Community College of Baltimore County

Transforming the library website (catalog) to a responsive mobile site

Screen size: 1280px in width

660px in width

320 px in width

1 Why a responsive site?

2013 → 2015

3% → 11%

Library's **mobile usage** continues to grow.
Responsive designs adapt to multiple screen sizes.

2 Solutions we tried.

1. Purchasing III's mobile product, AirPAC.
2. Adopting one of mobile design templates developed by other librarians.
3. Modifying CSS

3 Experienced limitations of AirPAC

1. AirPAC provides a seperate home page for small devices only. The mobile homepage's content is different from the regular homepage. Because librarians have no privilege to edit the homepage, they need to ask III to modify the content.
2. AirPAC doesn't adapt to various in-between screen sizes.

4 Experienced limitations of using a design template

1. It resulted in a whole change of the website. Therefore, it took a lot of efforts and time to readjust design elements for my institution's design requirement.
2. In particular, the main homepage's structure and design elements didn't work well with CCBC's needs.

Modifying CSS (styles.css in WebPac)

5 Pre-requisite: Setting The Viewport in HTML

```
<meta name="viewport"
content="width=device-width, initial-scale=1.0">
```

width=device-width : To set the width of the page to follow the screen-width of the device

initial-scale=1.0 : To set the initial zoom level when the page is first loaded by the browser.

Basic Rules

1. Avoid using large fixed width elements. Consider using relative width/height values.
2. Use CSS media queries to apply different styling for small and large screens.

6 Media Query in CSS to add breakpoints

- Code example:

```
@media only screen and (max-width: 500px) {
  #header {   background-image:
  url(/screens/images/Library-Header2_sm_v1.jpg);
        }
}
```

- Meaning:
If the browser window is smaller than 500px, the header image will be replaced with a smaller one.

7 Multiple progressive breakpoints for size of elements

```
@media screen and (max-width: 500px) {
    *** CSS code  ***
}

@media screen and (max-width: 700px) {
    *** CSS code ***
}

@media screen and (max-width: 900px) {
    *** CSS code ***
}
```

and so on...

8 Collapsible menu for small devices (toplogo.html)

[1] Add a li for menu icon and hide this for a screen bigger than 700px

```
<li class="icon-1">
  <a href="javascript:void(0);" onclick="topnav_1_ad
        dclass()">&#9776;</a>
</li>
```

[2] Hide all other li except the first li and show the menu icon for a screen smaller than 700px.

[3] By clicking the menu icon, you can toggle between the full menu and the collapsible menu.

What I experienced:

1. It didn't require as much of coding and time as I initially thought.
2. It was better to change/add codes little by little.
3. A browser's developer tool (F12) was a must to make my life easier.