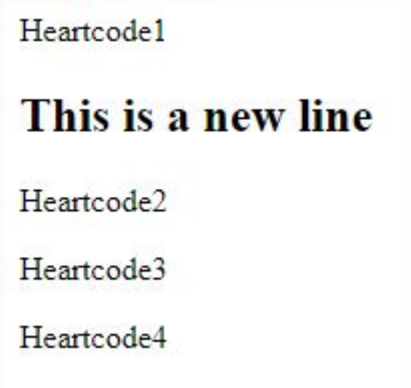
```
Heartcode1

Heartcode2

Heartcode3

Heartcode4

This is a new line
```

DOM - Create New HTML Elements

- Create a new line at a specific location



```html
<div id="test">
    <p>Heartcode1</p>
    <p>Heartcode2</p>
    <p>Heartcode3</p>
    <p>Heartcode4</p>
</div>

<script>
    var new_element = document.createElement("h2")
    var content = document.createTextNode("This is a new line")
    new_element.appendChild(content)
    document.getElementById("test").insertBefore(new_element, document.getElementById("test").children[1])
</script>
```



Heartcode1

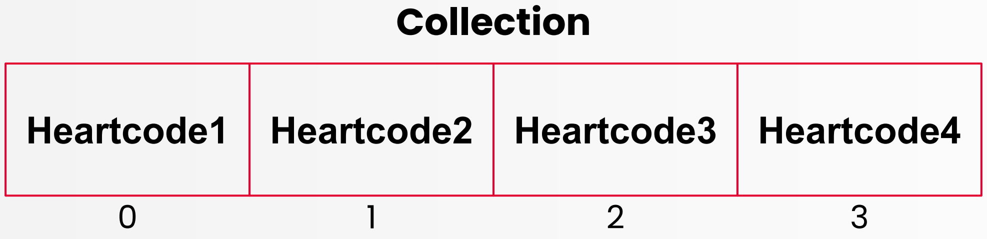
This is a new line

Heartcode2

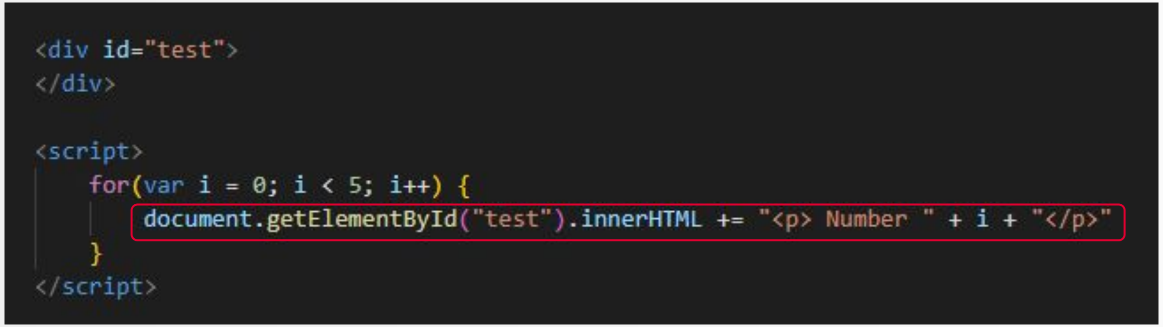
Heartcode3

Heartcode4

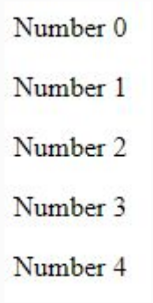
sample7.html



DOM – Adding Elements



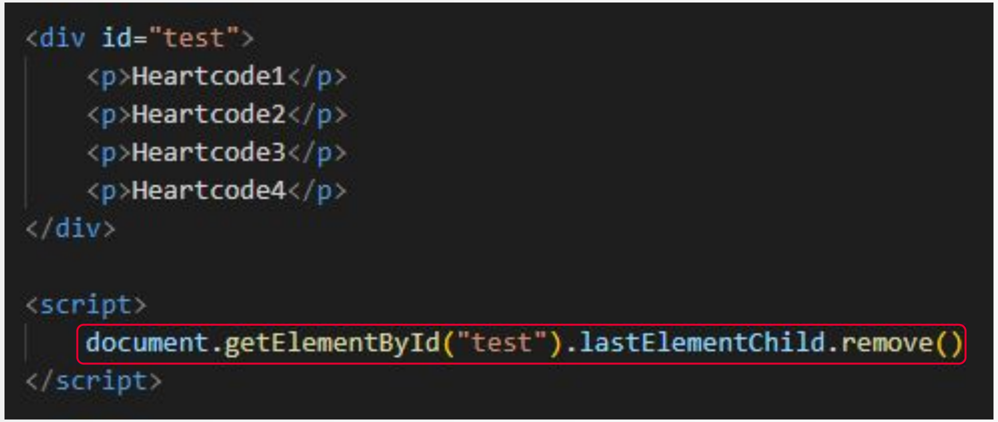
```
<div id="test">
</div>

<script>
    for(var i = 0; i < 5; i++) {
        document.getElementById("test").innerHTML += "<p> Number " + i + "</p>"
    }
</script>
```



```
Number 0

Number 1

Number 2

Number 3

Number 4
```

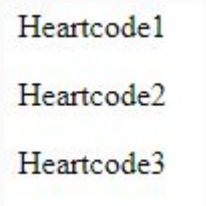
DOM - Remove HTML Elements

- Retrieve the element node and use remove()
    - lastElementChild
    - childNodes[index]



```
<div id="test">
    <p>Heartcode1</p>
    <p>Heartcode2</p>
    <p>Heartcode3</p>
    <p>Heartcode4</p>
</div>

<script>
    document.getElementById("test").lastElementChild.remove()
</script>
```

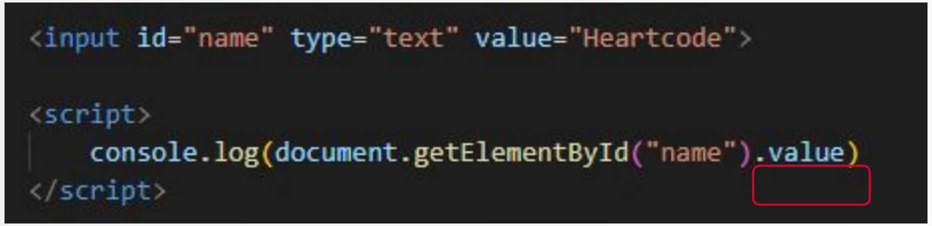
sample8.html



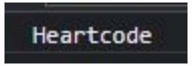
```
Heartcode1

Heartcode2

Heartcode3
```

DOM – Obtain Value

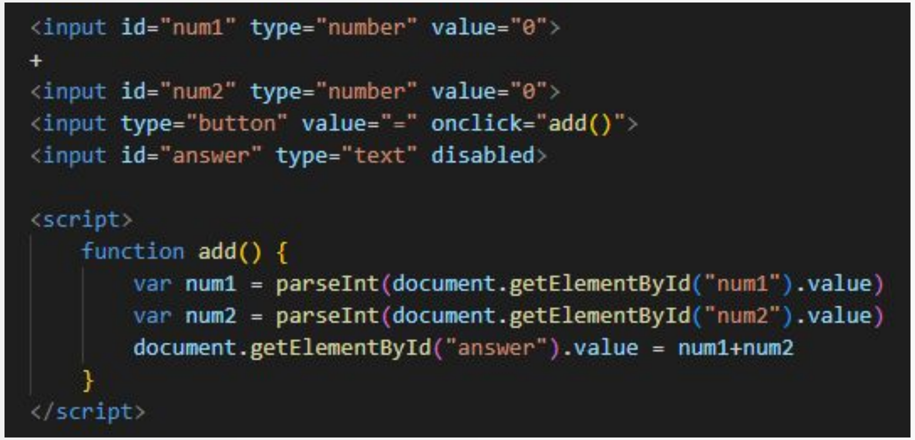
- Using .value



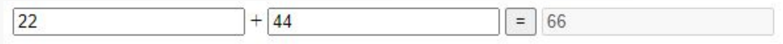
```html
<input id="name" type="text" value="Heartcode">

<script>
    console.log(document.getElementById("name").value)
</script>
```



```
Heartcode
```



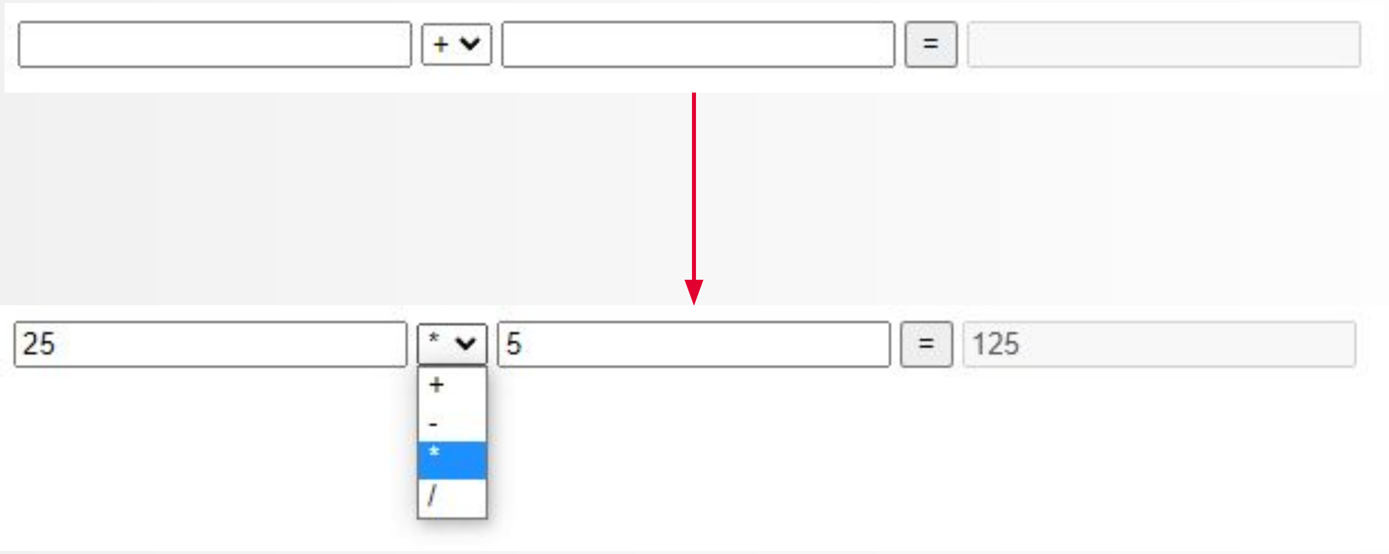
```html
<input id="num1" type="number" value="0">
+
<input id="num2" type="number" value="0">
<input type="button" value="=" onclick="add()">
<input id="answer" type="text" disabled>

<script>
    function add() {
        var num1 = parseInt(document.getElementById("num1").value)
        var num2 = parseInt(document.getElementById("num2").value)
        document.getElementById("answer").value = num1+num2
    }
</script>
```



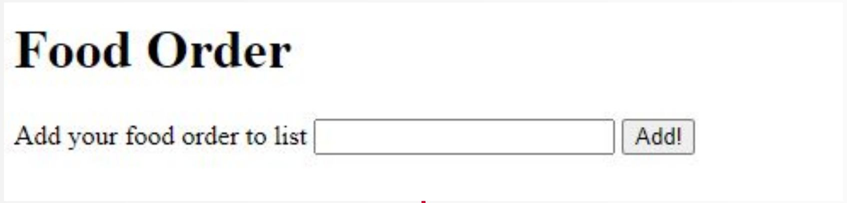
```
22 + 44 = 66
```

sample9.html

Exercise 1 – Calculator

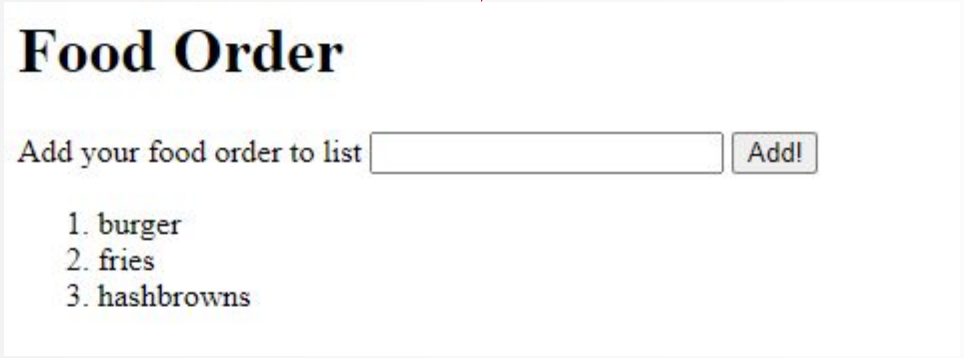


Exercise 2 – Food Order



Food Order

Add your food order to list



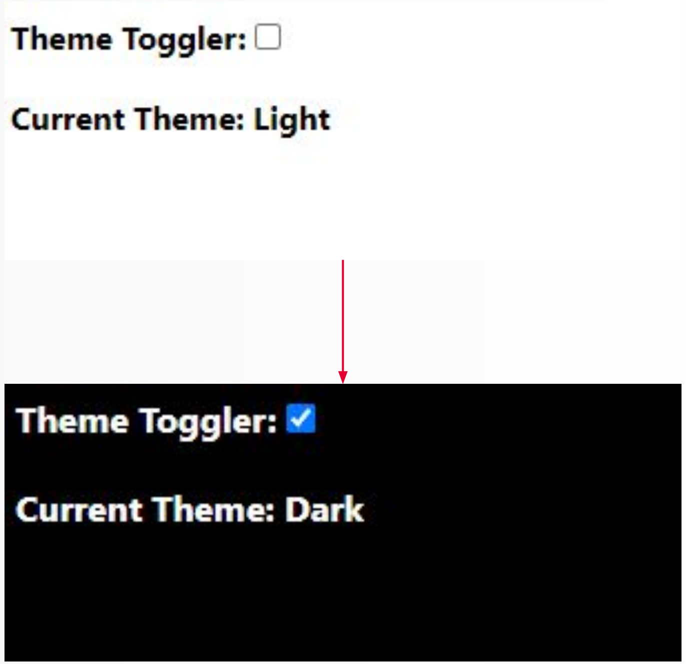
Food Order

Add your food order to list

1. burger
2. fries
3. hashbrowns

Theme Toggler

- **Toggle the theme of a webpage using a checkbox**
- When **checkbox is unchecked:**
  - Background color: white
  - Text color: black
  - Words displayed: “Current Theme: Light”


Theme Toggler: ☐

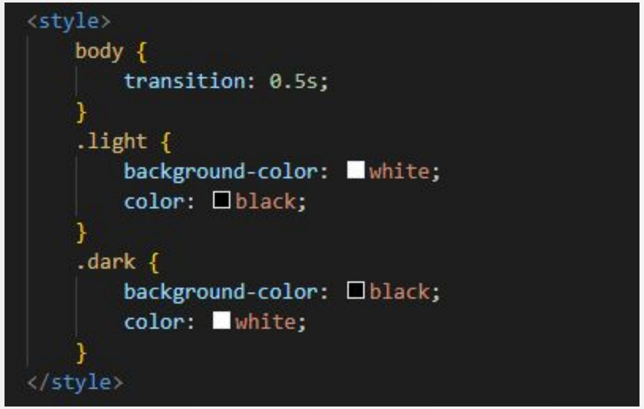
Current Theme: Light

Theme Toggler: ☑

Current Theme: Dark

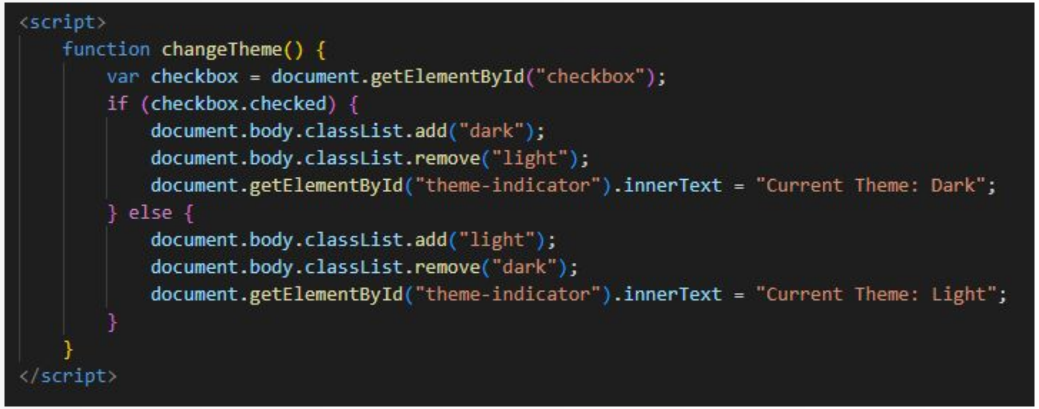
- When **checkbox is checked:**
  - Background color: black
  - Text color: white
  - Words displayed: “Current Theme: Dark”

Theme Toggler



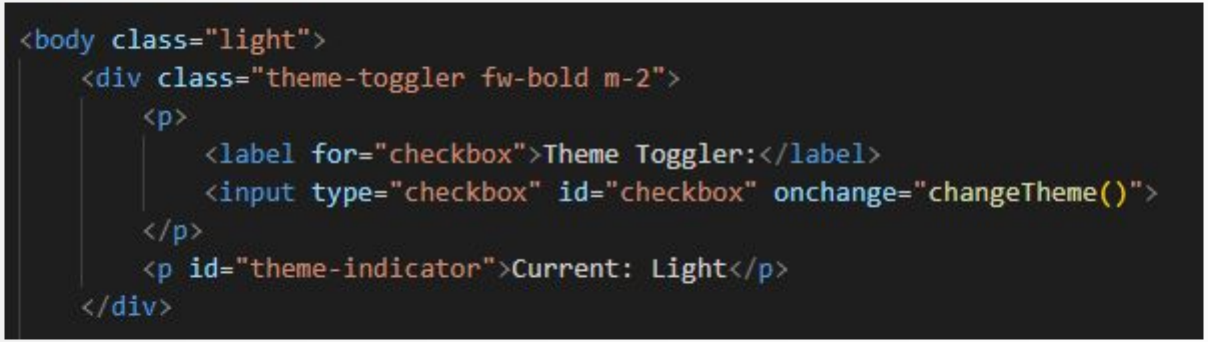
```
<style>
    body {
        transition: 0.5s;
    }
    .light {
        background-color: white;
        color: black;
    }
    .dark {
        background-color: black;
        color: white;
    }
</style>
```

CSS



```
<script>
    function changeTheme() {
        var checkbox = document.getElementById("checkbox");
        if (checkbox.checked) {
            document.body.classList.add("dark");
            document.body.classList.remove("light");
            document.getElementById("theme-indicator").innerText = "Current Theme: Dark";
        } else {
            document.body.classList.add("light");
            document.body.classList.remove("dark");
            document.getElementById("theme-indicator").innerText = "Current Theme: Light";
        }
    }
</script>
```

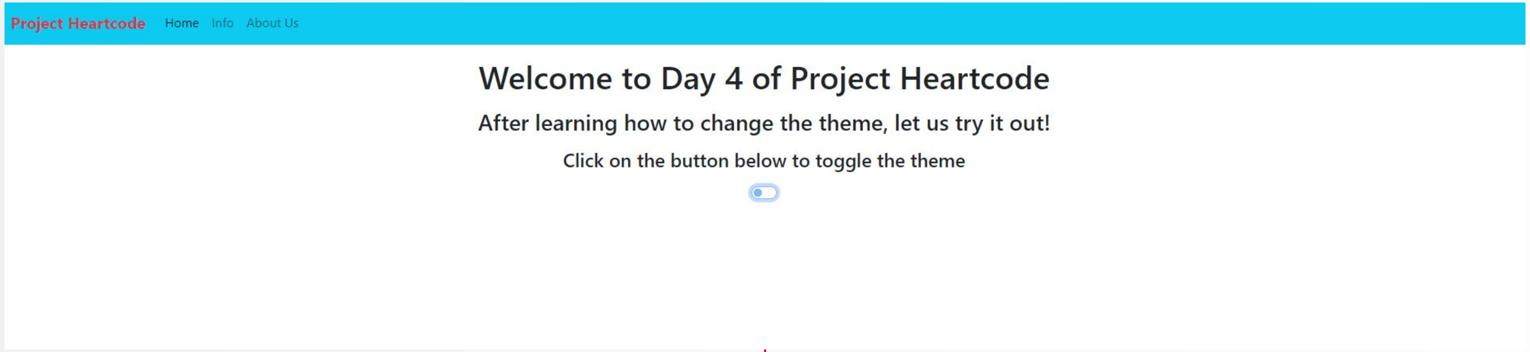
Javascript



```
<body class="light">
    <div class="theme-toggler fw-bold m-2">
        <p>
            <label for="checkbox">Theme Toggler:</label>
            <input type="checkbox" id="checkbox" onchange="changeTheme()">
        </p>
        <p id="theme-indicator">Current: Light</p>
    </div>
```

HTML

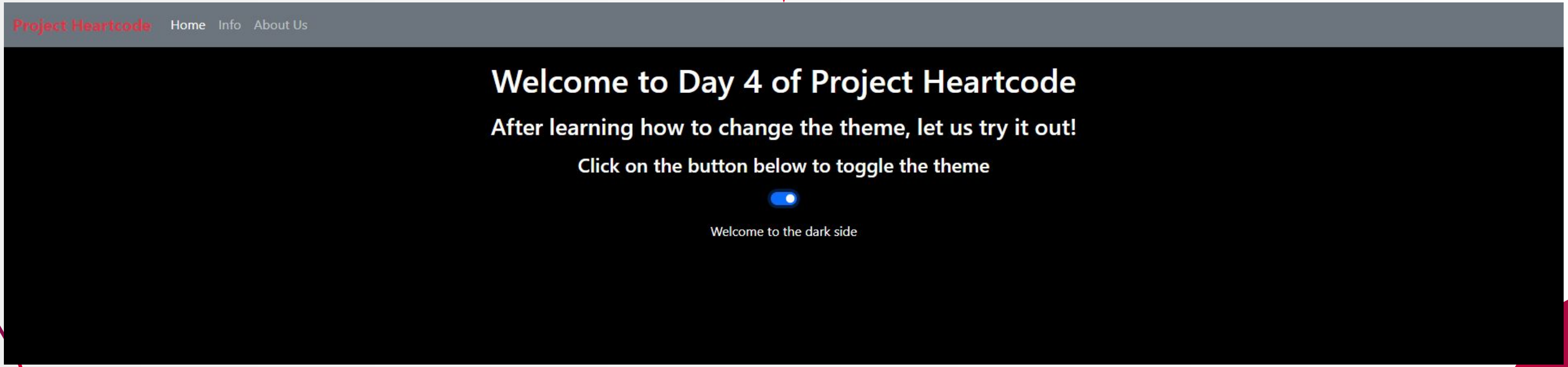
Exercise 3 – Theme Toggler



Welcome to Day 4 of Project Heartcode

After learning how to change the theme, let us try it out!

Click on the button below to toggle the theme



Welcome to Day 4 of Project Heartcode

After learning how to change the theme, let us try it out!

Click on the button below to toggle the theme

Welcome to the dark side

wooclap

https://wooclap.com
HEARTCODE3

03

Github & Github Pages

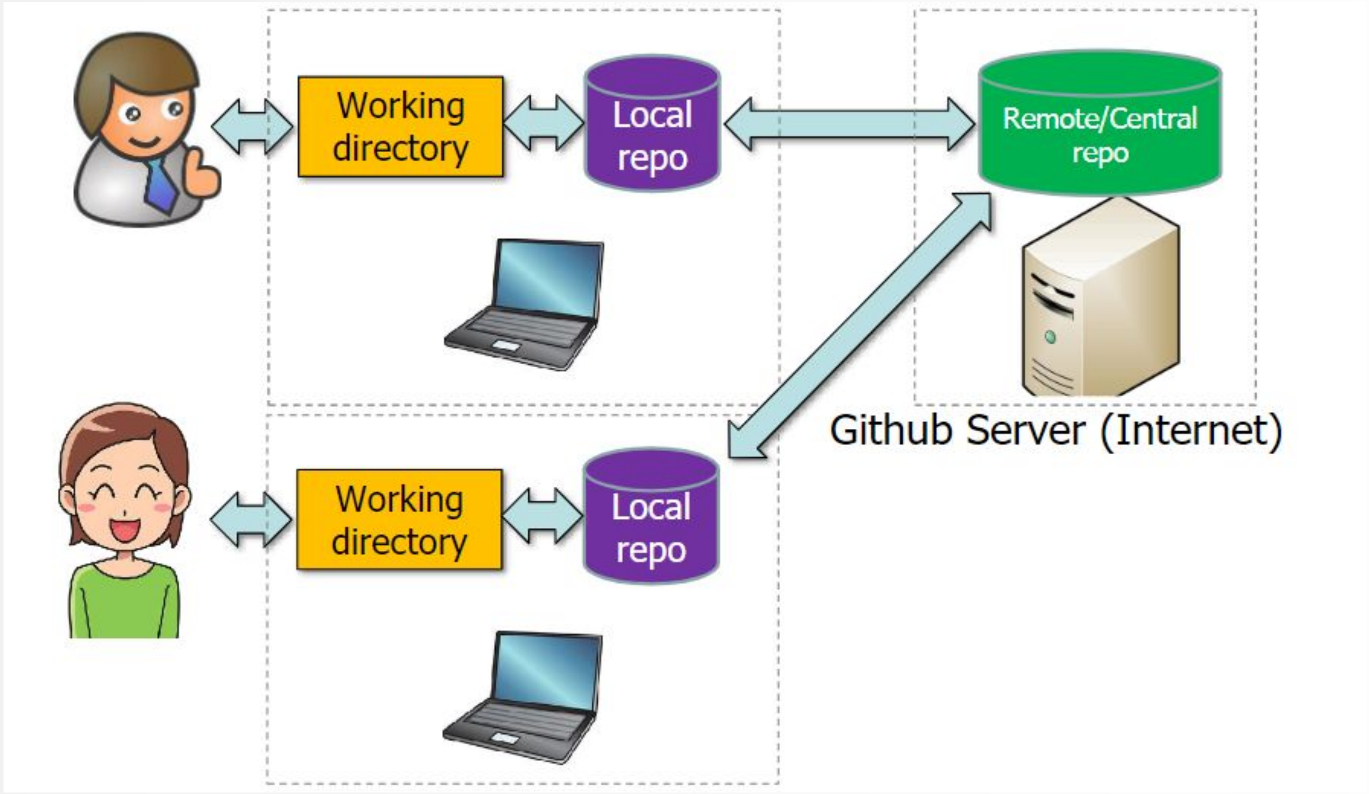
GitHub

Git

- Git is a distributed version control system
- Mechanisms to share source code and project materials
- Full control over when and what you ‘publish’ to the central code repository
- Systematically tracks history of changes
    - Allows you to revert to an earlier version

Git

Git – Collaboration

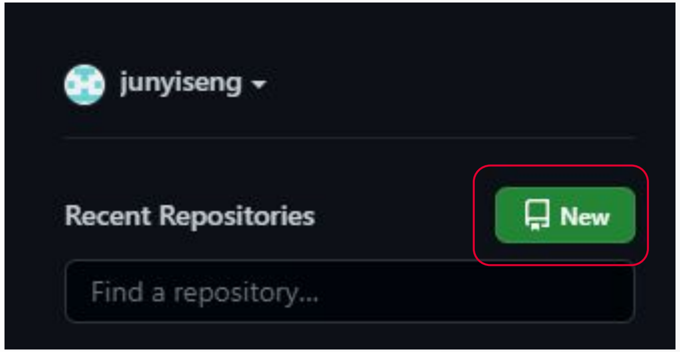


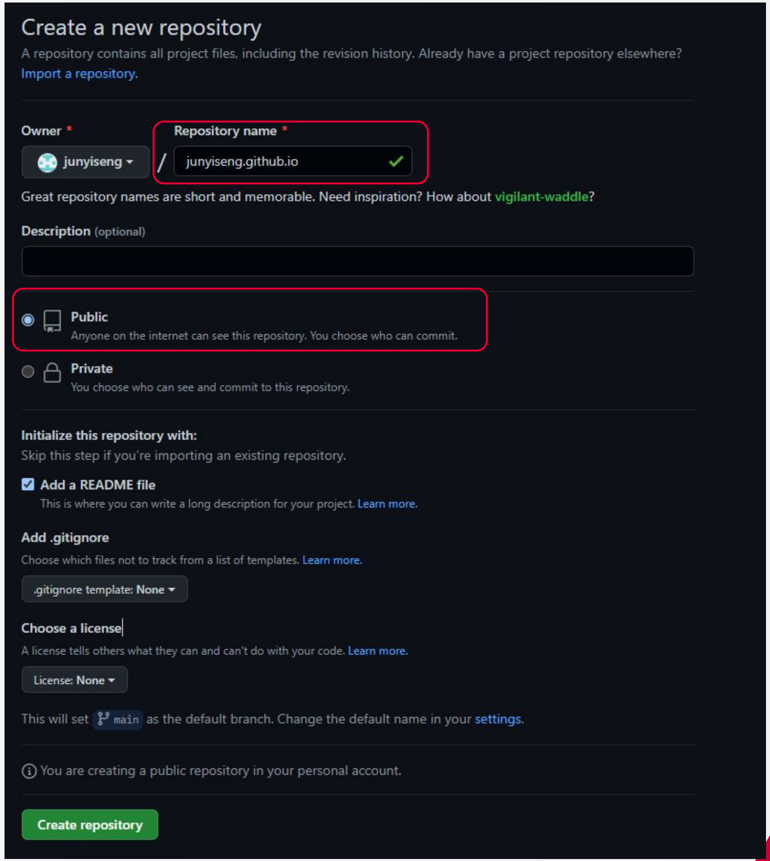
Key Acronyms for Github

- **Git Clone** - Getting a full local copy of the central repository
- **Git Pull** - Retrieves up-to-date copy of remote repo
- **Git Add** - Mark files to be staged for next commit
- **Git Commit** - Creates Snapshot of project's staged changes
- **Git Push** - All committed / staged changes in local repository are sent to remote repository

Github Demo

Create a New Repository





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * junyiseng / Repository name * junyiseng.github.io

Great repository names are short and memorable. Need inspiration? How about vigilant-waddle?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

[✓] Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

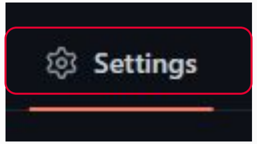
This will set `main` as the default branch. Change the default name in your settings.

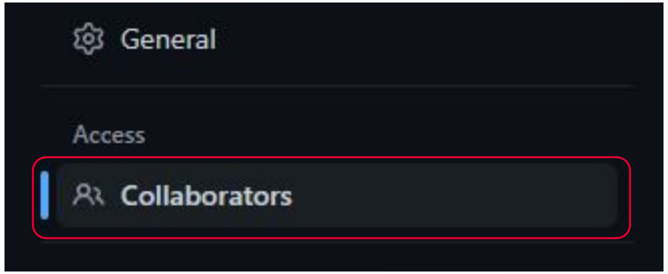
You are creating a public repository in your personal account.

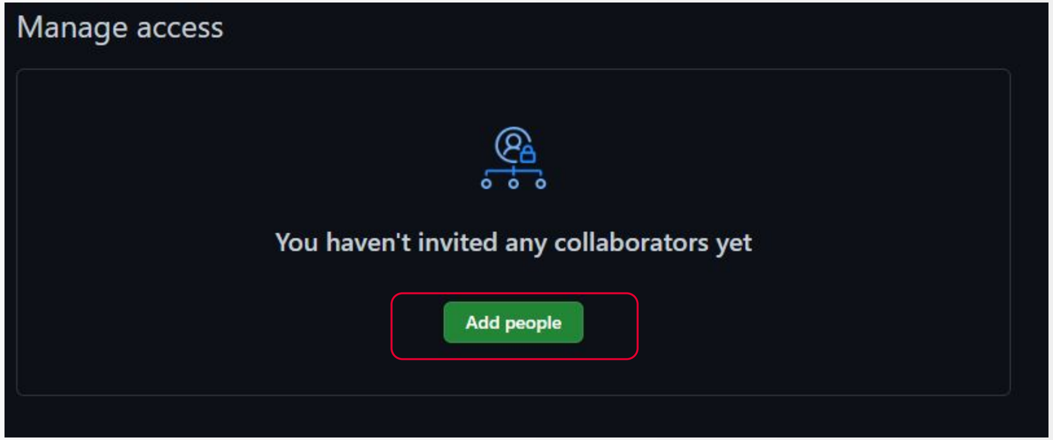
Create repository

1. Click on **New**
2. Enter the relevant information
   - **Repository Name (username.github.io)**
   - **Select "Public"**
3. Click on **Create Repository**

Adding Team Members (Collaborators)



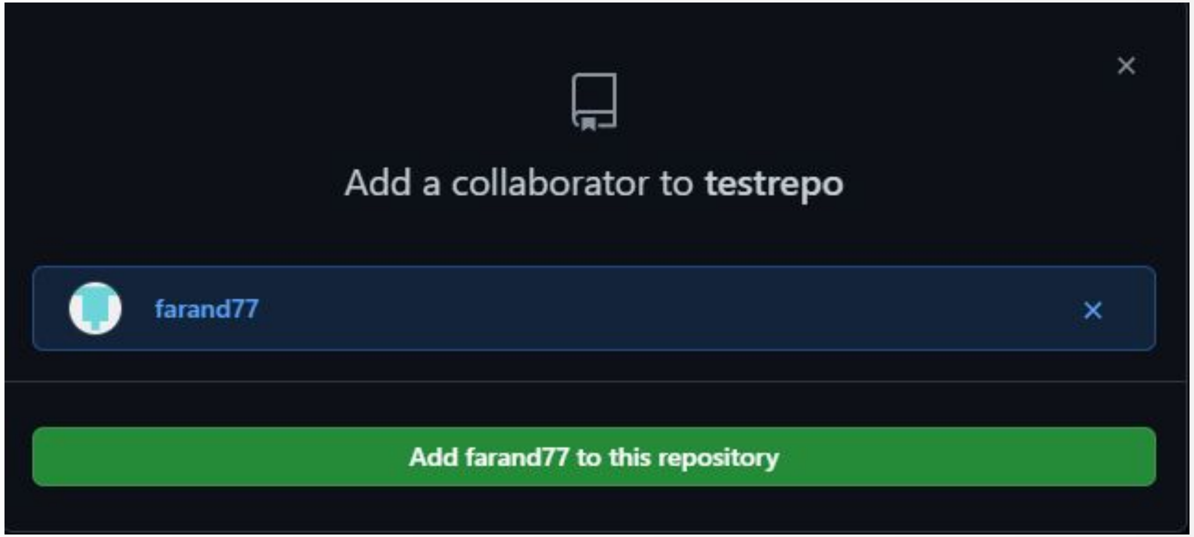



Manage access

You haven't invited any collaborators yet

1. Click on **Settings**
2. Under General, click on **Collaborators**
3. Under Manage Access, click on **Add People**

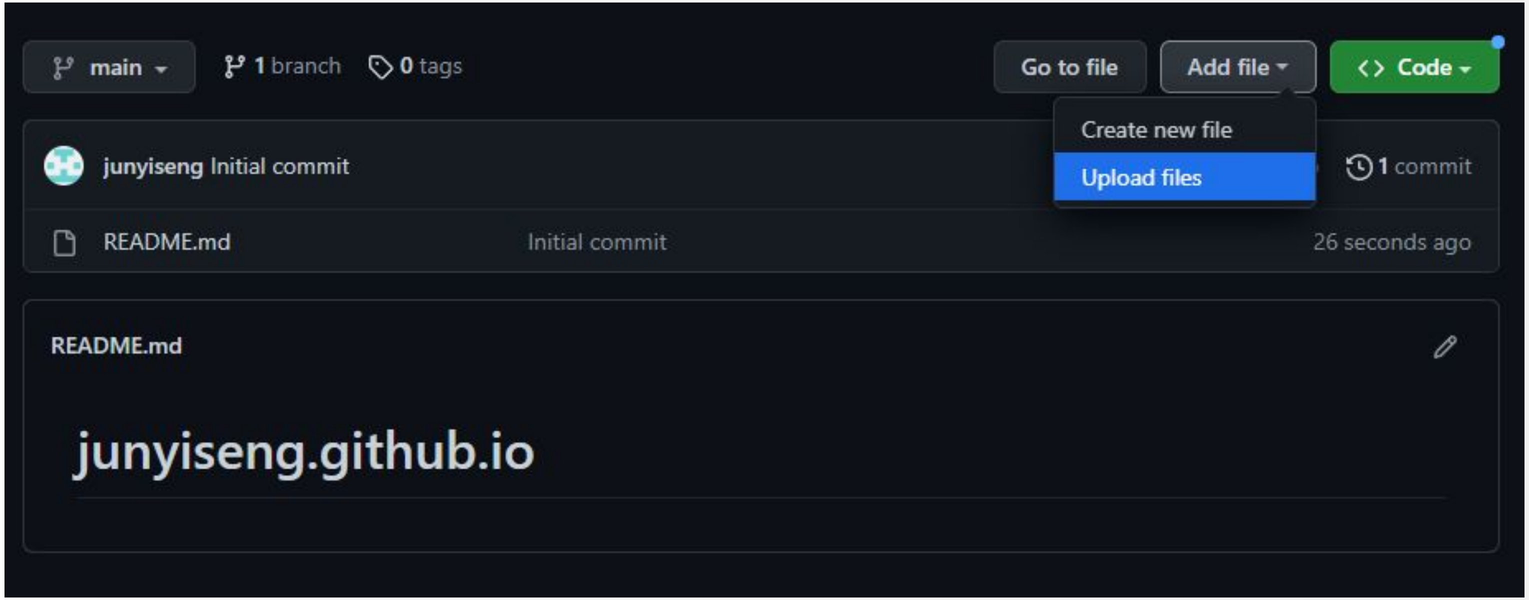
Adding Team Members (Collaborators)


Add a collaborator to **testrepo**

farand77

1. Search for the username of the team member you wish to add
2. Click on “Add ___ to this repository”
3. Added team members will receive the invitation via email

Uploading Files to Github



junyiseng Initial commit

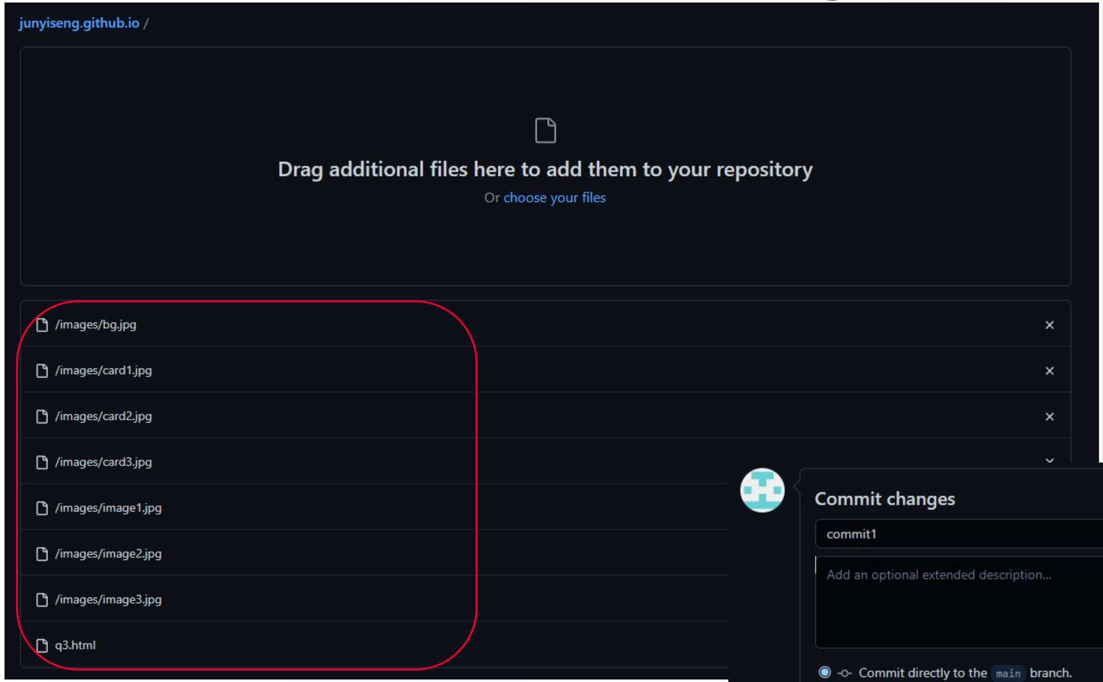
README.md Initial commit 26 seconds ago

README.md

junyiseng.github.io

1. On the homepage of the repo, click Add File → Upload Files

Uploading Files to Github

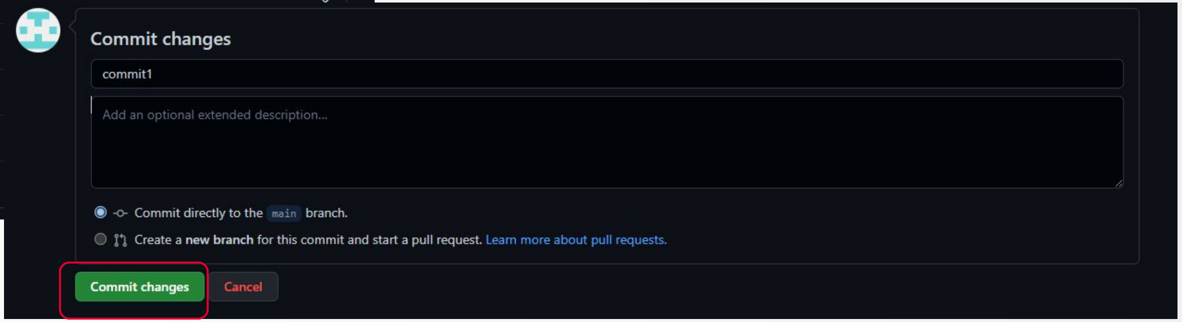


junyiseng.github.io /

Drag additional files here to add them to your repository

Or choose your files

/images/bg.jpg
/images/card1.jpg
/images/card2.jpg
/images/card3.jpg
/images/image1.jpg
/images/image2.jpg
/images/image3.jpg
q3.html



Commit changes

commit1

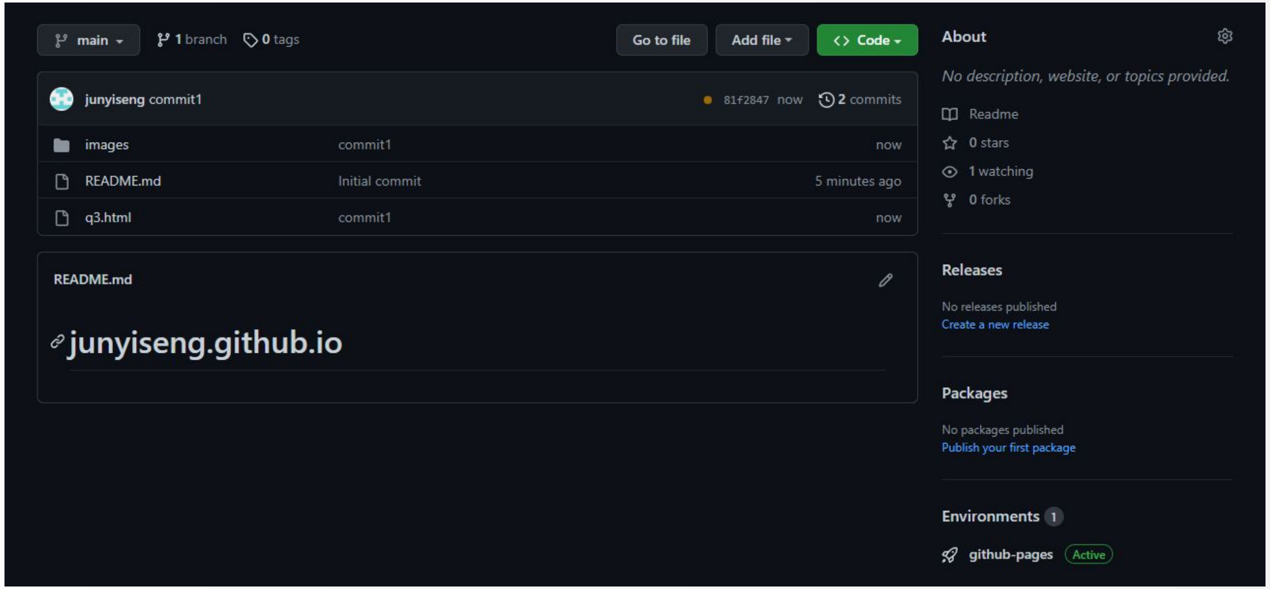
Add an optional extended description...

Commit directly to the `main` branch.

Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

1. Drag the files you wish to upload
2. Click on Commit Changes

Uploading Files to Github



junyiseng commit1 81f2847 now 2 commits

images	commit1	now
README.md	Initial commit	5 minutes ago
q3.html	commit1	now

README.md

junyiseng.github.io

About

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

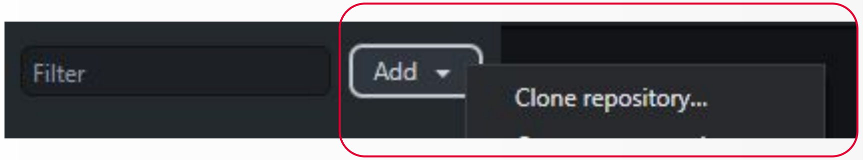
Publish your first package

Environments 1

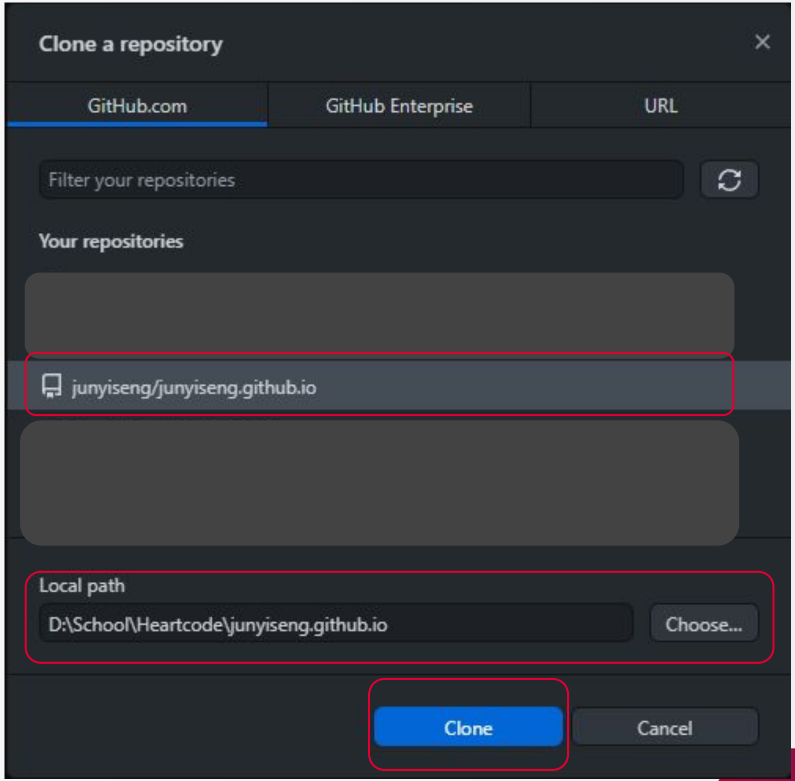
github-pages Active

1. The uploaded files will appear in the main repo page

Cloning GIT Repository



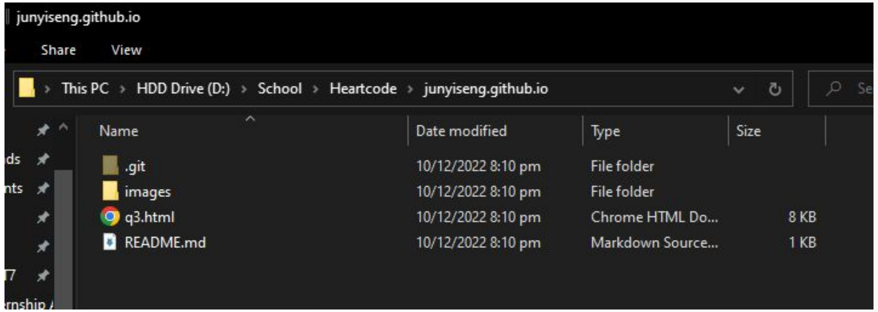
1. Open Github Desktop
2. Click on Add → Clone Repository
3. Choose the repository you wish to clone
4. Select the path to store the cloned repository in your laptop
5. Click on Clone



Local path: D:\School\Heartcode\junyiseng.github.io

Cloning GIT Repository

1. You should see the cloned repository on the left panel
2. Right click on it → Show in Explorer
3. The whole repository will appear in the path you have defined earlier on

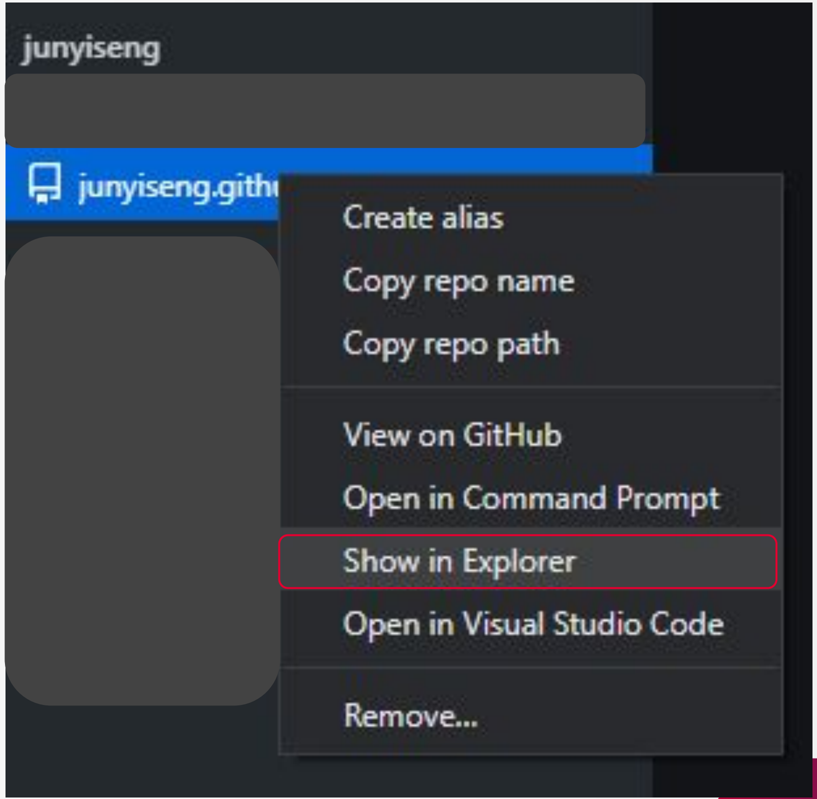


junyiseng.github.io

Share View

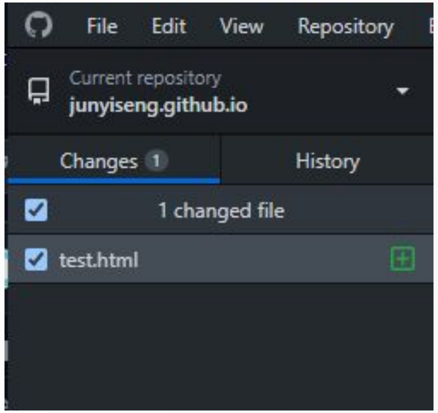
This PC › HDD Drive (D:) › School › Heartcode › junyiseng.github.io

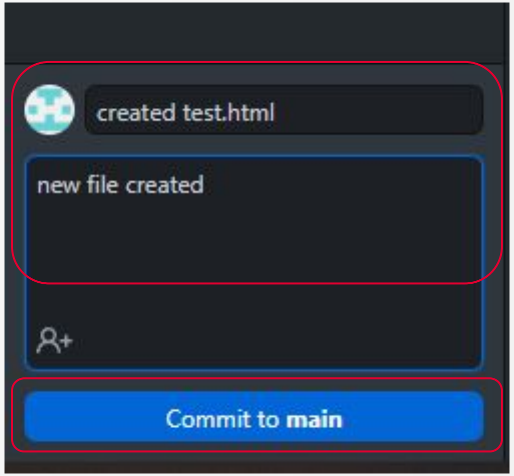
Name	Date modified	Type	Size
.git	10/12/2022 8:10 pm	File folder	
images	10/12/2022 8:10 pm	File folder	
q3.html	10/12/2022 8:10 pm	Chrome HTML Do...	8 KB
README.md	10/12/2022 8:10 pm	Markdown Source...	1 KB



Pushing Updates into Github

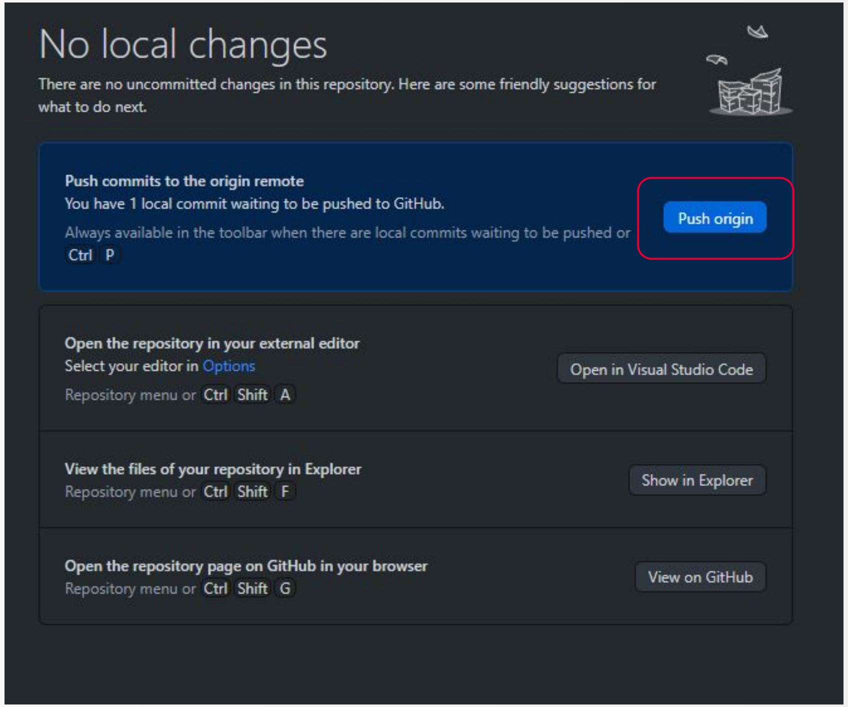
1. Changes done to files can be seen on the left panel
2. Write a description of the changes made
3. Click on Commit to Main





Pushing Updates into Github

1. Once the changes are confirm, you can push the commits to the Github Server



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or Ctrl P

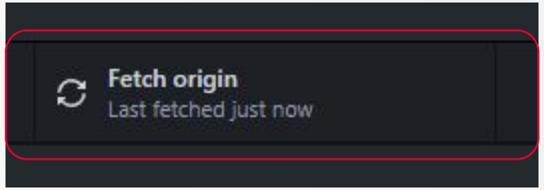
Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A

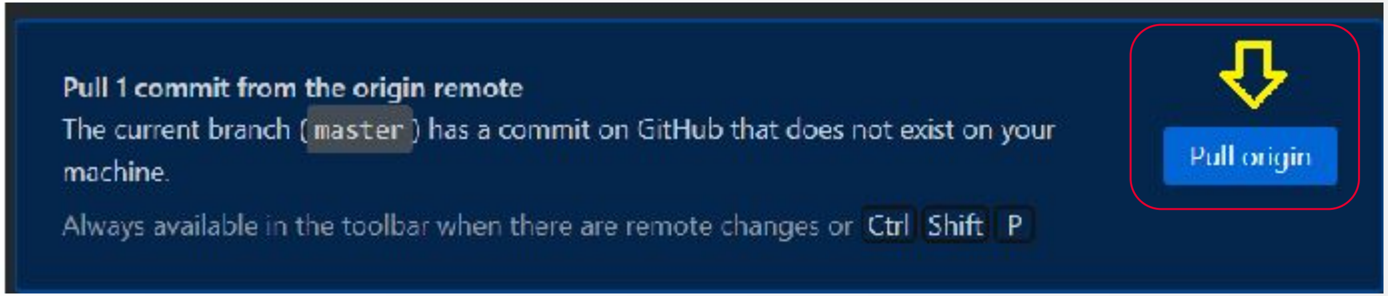
View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

Pulling Updates from Github

1. Click on **Fetch Origin**
2. If there is any updates, the below notification will appear
3. Click on Pull Origin



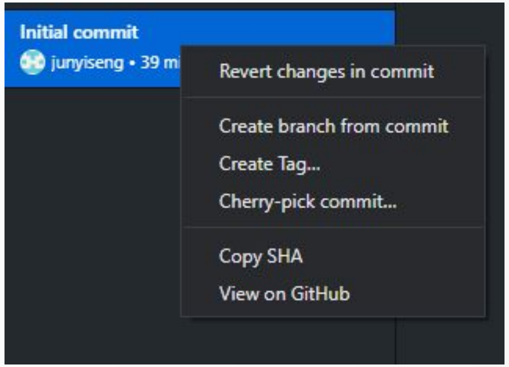

Pull 1 commit from the origin remote

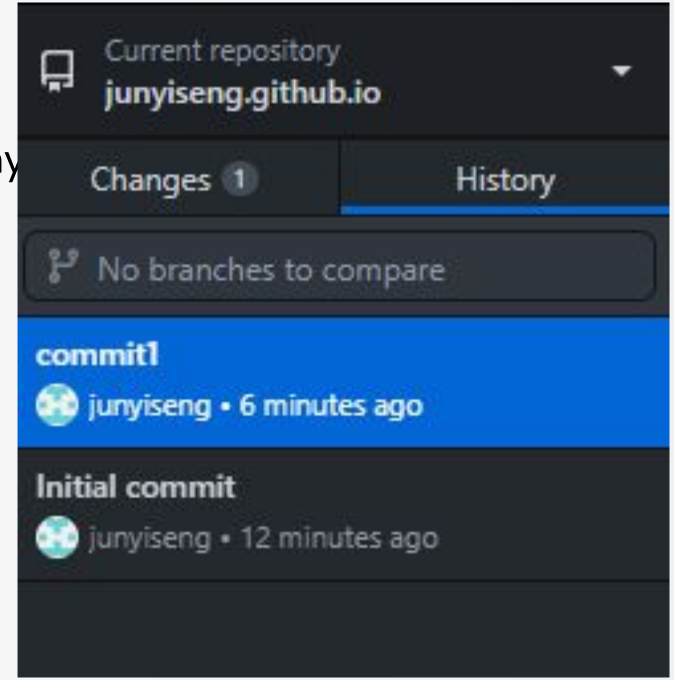
The current branch (`master`) has a commit on GitHub that does not exist on your machine.

Always available in the toolbar when there are remote changes or Ctrl Shift P

Looking at History

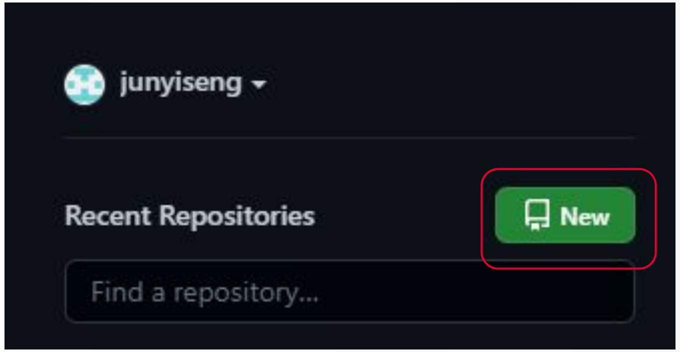
1. On the History Tab, can view what changes has been made recently
2. From this tab, you are able to revert to any previous history versions

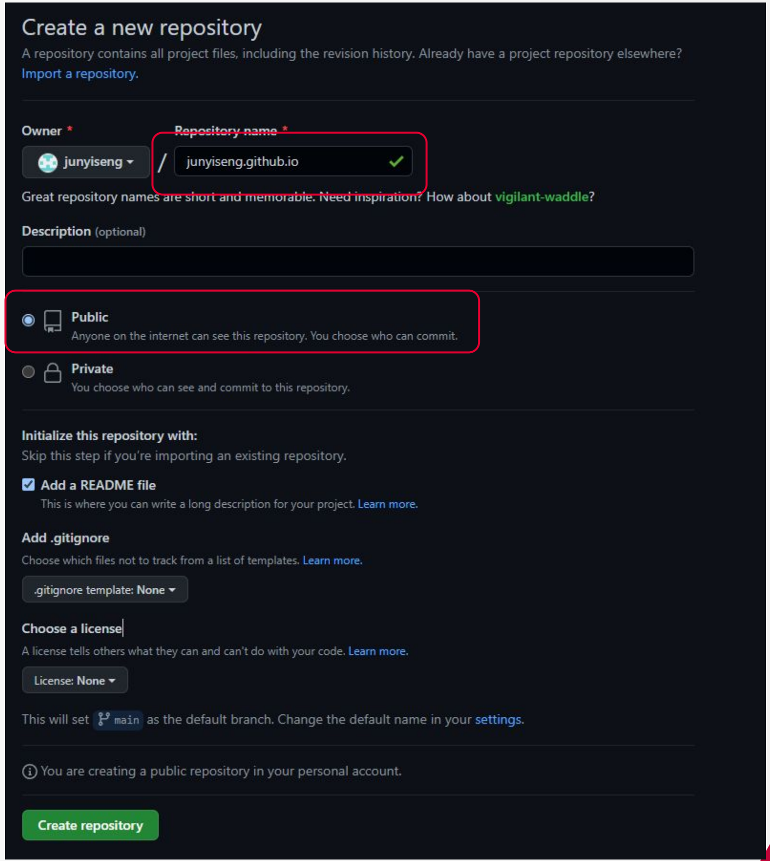




Deployment – Github Page

Create a New Repository





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * junyiseng / Repository name * junyiseng.github.io

Great repository names are short and memorable. Need inspiration? How about vigilant-waddle?

Description (optional)

(●) Public
Anyone on the internet can see this repository. You choose who can commit.

( ) Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

[✓] Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

This will set `main` as the default branch. Change the default name in your settings.

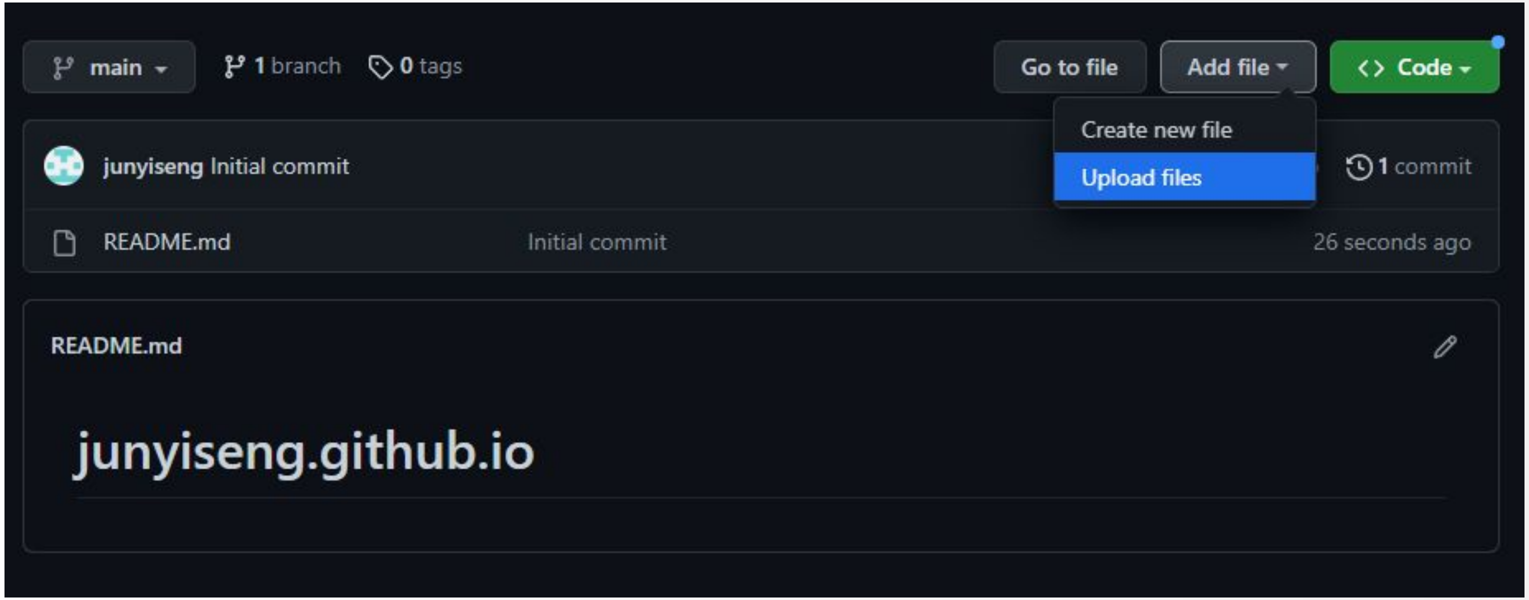
You are creating a public repository in your personal account.

Create repository

1. Click on **New**
2. Enter the relevant information
   - **Repository Name (username.github.io)**
   - **Select "Public"**
3. Click on **Create Repository**

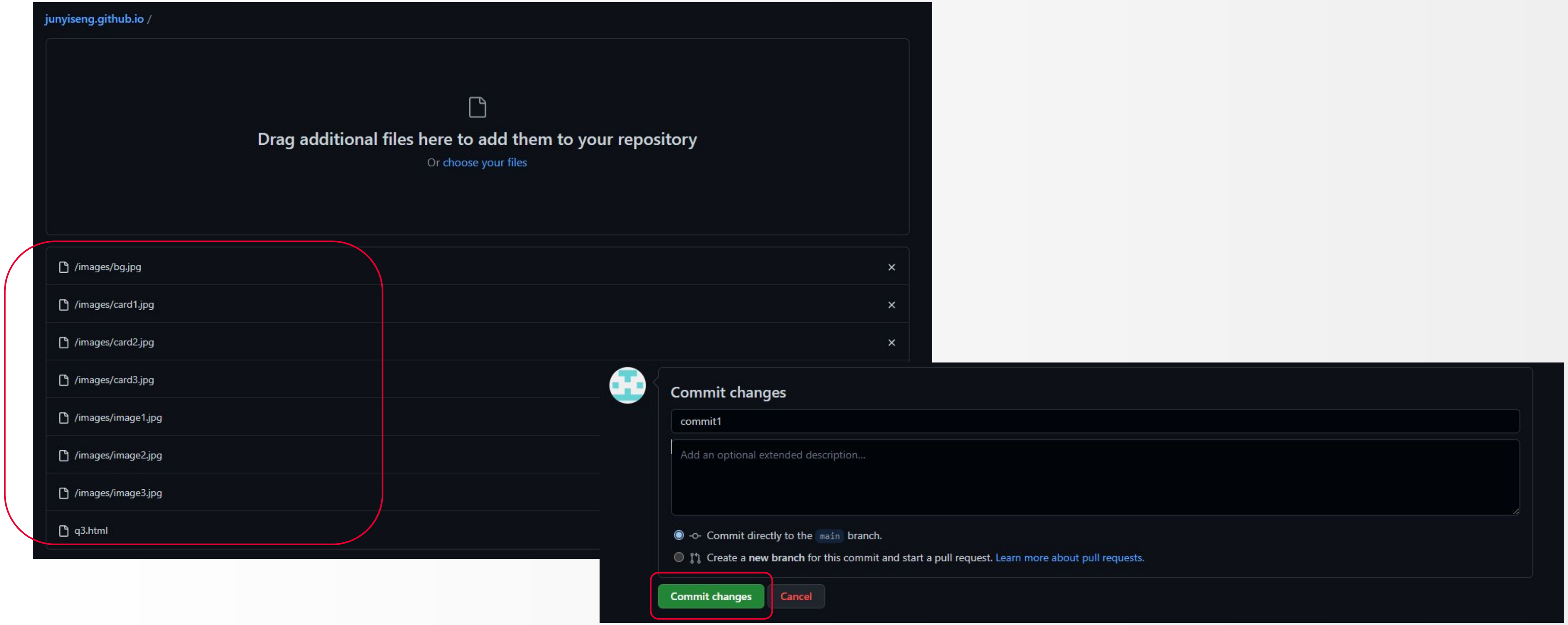
Note: It is a must to follow the repository name of (username.github.io)

Uploading Files to Github



1. On the homepage of the repo, click Add File → Upload Files

Uploading Files to Github



junyiseng.github.io /

Drag additional files here to add them to your repository

Or choose your files

/images/bg.jpg

/images/card1.jpg

/images/card2.jpg

/images/card3.jpg

/images/image1.jpg

/images/image2.jpg

/images/image3.jpg

q3.html

Commit changes

commit1

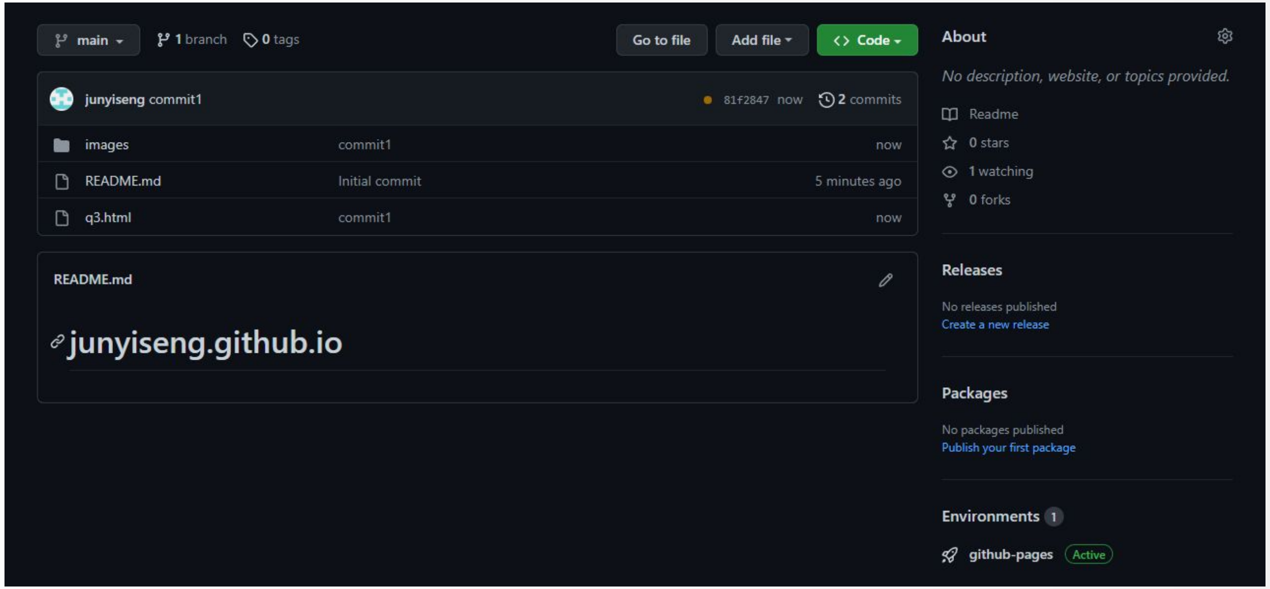
Add an optional extended description...

[● Commit directly to the `main` branch.]

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

1. Drag the files you wish to upload
2. Click on Commit Changes

Uploading Files to Github


main
1 branch
0 tags
Go to file
Add file
Code

junyiseng commit1 · 81f2847 now · 2 commits

images	commit1	now
README.md	Initial commit	5 minutes ago
q3.html	commit1	now

README.md

junyiseng.github.io

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

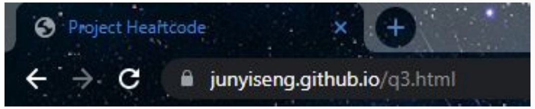
Environments 1

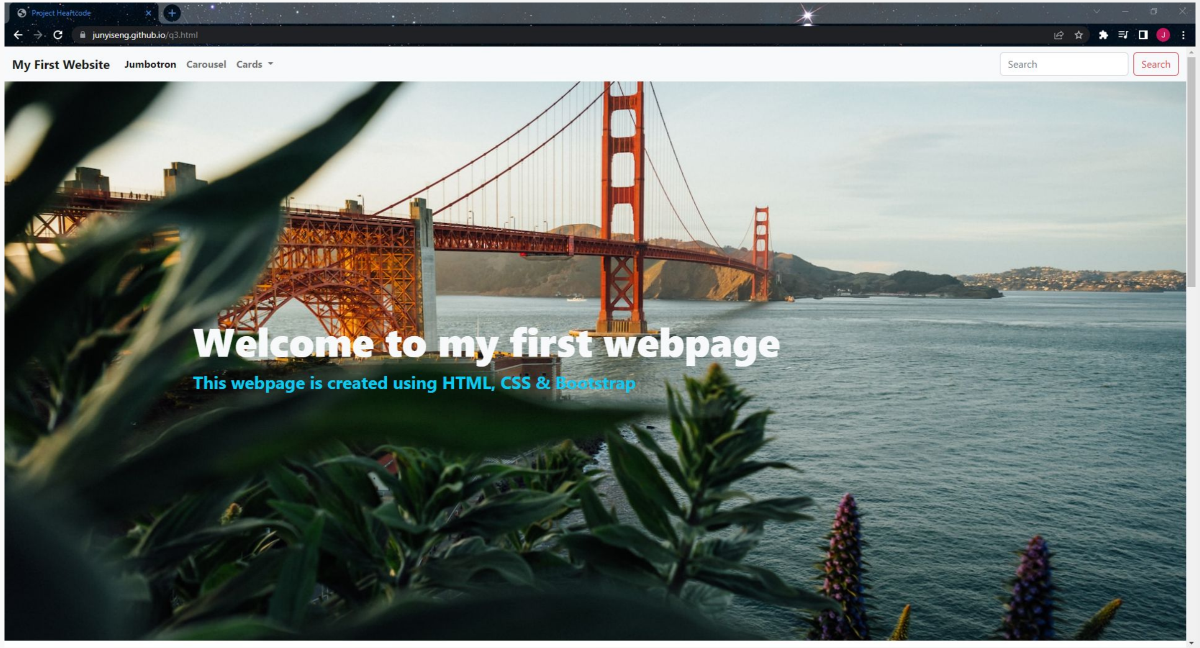
github-pages Active

1. The uploaded files will appear in the main repo page

Accessing the Website

1. On your browser, enter the following url https://username.github.io





Welcome to my first webpage

This webpage is created using HTML, CSS & Bootstrap

Note: It may take up to 10 minutes for the changes to be published on Github Pages

Submit ART

Include Name, Date, Time of taking ART on your ART kit and send the image to this google form! Test before coming daily!

Thank You!

Do you have any questions?