Department of
COMPUTER SCIENCE
HONG KONG BAPTIST UNIVERSITY

COMP7980 – Dynamic Web and Mobile Programming
COMP7270 – Web and Mobile Programming

Chapter 2 Bootstrap, JavaScript, and Form

Course Instructors: Dr. Ma Shichao, Dr. Zhang Ce, and Mr. Jiang Jintian

Announcement

- Apologize for the small font size in last class
  - Will change the screen resolution and use Zoom to share the screen

  Zoom link: https://hkbu.zoom.us/j/99775423344?pwd=8Wi6pJIZ29TDkzj0X4XnnbTTlEMjAj.1
- Choose Mac OS
  - Account name: guest
  - Account password:

Announcement

- Change of Instructors

Dr. MA Shichao (麻時鈔)

– Email: shichaoma@comp.hkbu.edu.hk
– Office: DLB 804
– Phone: 3411-7599

Dr. ZHANG Ce(張策)

– Email: cezhang@comp.hkbu.edu.hk
– Office: RRS 727
– Phone: 3411-5979

Mr. JIANG Jintian(姜錦田)

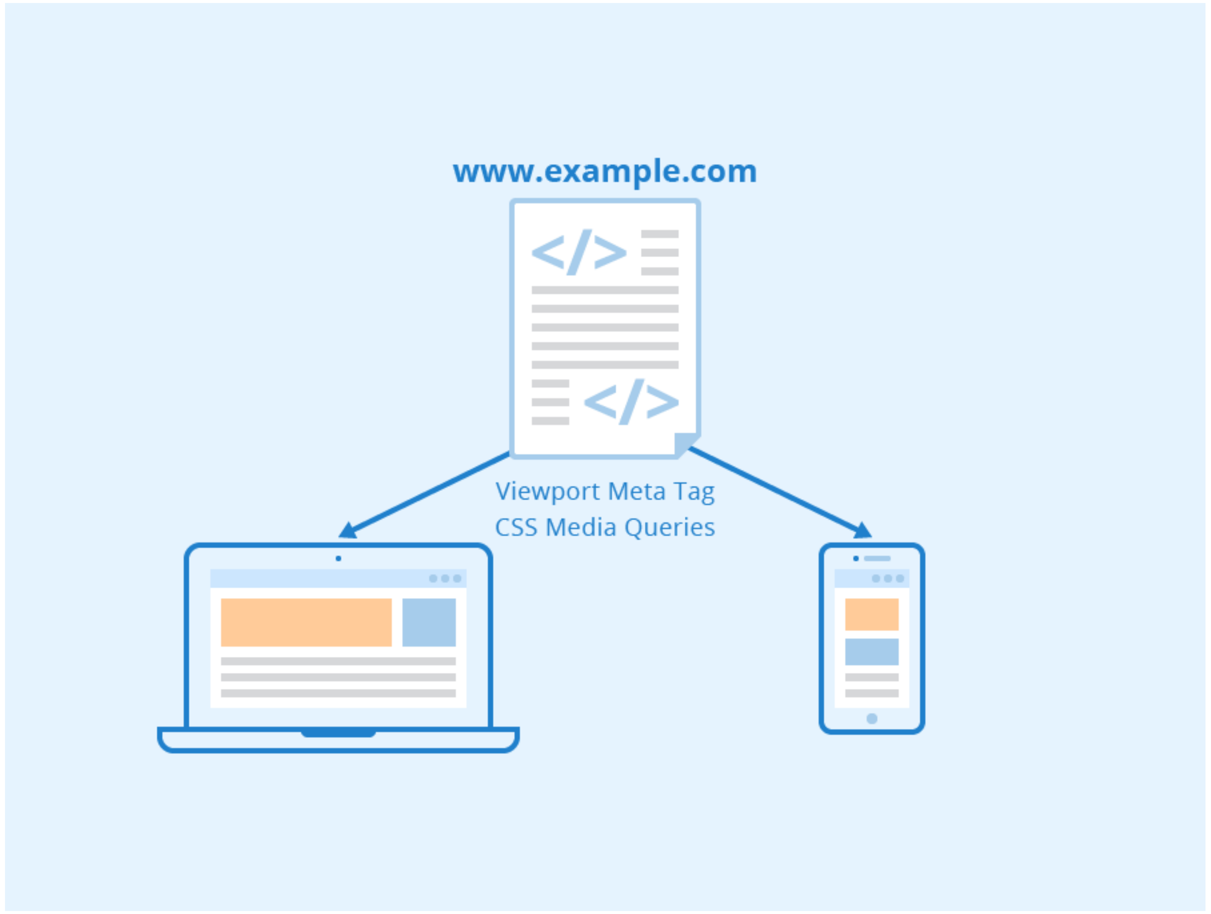
– Email: jintian-jiang@comp.hkbu.edu.hk
– Office: RRS 637
– Phone: 3411-6454

Agenda

- **Responsive Web Design** with **Bootstrap**
- Client-side **JavaScript**
- **HTML Form** Elements

Responsive Web Design

- **Responsive Web Design** (RWD) is an approach to web design aimed at crafting sites to provide an **optimal viewing** and interaction experience **across a wide range of devices**



Why RWD?

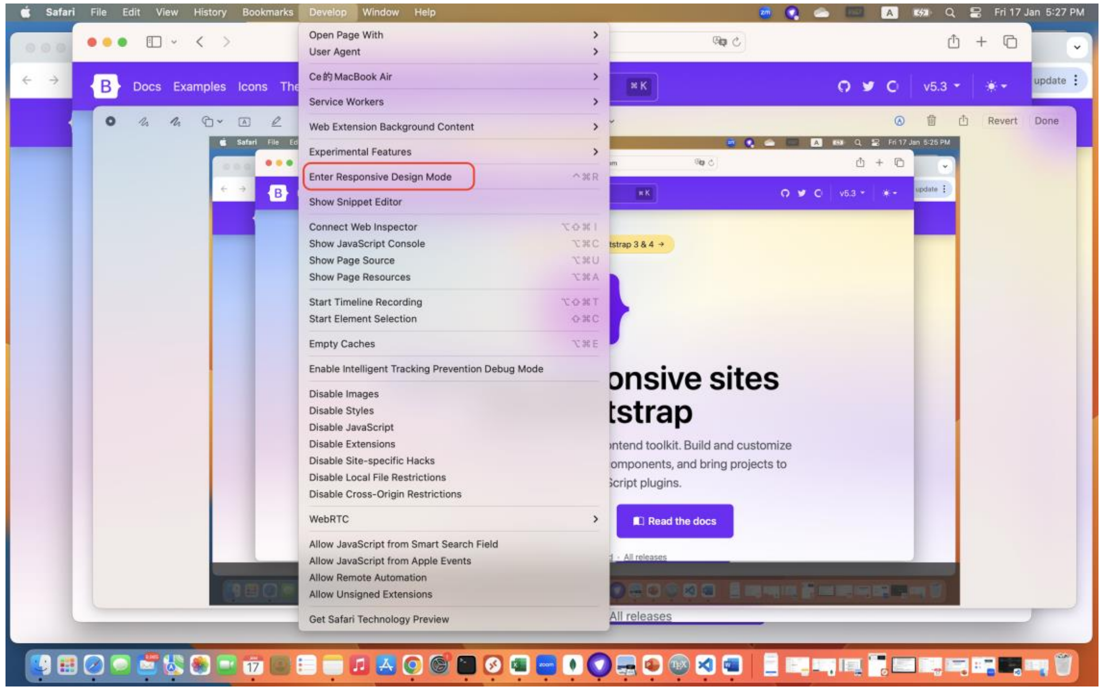
- It is becoming more important as the **amount of mobile traffic now accounts for more than half of total internet traffic**.
- This trend is so prevalent that Google has begun to **boost the ratings of sites that are mobile friendly** if the search was made from a mobile device.
    - This has the net effect of **penalizing sites that are not mobile friendly**.

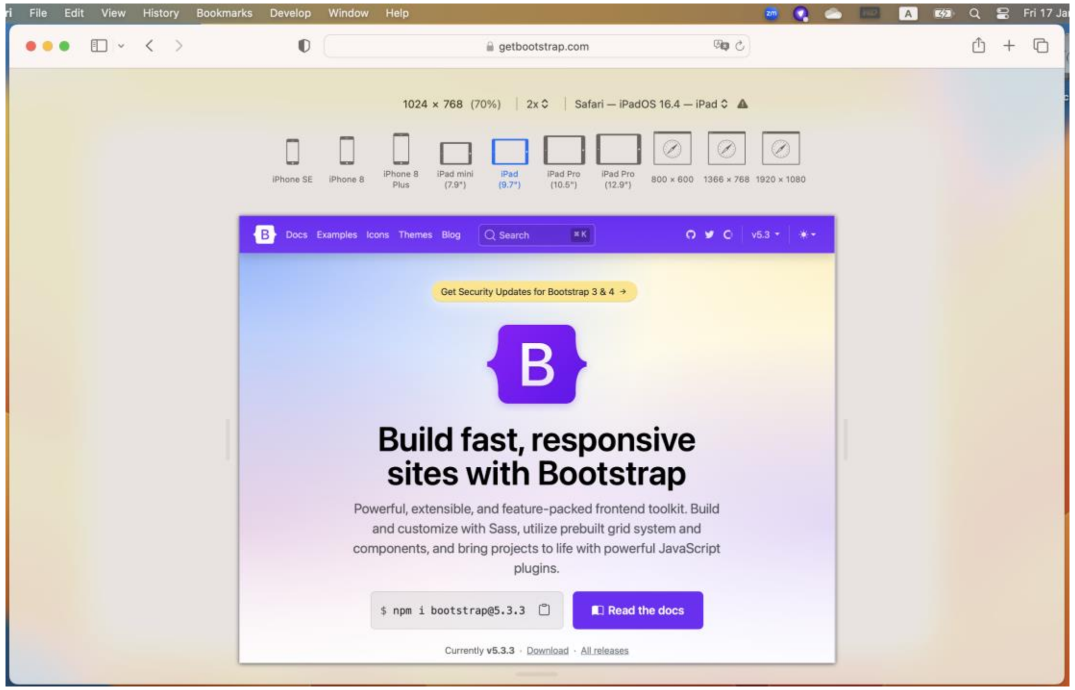
Bootstrap

- Bootstrap is a **popular front-end framework** that provides a collection of pre-designed HTML, CSS, and JavaScript components. It aims to simplify the process of building **responsive** and **mobile-first** web pages and applications.
- Bootstrap offers a **grid system**, **CSS styles**, and a wide range of **ready-to-use components** such as navigation bars, buttons, forms, modals, and more. These components are designed to be easily customizable and responsive, meaning they automatically **adapt to different screen sizes** and devices.

Responsive Mode on Browser

Safari


Safari File Edit View History Bookmarks Develop Window Help
Open Page With
User Agent
Ce的MacBook Air
Service Workers
Web Extension Background Content
Experimental Features
Enter Responsive Design Mode
Show Snippet Editor
Connect Web Inspector
Show JavaScript Console
Show Page Source
Show Page Resources
Start Timeline Recording
Start Element Selection
Empty Caches
Enable Intelligent Tracking Prevention Debug Mode
Disable Images
Disable Styles
Disable JavaScript
Disable Extensions
Disable Site-specific Hacks
Disable Local File Restrictions
Disable Cross-Origin Restrictions
WebRTC
Allow JavaScript from Smart Search Field
Allow JavaScript from Apple Events
Allow Remote Automation
Allow Unsigned Extensions
Get Safari Technology Preview
Revert
Done
Read the docs
All releases



Build fast, responsive sites with Bootstrap

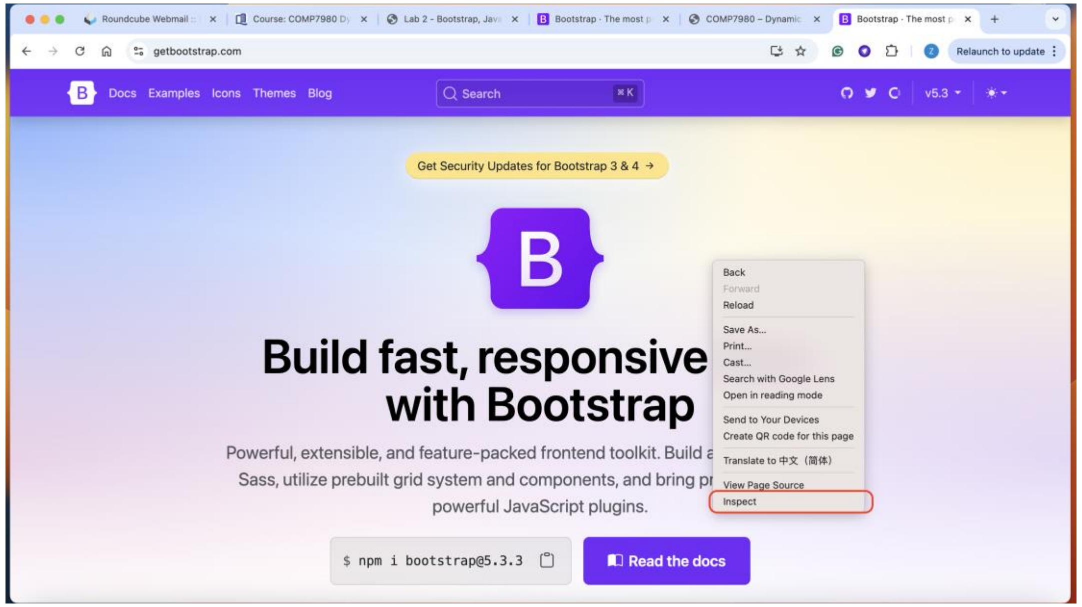
Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

```
$ npm i bootstrap@5.3.3
```

Currently v5.3.3 · Download · All releases

Responsive Mode on Browser

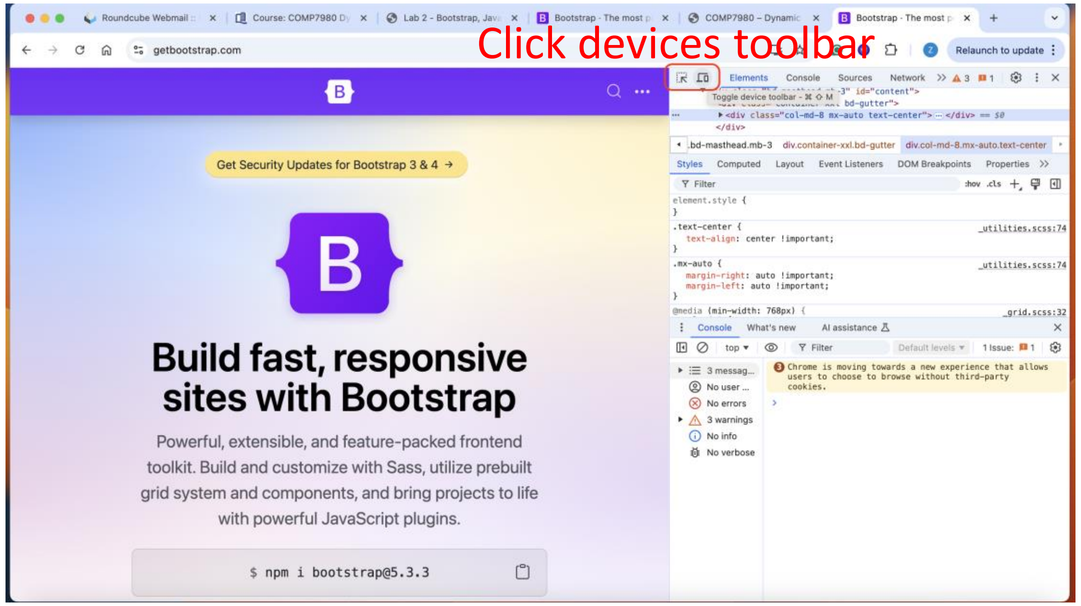
Chrome



Build fast, responsive with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build a Sass, utilize prebuilt grid system and components, and bring pr powerful JavaScript plugins.

```
$ npm i bootstrap@5.3.3
```



Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

```
$ npm i bootstrap@5.3.3
```

```
-3" id="content">
bd-gutter">
▸<div class="col-md-8 mx-auto text-center">…</div> == $0
</div>
```

```
element.style {
}
.text-center {                                   _utilities.scss:74
  text-align: center !important;
}
.mx-auto {                                       _utilities.scss:74
  margin-right: auto !important;
  margin-left: auto !important;
}
@media (min-width: 768px) {                      _grid.scss:32
```

```
Chrome is moving towards a new experience that allows
users to choose to browse without third-party
cookies.
```

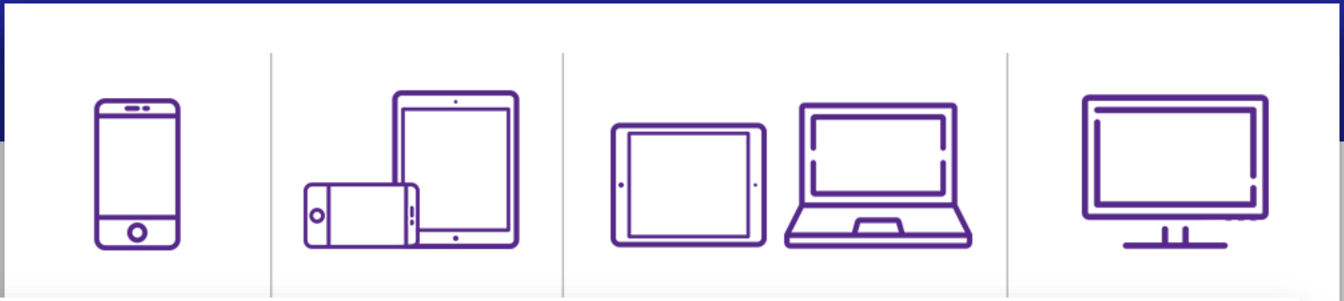
1. Mouse right-click, choose \`inspect\` button
2. Find \`devices toolbar\`, click it

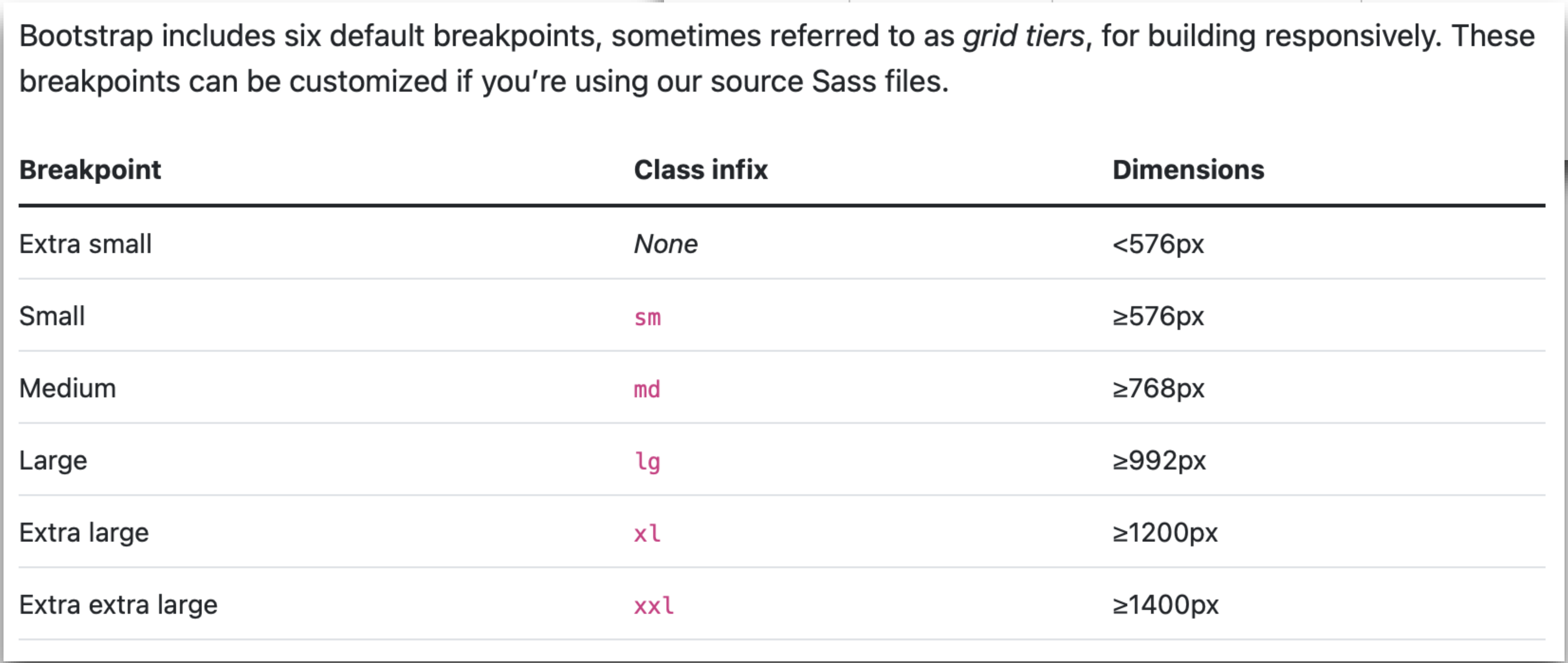
Bootstrap Grid System

- Bootstrap provides a **responsive grid system** that helps in creating flexible and responsive layouts. The grid system is based on a **12-column layout**, allowing developers to easily organize and structure content **across different screen sizes**.

- **Grid Classes**: Bootstrap provides **CSS classes** that define the layout and behavior of columns within the grid system. You can assign these classes to HTML elements to specify how they should behave at various screen sizes. The most commonly used grid classes are **col-xs-***, **col-sm-***, **col-md-***, and **col-lg-***, where * represents **the number of columns an element should span**.

Breakpoints





Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
Extra small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

Bootstrap Grid System

- **Responsive Behavior**: The grid system in Bootstrap is inherently responsive. By applying the appropriate grid classes, you can control how elements stack or rearrange themselves as the screen size changes. For example, **you can specify that an element should occupy two columns on large screens (col-lg-2), but span the entire width on extra small screens (col-12)**.
- **Nesting and Offset**: Bootstrap allows you to **nest columns inside other columns**, enabling more complex and nested layouts. You can also use offset classes to create spacing between columns or offset their position within the grid.

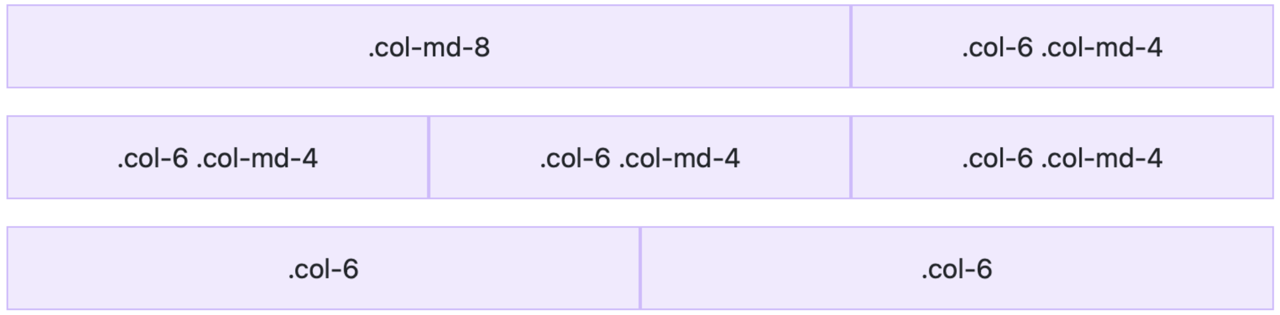
```
<div class="container text-center">
  <div class="row">
   <div class="col-md-8">.col-md-8</div>
   <div class="col-6 col-md-4">.col-6 .col-md-4</div>
 </div>

  <div class="row">
   <div class="col-6 col-md-4">.col-6 .col-md-4</div>
   <div class="col-6 col-md-4">.col-6 .col-md-4</div>
   <div class="col-6 col-md-4">.col-6 .col-md-4</div>
 </div>

  <div class="row">
   <div class="col-6">.col-6</div>
   <div class="col-6">.col-6</div>
 </div>
</div>
```

behave as a regular block-level element on mobile phone

Being viewed on tablets or desktops



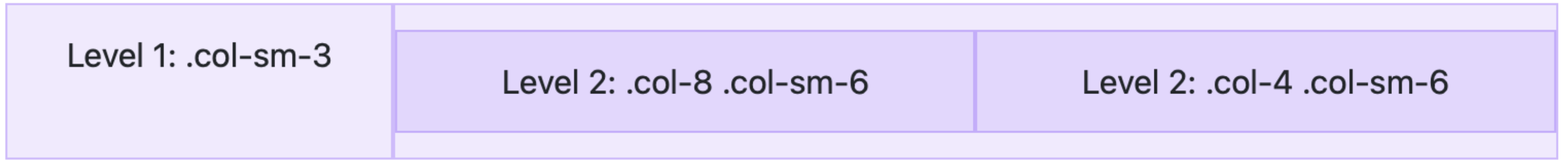
The column width defined for smaller screens **will be inherited for bigger screens** as well, **unless explicitly overridden**.

Bootstrap Grid System

- In the previous example, the **container** class creates a **responsive container** for the grid system. The row class represents a row within the grid, and the **col-*-*** classes define the columns.
- In Bootstrap 5, the "**col**" class **without specifying a column width** is a shorthand class that represents **an equal-width column**.

```html
<div class="row">
    <div class="col card">
        Card 1
    </div>
    <div class="col card">
        Card 2
    </div>
    <div class="col card">
        Card 3
    </div>
    <div class="col card">
        Card 4
    </div>
</div>
```

Nesting



- **Columns can be nested** inside one another to create more complex layouts.
- This allows you to **divide a column into multiple sub-columns**, each with its own width and content arrangement.

```
<div class="container text-center">
 <div class="row">
  <div class="col-sm-3">
   Level 1: .col-sm-3
  </div>
  <div class="col-sm-9">
   <div class="row">
    <div class="col-8 col-sm-6">
     Level 2: .col-8 .col-sm-6
    </div>
    <div class="col-4 col-sm-6">
     Level 2: .col-4 .col-sm-6
    </div>
   </div>
  </div>
 </div>
</div>
```

Bootstrap

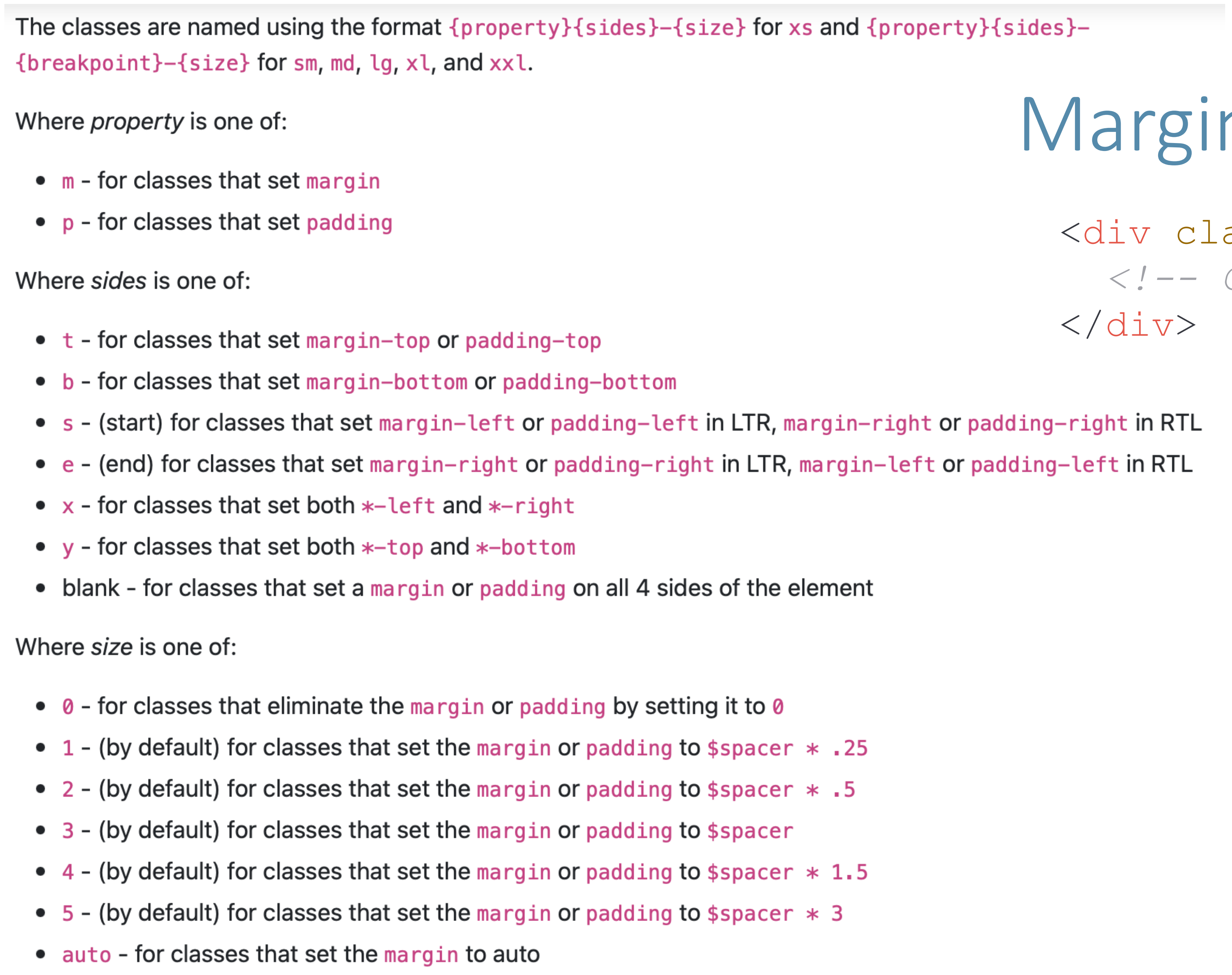
- Responsive CSS: Bootstrap includes **CSS classes** and utilities that enable the responsive behavior of elements. It allows you to **show or hide content** based on screen sizes, **adjust padding and margins**, and control the visibility of elements.

```
<div class="d-none d-md-block">
  This element is hidden on small screens but visible on medium and larger screens.
</div>

<div class="d-sm-none">
  This element is hidden on small screens but visible on medium and larger screens.
</div>

<div class="d-lg-none d-xl-block">
  This element is hidden on large screens but visible on extra-large screens.
</div>
```

Margin & Padding



The classes are named using the format `{property}{sides}-{size}` for `xs` and `{property}{sides}-{breakpoint}-{size}` for `sm`, `md`, `lg`, `xl`, and `xxl`.

Where *property* is one of:

- `m` - for classes that set `margin`
- `p` - for classes that set `padding`

Where *sides* is one of:

- `t` - for classes that set `margin-top` or `padding-top`
- `b` - for classes that set `margin-bottom` or `padding-bottom`
- `s` - (start) for classes that set `margin-left` or `padding-left` in LTR, `margin-right` or `padding-right` in RTL
- `e` - (end) for classes that set `margin-right` or `padding-right` in LTR, `margin-left` or `padding-left` in RTL
- `x` - for classes that set both `*-left` and `*-right`
- `y` - for classes that set both `*-top` and `*-bottom`
- blank - for classes that set a `margin` or `padding` on all 4 sides of the element

Where *size* is one of:

- `0` - for classes that eliminate the `margin` or `padding` by setting it to `0`
- `1` - (by default) for classes that set the `margin` or `padding` to `$spacer * .25`
- `2` - (by default) for classes that set the `margin` or `padding` to `$spacer * .5`
- `3` - (by default) for classes that set the `margin` or `padding` to `$spacer`
- `4` - (by default) for classes that set the `margin` or `padding` to `$spacer * 1.5`
- `5` - (by default) for classes that set the `margin` or `padding` to `$spacer * 3`
- `auto` - for classes that set the `margin` to auto

```
<div class="container my-4">
  <!-- Content here -->
</div>
```

Spacer = 1rem
= font-size
= 16px by default

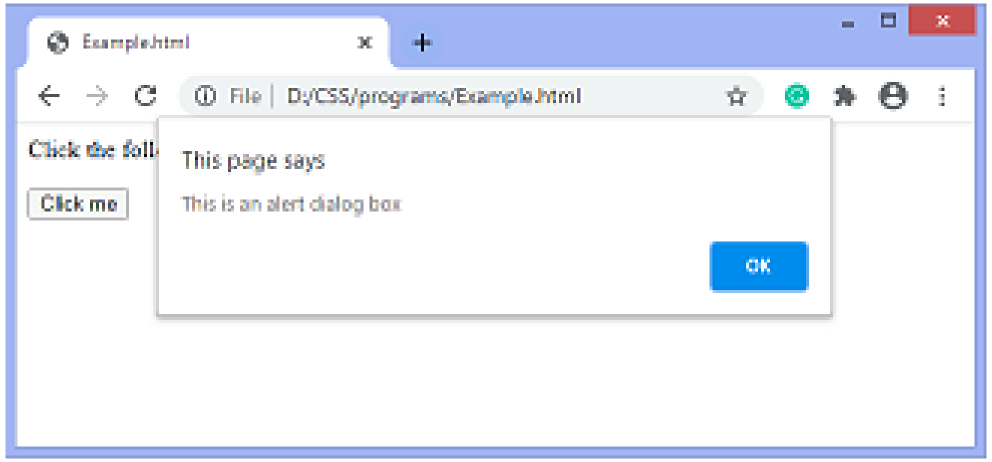
More on Bootstrap

- **Pre-styled Components**: Bootstrap offers a wide range of pre-designed UI components that are commonly used in web development, such as **navigation menus**, buttons, forms, **cards**, carousels, and more. These components come with **default styles and responsive behavior**, making it easier to create consistent and visually appealing interfaces.
- **JavaScript Plugins**: Bootstrap includes a set of JavaScript plugins that add **interactivity** and enhanced functionality to components. These plugins enable features like **dropdown menus**, **modals**, **tooltips**, carousels, and more, without requiring you to write complex JavaScript code from scratch.

Client-side Scripting with JavaScript

JavaScript

- JavaScript was designed to **add interactivity** to HTML pages
- Note that **JavaScript** and **Java** are two completely **different languages** in both concept and design!



This page says

This is an alert dialog box

- We use JavaScript to define the **behavior** of a webpage.

What can JavaScript do?

- **React to events**
  - Execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.
  - e.g., onchange, onclick, etc
- **Read and write HTML elements**
  - Read and change the content of an HTML element

```
function teamSelected(team) {
    var superheroElem = document.getElementById("superhero");
    superheroElem.options.length = 0;
}
```

Access HTML Element in DOM Tree

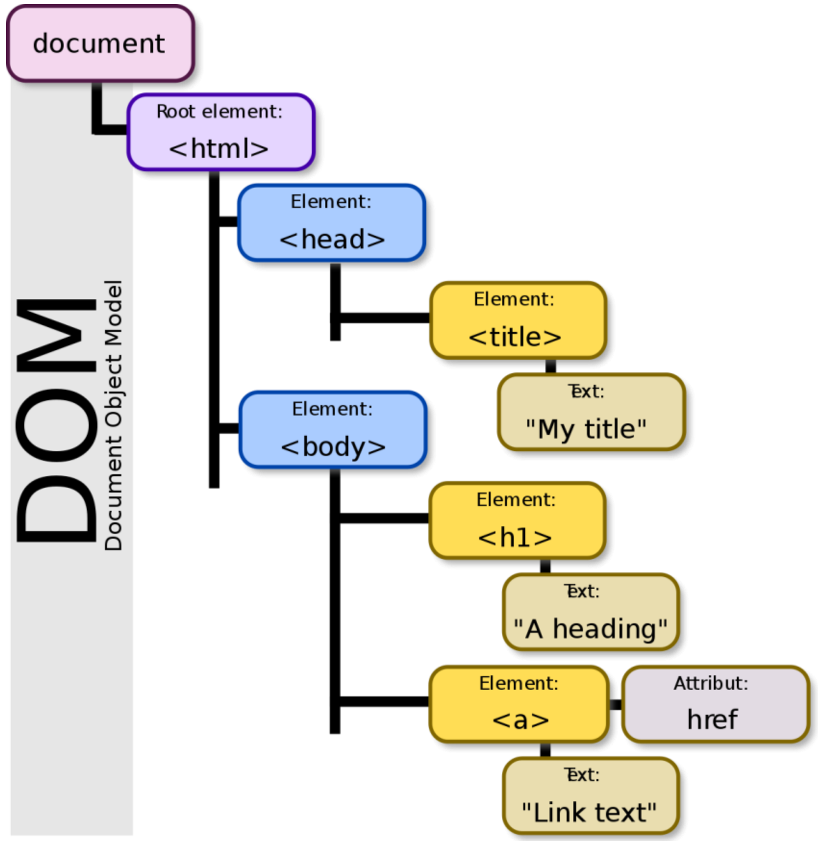
- To get an element **with a specified id**

```
document.getElementById("superhero")
```

- When a web page is loaded, the browser creates a **Document Object Model (DOM tree)** of the page.

- To access the **parent element** of an arbitrary node (`elem`) in the DOM tree, use:

```
elem.parentNode
```



Read and write HTML elements

- To assign a new **class name** to a particular ID element:

```
document.getElementById("myid").classList.add("active");
```

- To **edit a style property** of a particular ID element:

```
document.getElementById("myid").style.display = "none";
```

- To **modify the inner HTML**

```
document.getElementById("myid").innerHTML = "Inner content";
```

QuerySelector & QuerySelectorAll

- These methods return element(s) that **match a specified CSS selector(s)**.

Element Selector	`document.querySelector('td');`
Id Selector	`document.querySelector('#id2');`
Class Selector	`document.querySelectorAll('.sundays');`
Attribute Selector	`document.querySelectorAll('ol[start="5"]');`

QuerySelector & QuerySelectorAll

- The document method **querySelector**() returns **the first element** within the document that matches the specified selector, or group of selectors.
  - If no matches are found, null is returned.
- The document method **querySelectorAll**() returns **a collection of elements** within the document that matches the specified selector, or group of selectors.

Array

```
const avengers = ['Iron Man', 'Captain America',
                  'Thor', 'Hulk', 'Black Widow', 'Hawkeye'];
```

- An array is used to store and **manage a collection of elements in a specific order**.
- In JavaScript, arrays allow you to group multiple values under a single name and access those values using **numerical indices**.
- In the context of the person object, the **avengers** is an array property, you can access a specific element of the array by using the index notation **avengers[index]**. The index represents the position of the element in the array, starting from 0 for the first element.

for...of

```
const avengers = ['Iron Man', 'Captain America',
                  'Thor', 'Hulk', 'Black Widow', 'Hawkeye'];
```

- We can use a **for...of** loop to iterate through all the elements of an array

```
for (var avenger of avengers) {
  console.log(avenger);
}
```

```
Iron Man
Caption America
Thor
Hulk
Black Widow
Hawkeye
```

Form Elements

HTML Form Elements

- Input Element: The **<input>** element is used to create **text fields, checkboxes, radio buttons, and more**. It allows users to input data. The type attribute determines the specific type of input element.
- Type Attribute: The **type attribute** of the <input> element specifies the type of input control to display. Some common values include text for text fields, password for password fields, number for numeric input, checkbox for checkboxes, and radio for radio buttons.
- Select Element: The **<select>** element creates **a dropdown menu or list box**. It allows users to select options from a predefined list.
  The **<option>** elements inside the <select> element represent the **available choices**.

HTML Form Elements

- Client-side Validation: **HTML5 provides client-side validation features** that help validate form input **before submission**.
- Required Attribute: The **required** attribute is used to mark an input field as **mandatory**. When applied to an input element, it ensures that the user must provide a value in that field before submitting the form.
- Specify a **valid range** for number field, use max and min.

HTML Form Elements

- Disabled Attribute: The **disabled** attribute **disables an input element**, preventing the user from interacting with it.
- Button Element: The **<button>** element creates a clickable button.
  - The type attribute specifies the behavior of the button.
  - **type="submit"** is used to create **a submit button that triggers form submission**.