Basics of Responsive Web Design

To create a responsive website, we should know the below 3 main parts

1. Fluid Grid - It's a flexible width path. We should stop using pixel-based sizes, instead we use the em or percentage in the stylesheet. This feature help us to make designing for multiple screens easier. Here the column widths are proportional rather than fixed. Fluid web page design can be more user-friendly, because it adjusts to the user's set up.

For example: width: 1126px; will be width: 98%;

2. Flexible Images - The usage of fluid images causes the adjustment of the size to the parent block. The images will scale out according to the screen resolution/size. If the parent block is smaller than the size of image then the image is reduced proportionally.

The most common relative solution is to set the max-width of the image at 100%. The max-width style means that an image won't exceed the width of its container. Instead of specifying a width and height on the image tag, its best just to add the image tag without that information and rely on the max width.

3. Media Queries(@media) - Media queries allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser.

Media queries are used to write css for specific situations, which allows you to apply styles based on the information about device resolution. It can be set to detect such features as width, height, screen orientation, aspect-ratio and resolution. And also used to change the layout sizes and rules based on various devices. We have to specify some break points in the CSS.

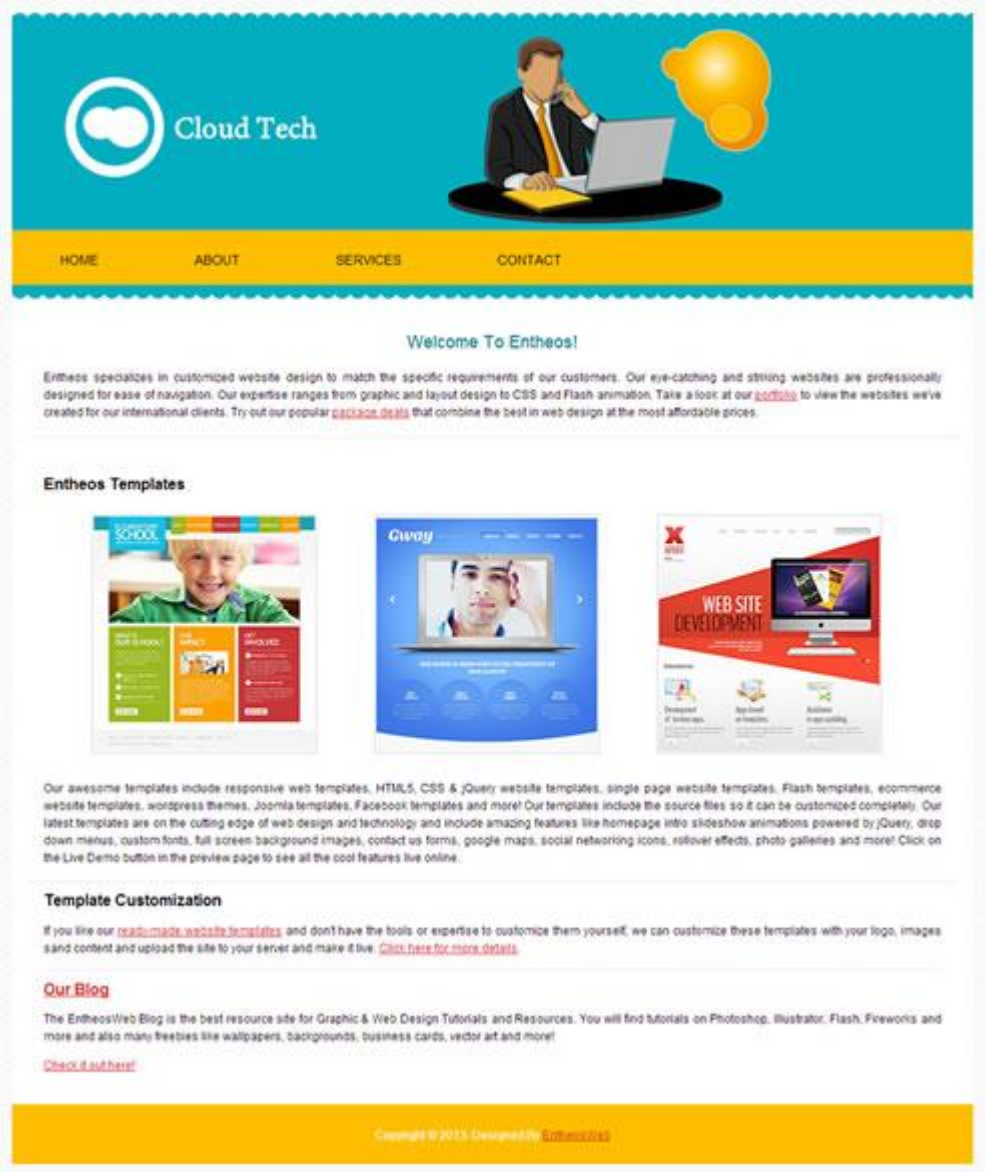
@media only screen and (max-width: 768px) {}
@media only screen and (max-width: 320px) {}

Today you will learn how to create a simple responsive website.

style.css - used for stylesheet files
images - used to store the used images

Tutorial

Step 1 : First let's design a simple website layout like the below image.



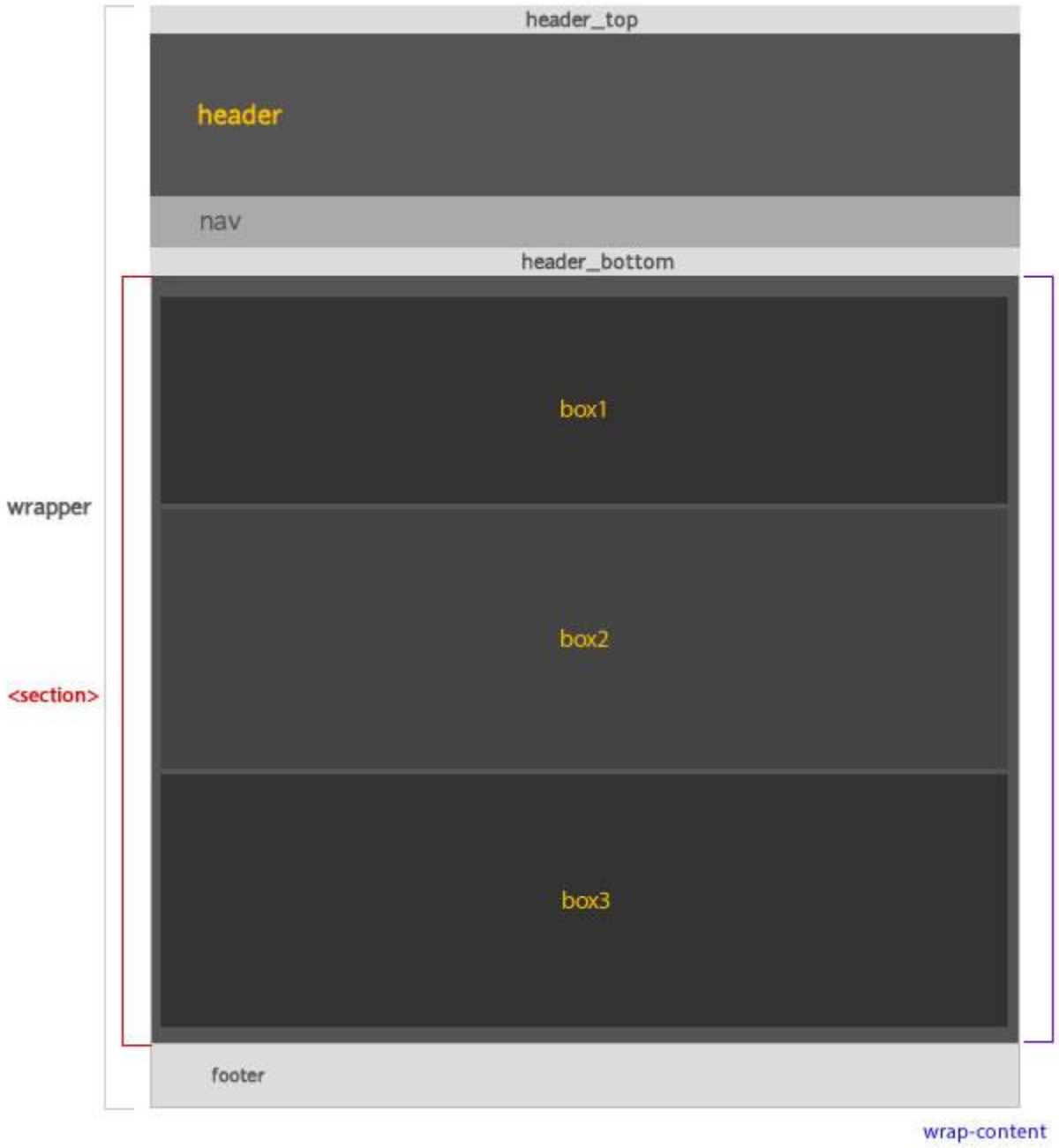
Step 2 : I have three different breakpoints setup to achieve various effects when resizing the browser window. So here in the tutorial the break point is 1126px for desktop, 768px for tablet and 320px for iphone.

Step 3 : You have to use the **viewport meta tag** in your **<head>** for the responsive website. The viewport meta tag enables web developers to indicate that the web page they built is optimized for mobile devices. It's generally used for responsive design to set the viewport width and initial-scale on mobile devices.

<meta name="viewport" content="width=device-width, maximum-scale=1.0, initial-scale=1.0", user-scalable=no" />

- **width** - device width of the viewport in pixels.
- **height** - device height of the viewport in pixels.
- **initial-scale** - sets the the initial scaling of the viewport. The 1.0 value displays an unscaled web page.
- **user-scalable** - specifies whether the user can scale the web page (zoom in or zoom out). Can get the yes or no values.
- **maximum-scale** or **minimum-scale** - sets the maximum or minimum scaling for the web page. Can get values between 0.25 to 10.0.

Step 4 : Next, I have designed a rough HTML Structure for the responsive page layout with a header, nav, wrapper, section, wrap-content, box and a footer. Building your site with these structures in mind makes it easy to imagine and code the styles.



The basic HTML structure is

```
<html>
   <head>

   <title>Responsive Website Tutorial</title>

   </head>
   <body>

   <header>   </header>
   <section>   </section>
   <footer>    </footer>

   </body>
</html>
```

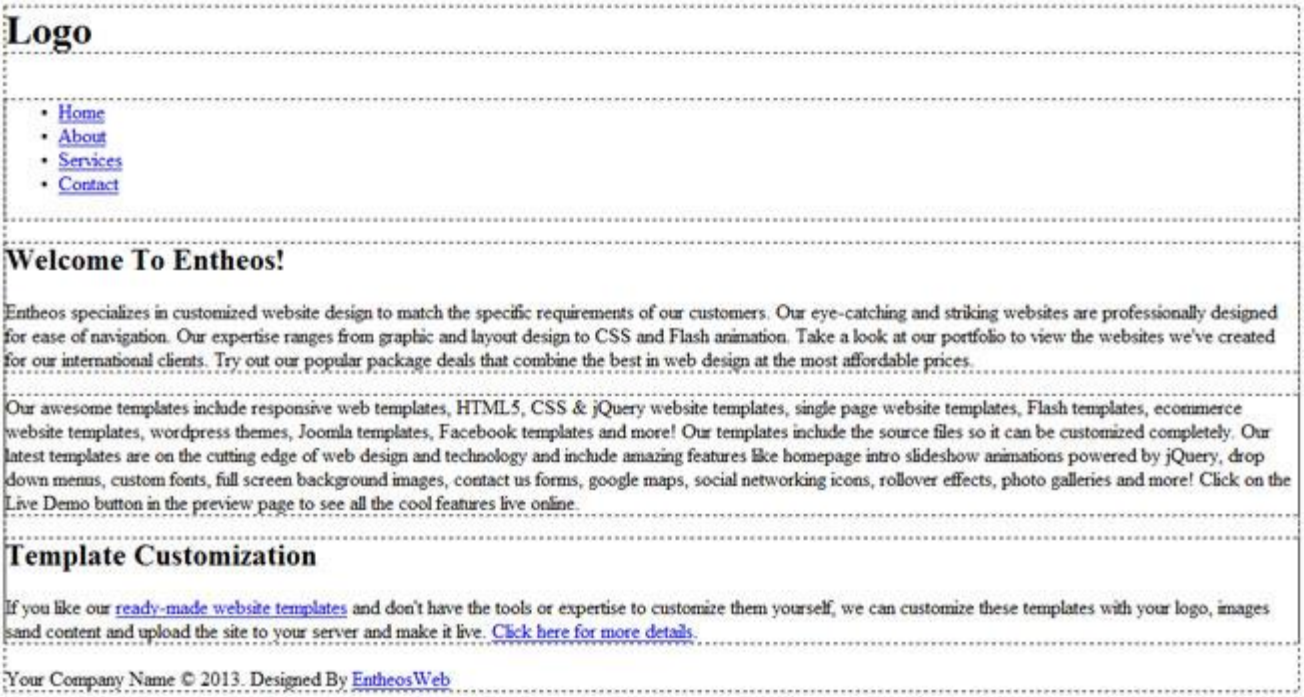
Step 5 : Let's begin our webpage with a HTML5 doctype and standard meta elements.

```
<!doctype html>

  <html lang="en">

  <head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

  <title>Free Responsive Website By EntheosWeb</title>

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

  </head>

  <body>

<div class="wrapper">

  <div id="header_top"></div>

  <div id="header"></div>

  <div id="nav"><ul>

  <li> <a href="../index.html" target="_self">Home</a> </li>

  <li> <a href="../about.html" target="_self">About</a> </li>

  <li> <a href="../services.html" target="_self">Services</a> </li>

  <li> <a href="../contact.html" target="_self">Contact</a> </li>

  </ul>  </div>

  <div id="header_bottom"></div>
  <section id="content">
  <div class="wrap-content text">
```

```
<div class="row box1">  </div>

<div class="row box2">  </div>

<div class="row box3">  </div>

</div></section>

<footer>  </footer>

</div>

</body>

</html>
```

Step 6 : Thus we have created the HTML file with CSS using the above html structure. Hence the result of this will be shown here

Step 7 : This step is needed to actually see how the webpage will look like with its content. If we add the content, logo and the other text, then the page will view as unordered like the below image, since we are yet to create the CSS style. So here is the html page without any styling.



Now we can include some styling code in the CSS files.

Always start with the desktop, so it will be easy for us to code. After this we need to start coding the styles for all the devices.

Step 8 : Open your style.css file and add CSS rules. We should build our website in a standard way, using HTML5 and CSS3. We can code for our header section first.

Step 9 : In the header_top and header_bottom style, I have assigned different background image to be repeated in the "x" direction. And also the company logo is placed in the header with the height of 218px.

Note: I have used 3 different header images for desktop, tablet and iphone with different sizes.

```
#header_top{

 width: 100%;

 background-image:url(images/top_blue_bg.jpg);

 background-repeat:repeat-x;

 height:12px;
 }

 #header{

 width: 100%;

 background-color:#00AEC3;

 background-image:url(images/header_image.jpg);

 background-repeat:no-repeat;

 height:218px;

 margin-left:0 0 0 0;

 }
 #header_bottom{

 width: 100%;

 background-image:url(images/bottom_blue_bg.jpg);

 background-repeat:repeat-x;

 height:15px;

}
```

Step 10 : The following code styles the navigation menu button:

```
#nav {
width: 100%;

background-color: #FFBF00;

height: 57px;
```

```
background-repeat: repeat;

}

#nav .wrap-nav{

 height: 30px;
 background: #333333;
 }

#nav ul{

 float:left;
 list-style:none;

 width:100%;

 }
#nav ul li{
 float:left;

 list-style:none;

 margin-right:80px;

 }
#nav ul li:hover {background: #00AEC3;}

 #nav ul li a{

 text-decoration: none;

 font-size: 14px;

 line-height: 16px;

 color: #000000;

 display: block;

 padding: 7px 10px;

 margin-bottom: 1px;

 z-index: 3;
 position: relative;

 text-transform: uppercase;

 font-family: Arial, Helvetica, sans-serif;

 }
```

Here we have coded the font type, size, color, text padding, background color, etc. according to your need.

Step 11 : Next, we can code the CSS styles for the three boxes where the main content will be placed.

text-align:left - The content of the page will be aligned on the left.
margin:5px 7px - Let's give margin of 5px to the top & bottom, and 7px to the left & right side of the box.
h2{text-align:center; } - The heading2 text is set to align center.

```
.box1{

margin:5px 7px;

padding:10px;

}
.box1 h2{text-align:center; }

.box2 {

margin:0px 7px;

padding:10px;

text-align:left;

}

.box3 {

margin:5px 7px;

padding:10px;

text-align:left;
}
```

Step 12 : Then your seperator line code may look like this:

```
div.line {

  border-top-width:1px;

  border-top-style:dotted;
  border-top-color:#D7D7D7;

  }
```

Step 13 : Finally, we can code for the footer part and for its text here

```
footer {

background-image: url(images/top_yellow_bg.jpg);

background-repeat: repeat-x;

}

.wrap-footer{
```

```
background-image: url(images/top_yellow_bg.jpg);
background-repeat: repeat-x;

 }

 .ftr {

  background:#FFBF00;

  font-family:Arial, Helvetica, sans-serif;

  padding:10px 0px;

  color:#ffffff;

  }

  .ftr p{

  font-size:12px;

  font-family:Arial, Helvetica, sans-serif;

  text-align:center;

  line-height:30px;

  }

  .ftr a{

  text-decoration:underline;

  color:#DA251D;

}
```

With that, we're all finished the coding for the desktop!

Step 14 : CSS Styling

The basic website uses this CSS:

```
/* CSS Document */


html, body {

height: 100%;
}
body{

background-image:url(images/cream_pixels.fw.png);

background-repeat:repeat;
margin-top:10px;
```

```
}
.wrapper {

max-width: 1126px;

background-color:#FFF;

width: 100%;

margin: 0 auto -20px;

}

img {
  padding:1px;

  border:1px solid #D7D7D7;

  background-color:#FFF;

  margin: 5px 15px 5px 50px;

  }



/* ------------------ header -------------------- */

#header_top{

  width: 100%;

  background-image:url(images/top_blue_bg.jpg);

  background-repeat:repeat-x;

  height:12px;
  }
  #header{

  width: 100%;
  background-color:#00AEC3;

  background-image:url(images/header_image.jpg);

  background-repeat:no-repeat;
  height:218px;

  margin-left:0 0 0 0;

  }
  #header_bottom{

  width: 100%;

  background-image:url(images/bottom_blue_bg.jpg);

  background-repeat:repeat-x;
```

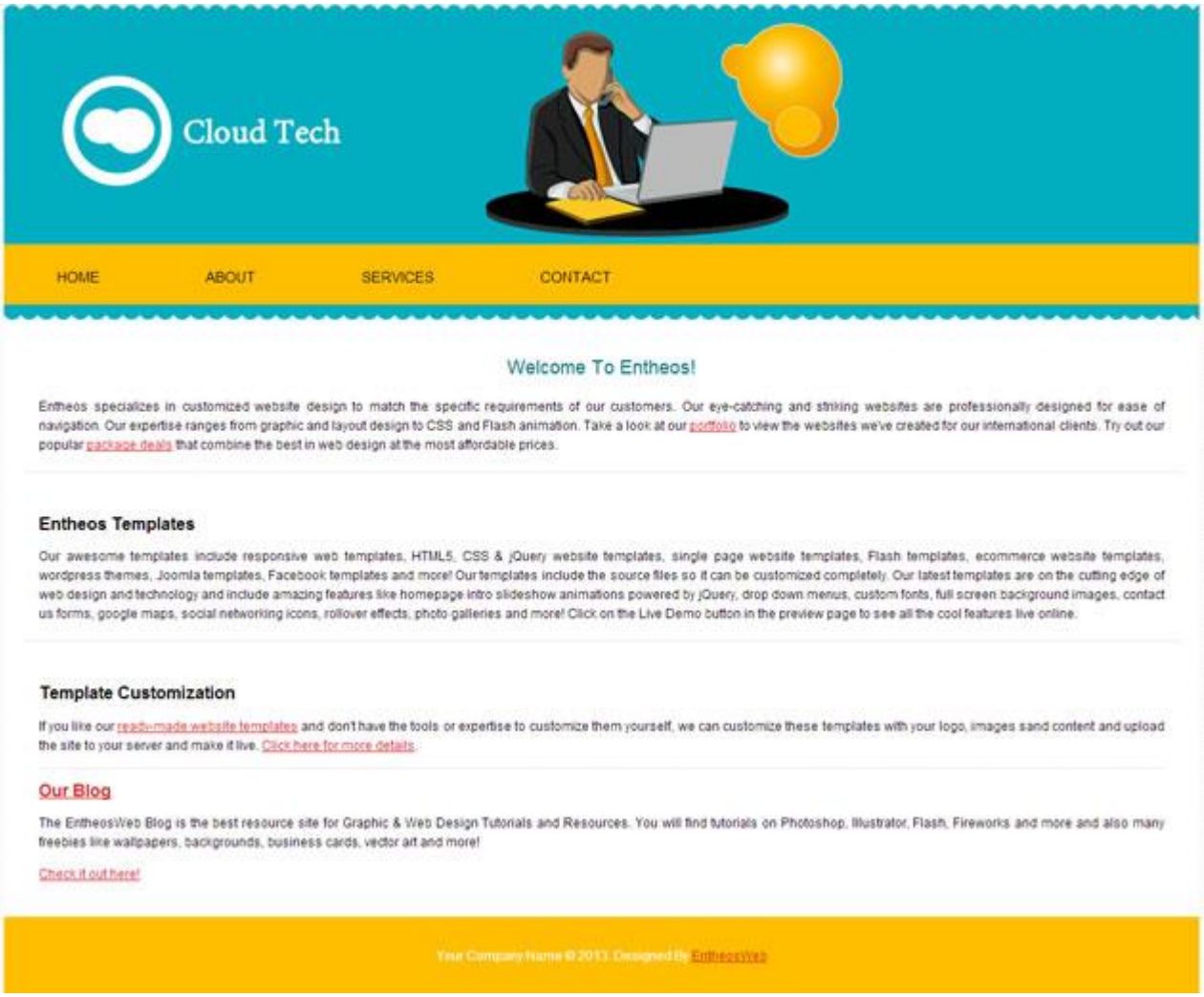
```
 height:15px;
 }

/* ------------------ navigation ------------------- */
#nav {
 width: 100%;
 background-color: #FFBF00;
 height: 57px;
 background-repeat: repeat;
 }
 #nav .wrap-nav{
 height: 30px;
 background: #333333;
 }
 #nav ul{
 float:left;
 list-style:none;
 width:100%;
 }
 #nav ul li{
 float:left;
 list-style:none;
 margin-right:80px;
 }
 #nav ul li:hover {background: #00AEC3;}
 #nav ul li a{
 text-decoration: none;
 font-size: 14px;
 line-height: 16px;
 color: #000000;
 display: block;
```

```
padding: 7px 10px;

margin-bottom: 1px;
z-index: 3;

position: relative;

text-transform: uppercase;

font-family: Arial, Helvetica, sans-serif;

}



/* ------------------ content ------------------- */

h2{

font-size:16px;

font-family:Arial, Helvetica, sans-serif;

line-height:16px;
}
h3{
font-family: Arial, Helvetica, sans-serif;

font-size: 110%;

font-weight: normal;

text-align: center;

color: #008493;

text-shadow: 0 1px 1px #dadada;

}
p{
font-size:12px;

font-family:Arial, Helvetica, sans-serif;
text-align:left;
line-height:18px;

text-align:justify;
 }
a{

color:#DA251D;

}
#content .wrap-content{ background: #ffffff;}

.box1 {
margin:5px 7px;
```

```
 padding:10px;
  }
 .box1 h2{text-align:center; }

 .box2 {

 margin:0px 7px;

 padding:10px;

 text-align:left;
 }

 .box3 {
 margin:5px 7px;

 padding:10px;

 text-align:left;

 }

div.line {

 border-top-width:1px;

 border-top-style:dotted;

 border-top-color:#D7D7D7;
 }



/* -------------------- column and row ------------------- */
.wraptext{margin:5px;}

.text .co1umn1, .text .co1umn2{float:left; display: inline-block;}

.text .wraptext{
 margin: 16px;

 }



/* ------------------ footer -------------------- */

footer {

 background-image: url(images/top_yellow_bg.jpg);

 background-repeat: repeat-x;

 }
 .wrap-footer{
 background-image: url(images/top_yellow_bg.jpg);
 background-repeat: repeat-x;
```

```
  }
.ftr {
  background:#FFBF00;
  font-family:Arial, Helvetica, sans-serif;
  padding:10px 0px;
  color:#ffffff;
  }
  .ftr p{
  font-size:12px;
  font-family:Arial, Helvetica, sans-serif;
  text-align:center;
  line-height:30px;
  }
  .ftr a{
  text-decoration:underline;
  color:#DA251D;
}
```

Step 15 : So here's what our page looks like now.



The responsive website layout, which we created in the desktop view.

Step 16 : We can then get start with CSS media queries to add the responsive functionality to our design. Adding Media Queries is the most interesting feature of the responsive web design!

Let's start! We will create some rules for different viewport widths.

Step 17 : Coding for tablet using media queries.

Set max-width for your tablet as **768px.** The screen size above the max-width of 768px will show the desktop version and below that size will show the tablet version.

```
@media only screen and (max-width: 768px) {

body{

 margin-top:10px;

 }

 #header{
 width: 100%;

 background-color:#00AEC3;

 background-image:url(images/header_image_big.jpg);

 background-repeat:no-repeat;

 height:214px;
 margin-left:0 0 0 0;
```

```
display:block;
}
#nav {

width: 100%;

background:#FFBF00;

position: relative;

border-bottom: 10px solid #FFBF00;
}

#nav ul{

float:left;
list-style:none;

width:100%;

display: block;

position:relative;

}
#nav ul li a{

text-decoration: none;

font-size: 12px;

line-height: 10px;

color: #000000;

display: block;

margin:0;

padding:5px;

list-style:none;
z-index: 1;
position:relative;
text-transform: uppercase;

font-family: Arial, Helvetica, sans-serif;

clear: left;

}
#header_bottom{

width: 100%;

position:relative;

z-index:3;
```

```
display:block;

}

#content .wrap-content{
background: #ffffff;

display: block;

width: 100%;

}

.text .wraptext{
margin: 16px;
}
.box1 .box2 .box3 {
margin:0px 7px;

padding:5px;

}

}
```

Here we need not to code for the entire layout. We can change some of the styles to be adjusted according to the tablet size. Use the style "display : block" to align the items one below the other in the smaller resolution. Also reduce the padding and margin size to fit the size.

The created responsive website layout, in the tablet view.

Step 18 : Coding for smartphones using media queries.

The smartphone layout is narrower than the original content width, so this div also needs altering with a new declaration in the media queries CSS file.

With that, we have a nice big image at the top of our page that automatically adjusts or replace with other as the page width is reduced!

Step 19 : For **320px** or less (iphone screen), we will display our navigation items in one column with 4 rows as a block. CSS allows us to show and hide content. The smart phone display area sizes are very small in size comparing to the desktop or tablet, so it's necessary to hide some unwanted items from the layout like ad, news and more!

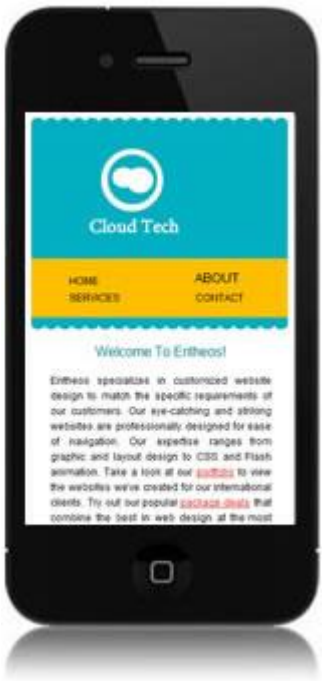
And finally code for the iphone

```
@media only screen and (max-width: 320px) {


body{
  margin-top:1px;
  }

  #header{

  max-width: 100%;

  background-image:url(images/header_image_small.jpg);

  height:160px;

  display:block;

  }
 #nav {

  height: auto;

  position: relative;
  border-bottom: 97px solid #FFBF00;

  }

#nav .wrap-nav{

  background:#FFBF00;

  background-repeat: repeat-x;
  display: block;

  height: auto;
  }

#nav ul{
  list-style:none;

  height: auto;

  width:100%;
```

```
display: block;

position:relative;

}


#nav ul li a{

text-decoration: none;

font-size: 12px;
line-height: 10px;

color: #000000;
display: block;

margin:0;

padding:5px;

list-style:none;

z-index: 3;

position:inherit;

text-transform: uppercase;

font-family: Arial, Helvetica, sans-serif;
clear: left;

}


.text .wraptext{

margin: 5px;
}

.box1 .box2 .box3  {

margin:0px 7px;

padding:5px;

display:block;

}
h3{
font-family: Arial, Helvetica, sans-serif;
font-size: 100%;

}
img {
padding:1px;
border:1px solid #D7D7D7;
background-color:#FFF;
```

```
margin: 0px 0px 0px 0px;

}

}
```

The created responsive website layout, in the iphone view.



Step 20 : Finally, I have saved this responsive website layout as a dreamweaver template(.dwt) using Adobe Dreamweaver, since it's very easy to do the common changes in all the pages at onego. Anyway you can also use the responsive website as it is.

Step 21 : You can scale your browser to see the responsive layouts for yourself.