9. Responsive Webpages

Making a webpage responsive to the screen size is an important features in web technology. Nowadays, web pages open on devices with different screen sizes, for example a **desktop computer** screen has a width of 1200px, a Samsung Galaxy Tab 10.1 tablet has the width of 900px, and the width of a Samsung Galaxy J7 smartphone is 720px . Therefore, it is important that when you design a webpage, the elements on the webpage can adjust to the screen of the device. You can visit: http://screensiz.es/ to view more devices width.

For app view development, it is an important to add the following line:

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

The **`viewport`** is the user's visible area of a web page. It varies with the device, and will be smaller on a mobile phone than on a computer screen. **`viewport`** element gives the browser instructions on how to control the page's dimensions and scaling.

The **`width=device-width`** part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The **`initial-scale=1.0`** part sets the initial zoom level when the page is first loaded by the browser.

Without the viewport meta tag

With the viewport meta tag

@media query

Media query is a CSS technique introduced in CSS3 and it is used to make responsive pages.

It uses the @media rule to include a block of CSS properties only if a certain condition is true.

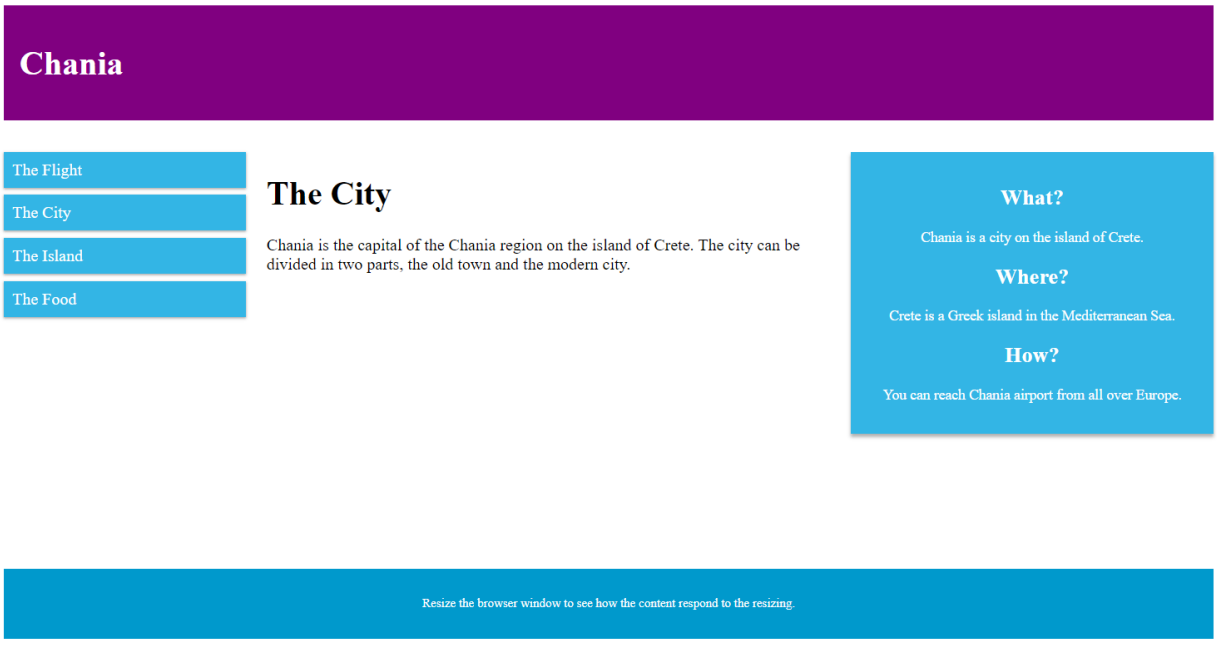
When using @media, instead of changing styles when the width gets *smaller* than 720px, we should change the design when the width gets *larger* than 720px. This will make our design Mobile First. The syntax code will look as:

```
@media only screen and (min-width: 720px){

}
```

Between the curly brackets should go the CSS attributes of the elements that will changed when the screen has the width of 720px or greater.

Mobile First means designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices). Some web developer prepares to design a mobile view first as it moves toward the tablet's, laptop's, and desktop's screen size. Therefore, when we apply @media query, the screen size has property **`min-width: 720px`**. On the other hand, since the material in this lab manual was designed from a desktop computer screen view, then we can design from the desktop computer screen toward the tablet's and smartphone's screen size. For this, instead of using **`min-width:720px`** we use **`max-width: 720px`**

Example 9.1) Create the following layout for a desktop screen:



HTML

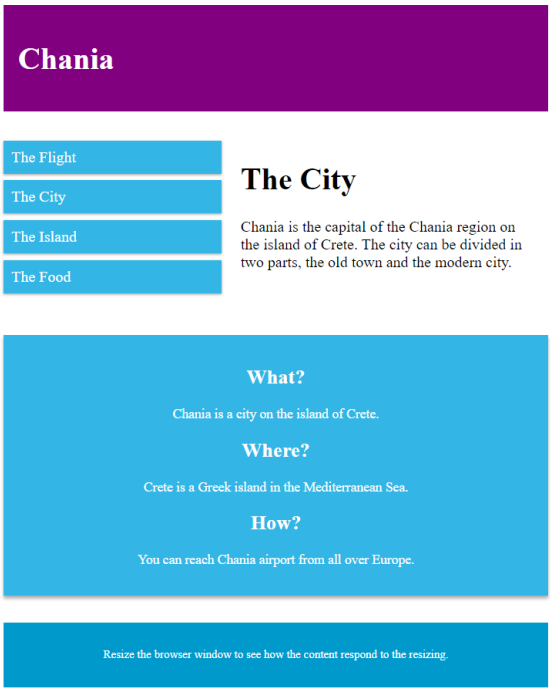
```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Responsive Webpage by Prof. Wu</title>
    <link rel="stylesheet" href="index.css" type="text/css">
  </head>
  <body>

  <div class="header">
    <h1>Chania</h1>
  </div>

  <div class="container">
    <div class="col_left">
      <ul>
        <li>The Flight</li>
        <li>The City</li>
        <li>The Island</li>
        <li>The Food</li>
      </ul>
    </div>

    <div class="col_middle">
      <h1>The City</h1>
       <p>Chania is the capital of the Chania region on the island of Crete. The
       city can be divided in two parts, the old town and the modern city.</p>
    </div>

    <div class="col_right">
      <div class="aside">
        <h2>What?</h2>
        <p>Chania is a city on the island of Crete.</p>
        <h2>Where?</h2>
        <p>Crete is a Greek island in the Mediterranean Sea.</p>
        <h2>How?</h2>
        <p>You can reach Chania airport from all over Europe.</p>
      </div>
    </div>
  </div> <!-- end of container -->

  <div class="footer">
       <p>Resize the browser window to see how the content respond to the
       resizing.</p>
  </div>

  </body>
</html>
```

index.css

```
body{padding: 20px;}
* { box-sizing: border-box; }
.header {
  background-color: purple; color: white; padding: 15px;
}
.container{
  width: 100%; height: 400px; margin-top: 30px;
}
/*-- left column --*/
.col_left{
  width: 20%; float: left; height: auto;
}
.col_left ul {
  list-style-type: none; margin: 0; padding: 0;
}
.col_left li {
  padding: 8px;
  margin-bottom: 7px;
  background-color: #33b5e5;
  color: #ffffff;
  box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);
}
.col_left li:hover { background-color: #0099cc;
}
/*-- middle column --*/
.col_middle{
  width: 50%;
  height: auto;
  float: left;
  padding: 0px 20px 20px 20px;
}
/*-- right column --*/
.col_right{
  width: 30%;
  height: auto;
  float: left;
}
.aside {
  background-color: #33b5e5;
  padding: 15px;
  color: white;
  text-align: center;
  font-size: 14px;
  box-shadow: 0 3px 3px rgba(0,0,0,0.3);
}
.footer {
  background-color: #0099cc;
  color: white;
  text-align: center;
  font-size: 12px;
  padding: 15px;
}
```

Breakpoint of a tablet screen size view

Once the desktop view is set, we can create a responsive webpage to a tablet of 1024px of width. The following layout for a tablet screen with a maximum width of 1024px:



In CSS, we can write a media query with a maximum width of **`1024px`** as the following

```
@media only screen and (max-width: 1024px){

}
```

Within the curly brackets, we will need to change the height of the **`container`** element to **`550px`**. We need to change the **`height`** of the container, otherwise the **`.col_right`** will overlap. Once the **`height`** of the **`container`** element is set, we need to set the **`.col_left`** element's **`width`** to **`40%`** of the container element, **`.col_middle`** element's **`width`** to **`60%`**, and the **`.col_right`** element's **`width`** to **`100%.`**

We can add the media query code to the end of the index.css

index.css

```
* Responsible to a tablet of 1024px width*/
@media only screen and (max-width: 1024px) {
  .container{ height: 550px; }
  .col_left{ width: 40%; }
  .col_middle{ width: 60%; }
  .col_right{
    width: 100%;
    margin-top: 30px;
    margin-bottom: 50px;
  }
}
```

When we refresh the internet browser, we can see the change or rearrangement of the elements to fit the window screen of at least **1024px**.

Breakpoint of a smartphone screen size view

Once the tablet view is set, we can create a responsive webpage to a smartphone screen width of 720px. The following layout for a smartphone screen with a maximum width of 720px:

index.css

```
/* Responsive webpage of of 720px width*/
@media only screen and (max-width: 720px) {
  .container{ height: 700px; }
  .col_left{ width: 100%; }
  .col_middle{ width: 100%; }
  .col_right{
    width: 100%;
    margin-top: 30px;
    margin-bottom: 50px;
  }
}
```

When we refresh the internet browser, we can see the change or rearrangement of the elements to fit the window screen of at least **1024px.** If we keep decreasing the width of the internet browser, we can see the elements to rearrange again to fit the window screen of at least **720px**.