Front-end | CSS

Session 2

Compatibility

Length Units

@media

by Mohammad Amin H.B. Tehrani - Reza Yazdani

www.maktabsharif.ir

Compatibility

Compatibility

Intro

- **Browser compatibility:** The term browser compatibility refers to the ability of a certain website to appear fully functional on different browsers that are available in the market. This means that the website's HTML coding, as well as the scripts on that website, should be compatible to run on the browsers.

- **Device compatibility (Responsive):** Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

Browser compatibility

Solutions:

While preparing the page style properties, you should also pay attention to preparing the specific properties of each browser engine.

Example:

```
-webkit-align-content: ...; /* For Safari, google chrome, Opera ... */
-moz-animation: ...; /* For mozilla */
-ms-behavior: ...; /* For Microsoft (Edge) */
```

Also you can check properties browser support

Device compatibility: Responsive

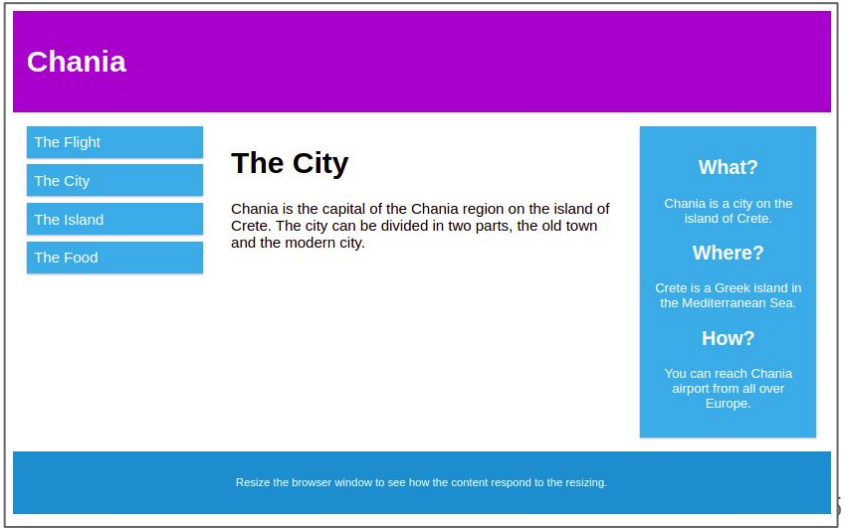
What is Responsive Web Design?

Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

Try it!



Viewport

Setting The Viewport

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.

You should include the following <meta> viewport element in all your web pages:

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

This 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.

Length Units

Length Units

Intro

CSS has several different units for expressing a length.

Many CSS properties take "length" values, such as width, margin, padding, font-size, etc.

Length is a number followed by a length unit, such as 10px, 2em, etc.

- **Absolute Lengths:** The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

- **Relative Lengths:**Relative length units specify a length relative to another length property. Relative length units scale better between different rendering medium.

```
h1 {
    font-size: 60px;
}

p {
    font-size: 1cm;
    line-height: 50%;
}
```

Absolute Lengths

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit	Description
cm	Centimeters Try it
mm	Millimeters Try it
in	inches (1in = 96px = 2.54cm) Try it
px *	pixels (1px = 1/96th of 1in) Try it
pt	points (1pt = 1/72 of 1in) Try it
pc	picas (1pc = 12 pt) Try it

Length Units

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scale better between different rendering medium.

Unit	Description	
em	Relative to the font-size of the element (2em means 2 times the size of the current font)	Try it
ex	Relative to the x-height of the current font (rarely used)	Try it
ch	Relative to the width of the "0" (zero)	Try it
rem	Relative to font-size of the root element	Try it
vw	Relative to 1% of the width of the viewport*	Try it
vh	Relative to 1% of the height of the viewport*	Try it
vmin	Relative to 1% of viewport's* smaller dimension	Try it
vmax	Relative to 1% of viewport's* larger dimension	Try it
%	Relative to the parent element	Try it

@media

@media

Intro

The @media rule is used in media queries to apply different styles for different media types/devices.

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

```
@media only screen and (max-width: 600px)
{
    body {
        background-color: lightblue;
    }
}
```

Using media queries are a popular technique for delivering a tailored style sheet (responsive web design) to desktops, laptops, tablets, and mobile phones.

You can also use media queries to specify that certain styles are only for printed documents or for screen readers (mediatype: print, screen, or speech).

In addition to media types, there are also media features. Media features provide more specific details to media queries, by allowing to test for a specific feature of the user agent or display device. For example, you can apply styles to only those screens that are greater, or smaller, than a certain width.

@media

Syntax

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {
  CSS-Code;
}
```

meaning of the not, only and and keywords:

not: The not keyword inverts the meaning of an entire media query.

only: The only keyword prevents older browsers that do not support media queries with media features from applying the specified styles. It has no effect on modern browsers.

and: The and keyword combines a media feature with a media type or other media features.

They are all optional. However, if you use not or only, you must also specify a media type.

```
<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">
<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">
....
```

@media

Example: Responsive Web Page using

Try it!