

## CP1406 Week 6 Practical – Media Queries and CSS Layout

Recall that a media query detects the media type (such as screen or print) and the capabilities of the device that the browser is running on (such as the size of the viewport in pixels or its orientation of portrait versus landscape). Based on the information, the media query applies styles that work well for that viewport. Last week, you created a responsive layout and added style rules for a mobile viewport. These style rules were not included within a media query so that tablet and desktop viewports will use them by default. To add style rules that target tablet or desktop viewports in particular, we will create a media query for each viewport.

For today's practical, you need to submit:

- A single zip file containing
  - The **Fitness folder**. The "Fitness" folder includes:
    - html files (index.html, template.html, contact.html, and aboutus.html) you have created for the site
    - a CSS file
    - the "images" folder contains images for the "Fitness" website.
  - The **Training Zone** folder. The "Training Zone" folder includes:
    - HTML files you have created for the site
    - a CSS file for the "Training Zone" website
    - the "images" folder contains images for the "Training Zone" website
  - Part 7 html file.

### Part 1 – Create a Media Query for a Tablet Viewport

You have already completed the mobile style rules and now need to create a media query to target tablet devices. The media query will adjust the page layout to better display content when viewed on tablet device.

- Open **styles.css** file from the css folder inside the "Fitness" folder to prepare to insert a media query.
- Above the **/\* #exercises {** selector line, add the following code (Figure 1):

```
/* Media Query for Tablet Viewport */
@media screen and (min-width: 630px) { /*insert a media query*/

}
```

```
/* Media Query for Tablet Viewport */
@media screen and (min-width: 630px) {

}

/* #exercises {
  clear: left;
  border-top: 1px solid #000;
  border-bottom: 1px solid #000;
```

Figure 1

## Part 2 – Showing and Hiding Content for a Tablet Viewport

The content within the main area of the Forward Fitness Club webpages is ideal for a mobile user. In the HTML documents, this content appears in a `div` element with an attribute and value of `class="mobile"`. You will create a new style rule to hide the mobile class in the tablet viewport with a display property and value of `none`. This is because a tablet screen is larger than a mobile screen, it can display more content.

- Place the insertion point at the end of the `@media screen and (min-width: 630px) {` line, and press the enter key twice to insert two new lines.
- On the second new line, indent if necessary, and then type the following code (also see Figure 2):

```
/* Tablet Viewport: Show tablet-desktop class, hide mobile class */  
.tablet-desktop {  
    display: block;  
}  
  
.mobile {  
    display: none;  
}
```

- Save your changes, open index.html in Google Chrome's device mode, and set the viewport width to 630px to simulate a tablet viewport. Scroll down to view your changes.

```
@media screen and (min-width: 630px) {  
  
    /* Tablet Viewport: Show tablet-desktop class, hide mobile class */  
    .tablet-desktop {  
        display: block;  
    }  
  
    .mobile {  
        display: none;  
    }  
  
}/* media query closing brace */
```

Figure 2

## Part 3 – Removing a Sticky Position for the Header for a Tablet Viewport

The purpose of the sticky header was to keep the company name and logo in view for a mobile device user. Because the screen size is larger for a tablet user, you can remove the sticky position. We can improve the layout for a tablet device user by making the header position static.

- Add the below code above the media query closing brace (also see Figure 3):

```
/* Tablet Viewport: Style rule for header */
header {
    position: static;
    padding-bottom: 2%;
}
```

- Save your changes, refresh `index.html` in Google Chrome's device mode, and then scroll down to view your changes

```
/* Tablet Viewport: Style rule for header */
header {
    position: static;
    padding-bottom: 2%;
}

}/* media query closing brace */

/* #exercises {
```

Figure 3

## Part 4 – Styling the Navigation Area for a Tablet Viewport

You will add a new style rule to display the list items in a single horizontal line and display the border between each link. This is to improve the webpage layout for a tablet viewport by displaying the navigation horizontally.

- Add the below code above the media query closing brace (also see Figure 4):

```
/* Tablet Viewport: Style rules for nav area */
nav li {
    border-top: none;
    display: inline-block;
    border-right: 1px solid #fff;
}

nav li:last-child {
    border-right: none;
}

nav li a {
    padding: 0.1em 0.75em;
}
```

```

/* Tablet Viewport: Style rules for nav area */
nav li {
  border-top: none;
  display: inline-block;
  border-right: 1px solid #fff;
}

nav li:last-child {
  border-right: none;
}

nav li a {
  padding: 0.1em 0.75em;
}
}/* media query closing brace */

```

Figure 4

## Part 5 – Designing for Desktop Viewports

You used a simple style sheet in the previous weeks to create a desktop layout. Now you will create a similar layout using responsive design and create a media query for a desktop viewport. The same desktop design principles apply for responsive design; use simple, intuitive navigation, clear images, and typography and apply the same colour scheme.

The desktop media query will contain styles to alter the website appearance for desktop users. The following steps add a media query to target desktop viewports.

- Below the closing brace `}` for the media query **@media screen and (min-width: 630px)**, add the below code on the second new line (see Figure 5):

```

/* Media Query for Desktop Viewport */
@media screen and (min-width: 769px) {

}

```

```

}/* media query for Tablet Viewport closing brace */

/* Media Query for Desktop Viewport */
▼ @media screen and (min-width: 769px) {

}

▼ /* #exercises {

```

Figure 5

You will create a two-column layout for the header and nav elements. You will start by creating a style rule for the header element that floats it to the left and sets the width to 25%. The tablet media query specifies bottom padding for the header. Because this style rule is inherited by the desktop media query, remove the bottom padding.

- At the end of the **@media screen and (min-width: 769px)** {}, add the following code (also see Figure 6):

```
/* Desktop Viewport: Style rule for header */
header {
  width: 25%;
  float: left;
  padding-bottom: 0;
}
```

```
/* Media Query for Desktop Viewport */
@media screen and (min-width: 769px) {

  /* Desktop Viewport: Style rule for header */
  header {
    width: 25%;
    float: left;
    padding-bottom: 0;
  }

}/* media query for Desktop Viewport closing brace */
```

Figure 6

- Save your changes, refresh **index.html** in Google Chrome's device mode, and then scroll down to view your changes.

You will now style the navigation area to appear as a single unified element, to the right of the header element, with navigation links evenly distributed within the area.

- Add the below code above the **media query for Desktop Viewpoint** closing brace (also see Figure 7):

```
/* Desktop Viewport: Style rules for nav area */
nav {
  float: right;
  width: 70%;
  margin: 4em 1em 0 0;
}
```

- Save your changes and refresh **index.html** in your browser to view your changes.

```

/* Desktop Viewport: Style rules for nav area */
nav {
  float: right;
  width: 70%;
  margin: 4em 1em 0 0;
}
}/* media query for Desktop Viewport closing brace */

```

Figure 7

The next step to create a single unified navigation area is to create a new style rule for the unordered list within the nav element. You will create a style rule that aligns the text to the right and style the main element for a desktop viewport. Then create a new style rule for the list items and list item links.

- Add the below code above the **media query for Desktop Viewpoint** closing brace (also see Figure 8):

```

nav ul {
  text-align: right;
}

nav li {
  border: none;
}

nav li a {
  padding: 0.5em 1em;
}

```

```

nav ul {
  text-align: right;
}

nav li {
  border: none;
}

nav li a {
  padding: 0.5em 1em;
}
}/* media query for Desktop Viewport closing brace */

```

Figure 8

- Save your changes, refresh **index.html** in Google Chrome's device mode, and then scroll down to view your changes.

The appearance of the main content area appears normal on the home page; however, when you view the About Us and Contact Us pages, the page layout appears distorted. This is due to the float

properties applied to the header and nav elements. To correct the layout issue, create a new style rule for the main element that specifies a clear property with a value of left.

Then you will adjust the font size for heading 1 elements in the main element to enhance their appearance to complement a multiple-column layout.

- Add the below code above the media query for Desktop Viewpoint closing brace (also see Figure 9):

```
main {  
    clear: left;  
}  
  
main h1 {  
    font-size: 1.8em;  
}
```

```
/* Desktop Viewport: Style Rules for main content */  
main {  
    clear: left;  
}  
  
main h1 {  
    font-size: 1.8em;  
}  
  
}/* media query for Desktop Viewport closing brace */
```

Figure 9

- Save your changes and open **about.html** in Google Chrome's device mode, and then scroll down to view your changes.

To create a three-column layout for the weights, cardio, and personal training div elements. You will make a three-column layout by create a new style rule for the **#weights**, **#cardio**, and **#training** selectors and specify float, width, and margin properties. Then create another style rule to clear the float for the exercises div element.

- Add the below code above the media query for Desktop Viewpoint closing brace (also see Figure 10):

```
#weights, #cardio, #training  
{  
    width: 29%;  
    float: left;  
    margin: 0 2%;  
}  
  
#exercises {  
    clear: left;  
}
```

- Save your changes and refresh about.html in your browser to view your changes

```
#weights, #cardio, #training
{
    width: 29%;
    float: left;
    margin: 0 2%;
}

#exercises {
    clear: left;
}

}/* media query for Desktop Viewport closing brace */
```

Figure 10

## Part 6 – Creating Tablet Viewport for your Training Zone webpages.

**Your Turn:** Now that you have some experience making the Forward Fitness Club webpages responsive by using media queries, make your Training Zone webpages suitable to be viewed on tablets.

## Part 7 – CSS Layouts

Spend some time learning more about CSS grids by playing with

[https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)

**Your Turn:** Download the and unzip **layout\_starter.zip** provided for the html file you will be working on for this part. We have embedded the starter CSS code in this for simplicity. Write the rest of the CSS needed to achieve the layout shown in the screenshot below (Figure 11):

**Note on colours:** #03B3A0; =  and #527da4; =



Figure 11



## Deliverables:

- A single zip file containing
  - The **Fitness folder**. The “Fitness” folder includes:
    - html files (index.html, template.html, contact.html, and aboutus.html) you have created for the site
    - a CSS file
    - the “images” folder contains images for the “Fitness” website.
  - The **Training Zone** folder. The “Training Zone” folder includes:
    - HTML files you have created for the site
    - a CSS file for the “Training Zone” website
    - the “images” folder contains images for the “Training Zone” website
  - Part 7 html file (with CSS embedded).