

Toggle Nav – Walkthrough Instructions

Introduction

This lesson will be taught in three stages:

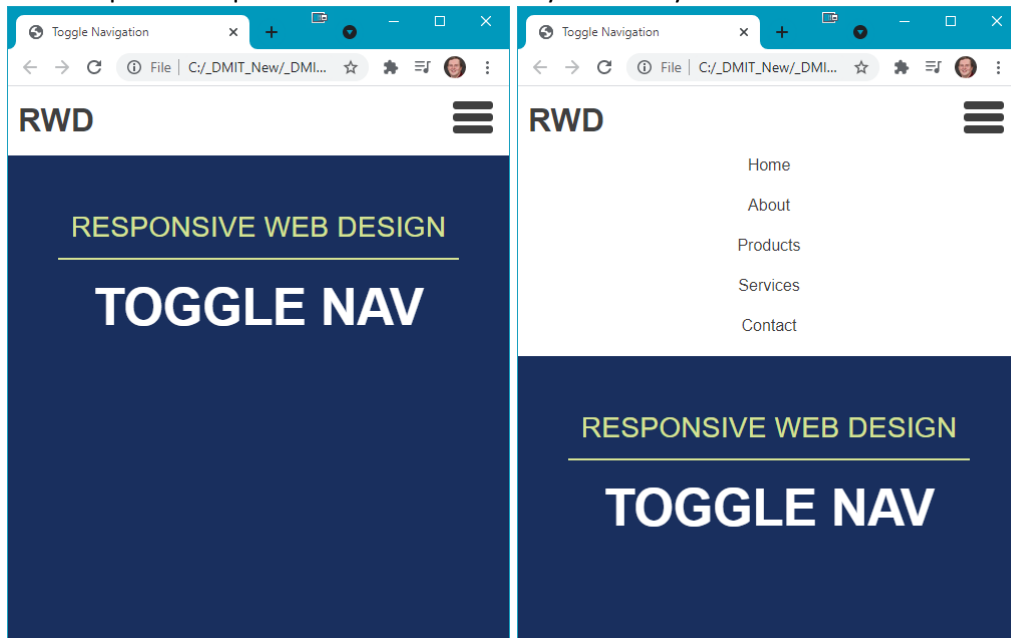
- Stage 1: Basic Toggle Nav
- Stage 2: Toggle Nav Version 2
- Stage 3: Toggle Nav Version 3

Demo Instructions

You can follow along with your instructor to complete this build and/or you can use this document as a guide in completing the demo build.

Steps – Stage 1: Basic Toggle Nav

1. Download the **demo-toggle-nav.zip** file from Moodle and extract its contents to a folder named **demo-toggle-nav**.
2. The completed output should look like when you resize your browser:



Note: When the browser is resized, the menu will start to wrap to the next row.

3. Add the following code to your **index.html**:

- a. Add the <header> block:

```
<header>
  <div class="flex-container">
    <h1><a href="index.html">RWD</a></h1>
    <button class="toggle-btn">
      <svg viewBox="0 0 100 80" width="40" height="40">
        <rect width="100" height="20" rx="8"></rect>
        <rect y="30" width="100" height="20" rx="8"></rect>
        <rect y="60" width="100" height="20" rx="8"></rect>
      </svg>
    </button>
  </div>
  <nav>
    <ul class="menu">
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Products</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</header>
```

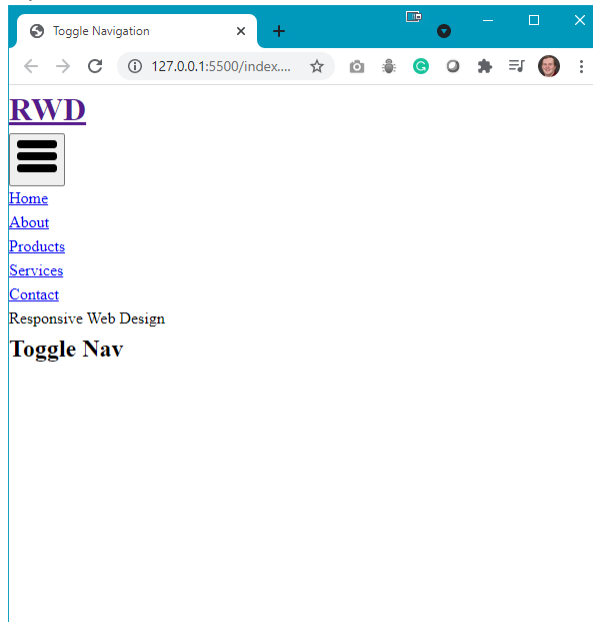
- b. Add the <section> block:

```
<section>
  <p>Responsive Web Design</p>
  <h2>Toggle Nav</h2>
</section>
```

- c. Add the following code, which **must** be the last line of code in the <body>:

```
<!-- link to the js file here - before the /body tag -->
<script src="js/main.js"></script>
```

Open **index.html** in the browser to see:



Question: What needs to be done now?

4. Open **styles.css** and add the following:

- a. You will need a **.flex-container** class:

```
.flex-container {  
  display: flex;  
  justify-content: space-between;  
}
```

- b. Start styling the header and nav:

```
/* ***** header & nav ***** */  
/* Step: 1 Style the default/showing state. Starting with the mobile view,  
style the layout and links for how they will look when visible */  
header {  
  background-color: #fff;  
  padding: 2%;  
}  
  
header a {  
  color: #3f3f3f;  
  text-decoration: none;  
  /* allows the clickable area to take up the full width and height of the li */  
  display: block;  
}
```

- c. Add styling for the menu and hover effects:

```
.menu a:hover,  
.menu a:active {  
  background-color: #gainsboro;  
  border-radius: 4px;  
}
```

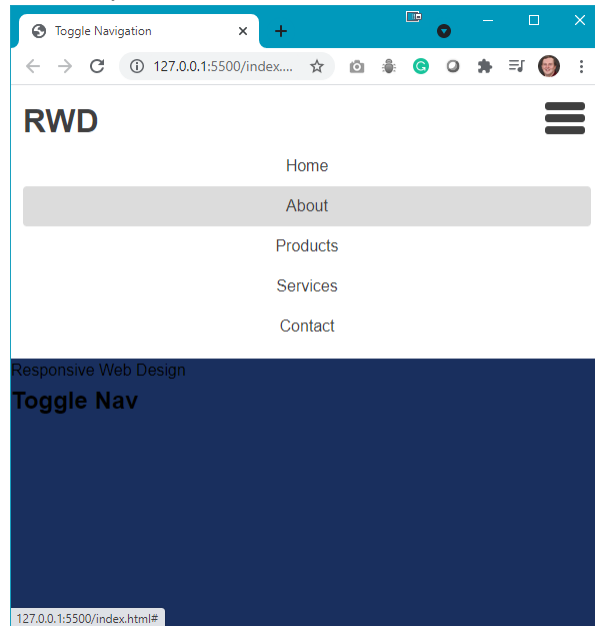
- d. Style the toggle button:

```
.toggle-btn {  
  border: none;  
  background-color: #fff;  
  outline: none;  
}  
  
.toggle-btn:hover {  
  cursor: pointer;  
}  
  
svg rect {  
  fill: #3f3f3f;  
}
```

- e. Style the menu list items:

```
.menu li {  
  /* the extra space from the additional line height, makes this a larger area  
and vertically centers the text */  
  line-height: 40px;  
  text-align: center;  
}
```

f. Refresh your browser to see:



Note: The *hamburger* menu icon does not collapse the menu; it is still fixed in place.

Question: What is the next step?

5. The next step is to hide the menu. Add the following rules to your **styles.css**:

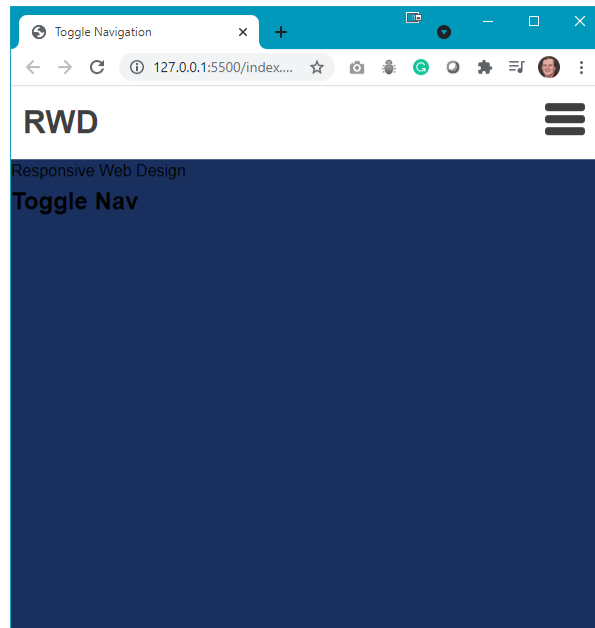
a. Style the **header nav**:

```
/* Step 2: Now let us hide the menu. Once it is hidden this is the default load stage.
If using display: none; - that property cannot be transitioned. */
header nav {
  max-height: 0;
  overflow: hidden;
  transition: max-height 0.7s ease-in-out;
}
```

b. Create a class that will show the menu:

```
/* Step 3: Create a special class that will show the menu. This class name exists only
in the CSS file for now. Next the javascript will add this class to the nav element for us.
Let us go and do that in the main.js file */
.show-nav {
  max-height: 12.5rem;
}
```

Refresh your browser to see (press the *hamburger* icon to see the changes):



Question: What is the next step?

6. Next the `<section>` needs some styling so add the following style rules:

a. Initial section styling:

```
/* ***** section ***** */
section {
  padding: 3.125rem;
  text-align: center;
  text-transform: uppercase;
}
```

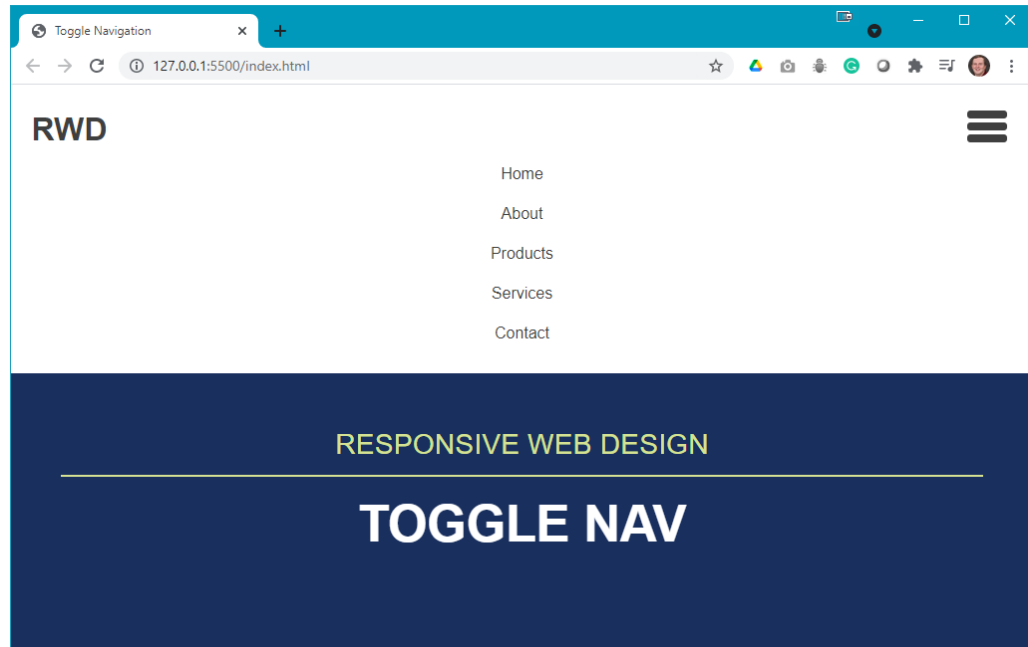
b. Style the paragraph:

```
section p {
  color: #d5e294;
  border-bottom: 2px solid #d5e294;
  padding-bottom: 10px;
  margin-bottom: 10px;
  font-size: 1.777rem;
}
```

c. Style the heading:

```
section h2 {
  color: #fff;
  font-size: 3.157rem;
  margin-bottom: 2rem;
}
```

Refresh your browser to see (you will need to resize your browser):



Question: What needs to be done next?

7. A media query is needed, thus add the following media query:

a. Hide the toggle button (*hamburger* icon):

```
/* Step 4: add a media query and show the nav by default (hide the toggle button)
and style it as a horizontal nav */
@media only screen and (min-width: 37.5em) {
  .toggle-btn {
    display: none;
  }
}
```

b. Show the nav by default, thus add the following to your media query:

```
/* we want to see the nav by default, so give it a height back */
header nav {
  max-height: 12.5rem;
  overflow: visible;
}
```

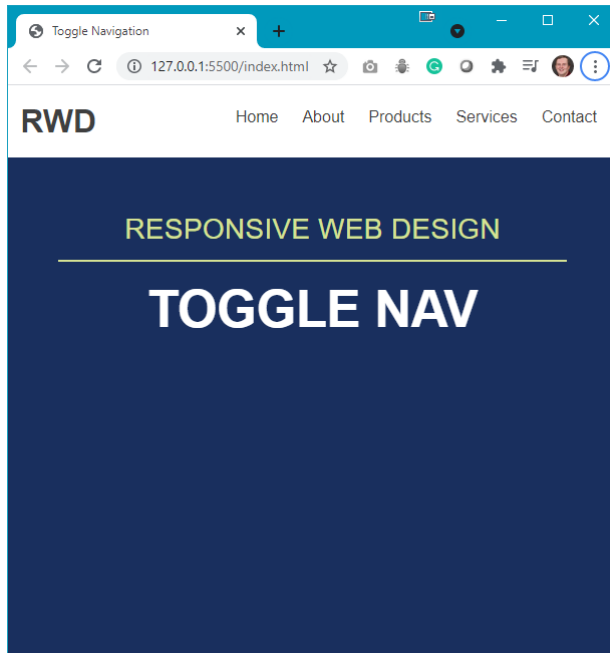
c. Add menu styling to your media query:

```
.menu,
header {
  display: flex;
  justify-content: space-between;
}

.menu li {
  margin-left: 0.5rem;
}

.menu li a {
  padding: 0 0.5rem;
}
```

8. The result, at a wider browser width, is:



Question: When the toggle button is pressed, the section content is pushed down. Would it not be better if the dropdown menu were an overlaid menu?

Steps – Stage 2: Toggle Nav Version 2

1. To get the desired effect make the following changes to your **styles.css** file:

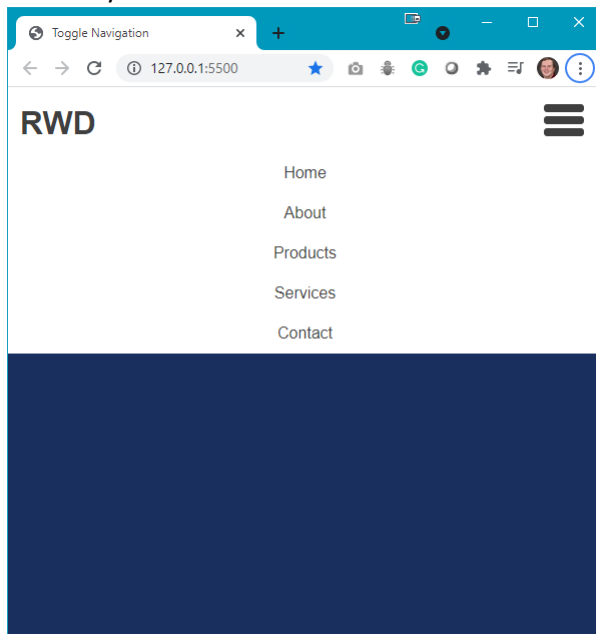
a. Modify the **header nav** style rule as shown below:

```
header nav {  
  
    /* style this for how it will look overlapped the main content */  
    /* this element does not have a parent container that is positioned relative,  
    therefore it is positioned in relationship to the browser screen */  
    position: absolute;  
    top: 4.125rem; /* the height of the closed header */  
    left: 0;  
  
    /* once positioning absolute, this disappears from the document flow and it  
    does not leave a space where it was. This does not have a background color,  
    because it is not content the header sees anymore */  
    background-color: rgba(255, 255, 255, 1);  
    width: 100vw;  
  
    max-height: 0;  
    overflow: hidden;  
    transition: max-height 0.7s ease-in-out;  
}
```

b. Add some padding to the menu (add this below the **header nav** style rule):

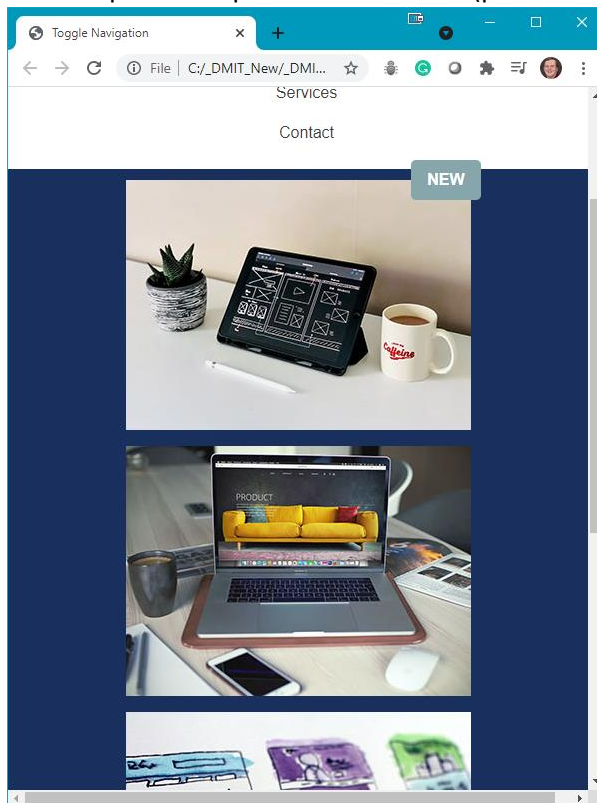
```
ul.menu {  
    padding-bottom: 1rem;  
}
```

2. Refresh your browser to see the menu now overlays the section content.



Steps – Stage 3: Toggle Nav Version 3

1. Download the **toggle-nav-3-images.zip** file from Moodle and extract the images to your **img** folder.
2. The completed output should look like (partial view shown):



3. The first step is to add the following to your **index.html**:

a. Add the following code below the `<h2>` in the `<section>`:

```
<div class="gallery">
  <div class="thumbnail">
    <span>New</span>
    
  </div>
  <div class="thumbnail">
    
  </div>
  <div class="thumbnail">
    
  </div>
  <div class="thumbnail">
    
  </div>
</div>
```

You should see the images below the heading.

4. Now add the following style rules below the section h2 style rule:

a. Style the **.gallery** class:

```
.gallery {
  display: flex;
  flex-flow: wrap;
  justify-content: center;
}
```

b. Style the **.thumbnail** class:

```
.thumbnail {
  margin: 0 0.5rem;
  position: relative;
}

.thumbnail img {
  margin-bottom: 1rem;
}
```

c. Style the `` element:

```
span {
  background-color: #87a6ab;
  color: #fff;
  font-weight: bold;
  padding: 0.5rem 1rem;
  border-radius: 5px;
  /* position this span over the image */
  position: absolute;
  /* The top, right, bottom, left values position in relation to the screen
  UNLESS the parent container is a relative positioned element */
  top: -20px;
  right: -10px;
}
```

5. Test the results in your browser; use different browser widths to see the full effects.