

HTML, CSS, and JS

BRINGING IT ALL TOGETHER

1

Bringing It All Together

- Now that we have a good understanding of the basic concepts of:
 - HTML – The content of our website
 - CSS – The style of our website
 - JavaScript – The functionality of coding
- It is time to bring all these concepts and together.
- Today, we will be looking at some further concepts of Web Development and bringing all of these concepts under one application.



2

HMTL Buttons

- Let's start by looking at adding in some buttons to a webpage and associating some functionality to that button with JavaScript.

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

- The code above 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.
- This is what we call in-line JavaScript.

3

onclick Event

- An onclick event will, as you might have guessed, execute a bit of JavaScript when the element has been clicked.
- We can access various other code blocks with this onclick event too. The example below shows the function "myEvent()" being called when the button is pressed.

Script block

```
<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>
```

4

Script Tags

- As you saw in the previous code example, a new tag has been introduced, the `<script>` tag.
- Inside this tag we are able to house various JavaScript code in any way we want.
- Do note, however, that this code will not execute on it's own! It will need some form of event to kick-start the execution such as onclick.

```
document.getElementById(div).innerHTML = errEmail;
else if (i==2)
{
var atpos=inputs[i].indexOf('@');
var dotpos=inputs[i].lastIndexOf('.');
if (atpos<1 || dotpos<atpos+2 || dotpos<inputs[i].length-1)
document.getElementById('errEmail').innerHTML = "Invalid email address";
else
document.getElementById(div).innerHTML = "Valid email address";
}
```

5

Using Scripts to Change CSS

- To be able to access elements of our webpage, we need to have identifiers on those elements.

```
<script type="text/javascript">
  function myEvent(){
    //alert('Running a function');
    document.getElementById('edit_p').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>
```


Get the element by it's
ID

ID of the paragraph

6




7

 10-15 Minutes

Task 1 – Webpage Elements

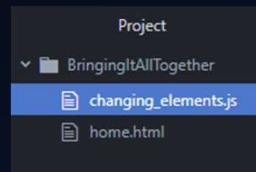
- Create a simple, new webpage that we will build on throughout this session.
- On this webpage, create 3 buttons that will activate **different methods** to change the style of your webpage.
- You can replicate what was shown in this tutorial, but a complete list of accessible styles can be found here:
 - https://www.w3schools.com/jsref/dom_obj_style.asp



8

Scripts – Best Practice

- The `<script>` tag is useful for simple code that needs to be executed, although best practice dictates that we store our scripts separately from our website.
- To do this, we must create a .js file within our website and 'include' it within the `<head>` of our document.



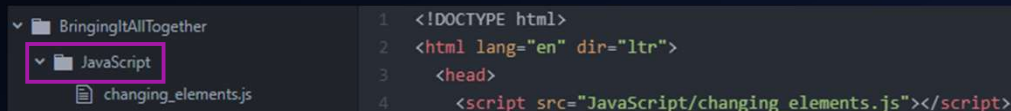
9

Scripts – Best Practice

- We can then move all our code into the .js file and create an `<script>` tag in our header pointing to the file.

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

- Taking this principle further, we should store all our .js files within it's own folder. This will change the 'src' path too.



10



3-5 Minutes

Task 2 – A Better Structure

- Change your project structure to have a folder for your JavaScript files.
- Move your existing JavaScript code to this new file.
- Move your `<script>` tag to the top of the `<head>` tag.
- Change your `<script>` tag to point to the new folder/file location.

```
▼ BringingItAllTogether
  ▼ JavaScript
    changing_elements.js
```

```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <script src="JavaScript/changing_elements.js"></script>
```

11

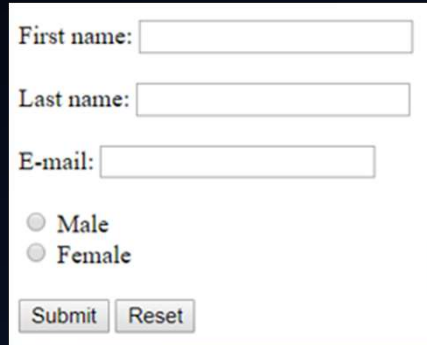
Any Questions So Far?



12

Forms

- Forms are a way of interacting with the user. We can ask the user for their input and, depending on their input, enact various actions and responses.



First name:

Last name:

E-mail:

☐ Male
☐ Female

13

Forms

- A form starts with the `<form>` tag, but it will need some additional properties to be able to do anything with that form data. In this instance, we will look at the 'onsubmit' property.

```
<form onsubmit="return analyseForm()">
```

'return' keyword

Method that will be execute
when the form is submitted

14

Forms – Input

- You have previously seen a `<label>` tag within a form:

```
<label for="fname">First name:</label><br>
```

- Accompanying this label will be an text input field which will allow the user to type their value into it:

```
<input type="text" id="fname" value=""><br>
```

- And finally, we will need a submit button to allow our code to evaluate the form data:

```
<input type="submit" name="submitForm" value="Submit">
```

15

Forms – JavaScript

- This will give use a very basic form layout.



- Finally, the code that is associated with this form's 'onsubmit' property is simple getting the value from the first name field and pushing an alert window to the user saying "hello <fname>".


```
function analyseForm(){
  window.alert("Hello " + document.getElementById('fname').value);
}
```

Accessing the value

16



17



10-15 Minutes

Task 3 – Forms

- Create a form on your homepage that asks the user to input:
 - First name
 - Last name
 - Email address
- Create a function that will execute when the form is submitted.
 - Have this function check to see if there is a value in all 3 fields.
 - Also have this function send an alert message **if all 3 values are present**:
 - “Hello, ‘first name’ ‘last name’. Your email ‘email address’ has been added to the mailing list”
- **STRETCH CHALLENGE**: If you complete the above and have some time left, edit your form to have style attributes applied to it:
https://www.w3schools.com/css/tryit.asp?filename=trycss_form_responsive

18

Forms – Radio Buttons

- Radio buttons forces the user to select 1 option from a list of options.
- We start by using the same `<input>` tag as the text field, but declaring the type as 'radio'.

```
<input type="radio"
```

19

Forms – Radio Buttons

- Each radio button should have an ID as we may want to access the element directly. We should also give the radio button a name.

```
<input type="radio" id="radio1" name="name_of_radio_collection"
```

- This name will be used on all radio buttons within the same collection. If the names don't match on your other radio buttons, they will be treated separately!

20

Forms – Radio Buttons

- We should finish our radio button off with a value. This will not be the text that is displayed on screen, but rather, something that we can associate with the radio button, accessed via JavaScript.

```
<input type="radio" id="radio1" name="name_of_radio_collection" value="Radio Option 1">
```

- Finally, we need to give the radio button a label so that the user knows what they are clicking.

```
<label for="name_of_radio_collection">Radio Option 1</label>
```

This must be the same as the 'name' of the radio button collection

21

Forms – Radio Buttons

- As radio buttons are considered a single selection from a list of choices, more than 1 radio button is necessary.

```
<input type="radio" id="radio1" name="name_of_radio_collection" value="Radio Option 1">
<label for="name_of_radio_collection">Radio Option 1</label><br>
<input type="radio" id="radio2" name="name_of_radio_collection" value="Radio Option 2">
<label for="name_of_radio_collection">Radio Option 2</label><br>
<input type="radio" id="radio3" name="name_of_radio_collection" value="Radio Option 3">
<label for="name_of_radio_collection">Radio Option 3</label>
```

- When the page loads, there won't be a default selected.

```

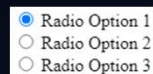
○ Radio Option 1
○ Radio Option 2
○ Radio Option 3
  
```

22

Forms – Radio Buttons

- For an option to be selected by default, we need to include the 'checked' property within the <input> tag.

```
<input type="radio" id="radio1" name="name_of_radio_collection" value="Radio Option 1" checked>
<label for="name_of_radio_collection">Radio Option 1</label><br>
```



- Doing this ensures that the user has selected one of the options.

23

Forms – Accessing Radio Buttons from JavaScript

- Within JavaScript, we are able to access the radio button collection elements by addressing the name of the collection and assigning that collection to a variable.

```
let testValues = document.getElementsByName("name_of_radio_collection");
```

getElementsByName


- You will need to then iterate over this collection until you find the one that has the 'checked' property.

```
for (var i = 0; i < testValues.length; i++){
  if (testValues[i].checked){
    console.log(testValues[i].value);
    break;
  }
}
```

24




25



10-15 Minutes

Task 4 – Radio Buttons

- Create a collection of radio buttons within your form to ask the user for their age range so that you can inform them of their minimum wage:
 - 16-17 – Has a minimum wage of £4.81
 - 18-20 – Has a minimum wage of £6.83
 - 21-22 – Has a minimum wage of £9.18
 - 23+ – Has a minimum wage of £9.50
- Iterate over the collection to find the 'checked' value.
- Once you have the value, use a case-switch to determine their minimum wage and show a message on the screen using alert().



26

Forms – Check Boxes

- A check box is a toggle-able input that is either yes or no.
- It's construction is identical to a radio button except we use "checkbox" as the type, including the 'checked' attribute.

```
<input type="checkbox" id="display" name="display_text" value="" checked>
<label for="display_text">Hide Text?</label>
```



27

Forms – Check Boxes

- We can use the check box within a form as a form element should we wish to process the information when the submit button has been pressed, exactly how we did with the radio buttons.
- However, we can also use it outside of a form and add an 'onclick' event listener to it instead.

```
<input onclick="hideUnhide()" type="checkbox" id="display" name="display_text" value="" checked>
<label for="display_text">Hide Text?</label>
```

28

Hiding and Unhiding Elements

- Imagine that we wanted a website that would change the displayed information depending on the buttons a user pressed.
- We can hide and unhide elements with our JavaScript! We do this by accessing a property called 'hidden'.

```
let elem = document.getElementById('edit_p');  
elem.hidden = true;
```

- How would we make this a toggle-able event?



29

Hiding and Unhiding Elements

- We can, of course, use an if-statement to toggle whether the element is hidden or not!

```
let elem = document.getElementById('edit_p');  
if (elem.hidden){  
  elem.hidden = false;  
} else {  
  elem.hidden = true;  
}
```

- Remember, we don't need to say '== true' in our if statements!

30

One Last Tip

- It may be the case that you need to execute some JavaScript when a page loads. You can do this by adding an 'onload' attribute to your <body> tag!

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

- Although this is useful, please be mindful of the end-user experience:
 - No-one wants to be greeted with a popup before they can see the page
 - No-one wants to be greeted with music when the page loads!

31

Any Questions So Far?



32



10-15 Minutes

Task 5 – Check Boxes

- Create a check box outside of your form that will have an 'onclick' method attached to it.
- This method should hide or unhide an aspect of your webpage such as a heading and a paragraph of text.
- **STRETCH CHALLENGE:** Use a check box to apply a light theme and a dark theme to your webpage. You will need to change all elements within the page:

```
let allElem = document.getElementsByTagName("*");
```

 * meaning 'all'