CHT2520 Advanced Web Programming

Matthew Mantle m.e.mantle@hud.ac.uk

Today's Session - Front-End Web Development (Responsive Web Design)

Front-End Web Development

- Front-End Web Technologies
  - HTML
  - CSS
  - JavaScript (JS)
- Run in a browser
- Look at CSS this week, JS next week

CSS

- CSS specifies the presentation (design) of a web page

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color:#646464;
    line-height: 2.0;
}
h1{
    color: #0295F3;
}
```

- We can select using tag names, ids, classes

CSS Good Practices

- Primarily select using the class selector

```
<div class="comment-holder">
<p id="comment" class="negative">This product broke the first time I used it.</p>
</div>
```

```
/*Good, this rule can be re-used in many different places*/
.negative{
    color:red;
}
```

- Avoid using the id, child and descendent selectors

  `#comment{...}` - Bad the rules can't be re-used

  `.comment-holder > .negative{...}` - Bad the rule can't be re-used

CSS Good Practices

- Create small re-usable CSS rules
  - Combine different classes for more complex styling

```
<li class="nav-opt">
<a class="nav-link light-green" href="about.html">FAQs</a>
</li>
<li class="nav-opt">
<a class="nav-link light-blue" href="products.html">Help</a>
</li>
```

```
.nav-opt{
    font-weight:bold;
}
.nav-link{
    text-decoration: none;
}
.light-green{
    color:#889c4d;
}
```

CSS Good Practices

- Many CSS properties are easy to use - color, line-height, margin, border etc.
- Controlling page layout is more challenging
  - Learn the basic `display` property.
  - There are many different options for page layout positioning, floats, flexbox, grids.
  - Favour modern techniques (flexbox and grids) over positioning and floats.

Mobile Web

- Mobile Web
  - Mobile devices accessing web sites e.g. smart phone, tablet
- Desktop
  - PC/Mac users using a physical keyboard and mouse
  - Includes laptop users
- Mobile web is bigger than desktop
  - Tipping point around 2016

Challenges of Mobile Web Design

- Smaller screen sizes
- User input
  - Touch input, no mouse, on-screen keyboard
- Mobile users tend to have slower internet connections
  - Increased latency, less bandwidth, pay for data?

Web Design Principles for Mobile

- Make buttons large and easy to select
  - Touch input
- Single columned design
  - Limited screen width
- Hide navigation on load
  - Limited screen space
- Limit text input in forms
  - Use radio buttons, checkboxes, limit 'free text' input form controls instead

Responsive Design

- The design responds to the device the user is using
  - The layout of the page changes based on the size of the display
    – E.g. Move from multi-columned design to single column, hide/show navigation
- Loads of examples e.g. https://www.vam.ac.uk/
  - Most modern websites are responsive
- Use developer tools to view pages in responsive mode

Implementing a Responsive Design

- Set the viewport

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="./css/style.css" type="text/css" rel="stylesheet" >
</head>
```

  - Resizes the content for the browser width

- Use fluid grids

  - set sizes using relative units e.g. em, fr, % etc.

Implementing a Responsive Design

- Media queries allow us to specify different CSS rules for different size displays

```
body{
    background-color:green;
}
@media screen and (min-width: 640px) {
    body{
        background-color:red;
    }
}
```

- On displays 640px or bigger the background will be red.
- It is considered good practice to design for 'mobile first'
  – Then add breakpoints to change the layout for larger screens

Implementing a Responsive Design

- Responsive Images
  - We can serve different images depending on the display size

```
<picture>
<source media="(min-width: 640px)" srcset="huddersfield-lrg.jpg">
<img src="huddersfield.jpg" alt="Photo of Huddersfield">
</picture>
```

Performance Considerations

- Page speed is really important on mobile!
  - ~2 seconds is the threshold
  - Site speed is a ranking factor in google
- There are different ways of measuring page load times e.g.
  - First Contentful Paint (FCP)
    - Any part of the page contents is rendered on screen
  - Largest Contentful Paint (LCP)
    - The largest image or block of text visible within the viewport is rendered on screen
- See: https://web.dev/vitals/#core-web-vitals for an overview of Google's metrics

Optimising Page Speed

- There are lots of things we can do to speed up a web site e.g.
  - Minimise CSS, HTML and JavaScript
  - Use CSS Instead of Images
  - Lazy load images and video
  - Use vector images where possible e.g. SVG
  - Optimise and use the correct image file format
  - Use the <picture> element to prevent unnecessary downloads

Practical Work

- Practical work - Implementing a Responsive Web Design
- Implementing a responsive design and showing understanding of front-end good practices/techniques is one topic you could explore for Assignment 2.