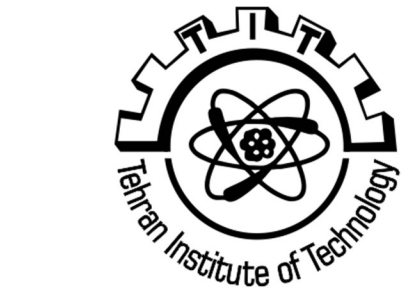


RESPONSIVE DESIGN

Tehran Institute of Technology

Zahra Mansoori
z.mansoori@gmail.com
http://tarsimm.com

Contents

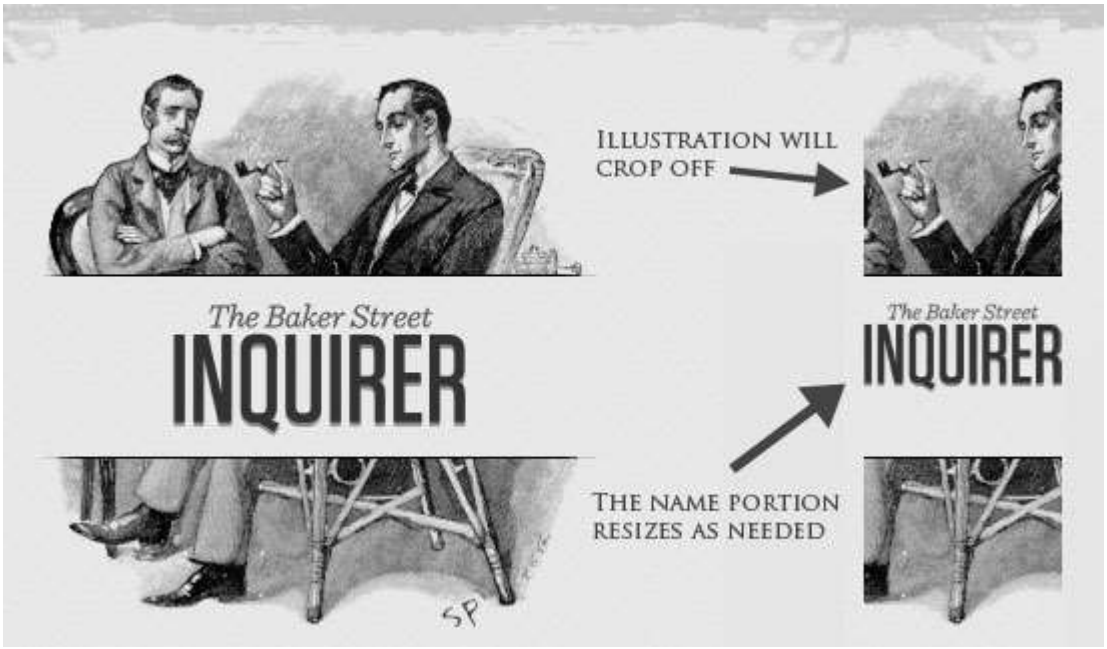
1. What Is Responsive Web Design? 3
2. Screen Size or Resolution? 3
3. Media queries 3
Example.1: 4
CSS 4
Example.2: 4
CSS 4
Exercise.1: 4
3.1. Media Types 5
3.2. Other usages of Media Query 5
4. CSS Framework 5
Foundation 6
5. Bootstrap 6
5.1. BootstrapCDN 6
CSS 6
JS 7
5.2. Local Bootstrap 7
6. Bootstrap Layouts 7
Examples: 8
Container 8
Container-Fluid 8
7. Bootstrap Grids 9
Example.1: 9
HTML 9
Example.2: 9
HTML 10
Example.3: 10
HTML 10
Example.4: 11
Extra small 11
Small 11

Medium 11
Large 11
Extra large 11
Extra extra large 11
HTML 11
8. Bootstrap Typography 11
Example.1: 12
9. Bootstrap Components 13

1. What Is Responsive Web Design?

Responsive Web design is the approach that suggests that design and development should respond to the user's behavior and environment based on screen size, platform and orientation.

It is an approach to web page creation that makes use of flexible layouts, flexible images and cascading style sheet and media queries. The goal of responsive design is to build web pages that detect the visitor's screen size and orientation and change the layout accordingly.



Picture.1: Responsive Design Example

2. Screen Size or Resolution?

When you buy a device, you will often see both screen size and resolution listed in the specs. The screen size is the physical measurement diagonally of the screen in inches. This is not to be confused with the resolution, which is the number of pixels on the screen often displayed as a width by height (i.e. 1024×768).

3. Media queries

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 be only for printed documents or for screen readers (mediatype: print, screen, or speech).

Example.1:

CSS

```
body {
  background-color: black;
}

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

@media only screen and (min-width: 601px) and (max-width: 1200px){
 body {
  background-color: blue;
 }
}
```

Example.2:

CSS

```
/* When the width is between 600px and 900px OR above 1100px - change the appearance of
<div> */
@media screen and (max-width: 900px) and (min-width: 600px), (min-width: 1100px) {
        div.example {
            font-size: 50px; padding: 50px;
            border: 8px solid black;
            background: yellow;
 } }
```

Exercise.1:

Create a webpage with top menu like the picture 2, and by using Media Query, at each screen size, one sub menu should appear (Picture 3).

Hint: Use “Visibility” property for making submenus appear or disappear.

Picture.1: Before using media query

Picture.2: After Using Media Query on Large display

3.1. Media Types

Value	*Description*
all	Default. Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

3.2. Other usages of Media Query

You can also have different *stylesheets* for different media, like this:

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

4. CSS Framework

A **CSS framework** is a library allowing for easier, more standards-compliant web design using the Cascading Style Sheets language. Most of these frameworks contain at least a grid. Frameworks that are more functional also come with more features and additional JavaScript based functions, but are mostly design oriented and focused around interactive UI patterns.

Picture.3: Top CSS Frameworks:

5. Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Mark Otto and Jacob Thornton created it at Twitter in mid-2010. Prior to being an open-sourced framework, Bootstrap was known as Twitter Blueprint. It served as the style guide for internal tools development at the company for over a year before its public release, and continues to do so today.

Picture.4: Bootstrap is Responsive

There are two ways to use bootstrap in your web development project:

BootstrapCDN and *Local Bootstrap*

5.1. BootstrapCDN

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"                                                integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
```

JS

Place the following <script>s near the end of your pages, right before the closing </body> tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bund
le.min.js"                                                integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
```

5.2. Local Bootstrap

Download ready-to-use compiled code from https://getbootstrap.com/ and drop into your project, which includes:

- Compiled and minified CSS bundles
- Compiled and minified JavaScript plugins

6. Bootstrap Layouts

Containers are the most basic layout element in Bootstrap and are required when using our default grid system. Containers are used to contain, pad, and (sometimes) center the content within them. While containers can be nested, most layouts do not require a nested container.

There are three kinds of Bootstrap Layouts:

- `container`, which sets a `max-width` at each responsive breakpoint
- `container-fluid`, which is `width: 100%` at all breakpoints
- `container-{breakpoint}`, which is `width: 100%` until the specified breakpoint

	Extra small *<576px*	***Small*** *≥576px*	***Medium*** *≥768px*	***Large*** *≥992px*	***Extra-large*** *≥1200px*	***Extra extra large***

						≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

Table.2: Different Layouts in Bootstrap

Examples:

Container

Responsive, fixed-width container:

```
<div class="container">
  <!-- Content here -->
</div>
```

Container-Fluid

Fluid-width (meaning it is `100%` wide all the time):

```
<div class="container-fluid">
  <!-- Content here -->
</div>
```

For extra information about layouts, refer to https://getbootstrap.com/ -> Layouts

7. Bootstrap Grids

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content.

Example.1:

One of three columns	One of three columns	One of three columns

HTML

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

Example.2:

1 of 2	2 of 2

1 of 3	2 of 3	3 of 3

HTML

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Example.3:

Col-8	Col-4

HTML

```
<div class="row">
  <div class="col-8">col-8</div>
  <div class="col-4">col-4</div>
</div>
```

Example.4:

Grid system of Bootstrap divides each area into 12 individual columns. You can adjust these columns by mentioning the ratio after **Col** or other prefixes mentioned at the table below.

	Extra small *<576px*	*Small* *≥576px*	*Medium* *≥768px*	*Large* *≥992px*	*Extra large* *≥1200px*	*Extra extra large* *≥1320px*
Max container width	None (auto)	540px	720px	960px	1140px	1320
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`

Table.3: Responsive Grid System

col-sm-8	col-sm-4

HTML

```
<div class="row">
  <div class="col-sm-8">col-sm-8</div>
  <div class="col-sm-4">col-sm-4</div>
</div>
```

For extra information about grids, refer to https://getbootstrap.com/ -> Grids

8. Bootstrap Typography

Bootstrap sets basic global display, typography, and link styles.

Default **font-family** for Bootstrap is "**Helvetica Neue", Helvetica, Arial, sans-serif**".

However, you can change it using below syntax:

```
body {
    font-family: 'Oswald', sans-serif !important;
}
```

Heading	Example
	h1. Bootstrap heading
	h2. Bootstrap heading
	h3. Bootstrap heading
	h4. Bootstrap heading
	h5. Bootstrap heading
	h6. Bootstrap heading

Picture.5: Different Heading sizes in Bootstrap

It also defines individual classes for each Headings. You need not to mention <H1> to have a heading Type1, just Use below syntax to have it.

```
class="h1"
```

Example.1:

h1. Bootstrap heading

h2. Bootstrap heading

h3. Bootstrap heading

h4. Bootstrap heading

h5. Bootstrap heading

h6. Bootstrap heading

Picture.6: Headings also have specific classes in Bootstrap

```
<p class="h1">h1. Bootstrap heading</p>
<p class="h2">h2. Bootstrap heading</p>
<p class="h3">h3. Bootstrap heading</p>
<p class="h4">h4. Bootstrap heading</p>
<p class="h5">h5. Bootstrap heading</p>
<p class="h6">h6. Bootstrap heading</p>
```

For extra classes, refer to https://getbootstrap.com/ -> typography

9. Bootstrap Components

Thereare several types of components introduced in Bootstrap, which defined by their specific classes, such as different types of **Buttons** and **Button Groups**, **Drop downs**, **Carousel**, **Collapse**, **Navs**, **Navbars**, **and Progress** etc**.** You can find related information in https://getbootstrap.com/ -> Components.