

Intro to Front End - Responsive Design & Deployment

Set up the Live Stream

Front End Fundamentals

- How's everyone finding advanced CSS?

Objectives

- Responsive Design: Implementing media queries
 - Parallax scroll animation
 - How to deploy your project
-

Advanced CSS Review

Using this HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <link href="index.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <section class="parent">
      <div class="child-a">
        <p>A</p>
        <p>B</p>
        <p>C</p>
      </div>

      <div class="child-b">
        <p>D</p>
        <p>E</p>
        <p>F</p>
      </div>
    </section>
  </body>
</html>
```

Stacking

Use a comma to select more than one tag, id or class.

Live Code

Add this CSS

```
.child-a,  
.child-b {  
  background: darkslategrey;  
  text-align: center;  
  color: white;  
}
```

Adjacent Sibling

Use a “ + “ to select a sibling that immediately follows the first tag.

```
p + p {  
  font-size: 30px;  
}
```

General Sibling

Use “ ~ “ to select all siblings of a type that follows an element.

```
div ~ div {  
  border: 5px solid black;  
}
```

Pseudo Selectors

Allows us to style elements in a specific state:

- :hover
- :first-of-type
- :nth-of-type(n)
- :first-child

- :last-child

```
.child-b:hover {  
  background-color: lightblue;  
}  
  
div p:first-child {  
  color: coral;  
}  
  
p:nth-of-type(2) {  
  font-size: 40px;  
}
```

Pseudo Elements

Style parts of the element itself

- ::selection
- ::first-letter

```
p::selection {  
  color: blue;  
}
```

Transitions

Transitions animate changes that are applied to html elements when triggered by an event like hover

Transition shortcut: use the transition property then list the values in this order:

- Property to transition
- Transition duration
- Transition timing function
- Transition delay

```
.child-b {  
  transition: background-color 1s ease-in-out 1s;  
}
```

Media Queries

Let's talk a bit about what responsive design means.

Where do you often browse web pages from?

Have you encountered bad responsive design?

Media Queries are how we make our webpages responsive to different screen sizes - especially on mobile.

We use this to apply different styles based on screen size.

Examples:

- [Exmouth](#)
- [David's Tea](#)
- [Slack](#)

Open up Responsive Design CodePen

This uses flexbox for the layout and percentage heights for the 3 main elements.

When resize the elements are kinda responsive but when narrow it gets all squishy. We'll use media queries to fix that.

Breakpoints

When we write media queries we set breakpoints ie. the point at which our responsive styles are triggered.

These points are:

- Large Screens or Desktops: Any screen over 1024px wide; a tablet being viewed in Landscape mode is within this breakpoint as well.
- Medium Screens or Tablets: Any screen between 768px and 1024px.
- Small Screens or Phones: There is much more variety among smartphone widths but the largest of them are 480px wide.

You can set specific styles for each range if you want or set more general breakpoints (ie: if you want your webpage to look the same on desktops and tablets)

Syntax

Each piece of the query can be broken down:

- `@media` Keyword that says we are applying responsive styles.
 - `[media-type]` The media type. This can be screen, print, or speech. We're only going to need to use screen. Print is for printers and speech is used for screen readers.
 - `(expression)` Size of the screen we'll apply styles to. Often 'max-width' and 'min-width' values. These values give us the breakpoints for various standard screen sizes. An iPad is min-width: 768px and an iPhone has a min-width: 375px;
 - `css selections` Target the elements you want to style in the usual ways
-

Media Queries

What will this do?

This will change the font-size of all divs to 24px if the user is viewing the page on a screen with a max width of 660px to account for larger phones.

Live Code

We will add styles for any screen (not printer or screen reader) that is smaller than an iPad.

Add Media Query

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

Add Header/footer resizing

```
header,  
footer {
```

```
height: 10%;  
font-size: 16px;  
}
```

Add height to .content-container

```
.content-container {  
  height: 80%;  
}
```

Add image width resize and flex-direction column

```
@media screen and (max-width: 768px) {  
  .content-container {  
    flex-direction: column;  
  }  
  
  img {  
    width: 150px;  
    height: auto;  
  }  
}
```

Parallax NEW CODE PEN

Does anyone know what parallax is?

It's like this trippy effect where the image stays in place and the content scrolls over top
This is created by manipulating the background property.

Some use CSS transitions, and others use javascript. If you were to google parallax, you'll find hundreds of tutorials but we will be keeping it simple!

Examples:

- [Snowfall](#)
- [Firewatch](#)

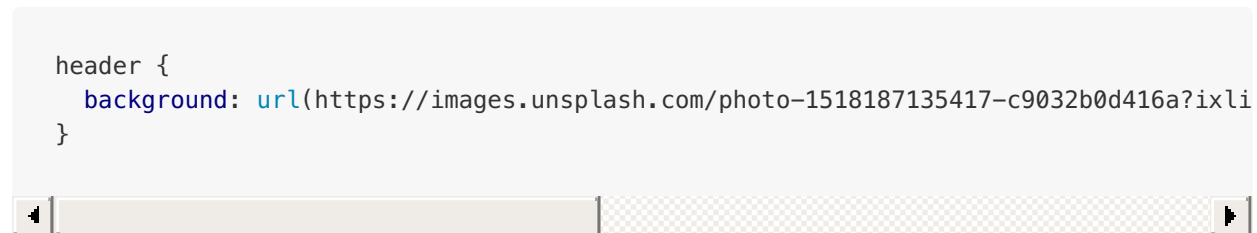
Background Images

Set Background Image:

You can set an image as a background.

“Background” is a keyword used as a property followed by a `url()` that contains the path to the file or the image’s url.

Change header background to an image and adjust height:



But of course now we can't see the image.

Set Background Size:

For Parallax we use `cover` which makes the image as large as possible without stretching it.

CSS crops either the image’s width or height so there is no empty space.

“contain” scales the image as large as possible without stretching OR cropping it

```
background-size: cover;
```

Set Background Attachment:

This property only works when a background image is specified. It determines if the image stays in place or scrolls with the content.

```
background-attachment: fixed;
```

Deployment

- Sign up for <https://getforge.com/>
- Create site on GetForge
- Zip your project folder on your computer
- Upload the zip file
- Navigate to your URL and