Software Development Course

Unit 1: Front End Web Technologies

Session 10

JavaScript: Introduction to the DOM

Recall.

1. What is the difference between a while loop and a do while loop?
2. Which iterator would you use to run a function on each item in an array?
3. How would you write a basic for loop?
4. What is an infinite loop?
5. How do you avoid an infinite loop?

TechTalent Academy Safeguarding Policy

*"Protecting an adult's right to live in **safety, free from abuse and neglect**. It is about people and organisations working together to **prevent and stop both the risks and experience of abuse or neglect**, while at the same time making sure that the **adult's wellbeing is promoted** including, where appropriate, having regard to their views, wishes, feelings and beliefs in deciding on any action. This must recognise that adults sometimes have complex interpersonal relationships and may be ambivalent, unclear or unrealistic about their personal circumstances."*

If you have a safeguarding concern, please raise this with your tutor or via the safeguarding link on our website:

https://www.techtalent.co.uk/safeguarding-statement

TechTalent's safeguarding lead is: **Max Ruddock**

techtalent academy

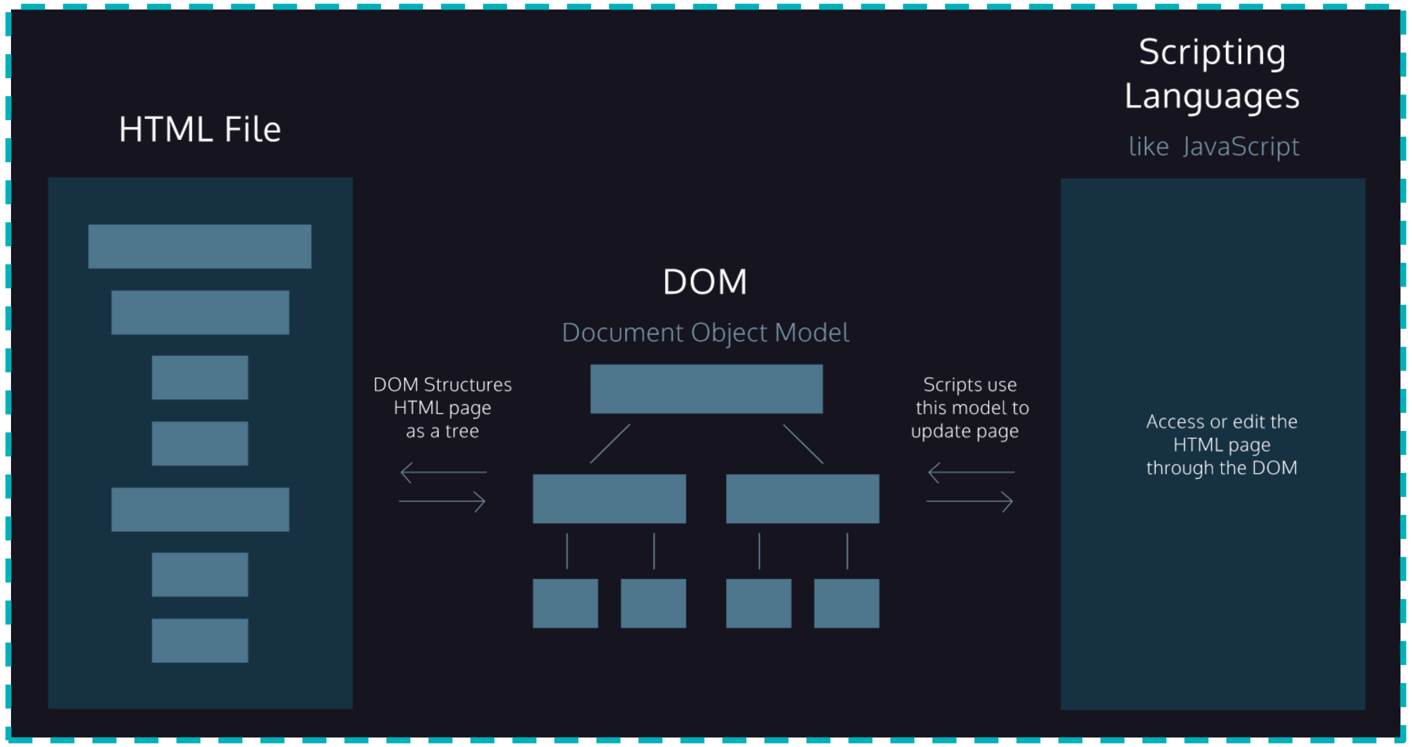
1. Introduction to the DOM

- Understanding and exploring the DOM Model
- Exploring the parent-child relationship to nodes and objects
- Accessing and updating the DOM

The DOM.
The Document Object Model

techtalent academy

- DOM – Document Object Model
- Allows programmers to conceptualise hierarchy.
- Used to access elements on a web page
- DOM is generated by the browser
- DOM allows JavaScript to access, modify, and update the structure of a HTML web page

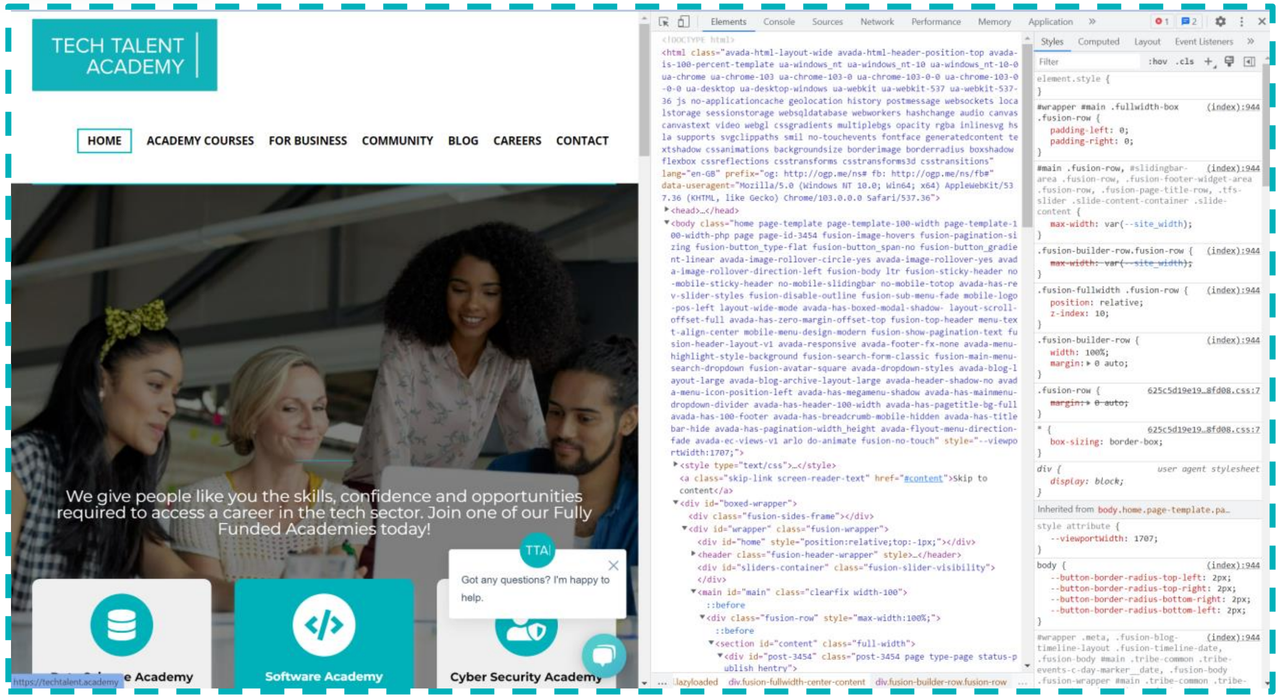

HTML File
DOM
Document Object Model
DOM Structures HTML page as a tree
Scripts use this model to update page
Scripting Languages
like JavaScript
Access or edit the HTML page through the DOM

techtalent academy

Activity.

Accessing and Updating the DOM

- We can use the console to update the DOM of a page in real time
- The DOM is a virtual representation of the code fed into the web browser – it is NOT the original file



We give people like you the skills, confidence and opportunities required to access a career in the tech sector. Join one of our Fully Funded Academies today!

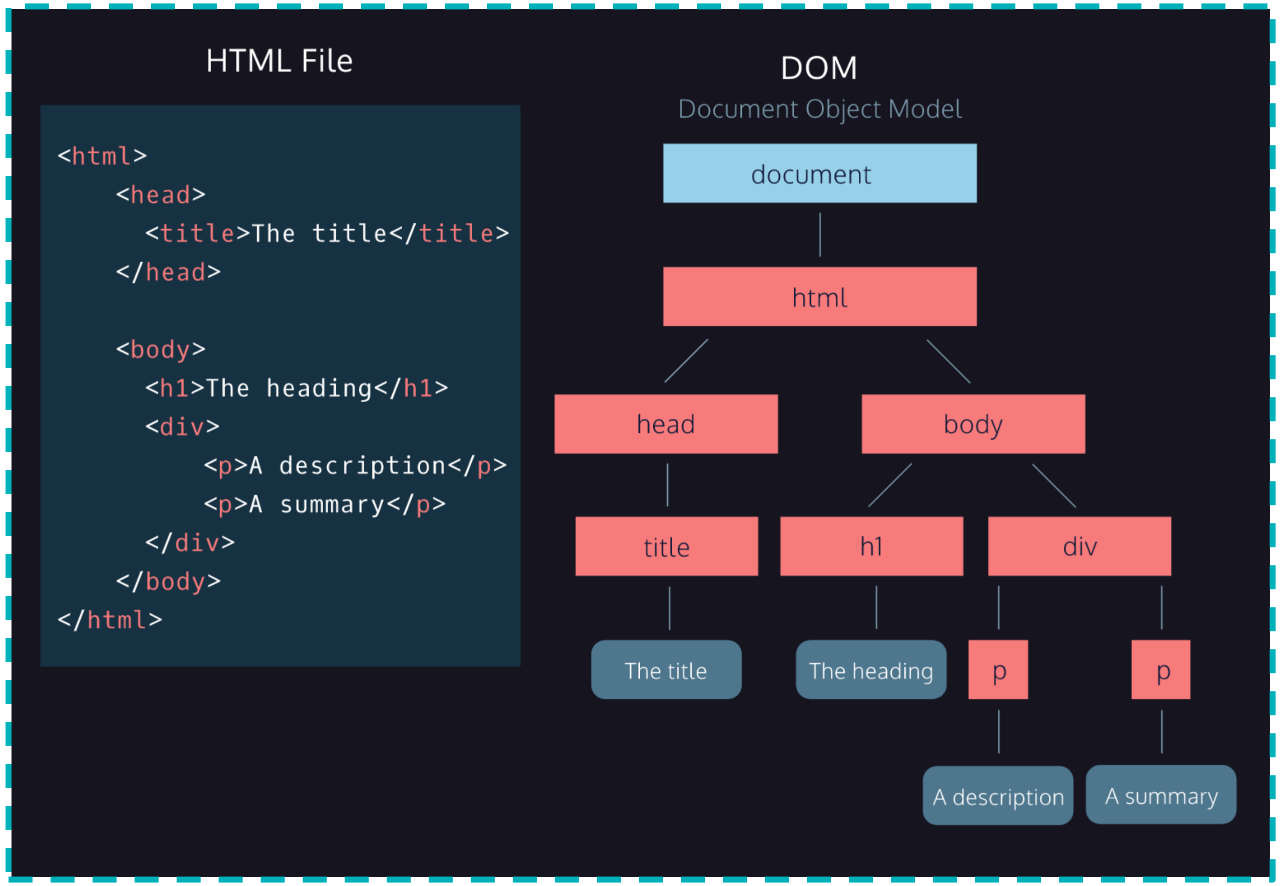
Got any questions? I'm happy to help.

```
<!DOCTYPE html>
<html class="avada-html-layout-wide avada-html-header-position-top avada-is-100-percent-template ua-windows_nt ua-windows_nt-10 ua-windows_nt-10-0 ua-chrome ua-chrome-103 ua-chrome-103-0 ua-chrome-103-0-0 ua-chrome-103-0-0-0 ua-desktop ua-desktop-windows ua-webkit ua-webkit-537 ua-webkit-537-36 js no-applicationcache geolocation history postmessage websockets localstorage sessionstorage websqldatabase webworkers hashchange audio canvas canvastext video webgl cssgradients multiplebgs opacity rgba inlinesvg hsla supports svgclippaths smil no-touchevents fontface generatedcontent textshadow cssanimations backgroundsize borderimage borderradius boxshadow flexbox cssreflections csstransforms csstransforms3d csstransitions" lang="en-GB" prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb#" data-useragent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36">
 ▶<head>…</head>
 ▼<body class="home page-template page-template-100-width page-template-100-width-php page page-id-3454 fusion-image-hovers fusion-pagination-sizing fusion-button_type-flat fusion-button_span-no fusion-button_gradient-linear avada-image-rollover-circle-yes avada-image-rollover-yes avada-image-rollover-direction-left fusion-body ltr fusion-sticky-header no-mobile-sticky-header no-mobile-slidingbar no-mobile-totop avada-has-rev-slider-styles fusion-disable-outline fusion-sub-menu-fade mobile-logo-pos-left layout-wide-mode avada-has-boxed-modal-shadow- layout-scroll-offset-full avada-has-zero-margin-offset-top fusion-top-header menu-text-align-center mobile-menu-design-modern fusion-show-pagination-text fusion-header-layout-v1 avada-responsive avada-footer-fx-none avada-menu-highlight-style-background fusion-search-form-classic fusion-main-menu-search-dropdown fusion-avatar-square avada-dropdown-styles avada-blog-layout-large avada-blog-archive-layout-large avada-header-shadow-no avada-menu-icon-position-left avada-has-megamenu-shadow avada-has-mainmenu-dropdown-divider avada-has-header-100-width avada-has-pagetitle-bg-full avada-has-100-footer avada-has-breadcrumb-mobile-hidden avada-has-titlebar-hide avada-has-pagination-width_height avada-flyout-menu-direction-fade avada-ec-views-v1 arlo do-animate fusion-no-touch" style="--viewportWidth:1707;">
   ▶<style type="text/css">…</style>
    <a class="skip-link screen-reader-text" href="#content">Skip to content</a>
   ▼<div id="boxed-wrapper">
      <div class="fusion-sides-frame"></div>
     ▼<div id="wrapper" class="fusion-wrapper">
        <div id="home" style="position:relative;top:-1px;"></div>
       ▶<header class="fusion-header-wrapper" style>…</header>
        <div id="sliders-container" class="fusion-slider-visibility">
        </div>
       ▼<main id="main" class="clearfix width-100">
          ::before
         ▼<div class="fusion-row" style="max-width:100%;">
            ::before
           ▼<section id="content" class="full-width">
             ▼<div id="post-3454" class="post-3454 page type-page status-publish hentry">
… lazyloaded   div.fusion-fullwidth-center-content   div.fusion-builder-row.fusion-row   …
```

```
element.style {
}
#wrapper #main .fullwidth-box                 (index):944
.fusion-row {
    padding-left: 0;
    padding-right: 0;
}
#main .fusion-row, #slidingbar-               (index):944
area .fusion-row, .fusion-footer-widget-area
.fusion-row, .fusion-page-title-row, .tfs-
slider .slide-content-container .slide-
content {
    max-width: var(--site_width);
}
.fusion-builder-row.fusion-row {              (index):944
    ~~max-width: var(--site_width);~~
}
.fusion-fullwidth .fusion-row {               (index):944
    position: relative;
    z-index: 10;
}
.fusion-builder-row {                         (index):944
    width: 100%;
    margin: ▶ 0 auto;
}
.fusion-row {                   625c5d19e19…8fd08.css:7
    ~~margin: ▶ 0 auto;~~
}
* {                             625c5d19e19…8fd08.css:7
    box-sizing: border-box;
}
div {                            user agent stylesheet
    display: block;
}
Inherited from body.home.page-template.pa…
style attribute {
    --viewportWidth: 1707;
}
body {                                        (index):944
    --button-border-radius-top-left: 2px;
    --button-border-radius-top-right: 2px;
    --button-border-radius-bottom-right: 2px;
    --button-border-radius-bottom-left: 2px;
}
#wrapper .meta, .fusion-blog-                 (index):944
timeline-layout .fusion-timeline-date,
.fusion-body #main .tribe-common .tribe-
events-c-day-marker__date, .fusion-body
.fusion-wrapper #main .tribe-common .tribe-
```

The DOM. Nodes

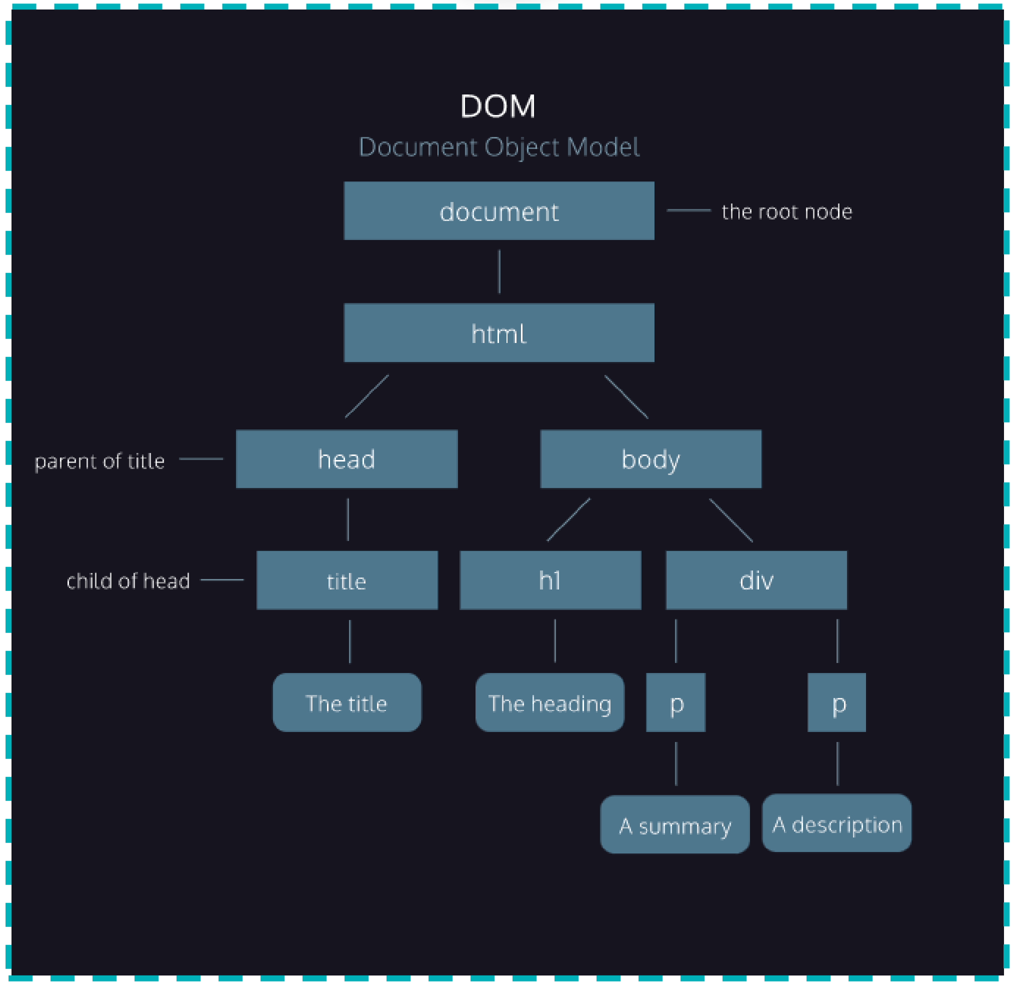
- Top most node is referred to as the 'root node'
- Sharp-edged rectangles are element nodes
- Rounded edge rectangles are text nodes
- When modifying a web page, the script will mostly interact with the DOM elements (occasionally text nodes)


HTML File

```
<html>
    <head>
      <title>The title</title>
    </head>

    <body>
      <h1>The heading</h1>
      <div>
          <p>A description</p>
          <p>A summary</p>
      </div>
    </body>
</html>
```

The DOM.
Parent-Child Relationship

- A parent node is any node that is a direct ancestor of another node
- A child node is a direct descendant of another node, called the parent node



techtalent
academy

2. The Script Tag

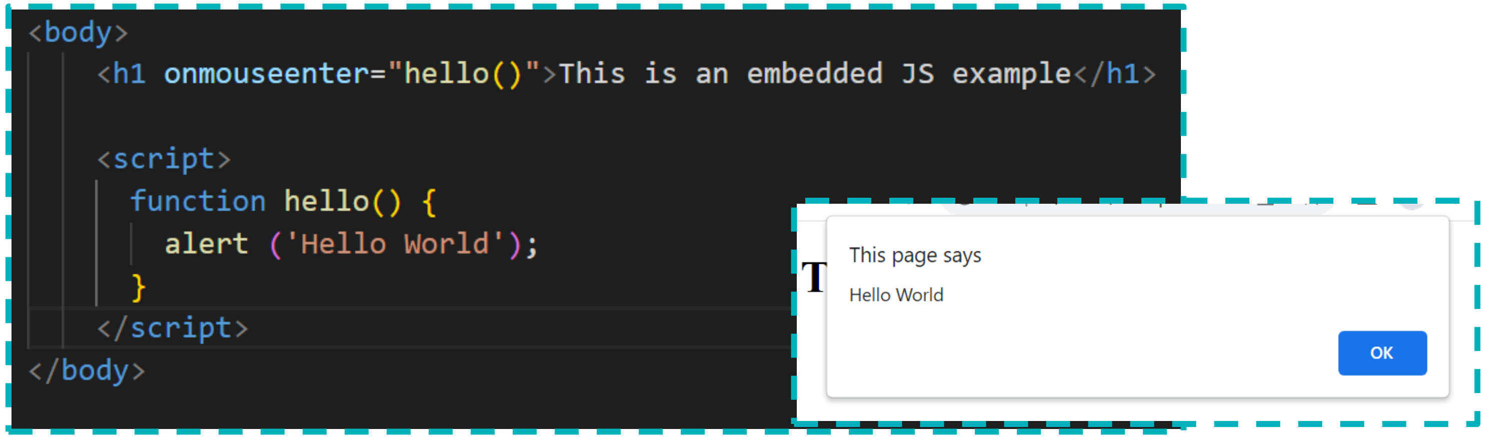
- Exploring the different methods to include JavaScript into a HTML webpage
- Using HTML buttons to add JavaScript functionality to a webpage

Scripts.

Inline and Internal JavaScript

techtalent academy

- Like CSS, you can have internal, external and inline JavaScript
- The <script></script> tag allows JavaScript code to be added to a HTML file
- Works like most tag within HTML
- Can be placed anywhere within a HTML file
- By default the web browser will load the script when loading the HTML file (top to bottom)

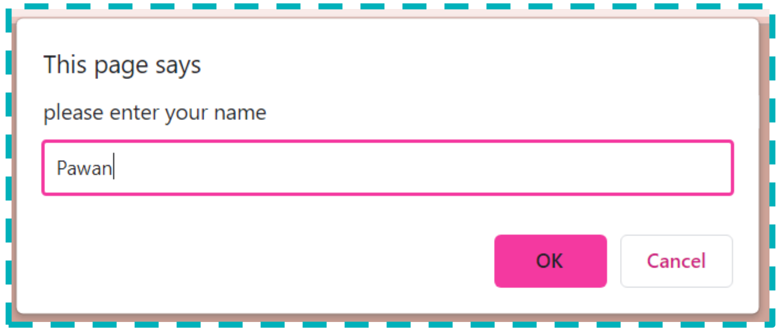


```
<body>
    <h1 onmouseenter="hello()">This is an embedded JS example</h1>

    <script>
      function hello() {
        alert ('Hello World');
      }
    </script>
</body>
```

```
This page says
Hello World
```

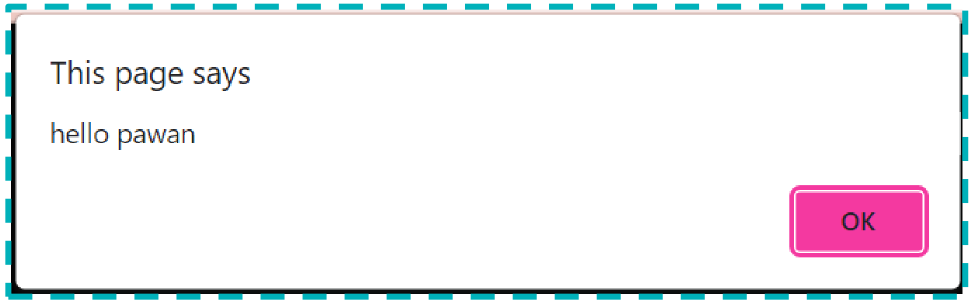
Scripts.
prompt() and alert()

- Console is for developers
- We use prompt() and alert() to get and present information to users
- (not best practice but good to know)
- Information gathered from prompt() can be stored as a variable

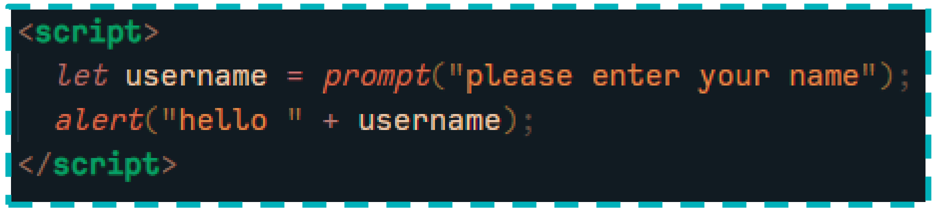

This page says

please enter your name

Pawan


This page says

hello pawan

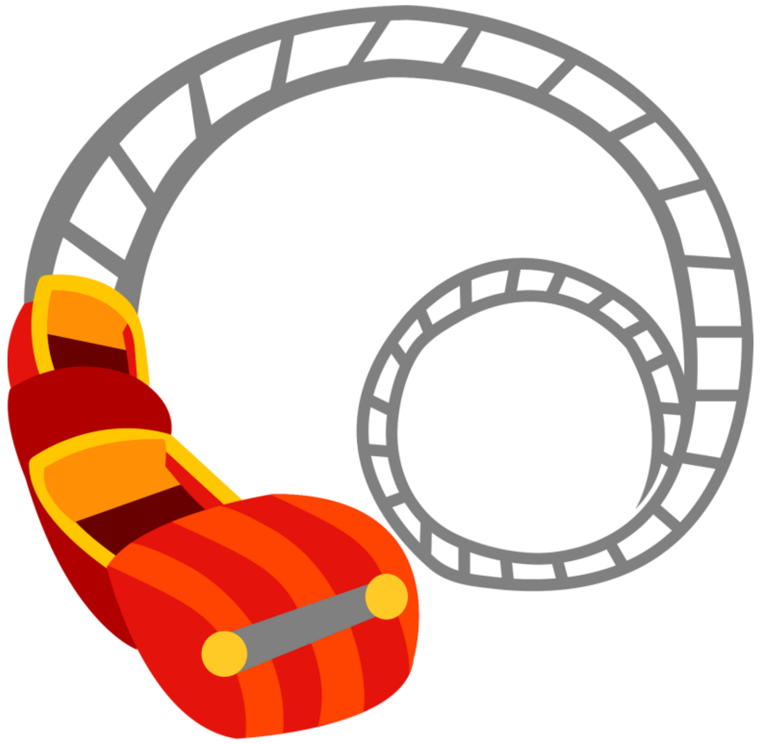


```
<script>
  let username = prompt("please enter your name");
  alert("hello " + username);
</script>
```

Activity.
User Loops

- Create a loop that will ask the user for number and then print out the following sentence that number of times:
  - “The value of i is: i” where the second i is replaced with the value e.g. ‘The value of i is: 1” etc

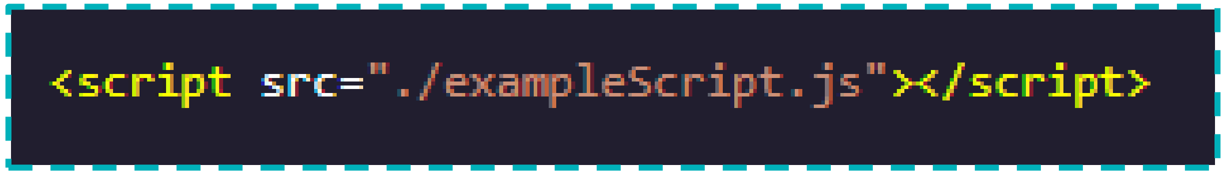
Challenge: Create a loop that will print the number of times the loop has been executed, and only exit when the user inputs the word “stop”.

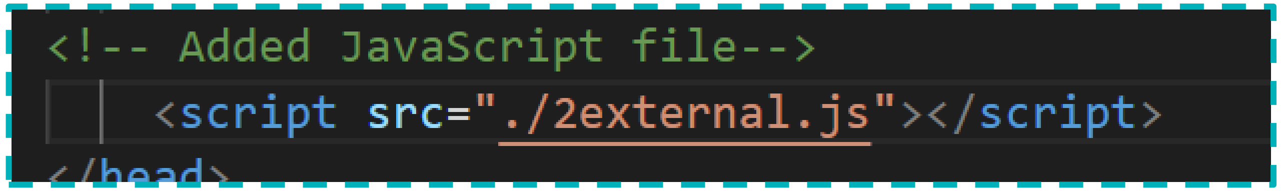


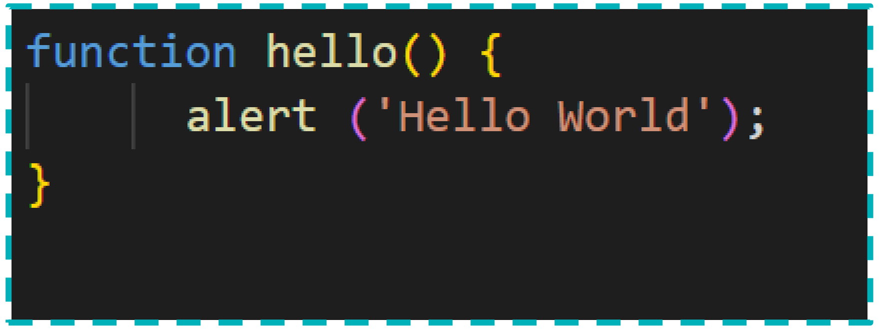
techtalent academy

Scripts.
External scripts

- SoC (Separation of Concerns)
  - Having JavaScript written in-line with HTML can be messy and can cause concerns when the code does not run as intended.

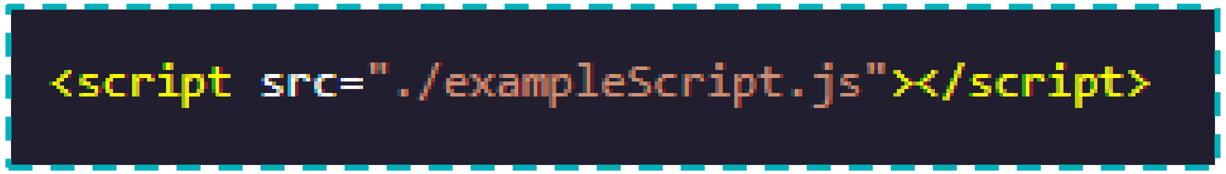

```
<script src="./exampleScript.js"></script>
```


```
<!-- Added JavaScript file-->
    <script src="./2external.js"></script>
</head>
```

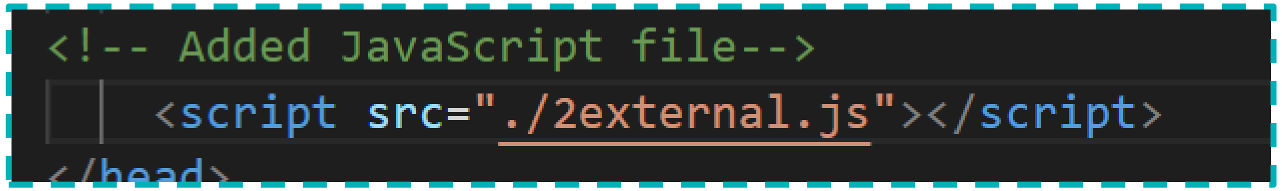

```
function hello() {
    alert ('Hello World');
}
```

Scripts.
External scripts

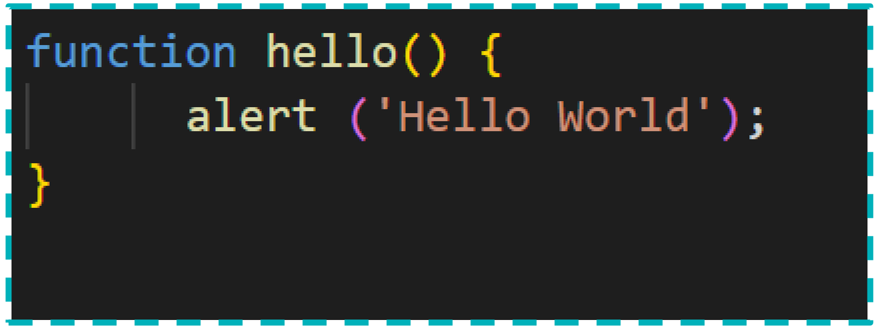
- SoC (Separation of Concerns)
  - Having JavaScript written in-line with HTML can be messy and can cause concerns when the code does not run as intended.



```
<script src="./exampleScript.js"></script>
```



```
<!-- Added JavaScript file-->
    <script src="./2external.js"></script>
</head>
```



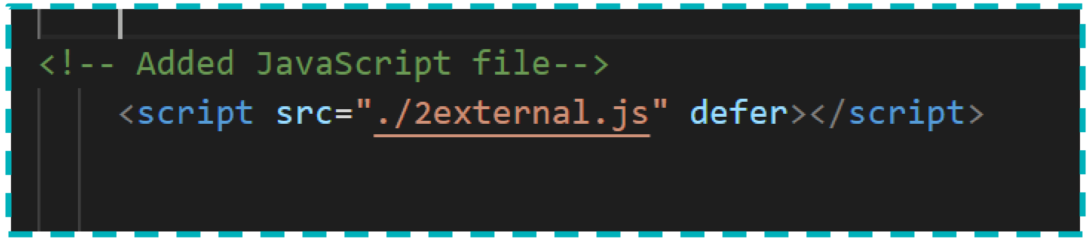
```
function hello() {
    alert ('Hello World');
}
```

Scripts.
Parsing

- All browsers come equipped with HTML parsers that help render the elements accordingly
- Elements, including the <script> element, are by default parsed in the order they appear in the HTML file
- When the HTML parser encounters a <script> element, it loads the script then executes its contents before parsing the rest of the HTML
- Important points:
  - The HTML parser does NOT process the next element in the HTML file until it loads and executes the <script> element, thus leading to a delay in load time and often resulting in poor user experience
  - Scripts are loaded sequentially, so if one script is dependent on another script, they should be placed in that very order inside the HTML

Scripts.
Defer Attribute

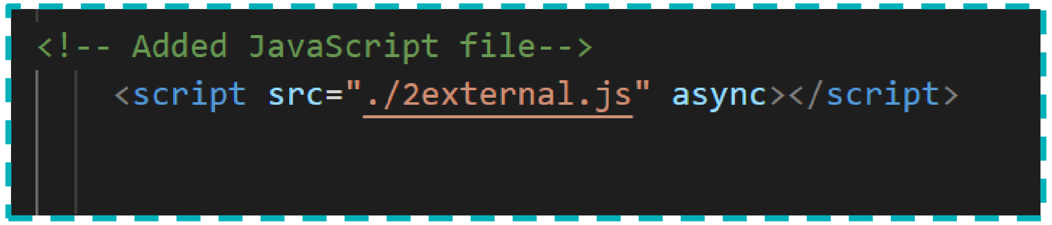
- Adding the defer attribute specifies scripts should be executed after the HTML file is completely parsed
- Added inside the open <script> tag
- HTML parser loads the script but defers the actual execution until all HTML elements have been parsed.
- Used when script requires interaction with the DOM
- Defer attribute is only for external scripts



```
<!-- Added JavaScript file-->
    <script src="./2external.js" defer></script>
```

Scripts.
Async Attribute

- Adding the async attribute specifies scripts should be downloaded and parsed in parallel along with the HTML file
- The script will execute as soon as it is available.
- Added inside the open <script> tag
- Defer attribute is only for external scripts



```
<!-- Added JavaScript file-->
    <script src="./2external.js" async></script>
```

Scripts.
Best Practice

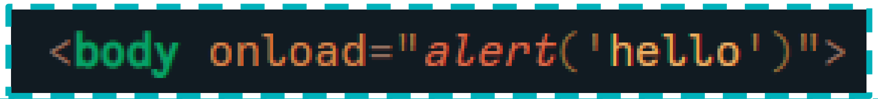
- Although we can use the <script> tag for simple code, best practice is that we store our scripts externally.
- To do this, create a .js file locally (usually script.js) and include it within the <head> of our document:

```
<script src="script.js"></script>
```

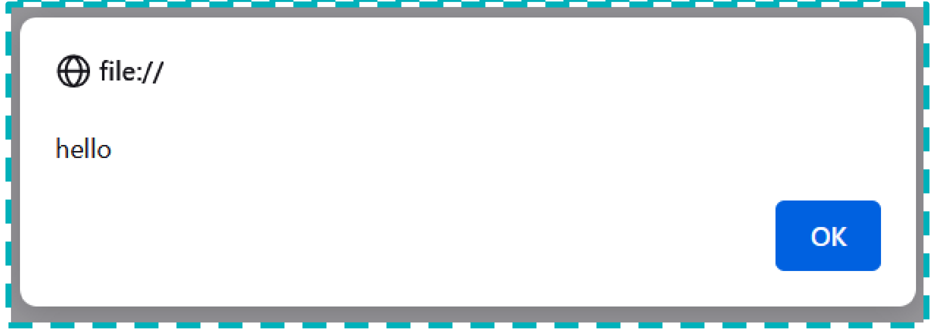
- Multiple script files should be included in its own folder e.g. 'Javascript', which would change the 'src' path too: Javascript/script.js

Scripts.
Events

- An event is something that happens to an element
- JavaScript 'listens' out for these events and then 'reacts' to them
- For example <body onload="alert('hello!')"> will execute a pop-up when the onload event occurs (not recommended however!)
- Some of the most common events are: onmouseover, onmouseout, onclick, onchange etc, but there are lots more: https://www.w3schools.com/jsref/dom_obj_event.asp



```
<body onload="alert('hello')">
```

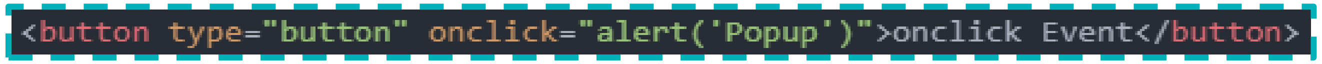


file://

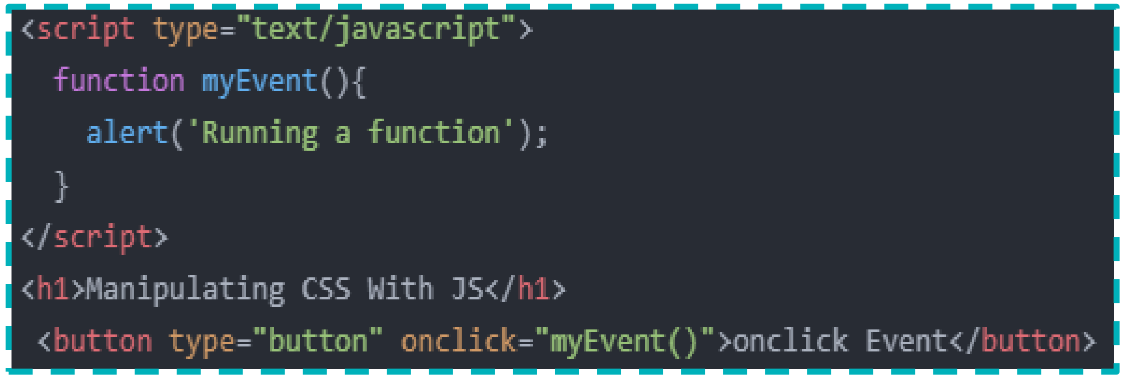
hello

Scripts.
onclick events

- Let's start by looking at adding in some buttons to a webpage and associating some functionality to that button with JavaScript.
- The code on the right has two properties associated with it: it's type and an onclick event. We have set the event to create an alert dialogue box that says the word "popup" when the button is pressed.
- *Mini-task: create a button that pops up an alert box*



```html
<button type="button" onclick="alert('Popup')">onclick Event</button>
```



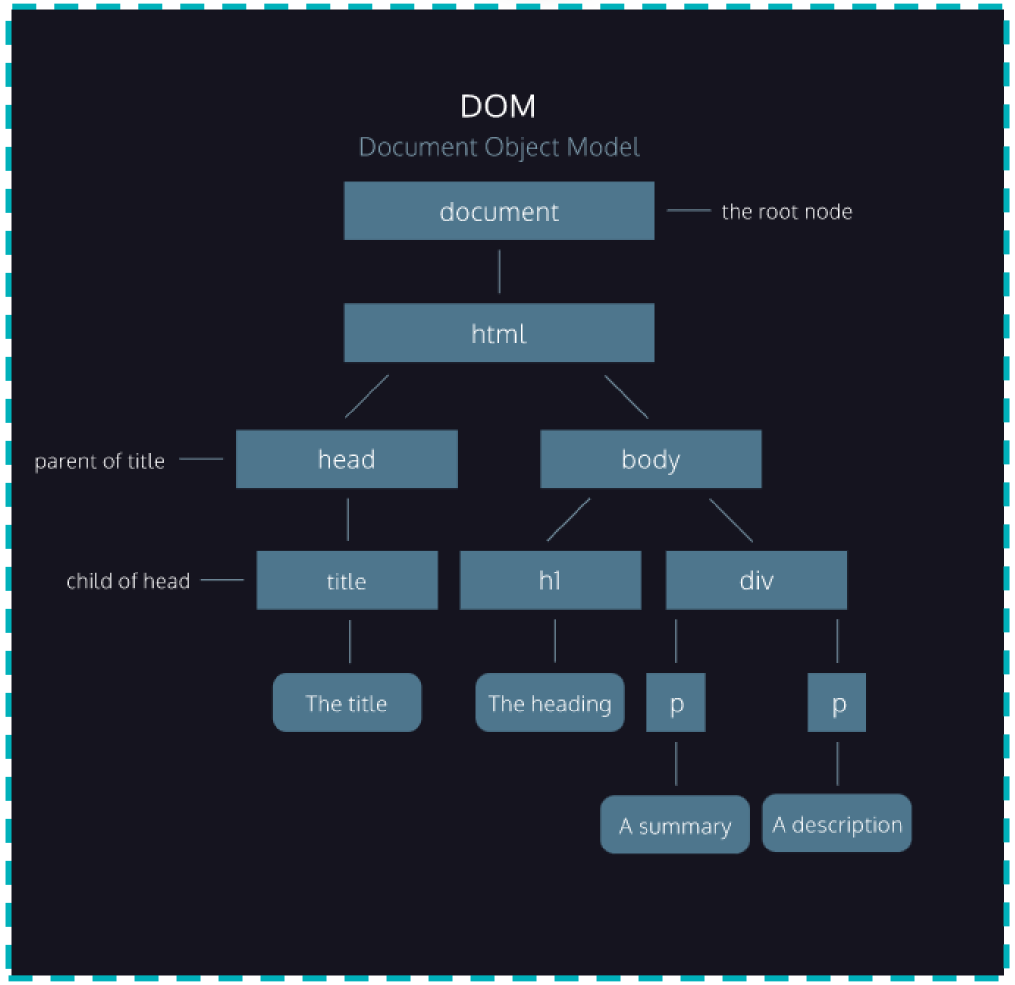
```html
<script type="text/javascript">
  function myEvent(){
    alert('Running a function');
  }
</script>
<h1>Manipulating CSS With JS</h1>
 <button type="button" onclick="myEvent()">onclick Event</button>
```

3. Grabbing Elements

- Exploring the different methods to grab elements from the DOM

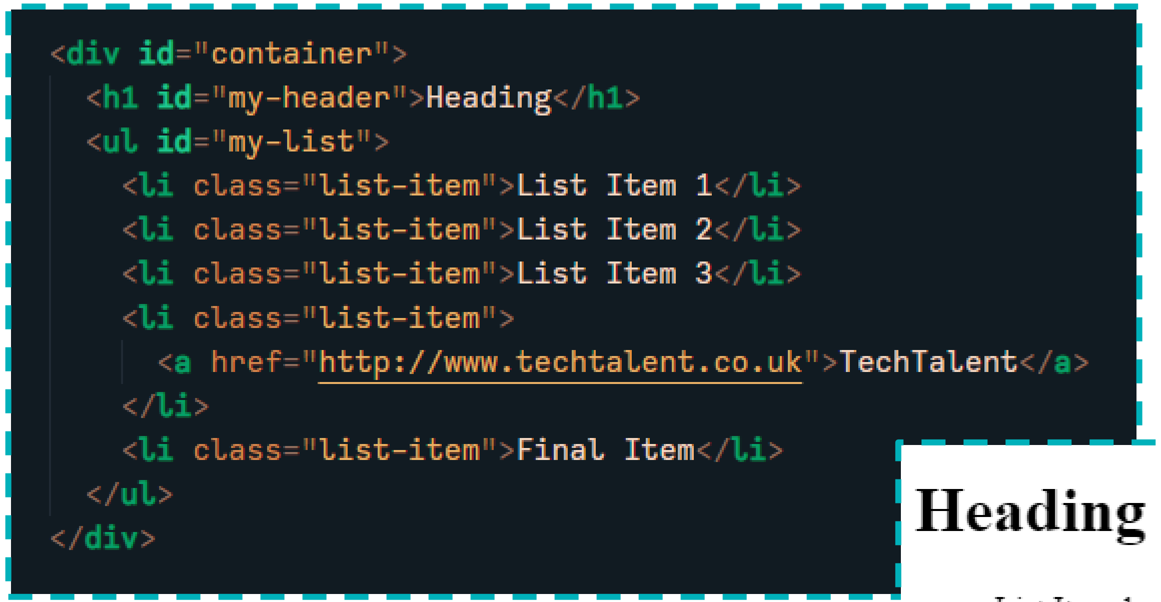
Grabbing Elements.
DOM Tree

- A reminder that elements are arranged in a tree structure, with each element being a **node**
- There are different ways of ‘grabbing’ these different items

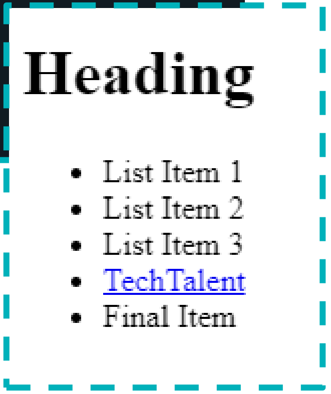


Grabbing Elements.
Code Along

- A reminder that elements are arranged in a tree structure, with each element being a **node**
- There are different ways of 'grabbing' these different items
- Open the '3.1 Grabbing Elements Task' HTML file in a browser and open up the **console**.



```html
<div id="container">
  <h1 id="my-header">Heading</h1>
  <ul id="my-list">
    <li class="list-item">List Item 1</li>
    <li class="list-item">List Item 2</li>
    <li class="list-item">List Item 3</li>
    <li class="list-item">
      <a href="http://www.techtalent.co.uk">TechTalent</a>
    </li>
    <li class="list-item">Final Item</li>
  </ul>
</div>
```



Heading

- List Item 1
- List Item 2
- List Item 3
- TechTalent
- Final Item

Grabbing Elements.
Selecting HTML Activity

- **Select the following items:**
- The heading
- The anchor tag
- The last item on the list
- The array of the list items

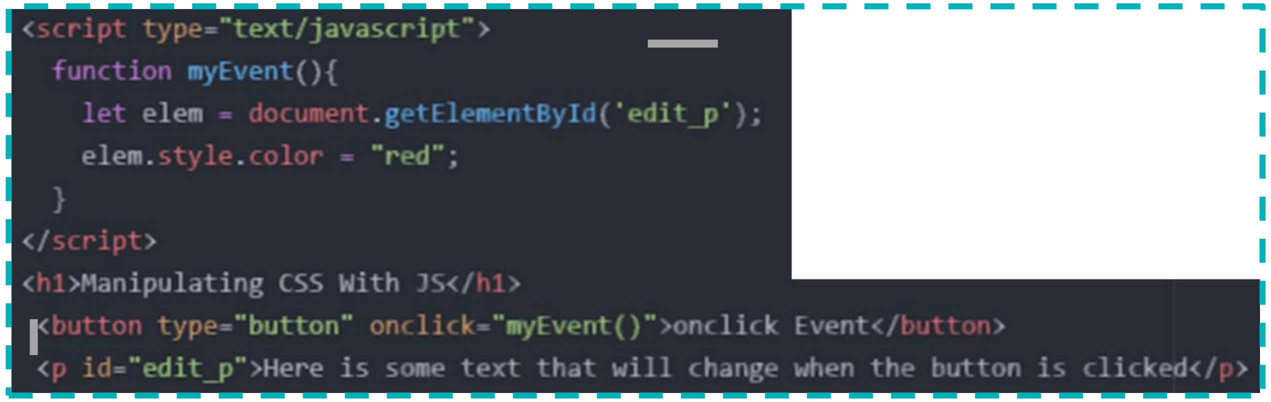
techtalent academy

Grabbing Elements.
Changing content

- Step 1: grabbing an element. You need to store it as a variable in order to do things with it
- Step 2: calling a method on it to change it
- To be able to access elements of our webpage, we need to have **identifiers** on those elements.

```
const myBody = document.getElementByTagName('body');
const myTitle = document.getElementById('title');
```

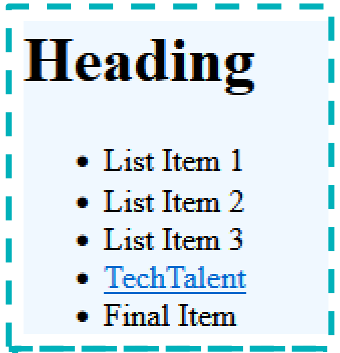
```
myBody[0].innerHTML = "<p>new HTML to replace old HTML</p>";

myTitle.textContent = "new Title";
```



```
<script type="text/javascript">
  function myEvent(){
    let elem = document.getElementById('edit_p');
    elem.style.color = "red";
  }
</script>
<h1>Manipulating CSS With JS</h1>
<button type="button" onclick="myEvent()">onclick Event</button>
<p id="edit_p">Here is some text that will change when the button is clicked</p>
```

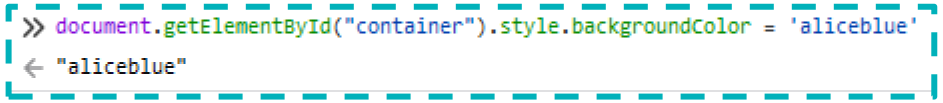
Grabbing Elements.
Changing content

- .textContent – accepts a string; changes the text
- .innerHTML – accepts a string; changes the HTML
- .style – any CSS property can be changed using dot notation and camelCase

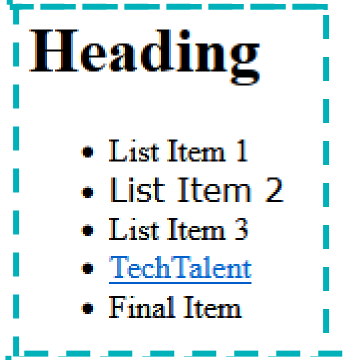


Heading

- List Item 1
- List Item 2
- List Item 3
- TechTalent
- Final Item

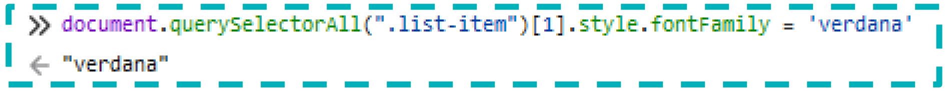


```
>> document.getElementById("container").style.backgroundColor = 'aliceblue'
← "aliceblue"
```

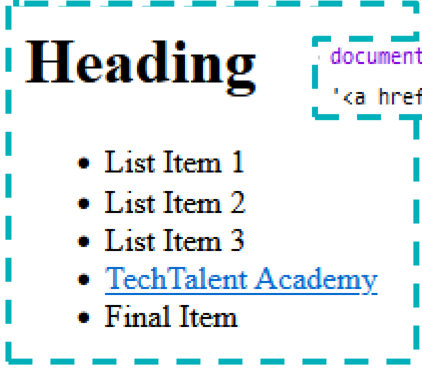


Heading

- List Item 1
- List Item 2
- List Item 3
- TechTalent
- Final Item

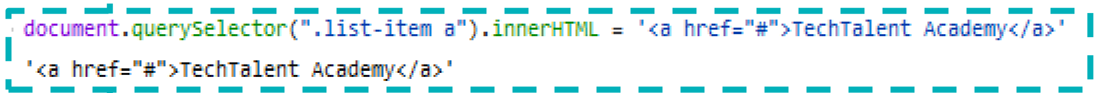


```
>> document.querySelectorAll(".list-item")[1].style.fontFamily = 'verdana'
← "verdana"
```



Heading

- List Item 1
- List Item 2
- List Item 3
- TechTalent Academy
- Final Item



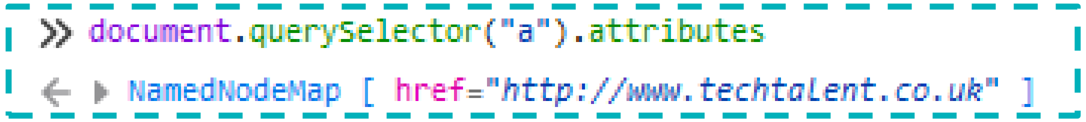
```
document.querySelector(".list-item a").innerHTML = '<a href="#">TechTalent Academy</a>'
'<a href="#">TechTalent Academy</a>'
```

Grabbing Elements.
Changing CSS

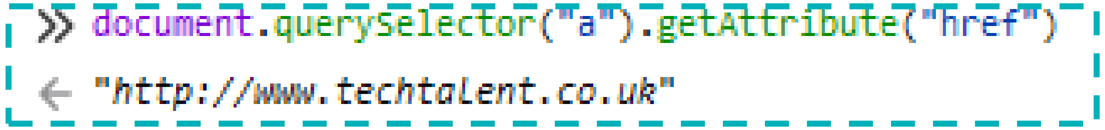
- On this webpage, create 3 buttons that will activate different methods to change the style of your webpage.
- This page holds a complete list of accessible styles. What else can you change? https://www.w3schools.com/jsref/dom_obj_style.asp
- **Challenge**: You can also create new elements. Research and add another element to the list

Grabbing Elements.
Changing attributes

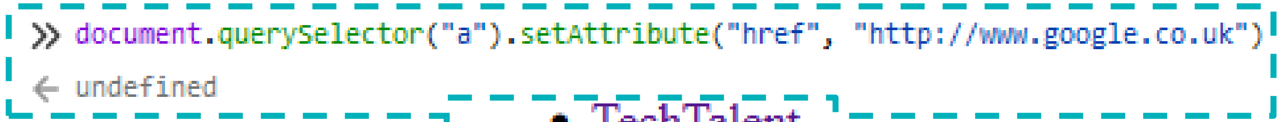
- .attributes – gives a list of elements attached to an element
- .getAttribute(“”) – gets back values
- .setAttribute(“”, “”) – takes two parameters: what attribute you want to access, and what you want to set it to



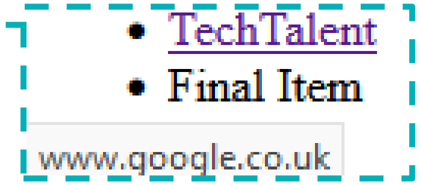
```
>> document.querySelector("a").attributes
<- ▶ NamedNodeMap [ href="http://www.techtalent.co.uk" ]
```



```
>> document.querySelector("a").getAttribute("href")
<- "http://www.techtalent.co.uk"
```



```
>> document.querySelector("a").setAttribute("href", "http://www.google.co.uk")
<- undefined
```

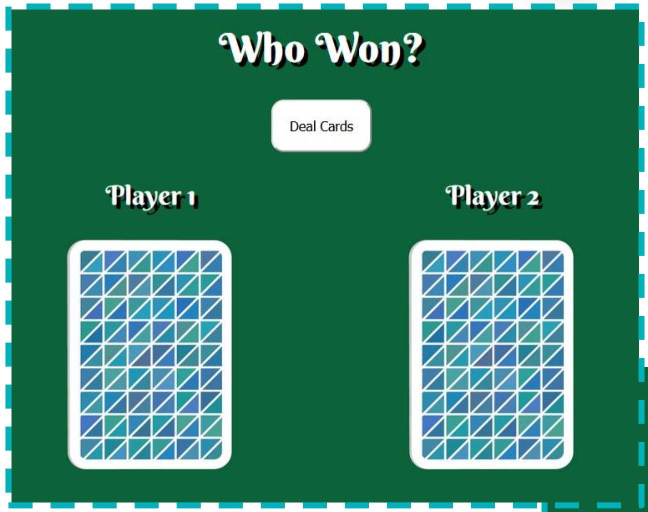


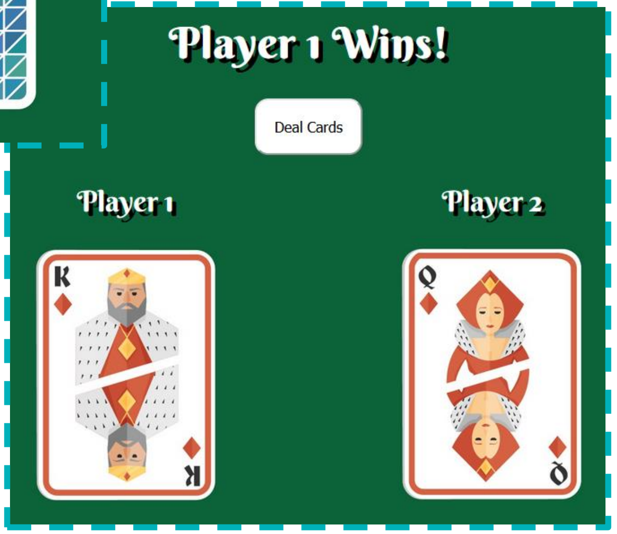
- TechTalent
- Final Item

www.google.co.uk

Activity.
Playing Cards

- You have been given some starter code to create a playing card game.
- Every time you refresh the screen, a new card – either a Jack, Queen, King, Ace or Joker – is chosen.
- Whoever draws the highest card, wins.
- There is a README file in the starter code with further instructions, and the completed code, too.



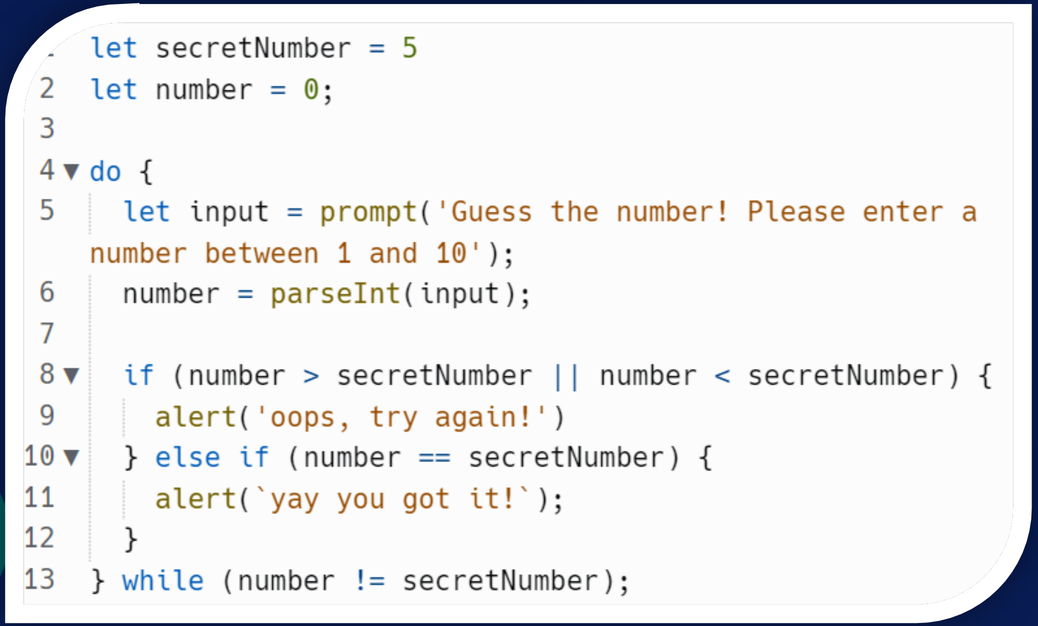


techtalent
academy

Review.

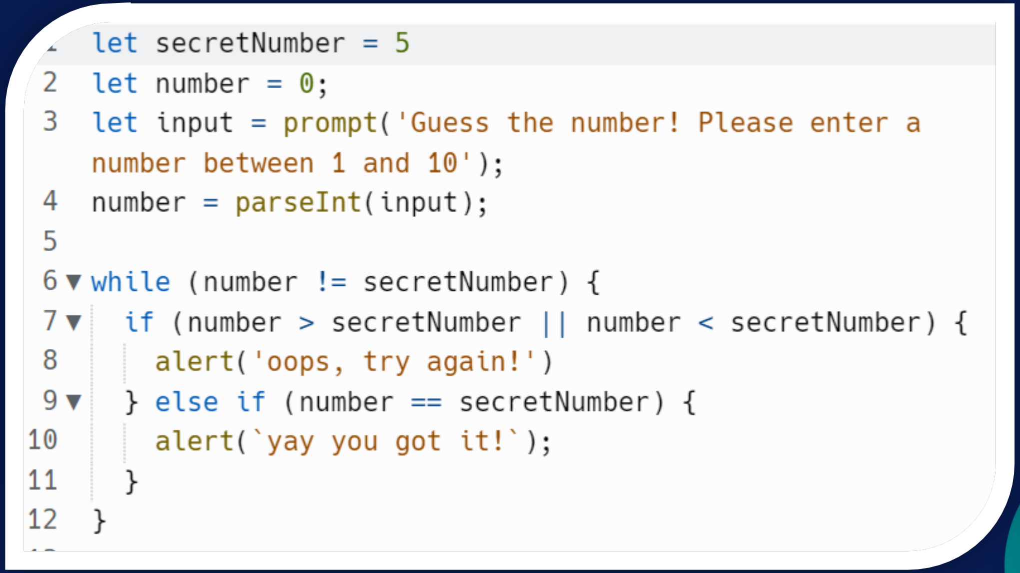
Which of the following pieces of code will run correctly?

a)



```
let secretNumber = 5
let number = 0;

do {
  let input = prompt('Guess the number! Please enter a number between 1 and 10');
  number = parseInt(input);

  if (number > secretNumber || number < secretNumber) {
    alert('oops, try again!')
  } else if (number == secretNumber) {
    alert(`yay you got it!`);
  }
} while (number != secretNumber);
```

b)



```
let secretNumber = 5
let number = 0;
let input = prompt('Guess the number! Please enter a number between 1 and 10');
number = parseInt(input);

while (number != secretNumber) {
  if (number > secretNumber || number < secretNumber) {
    alert('oops, try again!')
  } else if (number == secretNumber) {
    alert(`yay you got it!`);
  }
}
```

techtalent academy

Where Talent Meets Tech

Birmingham | 6 Brindley Place, B1 2JB
Bristol | 40 Berkley Square, BS8 1HP
London | 77-79 New Cavendish Street, W1W 6XB

Website: techtalent.co.uk
Email: hello@techtalent.academy

techtalent