

# the Master Course

{C0DENATION}

# Web Technologies

## Intro to CSS

{CODENATION}

# HTML

## First Things First...

... write a function called **dataChecker** which takes **two parameters** (string, rank). If the string is **equal to 'codenation'** and the rank is **equal to 1**, then log to the console a template string. If the string is 'codenation' but the rank **isn't 1** log to the console a relevant template string.

Finally, if both parameters are incorrect log a relevant template string to the console.

**Extension:** If you've finished, try to create a CSS file and link it to the your HTML file

{ C0DENATION }

# HTML

## First Things First...

```
const dataChecker = (string, rank) => {  
  if (string == "codenation" && rank == 1) {  
    console.log(`Both ${string} and the ranking ${rank} are correct`);  
  } else if (string == "codenation" && rank != 1) {  
    console.log(`${string} is correct but the ranking ${rank} isn't`);  
  } else {  
    console.log(`Both ${string} and/or the ranking ${rank} are wrong`);  
  }  
}  
  
dataChecker("codenation", 1);
```

# Learning Objectives

**To understand what CSS is and what it's used for**

**To understand what a few CSS syntax and selectors are and be able to write your own**

**To be familiar with some common CSS properties**

CSS

# Cascading Style Sheets



CSS

How do we...

... add **CSS!**

# Inline Style

...least common method and  
**should be avoided**

CSS

# Internal Style Sheet

...another uncommon method  
and **should be avoided**

# External Style Sheet

...the **most common method** and  
should be used **most of the time**







CSS

So lets...

... add an **External Style Sheet!**

# Adding an External Style Sheet

CSS

create a new file and name it **main.css**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="main.css">
  <title>Website title</title>
</head>
<body>
  Content here...
</body>
</html>
```

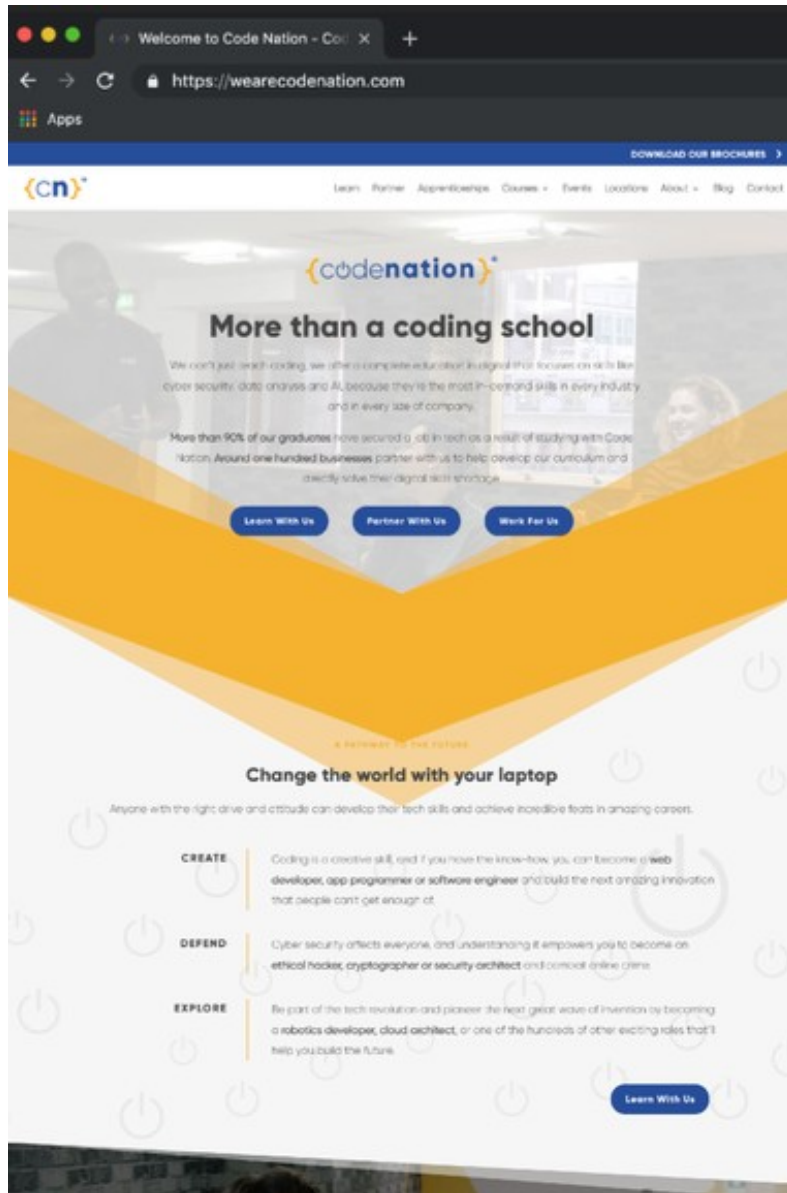
Make sure the path to the css file is correct



CSS

# A little tip!

... this is an important one!



# CSS

Always **start from the top** and **work to the bottom** when doing any styling

Think about the **whole body** (font, color). Think about **Navigation**. Think about **Headings**. Think about **Paragraphs**.





# HTML

## CSS Syntax

... rules are formed using a list of **properties** which we give values to.



# HTML

## CSS Properties

... apply to specific HTML Elements  
which we target using **selectors**.




# HTML

## CSS Selectors

...are used to find HTML Elements  
based on their **tag name, id, class,**  
**attribute** etc.

# The whole html!

```
* {  
  background-color: aliceblue;  
  font-size: 12px;  
}  
  
* {  
  property: value;  
  property: value;  
}
```





# Element Selector

```
h1 {  
    color: red;  
    font-size: 12px;  
}
```

```
elementtag {  
    property: value;  
    property: value;  
}
```

## ID Selector

```
#aboutme {  
    color: blue;  
    font-size: 20px;  
}
```

```
#idname {  
    property: value;  
    property: value;  
}
```

## Class Selector

```
.container {  
    background-color: firebrick;  
}
```

```
.classname {  
    property: value;  
}
```

## Group Selectors

You can set the same styling to different groups

```
h1, h2, p{  
  color: violet;  
}
```





HTML

# Defining in HTML

...through IDs, classes and more!

# IDs

CSS

Unique, cannot start with a number

```
<p id="para1">About Me</p>
#para1 {
  color: blue;
}
```

# Classes

Can have more than one style so add as many as required

```
<p class="center large">Hello World</p>
.center {
  text-align: center;
}
* another set of styling for "large"
```

# Example of CSS multiple classes

CSS

index.html

```
<p class = "makeMeRed">Hello</p>
<p class = "char1 makeMeRed">Hello</p>
<p class = "char2 makeMeRed">Hello</p>
<p class = "char1">Hello</p>
```

style.css

```
.makeMeRed {
    color: red;
}
.char1{
    font-family: 'Courier New', Courier, monospace;
}
.char2{
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}
```

Hello

Hello

Hello

Hello



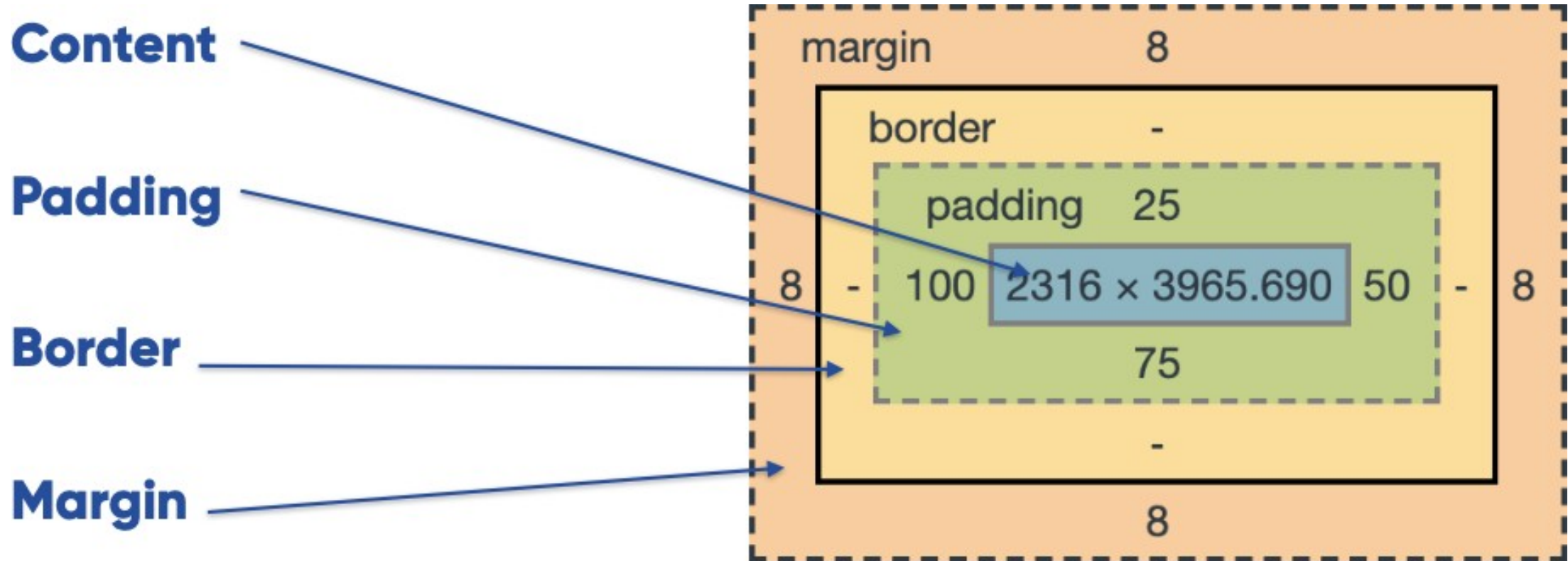
CSS

# The All Important ...box model



# Box Model: Design & Layout

CSS



# Box Model: Design & Layout



## Content

>The content of the box, where text and images appear.

## Padding

>Clears an area **around** the content. The Padding is transparent

## Border

>A border that goes around the padding and content.

## Margin

>Clears and area outside the border. The Margin is transparent.

# Box Sizing

CSS

Defines how the **width and height** of an element are calculated, should they include padding and border?

```
* {  
    box-sizing: border-box;  
}
```

# Padding Example

CSS

```
* {  
    box-sizing: border-box;  
    padding: 25px 50px 75px 100px;  
}
```

- Padding applied: top 25, right 50, bottom 75, left 100
- Three values: top, right and left, bottom
- Two values: top and bottom
- One value: all for paddings



CSS

**A little extra**  
...element sizing

# Height & Width of Elements

CSS

You can set the sizes of any element to how you like. These are the main ones to look for:

**Auto:** The browser calculates the width

**Length:** Defines the width in px for example

**%:** Defines the width in percentage of the containing block

```
img {  
  width: 100%;  
}
```

Set the image to 100% of the original size



CSS

**A little extra**  
...colors

# Three Main Color Values

CSS

**String**  
Tomato



**HEX**  
#ff6347



**RGB**  
255, 99, 71



There are many many more... (RGBA, HSL, HSLA)

{ CODENATION }





CSS

**A little extra**  
...fonts

## Font Properties

### Font Family

... changes font

### Font-style

... changes font style  
(normal, italicise).

### Font-size

... changes font size

### Font-weight

... changes font size  
width with pixels.

There are many many more.... (text-transform etc)

# Google Fonts

We can import fonts that are not commonly used in HTML.

[fonts.google.com](https://fonts.google.com) is the one to use.

```
@import url('https://fonts.googleapis.com/css?family=Roboto+Mono:400,500');  
  
h1 {  
  font-family: 'Roboto Mono', monospace;  
}
```

**Lets take a look...**



CSS

# A little extra

...media queries

# Media Queries

Media queries allow us to **change styles depending on particular parameters** like screen width or device type.

```
@media (max-width: 800px) {  
  .sidebar {  
    display: none;  
  }  
  
  .main {  
    width: 80%;  
  }  
}  
  
@media (min-width: 30em) and (orientation: landscape) {  
  #container {  
    flex-direction: column;  
    justify-content: center;  
  }  
}
```

Lets take a look...

# Learning Objectives

**To understand what CSS is and what it's used for**

**To understand what a few CSS syntax and selectors are and be able to write your own**

**To be familiar with some common CSS properties**

# Activity

# HTML

Add **CSS to all HTML pages** you have created so far.

Add **Class Attributes** to elements if you want them to share certain CSS rules. If you are targeting just one element, use an **ID attribute**.

## Stretch

You can add IDs so you can jump to a part of a page.

Read more here: <https://blog.hubspot.com/marketing/jump-link-same-page>



CSS

## More info...

<https://www.w3schools.com/css/> [https://  
developer.mozilla.org/en-US/docs/Web/CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)



# For tomorrow...

... take a look at **Flexbox**.

# CSS

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)

<https://www.youtube.com/watch?v=fYq5PXgSsbE>

Can you **name a Flex property**?  
How can you **centre a div in Flex box**?