**Project: Coding Challenge**

**Lesson 1 Challenge 1**

A Hello, World! program is traditionally used to introduce coders to a new programming language or programming in general. It is a rite of passage in the world of programming that your first program in a new language outputs the text " Hello, World! " to the screen.

In particular you will

Display some text in the document.

**Result**

Result should look like the image below.

A white background with black and white text

Description automatically generated with medium confidence

**Steps**

* Add a <p> element in the area indicated on the editor on the left of the screen
* Add the text "Hello, World!" as content to the paragraph (Don't include the "" quotation marks in your answer)

***Code (index.html)***

*<!DOCTYPE html>*

*<html>*

*<head>*

*<title>Day 1 - Challenge 1</title>*

*</head>*

*<body>*

*<h1>My Day 1 challenge. I am Fazal !</h1>*

*<p> Hello, World! </p>*

*</body>*

*</html>*

**Lesson 1 - Challenge 2**

The challenge is to add the main heading and two paragraphs of content to a web page.

Specified words in each paragraph will be made to stand out.

**Result**

Your result should look like the image below.

A close-up of a sign

Description automatically generated

**Steps**

* Create an <h1> element and give it the following text content: " Why is HTML so Successful? "
* Create a <p> element and give it the following text content: " HTML is easy to learn ".
* Wrap the word " easy " in a <em> element.
* Create a <p> element and give it the following text content: " HTML is powerful ".
* Wrap the word " powerful " in a <strong> element.

**Tips**

Do a web search for the <em> and the <strong> element, and check out how they are used.

Why do we recommend searching? Well, at least once a day, even the most experienced developer will search the web looking for a solution to a problem they need to solve.

Knowing how to fine-tune your internet search words and phrases is a skill every developer needs to have.

***Code(index.html):***

*<html>*

*<head>*

*<title>HTML Content</title>*

*</head>*

*<body>*

*<!-- Place the <h1> element below here -->*

*<h1>Why is HTML so Successful?</h1>*

*<!-- Place a <p> element below here with the word easy wrapped in an <em> element -->*

*<p>HTML is <em>easy</em> to learn </p>*

*<!-- Place a <p> element below here with the word powerful wrapped in an <strong> element -->*

*<p>HTML is <strong>powerful</strong>*

*</body>*

*</html>*

**Lesson 2 - Challenge 1**

In particular, you will:

* style a <h1> element
* style a <p> element
* style an <em> element
* style a <strong> element
* You'll do this by using element selectors (style rules that have the same selector names as the HTML elements they are targeting).
* In this challenge, each style rule will contain one style declaration that will target its associated HTML element.

**Result**

Your result should look like the image below:

A yellow text on a white background

Description automatically generated

**Steps**

* We started with creating a style rule for the <body> element, using its HTML Element selector name creating a declaration giving it a font-size property setting the property value to 14px
* Create an h1 style rule within the < style > element and below the body
* Set the h1 rule color property to gold . - *Note the US spelling of color -*
* Create a p rule below the h1 rule
* Set the p rule color property to darkslategray
* Create an em rule below the p rule
* Set the em rule font-size property to 20px
* Create a strong rule below the em rule
* Set the strong rule font-size property to 24px

**Tips**

* Specifying a color by its name can be restricting given the millions of colours available to us on our screens.
* We did a web search to find out other ways of specifying color.
* In particular looked at how hex and RGB color is used in CSS
* Why do we recommend searching? Well, at least once a day, even the most experienced developer will search the web looking for a solution to a problem they need to solve.
* Knowing how to fine-tune your internet search words and phrases is a skill every developer needs to have

***Code(index.html)***

*<!DOCTYPE html>*

*<html>*

*<head>*

*<title>Day 2 Challenge 1</title>*

*<style>*

*body {*

*font-size: 14px;*

*}*

*/\* place your style rules below this comment \*/*

*h1*

*{*

*color: gold;*

*}*

*p {*

*color: darkslategray;*

*}*

*em {*

*font-size: 20px;*

*}*

*strong {*

*font-size: 24px;*

*}*

*</style>*

*</style>*

*</head>*

*<body>*

*<!-- give these elements some style -->*

*<h1> Why is HTML so successful</h1>*

*<p> Html is <em>easy</em></p>*

*<p>Html is <strong>fun</strong></p>*

*</body>*

*</html>*

**Lesson 2 - Challenge 2**

In the previous challenge, we styled some HTML elements. we targeted the HTML by creating associated element style rules inside the inline <style> element.

we used one style declaration for each of the rules. However, a style rule can contain more than one style declaration.

Also in the last challenge, we were encouraged to search the web for other formats available to the color property. By doing this we had found using RGB and hex formatting to specify millions of colors.

In this challenge, we will modify some of the rules created in your last challenge.

Also as part of this challenge, we will set some of the text colours using hex and RGB .

In particular, we will:

* Set the font type for the entire page
* Edit the heading text color
* Set all heading text to uppercase.
* Highlight emphasised text by assigning it a particular color.
* Highlight bold text by assigning it a particular color.

**Result**

Your result should look like the image below:

A yellow text on a white background

Description automatically generated

**Steps**

* Add a font-family property to the body rule. Set it to Arial change the style declaration for the h1 rule and use the hex format for the color property. Set the color to #ffd700
* Add a text-transform property to the h1 rule. Set the value to uppercase .
* Add a color property to the em rule using the RGB format. Set the color to rgb(0, 128, 0)
* Add a color property to the strong rule using the RGB format. Set the color to rgb(0, 0, 0)

**Tips**

* Did a web search for the two CSS properties ( text-transform & font-family ) needed to help complete the challenge.
* Why do we recommend searching? Well, at least once a day, even the most experienced developer will search the web looking for a solution to a problem they need to solve.
* Knowing how to fine-tune our internet search words and phrases is a skill every developer needs to have.

**Code (index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<title>Day 2 Challenge 2</title>*

*<style>*

*body {*

*font-size: 14px;*

*font-family: Arial;*

*}*

*h1 {*

*color: #ffd700;*

*text-transform: uppercase;*

*}*

*p {*

*color: darkslategray;*

*}*

*em {*

*font-size: 20px;*

*color:rgb(0, 128, 0);*

*}*

*strong {*

*font-size: 24px;*

*color:rgb(0, 0, 0);*

*}*

*</style>*

*</head>*

*<body>*

*<h1>Why is HTML so successful?</h1>*

*<p>Html is <em>easy</em></p>*

*<p>Html is <strong>fun</strong></p>*

*</body>*

*</html>*

**Lesson 2 Challenge 3**

Every HTML element can be assigned a unique identifier. This is done by including an id attribute and its associated value in the element's opening tag. Element ids should be unique to each element on a page. They should also be given a meaningful name that describes what the element is being used for. We can use CSS to target an id

In particular, you will:

* Add an id and its value to a div element
* Add an id and its value to the div's child paragraph element
* You'll then target these ids with unique styles

**Result**

Your result should look like the image below:

A screenshot of a history page

Description automatically generated

**Steps**

* A style rule that targets the div has created. Next we will need to create the second style rule to target the paragraph
* Add an id attribute to the < div > element and give it the value " heading" . Our #heading style rule will now target that div
* Add an id attribute to the <p> element within the < div > element, give it the value " uppercase"
* Create a style rule on line 14 that has an id selector name of #uppercase. This will target our paragraph
* Within that rule add a declaration that sets the text-transform property to uppercase

**Code(index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<title>Targeting an individual</title>*

*<style>*

*#heading {*

*font-family: Arial;*

*background-color: red;*

*color: white;*

*}*

*/\* create your rule here \*/*

*#uppercase {*

*text-transform: uppercase;*

*}*

*</style>*

*</head>*

*<body>*

*<!-- Add the correct id to this element -->*

*<div id="heading">*

*<h1 >Sporting History</h1>*

*<!-- add the correct id to this element -->*

*<p id="uppercase">Giants & Heroes</p>*

*</div>*

*<h3> Track & Field</h3>*

*<p id="uppercase">*

*Some sporting feats ridicule expectations, fewer violate logic but the rarest of all added a defying of gravity to a hat-trick of achievements. At 3.45pm on 18 October 1968 in Mexico City's Estadio Olmpico Universitario Bob Beamon accomplished all three.*

*</p>*

*<h3>Rugby</h3>*

*<p>*

*With the game locked at 10 - 10 with thirty minutes on the clock, a Munster scrum just outside the Biarritz line enabled Peter Stringer to do the unexpected and break on the blindside for an unforgettable try. A moment of sheer brilliance by the Munster and Ireland scrumhalf. The try proved to be crucial as Munster won by 23 - 19 to lift the European Cup in Cardiff.*

*</p>*

*<h3>The first Programmer</h3>*

*<p>*

*Ada Lovelace was an English mathematician and writer, chiefly known for her work on Charles Babbage's proposed mechanical general-purpose computer, the Analytical Engine. She was the first to recognise that the machine had applications beyond pure calculation, and published the first algorithm intended to be carried out by such a machine. As a result, she is sometimes regarded as the first to recognise the full potential of a "computing machine" and the first computer programmer.*

*</p>*

*<h3>The first Compiler</h3>*

*<p>*

*Grace Hopper was an American computer scientist and United States Navy rear admiral. One of the first programmers of the Harvard Mark I computer, she was a pioneer of computer programming who invented one of the first compiler related tools.*

*</p>*

*</body>*

*</html>*

**Lesson 2 - Challenge 4**

Every HTML element can be assigned a class identifier. This is done by including a class attribute and its associated value in the element's opening tag. Elements that are conceptually grouped in some way can be given the same class attribute value. They should also be given a meaningful name that describes what the element grouping is being used for. We can use CSS to target class attributes and apply a collective style to those elements.

In the last challenge, we applied id styling to a Sporting History web page. In particular, :

* Added an id and its value to a div element
* Added an id and its value to the div's child paragraph element
* Targeted these ids with unique styles
* This challenge will build upon the previous one.

In particular, also will:

* Group and style the sports-related content on our web page
* Group and style programming-related content on our web page

**Result**

Your result should look like the image below.

A screenshot of a computer

Description automatically generated

**Steps**

* Add a class attribute to the first two <h3> elements and give each of them the attribute value of sports-history
* Add a class attribute to the second two <h3> elements and give each of them the attribute value of computing-history
* Create a style rule beneath the existing id rules that has a corresponding class selector name of sports-history. This will target our sports content within that rule, add a declaration that sets the color property to orange
* Create another style rule has a corresponding class selector name of computing-history. This will target our programming content
* Within that rule add a declaration that sets the color property to green

**Code(Index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<title>Lesson 2 Challenge 4</title>*

*<style>*

*#heading{*

*font-family: Arial;*

*background-color: red;*

*color: white;*

*}*

*#uppercase {*

*text-transform: uppercase;*

*}*

*/\* Add your class rules here \*/*

*.sports-history {*

*color: orange;*

*}*

*.computing-history {*

*color: green;*

*}*

*</style>*

*</head>*

*<body>*

*<div id="heading">*

*<h1> Sporting History</h1>*

*<p id='uppercase'>*

*Giants & Heroes*

*</p>*

*</div>*

*<!-- add the correct class here -->*

*<h3 class="sports-history"> Track & Field</h3>*

*<p>*

*Some sporting feats ridicule expectations, fewer violate logic but the rarest of all added a defying of gravity to a hat-trick of achievements. At 3.45pm on 18 October 1968 in Mexico City’s Estadio Olímpico Universitario Bob Beamon accomplished all three.*

*</p>*

*<!-- add the correct class here -->*

*<h3 class="sports-history">Rugby</h3>*

*<p>*

*With the game locked at 10 - 10 with thirty minutes on the clock, a Munster scrum just outside the Biarritz line enabled Peter Stringer to do the unexpected and break on the blindside for an unforgettable try. A moment of sheer brilliance by the Munster and Ireland scrumhalf. The try proved to be crucial as Munster won by 23 - 19 to lift the European Cup in Cardiff.*

*</p>*

*<!-- add the correct class here -->*

*<h3 class="computing-history">The first Programmer</h3>*

*<p>*

*Ada Lovelace was an English mathematician and writer, chiefly known for her work on Charles Babbage's proposed mechanical general-purpose computer, the Analytical Engine. She was the first to recognise that the machine had applications beyond pure calculation, and published the first algorithm intended to be carried out by such a machine. As a result, she is sometimes regarded as the first to recognise the full potential of a "computing machine" and the first computer programmer.*

*</p>*

*<!-- add the correct class here -->*

*<h3 class="computing-history">The first Compiler</h3>*

*<p>*

*Grace Hopper was an American computer scientist and United States Navy rear admiral. One of the first programmers of the Harvard Mark I computer, she was a pioneer of computer programming who invented one of the first compiler related tools.*

*</p>*

*</body>*

*</html>*

**Lesson 3 Challenge 1**

In these upcoming challenges, we will create our own recipe. our first challenge is to create the core HTML in order to create our own recipe page.

In particular, you will:

* Add a header area with placeholder text for our recipe name and image
* Add a section with placeholder text for our ingredients list
* Add a section with placeholder text for preparation list
* Add a page footer with placeholder text for the content copyright

**Result**

Your result should look like the image below:

A white background with black text

Description automatically generated

**Steps**

* Add recipe name to the <title> element within the head.

Within the < body > element, add the following elements:

* Add a <header> element. Within the header add the placeholder text " recipe name, image, description"
* Add two <section> elements.
* Within the first <section> add the placeholder text " ingredients list"
* Within the second <section> add the placeholder text " preparation list"
* Add a <footer> element. Within the footer add the placeholder text " copyright"

**Code (index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<!-- Set the title here -->*

*<title>Receipe 1</title>*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<!-- Add your header here -->*

*<header>recipe name, image, description</header>*

*<!-- Add a section here -->*

*<section><p>ingredients list</p></section>*

*<!-- Add a section here -->*

*<section><p>preparation list</p></section>*

*<!-- Add your footer here -->*

*<footer>copyright</footer>*

*<script src="index.js"></script>*

*</body>*

*</html>*

**Lesson 3 - Challenge 2**

Now that we have boilerplate HTML in place, which includes the placeholder text. It is now time to start replacing the placeholder text with the required elements and our own recipe details . We are going to do this over the following challenges.

we are going to add elements to the header and first section

In particular, you will add:

* The recipe name
* The recipe description
* The ingredients heading
* The list of ingredients

**Result**

Your result should look similar to the image below

A white background with black text

Description automatically generated

**Steps**

* Remove the recipe name placeholder text and add an <h1> element to the <header> giving it your recipe name
* Remove the description placeholder text and add a <p> element to the <header> describing your recipe
* Remove the ingredients placeholder text and add a <h2> element to the first <section> . The heading text will be " ingredients"
* Remove the list placeholder text and add a <ul> element to the first <section>
* The <ul> should contain at least 4 <li> elements listing your recipe ingredients.

**Code (Index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<!-- Add your recipe title here -->*

*<title>My Recipe</title>*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<header>*

*<!-- Add your <h1> here -->*

*<h1>How to make Tea</h1>*

*<!-- we will add the image in a later challenge -->*

*image*

*<!-- Add your <p> here -->*

*<p>At the very least</p>*

*</header>*

*<section>*

*<!-- Add your <h2> here -->*

*<h2>ingredients</h2>*

*<!-- Add your <ul> list here -->*

*<ul>*

*<li>Tea bags</li>*

*<li>Water</li>*

*<li>milk</li>*

*<li>sugar</li>*

*</ul>*

*</section>*

*<section>*

*preparation*

*list*

*</section>*

*<footer>*

*copyright*

*</footer>*

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

*</body>*

*</html>*

**Lesson 3 - Challenge 3**

In the last challenge, we began removing the placeholder text and started adding some of the elements we needed for our recipe

Now have the following elements in place

* The recipe name
* The recipe description
* The ingredients heading
* The list of ingredients
* In this challenge, we will add some more of the HTML elements that are required.

In particular, we will add:

* The preparation heading
* The preparation steps
* The copyright text

**Result**

Your result should look similar to the image below (but with your own recipe text content):

A screenshot of a computer

Description automatically generated

**Steps**

* Replace the recipe name, ingredients heading, and ingredients list with the text you added to the last challenge
* Add a <h2> element to the second <section>, and text of your choice
* Add an <ol> element to the second <section>
* The < ol > should contain at least 4 <li> element listing your recipe instructions.
* The < footer > element should contain a <p> element with the text: Copyright Me 2020

**Code(index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<!-- Add your recipe title here -->*

*<title>Tea Recipe</title>*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<header>*

*<h1>How to make Tea</h1>*

*<!-- we will add the image in a later challenge -->*

*image*

*<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.*

*On top of that, studies have shown teas can help protect your teeth and your heart</p>*

*</header>*

*<section>*

*<h2>ingredients</h2>*

*<ul>*

*<li>Tea Bag</li>*

*<li>Water</li>*

*<li>Milk - Dairy/Plant based(Optional)</li>*

*<li>Sugar/Honey (Optional)</li>*

*</ul>*

*</section>*

*<section>*

*<!-- Add your <h2> here -->*

*<h2>preparation</h2>*

*<!-- Add your <ol> list here -->*

*<ol>*

*<li>1.jjj</li>*

*<li>2.jjj</li>*

*<li>3.jjj</li>*

*<li>4.jjj</li>*

*</ol>*

*</section>*

*<footer>*

*<!-- Add your <p> here -->*

*<p>Copyright Me 2020</p>*

*</footer>*

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

*</body>*

*</html>*

**Lesson 4 - Challenge 1**

In the last two challenges, we added the following to your recipe page:

* The recipe name
* The recipe description
* The ingredients heading
* The list of ingredients
* The preparation heading
* The preparation steps
* The copyright text
* All that's left in terms of content is the adding of a recipe image to the page.
* Our challenge is to do just that.

**Result**

Your result should look similar to the image below (but with the option using your own image content):

A person holding a cup of tea

Description automatically generated

**Steps**

* Within the < header > element remove image the placeholder text and add an <img> element. Set its src attribute value to " <https://codeinstitute.s3-eu-west> 1.amazonaws.com/5DCC/images/Content/oSVKMED.jpg ",
* Or can choose our own image. If we do, will make sure the image has a .png or .jpg file extension name
* Give < img > element an alt attribute. The alt attribute should be a short description of the image. Eg. *alt="A man drinking tea"*, The inclusion of the alt attribute helps screen readers determine content meaning for sight-impaired users

**Code(index.html)**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width">

<!-- Add your recipe title here -->

<title>Tea Recipe</title>

</head>

<body>

<header>

<h1>How to make Tea</h1>

<!-- Add the image here -->

<img src="https://codeinstitute.s3-eu-west-1.amazonaws.com/5DCC/images/Content/oSVKMED.jpg" alt="A man drinking tea">

<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.

On top of that, studies have shown teas can help protect your teeth and your heart</p>

</header>

<section>

<h2>ingredients</h2>

<ul>

<li>Tea Bag</li>

<li>Water</li>

<li>Milk - Dairy/Plant based(Optional)</li>

<li>Sugar/Honey (Optional)</li>

</ul>

</section>

<section>

<h2>preparation</h2>

<ol>

<li>

Run the tap a little so the water's nicely aerated, and only boil it once to keep the oxygen level up. Oxygen in water helps flavour!

</li>

<li>

Pop a tea bag into your mug (<em>always</em> a big mug)

</li>

<li>

Pour the hot water over the tea bag and stir briefly.

</li>

<li>

Tea needs time to unlock all its flavour, so give it 3-4 minutes to do its thing. This is a perfect time to grab a sneaky cookie or daydream about vacations.

</li>

<li>

Before removing the tea bag, gently squish it with a spoon against the side of the mug. Not too much or you'll make it bitter.

</li>

<li>

If you want, throw in some milk or sugar or honey or nothing else at all.

</li>

<li>

</li>

</ol>

</section>

<footer>

<p>Copyright Me 2020</p>

</footer>

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

</body>

</html>

**Lesson 4 - Challenge 2**

Having added all the HTML we require for the recipe page it is now time to start styling the page. The most common and most useful location for CSS style rules in a separate file that is linked to one or more HTML documents that are targets of the styles. Why? Because a central change made to a style will update all the web pages that link to and use that style. Central location changes are easier to manage than multiple duplicated style content saved across many documents. So let's do that now.

In particular, will:

* Style the default text font
* Size the main heading
* Color the headings
* Make the heading text uppercase

**Result**

Your result should look similar to the image below.

A screenshot of a cell phone

Description automatically generated

**Steps**

* In the panel on the left, we will see there are two tabs, one of which is named index.css. Note the .css
* Our HTML document named index.html links to the CSS file on line 7.
* Click on the index.css file to view/edit/add style rules.
* Create a style rule with a type selector that targets the <body> element
* Inside the body rule set the font family to Arial
* Inside the body rule set the color property to slategray
* Create a style rule with a type selector that targets the <h1> element
* Inside the h1 rule set the font-size property to 300%
* Create a style rule with a type selector that targets both <h1> and <h2> elements
* Inside the h1, h2 rule set the color property to orange
* Inside this rule set the text-transform property to uppercase
* Inside this rule set the font-weight property to 100
* Create a style rule with a type selector that targets the <img> element
* Inside the img rule set the width property to 100%

**Code(Index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<title>Tea Recipe</title>*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<header>*

*<h1>How to make Tea</h1>*

*<img src="https://codeinstitute.s3-eu-west-1.amazonaws.com/5DCC/images/Content/oSVKMED.jpg" alt="A man drinking tea">*

*<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.*

*On top of that, studies have shown teas can help protect your teeth and your heart</p>*

*</header>*

*<section>*

*<h2>ingredients</h2>*

*<ul>*

*<li>Tea Bag</li>*

*<li>Water</li>*

*<li>Milk - Dairy/Plant based(Optional)</li>*

*<li>Sugar/Honey (Optional)</li>*

*</ul>*

*</section>*

*<section>*

*<h2>preparation</h2>*

*<ol>*

*<li>*

*Run the tap a little so the water's nicely aerated, and only boil it once to keep the oxygen level up. Oxygen in water helps flavour!*

*</li>*

*<li>*

*Pop a tea bag into your mug (<em>always</em> a big mug)*

*</li>*

*<li>*

*Pour the hot water over the tea bag and stir briefly.*

*</li>*

*<li>*

*Tea needs time to unlock all its flavour, so give it 3-4 minutes to do its thing. This is a perfect time to grab a sneaky cookie or daydream about vacations.*

*</li>*

*<li>*

*Before removing the tea bag, gently squish it with a spoon against the side of the mug. Not too much or you'll make it bitter.*

*</li>*

*<li>*

*If you want, throw in some milk or sugar or honey or nothing else at all.*

*</li>*

*<li>*

*</li>*

*</ol>*

*</section>*

*<footer>*

*<p>Copyright Me 2020</p>*

*</footer>*

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

*</body>*

*</html>*

*Code(index.css)*

*body {*

*font-family:Arial;*

*color:slategray;*

*}*

*h1 {*

*font-size:300%;*

*}*

*h1,h2 {*

*color: orange;*

*text-transform:uppercase;*

*font-weight:100;*

*}*

*img {*

*width:100%;*

*}*

**Lesson 4 - Challenge 3**

In the last challenge we :

* Styled the default text font
* Sized the main heading
* Colored the headings
* Made the heading text uppercase

In this challenge, we will style the page footer

* Also as with the last challenge, we will write your styles in index.css

**Result**

Your result should look similar to the image below.

A screenshot of a computer

Description automatically generated

**Steps**

* Create a style rule with a type selector that targets the <footer> element
* Inside the footer rule set the margin-top to 70px
* Inside the footer rule set the background-color to orange
* Inside the footer rule set the color to white
* Create a style rule with a type selector that targets the <p> child of the <footer>
* Inside the footer p rule set the text-align to center
* Inside the footer p rule set the padding-top to 15px
* Inside the footer p rule set the padding-bottom to 15px

**Code (index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<title>Tea Recipe</title>*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<header>*

*<h1>How to make Tea</h1>*

*<img src="https://codeinstitute.s3-eu-west-1.amazonaws.com/5DCC/images/Content/oSVKMED.jpg" alt="A man drinking tea">*

*<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.*

*On top of that, studies have shown teas can help protect your teeth and your heart*

*</p>*

*</header>*

*<section>*

*<h2>ingredients</h2>*

*<ul>*

*<li>Tea Bag</li>*

*<li>Water</li>*

*<li>Milk - Dairy/Plant based(Optional)</li>*

*<li>Sugar/Honey (Optional)</li>*

*</ul>*

*</section>*

*<section>*

*<h2>preparation</h2>*

*<ol>*

*<li>*

*Run the tap a little so the water's nicely aerated, and only boil it once to keep the oxygen level up. Oxygen in water helps flavour!*

*</li>*

*<li>*

*Pop a tea bag into your mug (<em>always</em> a big mug)*

*</li>*

*<li>*

*Pour the hot water over the tea bag and stir briefly.*

*</li>*

*<li>*

*Tea needs time to unlock all its flavour, so give it 3-4 minutes to do its thing. This is a perfect time to grab a sneaky cookie or daydream about vacations.*

*</li>*

*<li>*

*Before removing the tea bag, gently squish it with a spoon against the side of the mug. Not too much or you'll make it bitter.*

*</li>*

*<li>*

*If you want, throw in some milk or sugar or honey or nothing else at all.*

*</li>*

*<li>*

*</li>*

*</ol>*

*</section>*

*<footer>*

*<p>Copyright Me 2020</p>*

*</footer>*

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

*</body>*

*</html>*

**Code(index.css)**

*body {*

*font-family: Arial;*

*color: slategray;*

*}*

*h1 {*

*font-size: 300%;*

*}*

*h1,*

*h2 {*

*color: orange;*

*text-transform: uppercase;*

*font-weight: 100;*

*}*

*img {*

*width: 100%;*

*}*

*/\* Add your styles here \*/*

*footer {*

*margin-top:70px;*

*background-color:orange;*

*color:white;*

*}*

*footer p{*

*text-align: center;*

*padding-top: 15px;*

*padding-bottom: 15px;*

*}*

**Lesson 5 - Challenge 1**

Our recipe site is really coming together now. We next need to make sure that its design and structural integrity is maintained across device sizes. Our challenge is to make our recipe page be responsive to the size of the screen that it is viewed on.

In particular, we will:

* Add an outer container that wraps around all the content of the page
* Target that container with a style that sets the container width and also centers the container
* Target that container with a media query that contains styles that are triggered once the screen size goes below a certain value

**Result**

Your result should look similar to the image below

A person holding a cup

Description automatically generated

**Steps**

* On line 13 a <div> element that acts as the parent for all content has been created. Add an id attribute to that <div> element with the value of container
* Create a style rule with an id selector that targets the container id
* Inside the #container rule set the width to 60%
* Inside the #container rule set the margin to auto
* Create a media query that will apply styles for screen size up to a max-width of 768px
* Reuse the #container style rule inside the media query and set the width to 90%
* Reuse the h1 style rule inside the  media query and set the font-size to 200%

**Tips**

* Drag the panel separator bar to the left of this text to resize the result pane and see the media queries being triggered

**Code(index.html)**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width">

<title>Tea Recipe</title>

<link href="index.css" rel="stylesheet" type="text/css" />

</head>

<body>

<!-- Add the id to this div -->

<div id="container">

<header>

<h1>How to make Tea</h1>

<img src="https://codeinstitute.s3-eu-west-1.amazonaws.com/5DCC/images/Content/oSVKMED.jpg" alt="A man drinking tea">

<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.

On top of that, studies have shown teas can help protect your teeth and your heart</p>

</header>

<section>

<h2>ingredients</h2>

<ul>

<li>Tea Bag</li>

<li>Water</li>

<li>Milk - Dairy/Plant based(Optional)</li>

<li>Sugar/Honey (Optional)</li>

</ul>

</section>

<section>

<h2>preparation</h2>

<ol>

<li>

Run the tap a little so the water's nicely aerated, and only boil it once to keep the oxygen level up. Oxygen in water helps flavour!

</li>

<li>

Pop a tea bag into your mug (<em>always</em> a big mug)

</li>

<li>

Pour the hot water over the tea bag and stir briefly.

</li>

<li>

Tea needs time to unlock all its flavour, so give it 3-4 minutes to do its thing. This is a perfect time to grab a sneaky cookie or daydream about vacations.

</li>

<li>

Before removing the tea bag, gently squish it with a spoon against the side of the mug. Not too much or you'll make it bitter.

</li>

<li>

If you want, throw in some milk or sugar or honey or nothing else at all.

</li>

<li>

</li>

</ol>

</section>

<footer>

<p>Copyright Me 2020</p>

</footer>

</div>

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

</body>

</html>

**Code(index.css)**

body {

font-family: Arial;

color: slategray;

}

h1 {

font-size: 300%;

}

h1,

h2 {

color: orange;

text-transform: uppercase;

font-weight: 100;

}

img {

width: 100%;

}

footer {

margin-top: 70px;

background-color: orange;

color: white;

}

footer p {

text-align: center;

padding-top: 15px;

padding-bottom: 15px;

}

/\* add your styles here \*/

#container {

width: 60%;

margin:auto;

}

@media screen and (max-width: 768px){

#container {

width:90%;

}

h1 {

font-size:200%;

}

}

**Lesson 5 - Challenge 2**

Discerningly placed icons can draw the user's eye to important content on a site.

Our challenge is to add external font icons to your recipe.

In particular, we will:

* Place an icon within your ingredients sub-heading
* Place an icon within your preparation sub-heading

**Result**

Your result should look similar to the image below

A screenshot of a recipe

Description automatically generated

**Steps**

* Copy the font awesome link and paste it into the head area of your HTML : <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" type="text/css" /> Place it above the link to the index.css file.
* Go To the Font Awesome website below and choose fonts from the 4.7 icon set: *https://fontawesome.com/v4.7.0/icons/*
* Add a font icon to the ingredients sub-heading
* Add a font icon to the preparation sub-heading

**Code(index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<title>repl.it</title>*

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" type="text/css" />*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<div id="container">*

*<header>*

*<h1>How to make tea</h1>*

*<img src="https://i.imgur.com/oSVKMED.jpg">*

*<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.*

*On top of that, studies have shown teas can help protect your teeth and your heart</p>*

*</header>*

*<section>*

*<h2>ingredients<i class="fa fa-superpowers" aria-hidden="true"></i></h2>*

*<ul>*

*<li>Tea bag</li>*

*<li>Water</li>*

*<li>Milk - Dairy/Plant-based (Optional)</li>*

*<li>Sugar/Honey (Optional)</li>*

*</ul>*

*</section>*

*<section>*

*<h2>preparation<i class="fa fa-bath" aria-hidden="true"></i></h2>*

*<ol>*

*<li>*

*Run the tap a little so the water's nicely aerated, and only boil it once to keep the oxygen level up. Oxygen in water helps flavour!*

*</li>*

*<li>*

*Pop a tea bag into your mug (<em>always</em> a big mug)*

*</li>*

*<li>*

*Pour the hot water over the tea bag and stir briefly.*

*</li>*

*<li>*

*Tea needs time to unlock all its flavour, so give it 3-4 minutes to do its thing. This is a perfect time to grab a sneaky cookie or daydream about vacations.*

*</li>*

*<li>*

*Before removing the tea bag, gently squish it with a spoon against the side of the mug. Not too much or you'll make it bitter.*

*</li>*

*<li>*

*If you want, throw in some milk or sugar or honey or nothing else at all.*

*</li>*

*<li>*

*</li>*

*</ol>*

*</section>*

*<footer>*

*<p>Copyright Me 2018</p>*

*</footer>*

*</div>*

*<script src="index.js"></script>*

*</body>*

*</html>*

*Code(index.css)*

*body {*

*font-family: Arial;*

*color: SlateGray;*

*}*

*h1 {*

*font-size: 300%;*

*}*

*h2 {*

*color: Orange;*

*text-transform: uppercase;*

*font-weight: 100;*

*}*

*img {*

*width: 100%;*

*}*

*footer {*

*margin-top: 70px;*

*background-color: Orange;*

*color: White;*

*}*

*footer p {*

*text-align: center;*

*padding-top: 15px;*

*padding-bottom: 15px;*

*}*

*#container {*

*width: 60%;*

*margin: auto;*

*}*

*@media screen and (max-width: 768px) {*

*#container {*

*width: 90%;*

*}*

*h1 {*

*font-size: 200%;*

*}*

*}*

**Lesson 5 Challenge 3**

At this point, we have all our HTML structure and CSS styling in place. Well done for getting this far! In this challenge, we will add some JavaScript interactivity to the recipe site. Interactivity can be as simple or complex as needed. Our interactivity is used to draw the user's attention to the recipe ingredients and preparation steps when moving their mouse on the page

In particular, we will:

* Trigger an event that increases the size of the ingredients Font Awesome icon when a user's mouse passes over the ingredients section
* Trigger an event that returns the ingredients Font Awesome icon to normal size when a user's mouse passes out of the ingredients section
* Trigger an event that increases the size of the preparation Font Awesome icon when a user's mouse passes over the preparation section
* Trigger an event that returns the preparation Font Awesome icon to normal size when a user's mouse passes out of the preparation section

**Result**

Your result should look similar to the image below

A screenshot of a cell phone

Description automatically generated

**Steps**

* Take a look at lines 22 and 33 . We'll see that there are two of what are called event listeners added as attributes to the ingredients and preparation sections . Their job is to listen for and react to two particular events . One reacts when a user's mouse passes into the section area ( onmouseover ) and the second reacts when a user's mouse passes out of the section ( onmouseout ).
* When one of these events occurs the appropriate event listener calls a function . For example, when the user's mouse passes into the ingredients section the onmouseover event listener will call a function named ingredientsHover() . This is function is written by the programmer.
* We will create a function with the same name as the function referred to in the event listener.
* Inside that function, we will write the code that resizes the icon when the event happens.

We will write four functions.

* Two for the ingredients section and two for the preparation section.
* A <script> element has been created. We will add your JavaScript functions within that element Ingredients
* Create the function that is called when the ingredients onmouseover event is triggered. The code is shown below. Notice how the function name tries to indicate its purpose

function ingredientsHover(){

}

* Add the code that changes the ingredients icon size. This will be placed inside the function's curly braces {}. The code is shown below. Try and understand how it works. Read it from left to right. The document represents the web page document.getElementById("ingredients").style.fontSize = "300%";
* Create the function that is called when the ingredients onmouseout event is triggered. The code is shown below

function ingredientsNormal(){

}

* Add the code that resets the ingredients icon size to its default size. This will be placed inside the function's curly braces {}. The code is shown below. Try to understand how it works.

document.getElementById("ingredients").style.fontSize = "100%";

**Preparation**

Create the function that is called when the preparation onmouseover event is triggered. The code is shown below

function preparationHover(){

}

Add the code that changes the preparation icon size. This will be placed inside the function's curly braces {}. The code is shown below. Try to understand how it works.

document.getElementById("preparation").style.fontSize = "300%";

Create the function that is called when the preparation onmouseout event is triggered. The code is shown below

function preparationNormal(){

}

Add the code that resets the preparation icon size to its default size. This will be placed inside the function's curly braces {}. The code is shown below. Try to understand how it works.

document.getElementById("preparation").style.fontSize = "100%";

And That's it!

Well done on completing your challenges. We hope this inspired you to explore the world of software development even further. Whatever you decide to do next, do what you love. Life is too short to do anything less.

**Code(index.html)**

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width">*

*<title>Tea Recipe</title>*

*<!-- add the font awesome link here -->*

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" type="text/css" />*

*<link href="index.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<div id="container">*

*<header>*

*<h1>How to make Tea</h1>*

*<img src="https://codeinstitute.s3-eu-west-1.amazonaws.com/5DCC/images/Content/oSVKMED.jpg" alt="A man drinking tea">*

*<p>At the very least, tea is a flavourful way of getting enough fluid into your body each day.*

*On top of that, studies have shown teas can help protect your teeth and your heart</p>*

*</header>*

*<section onmouseover="ingredientsHover()" onmouseout="ingredientsNormal()">*

*<!-- add a fontawesome icon here -->*

*<h2>ingredients <i id="ingredients" class="fa fa-coffee" aria-hidden="true"></i></h2>*

*<ul>*

*<li>Tea Bag</li>*

*<li>Water</li>*

*<li>Milk - Dairy/Plant based(Optional)</li>*

*<li>Sugar/Honey (Optional)</li>*

*</ul>*

*</section>*

*<section onmouseover="preparationHover()" onmouseout="preparationNormal()">*

*<!-- add a fontawesome icon here -->*

*<h2>preparation <i id="preparation" class="fa fa-check-circle" aria-hidden="true"></i></h2>*

*<ol>*

*<li>*

*Run the tap a little so the water's nicely aerated, and only boil it once to keep the oxygen level up. Oxygen in water helps flavour!*

*</li>*

*<li>*

*Pop a tea bag into your mug (<em>always</em> a big mug)*

*</li>*

*<li>*

*Pour the hot water over the tea bag and stir briefly.*

*</li>*

*<li>*

*Tea needs time to unlock all its flavour, so give it 3-4 minutes to do its thing. This is a perfect time to grab a sneaky cookie or daydream about vacations.*

*</li>*

*<li>*

*Before removing the tea bag, gently squish it with a spoon against the side of the mug. Not too much or you'll make it bitter.*

*</li>*

*<li>*

*If you want, throw in some milk or sugar or honey or nothing else at all.*

*</li>*

*<li>*

*</li>*

*</ol>*

*</section>*

*<footer>*

*<p>Copyright Me 2020</p>*

*</footer>*

*</div>*

*<script>*

*function ingredientsHover(){*

*document.getElementById("ingredients").style.fontSize = "300%";*

*}*

*function ingredientsNormal(){*

*document.getElementById("ingredients").style.fontSize = "100%";*

*}*

*function preparationHover(){*

*document.getElementById("preparation").style.fontSize = "300%";*

*}*

*function preparationNormal(){*

*document.getElementById("preparation").style.fontSize = "100%";*

*}*

*</script>*

*</body>*

*</html>*

*Code(index.css)*

*body {*

*font-family: Arial;*

*color: slategray;*

*}*

*h1 {*

*font-size: 300%;*

*}*

*h1,*

*h2 {*

*color: orange;*

*text-transform: uppercase;*

*font-weight: 100;*

*}*

*img {*

*width: 100%;*

*}*

*footer {*

*margin-top: 70px;*

*background-color: orange;*

*color: white;*

*}*

*footer p {*

*text-align: center;*

*padding-top: 15px;*

*padding-bottom: 15px;*

*}*

*#container {*

*width: 60%;*

*margin: auto;*

*}*

*@media screen and (max-width: 768px) {*

*#container {*

*width: 90%;*

*}*

*h1 {*

*font-size: 200%;*

*}*

*}*