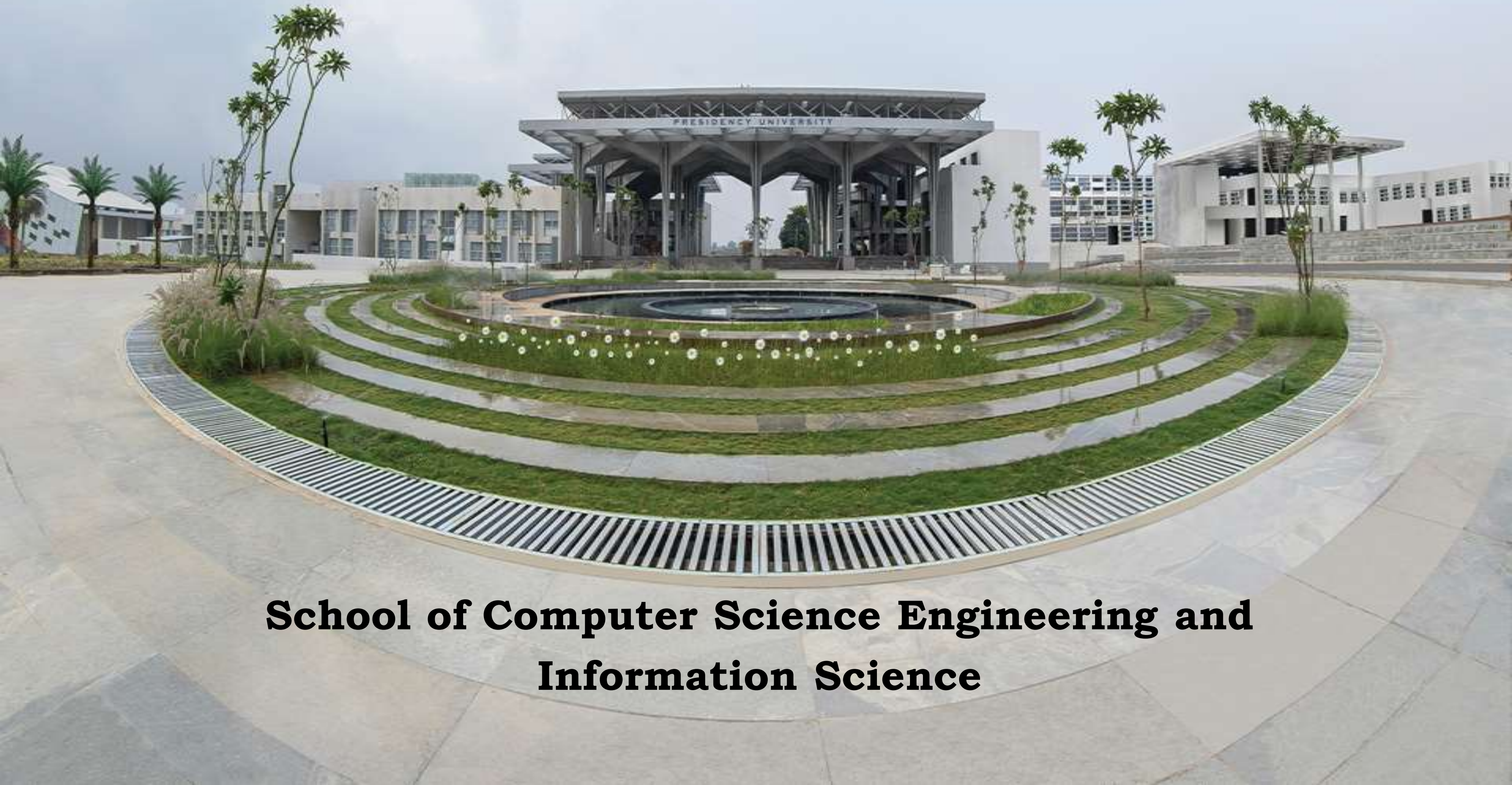
PRESIDENCY UNIVERSITY

CSE3150 – Front-end Full Stack Development



School of Computer Science Engineering and Information Science

Module 2 - Responsive web design

BootStrap for Responsive Web Design; **JavaScript – Core syntax, HTML DOM, objects, classes, Async;** Ajax and jQuery Introduction

What is Javascript?

- a lightweight programming language ("scripting language")
  - used to make web pages interactive
  - insert dynamic text into HTML (ex: user name)
  - **react to events** (ex: page load **user click**)
  - get information about a user's computer (ex: browser type)
  - perform calculations on user's computer (ex: form validation)

Javascript vs Java

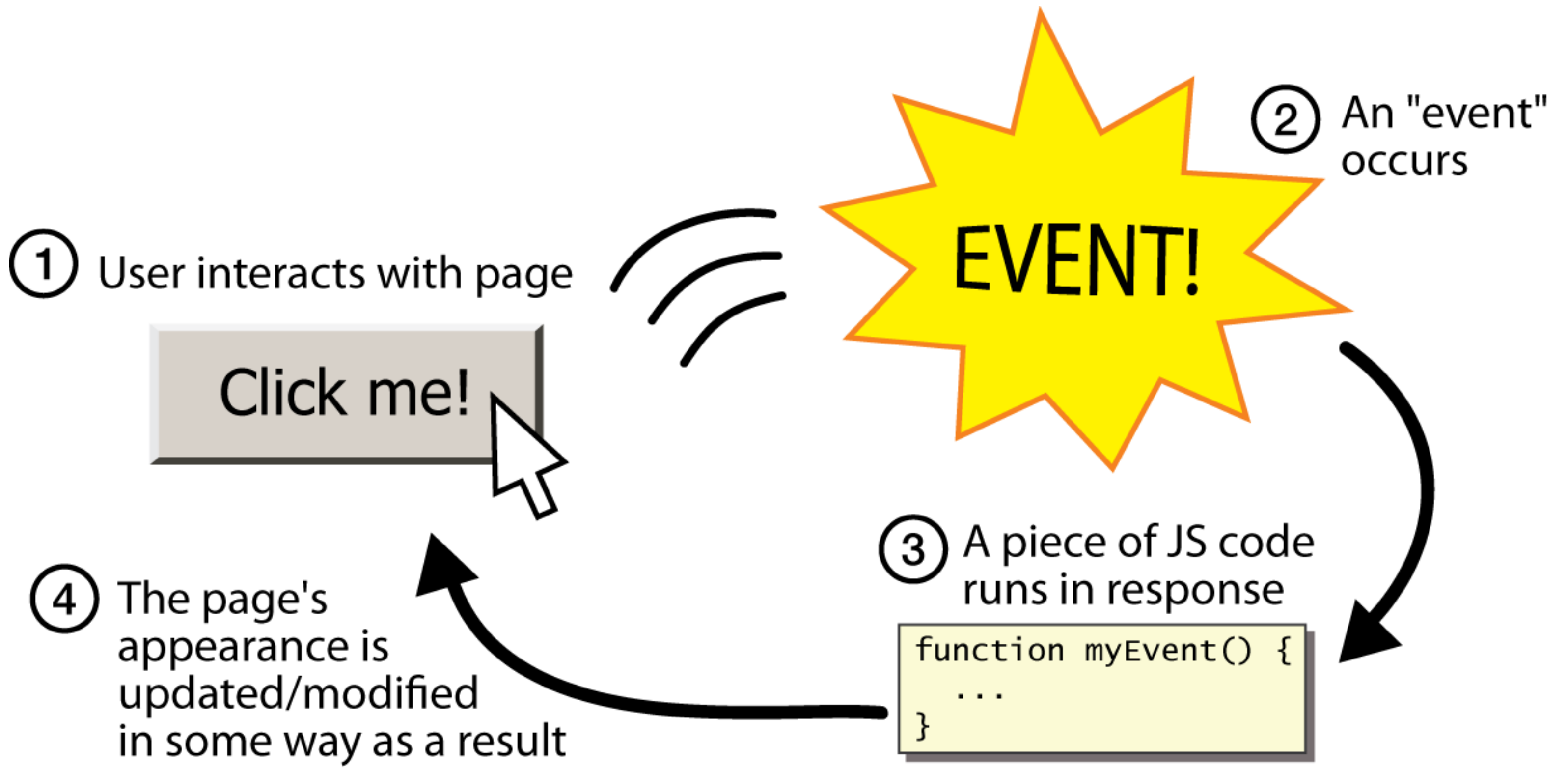
- interpreted, not compiled
- more relaxed syntax and rules
  - fewer and "looser" data types
  - variables don't need to be declared
  - errors often silent (few exceptions)
- key construct is the function rather than the class
  - "first-class" functions are used in many situations
- contained within a web page and integrates with its HTML/CSS content

Linking to a JavaScript file: `script`

```
<script src="filename" type="text/javascript">
</script>
```

- script tag should be placed in HTML page's head
- script code is stored in a separate .js file
- JS code can be placed directly in the HTML file's body or head (like CSS)

Event-driven programming


1 User interacts with page
Click me!
EVENT!
2 An "event" occurs
3 A piece of JS code runs in response
4 The page's appearance is updated/modified in some way as a result

```
function myEvent() {
  ...
}
```

Event-driven programming

- you are used to programs start with a main method (or implicit main like in PHP)
- JavaScript programs instead wait for user actions called *events* and respond to them
- event-driven programming: writing programs driven by user events
- Let's write a page with a clickable button that pops up a "Hello, World" window...

Event handlers

```
<element attributes onclick="function();">...
```

```
<button onclick="myFunction();">Click me!</button>
```

- JavaScript functions can be set as event handlers
  - when you interact with the element, the function will execute
- onclick is just one of many event HTML attributes we'll use
- but popping up an alert window is disruptive and annoying
  - A better user experience would be to have the message appear on the page...

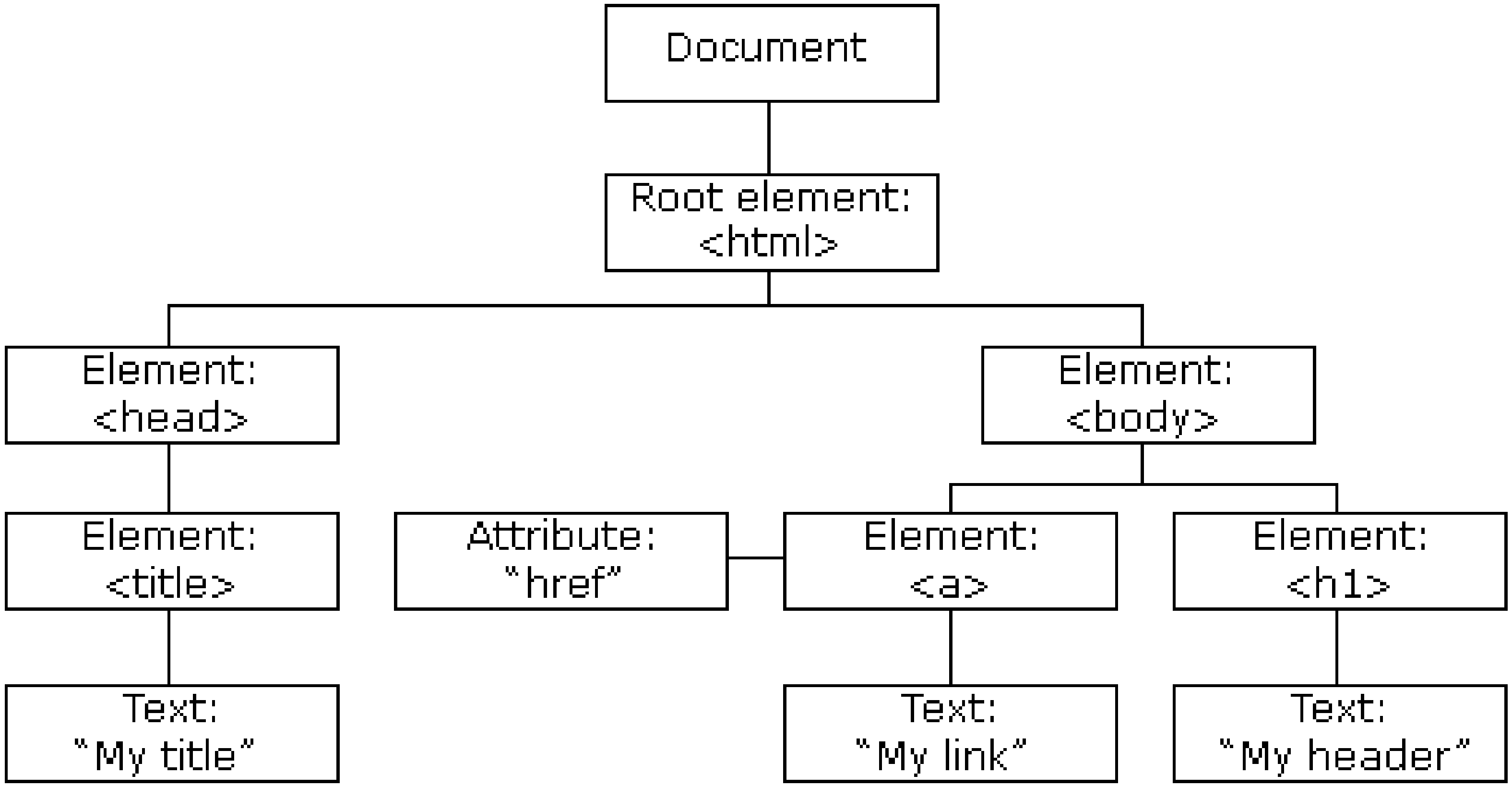
JavaScript functions

```
function name() {
statement ;
statement ;
...
statement ;
}                                                    JS
```

```
function myFunction() {
      alert("Hello!");
      alert("How are you?");
}                                                    JS
```

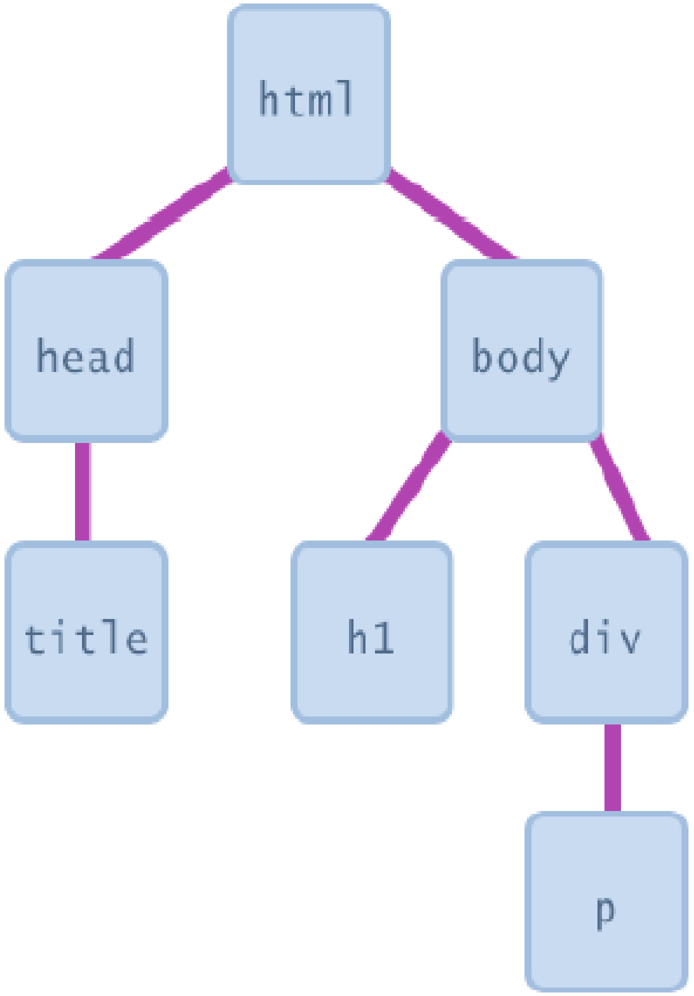
- the above could be the contents of example.js linked to our HTML page
- statements placed into functions can be evaluated in response to user events

The HTML DOM (Document Object Model)

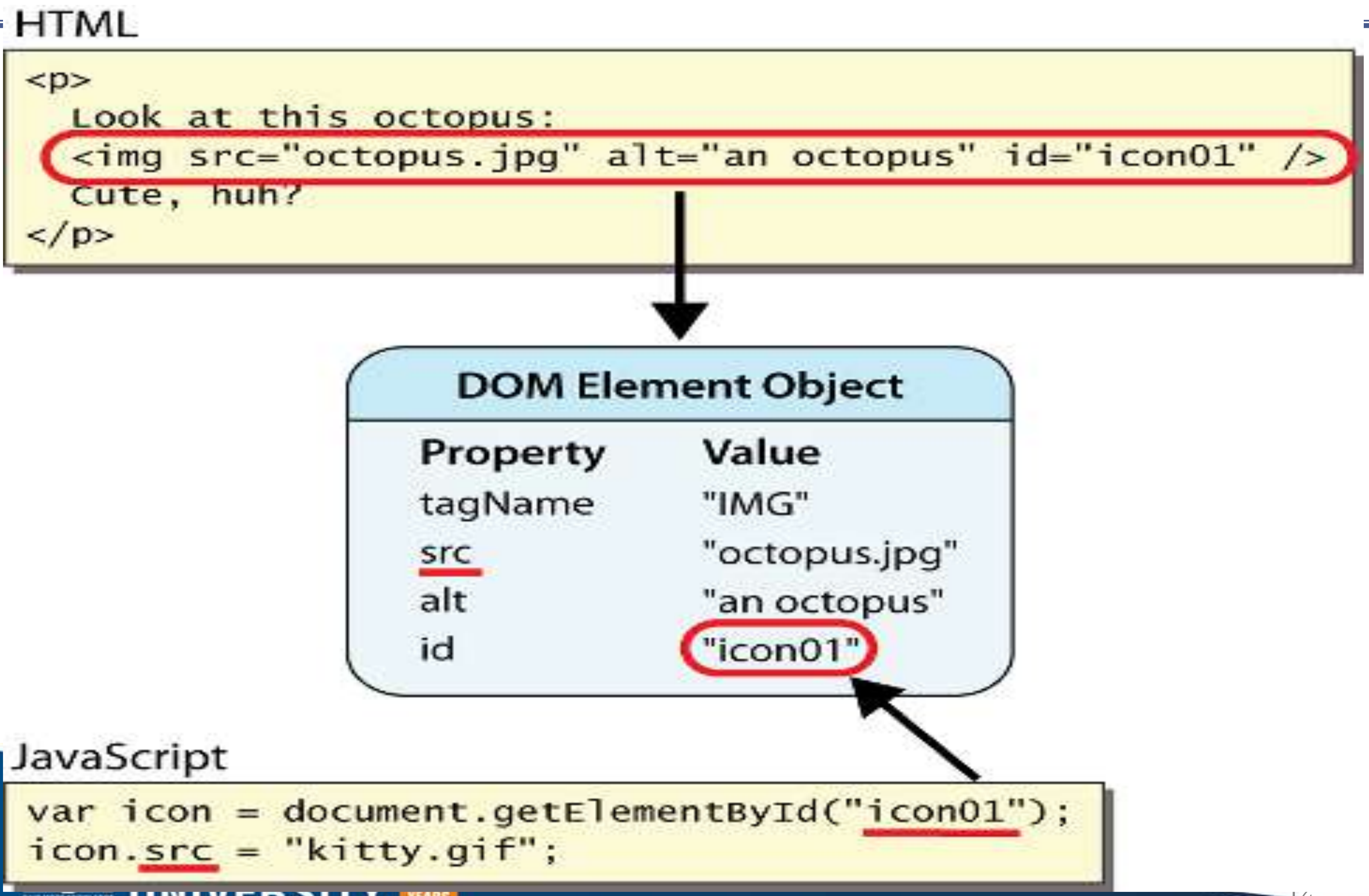


Document Object Model (DOM)

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML element and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML event in the page



DOM element objects



```
<p>
  Look at this octopus:
  <img src="octopus.jpg" alt="an octopus" id="icon01" />
  Cute, huh?
</p>
```

Property	Value
tagName	"IMG"
src	"octopus.jpg"
alt	"an octopus"
id	"icon01"

```
var icon = document.getElementById("icon01");
icon.src = "kitty.gif";
```

Accessing elements: `document.getElementById`

- document.getElementById returns the DOM object for an element with a given id
- can change the text inside most elements by setting the innerHTML property
- can change the text in form controls by setting the value property

Accessing elements: `document.getElementById`

- In the DOM, all HTML elements are defined as objects.

```
<body>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "
Hello World!";
</script>
</body>
```

The innerHTML property is useful for getting or replacing the content of HTML elements.

Accessing elements: document.getElementById

```
var name = document.getElementById("id");
```

```
<button onclick="changeText();">Click me!</button>
<span id="output">replace me</span>
<input id="textbox" type="text" />
</body>
<script>
function changeText() {
		var span = document.getElementById("output");
		var textBox = document.getElementById("textbox");
		textbox.style.color = "red";
}
</script>
```

Changing element style: `element.style`

Attribute	Property or style object
color	color
padding	padding
background-color	backgroundColor
border-top-width	borderTopWidth
Font size	fontSize
Font famiy	fontFamily

```
function changeText() {
        //grab or initialize text here

        // font styles added by JS:
        text.style.fontSize = "13pt";
        text.style.fontFamily = "Comic Sans MS";
        text.style.color = "red"; // or pink?
}                                                              JS
```

Reading Properties with JavaScript

Sample HTML

```
<ul id="t1">
<li> Item 1 </li>
</ul>
```

```
1. document.getElementById('t1').nodeName
2. document.getElementById('t1').nodeValue
3. document.getElementById('t1').firstChild.nodeName
4. document.getElementById('t1').firstChild.firstChild.nodeName
5. document.getElementById('t1').firstChild.firstChild.nodeValue
```

- Example 1 returns "ul"
- Example 2 returns "null"
- Example 3 returns "li"
- Example 4 returns "text"
  - A text node below the "li" which holds the actual text data as its value
- Example 5 returns " Item 1 "

JavaScript Output

JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML.
- Writing into the HTML output using document.write().
- Writing into an alert box, using window.alert().
- Writing into the browser console, using console.log().

Using innerHTML

To access an HTML element, JavaScript can use the document.getElementById(id) method.

The id attribute defines the HTML element. The innerHTML property defines the HTML content:

```
<html>
<body>
<h1>My First Web Page</h1>
<p>My First Paragraph</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = 5 +
6;
</script>
</body>
</html
```

Using document.write()

```
<script>
document.write(5 + 6);
</script>
```

```
<button type="button" onclick="document.write(5 + 6)">
Try it</button>
```

Displaying text

- The document.write() method writes a string of text to the browser

```
<script type="text/javascript">
  <!--
   document.write("<h1>Hello, world!</h1>");
  //-->
</script>
```

document.write()

Ends in a semicolon

```
document.write("<h1>Hello,world!</h1>");
```

Enclosed in quotes -- denotes a "string"

Using window.alert()

```
<script>
window.alert(7 +12);
</script>
```

Using console.log()

For debugging purposes, you can call the console.log() method in the browser to display data.

Comments in JavaScript

- Two types of comments
  - Single line
    - Uses two forward slashes (i.e. **//**)
  - Multiple line
    - Uses **/*** and ***/**

Language Basics

- JavaScript is case sensitive
  - onClick, ONCLICK, … are HTML, thus not case-sensitive
- Statements terminated by returns or semi-colons
  - x = x+1;

  "Blocks" of statements enclosed in { …}
- Variables
  - Define using the var statement
  - Define implicitly by its first use, which must be an assignment
    - Implicit defn has global scope, even if occurs in nested scope!

JavaScript Primitive Datatypes

- Boolean: true and false
- Number: 64-bit floating point
  - Similar to Java double and Double
  - No integer type
  - Special values NaN (not a number) and Infinity
- String: sequence of zero or more Unicode chars
  - No separate character type (just strings of length 1)
  - Literal strings using ' or " characters (must match)
- Special objects: null and undefined

4 Ways to Declare a JavaScript Variable

- Using var
- Using let
- Using const
- Using nothing

Variables

```
var name = expression;                                   JS
```

```
var clientName = "Connie Client";
var age = 32;
var weight = 127.4;                                       JS
```

- variables are declared with the var keyword (case sensitive)
- types are not specified, but JS does have types ("loosely typed")
  - `Number, Boolean, String, Array, Object, Function, Null, Undefined`
  - can find out a variable's type by calling `typeof()`

- let x = 5;
- let y = 6;
- let z = x + y;

When to Use?

- Always declare JavaScript variables with var,let, or const.
- The let and const keywords were added to JavaScript in 2015.
- If you want your code to run in older browsers, you must use var.
- If you want a general rule: always declare variables with const.
- If you think the value of the variable can change, use let.
  - const price1 = 5;
  - const price2 = 6;
  - let total = price1 + price2;

Number type

```
var enrollment = 99;
var medianGrade = 2.8;
var credits = 5 + 4 + (2 * 3);
                                                       JS
```

- integers and real numbers are the same type (no int vs. double)
- same operators: + - * / % ++ -- = += -= *= /= %=
- similar precedence to Java
- many operators auto-convert types: "2" * 3 is 6

Comments (same as Java)

```
// single-line comment
/* multi-line comment */
                                                        JS
```

- identical to Java's comment syntax
- recall: 4 comment syntaxes
  - HTML: <!-- comment -->
  - CSS/JS/PHP: /* comment */
  - Java/JS/PHP: // comment
  - PHP: # comment

Math object

```
var rand1to10 = Math.floor(Math.random() * 10 + 1);
var three = Math.floor(Math.PI);
                                                       JS
```

- methods: `abs, ceil, cos, floor, log, max, min, pow, random, round, sin, sqrt, tan`
- properties: `E, PI`

```
// Numbers:
let length = 16;
let weight = 7.5;
// Strings:
let color = "Yellow";
let lastName = "Johnson";
// Booleans
let x = true;
let y = false;
// Object:
const person = {firstName:"John", lastName:"Doe"};
// Array object:
const cars = ["Saab", "Volvo", "BMW"];
// Date object:
const date = new Date("2022-03-25");
```

Special values: null and undefined

```
var ned = null;
var benson = 9;
// at this point in the code,
// ned is null
// benson's 9
// caroline is undefined
                                              JS
```

- `undefined` : has not been declared, does not exist
- `null` : exists, but was specifically assigned an empty or null value
- Why does JavaScript have both of these?

Operators (self study)

- Refer all operators you studied in java

Logical operators

- > < >= <= && || ! == != === !==
- most logical operators automatically convert types:
  - 5 < "7" is true
  - 42 == 42.0 is true
  - "5.0" == 5 is true
- === and !== are strict equality tests; checks both type and value
  - "5.0" === 5 is false

JavaScript Strings

- <p>The length of the string is:</p>
- <p id="demo"></p>

- <script>
- let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
- document.getElementById("demo").innerHTML = text.length;
- </script>

JavaScript Functions

-
  ```
  function name(parameter1, parameter2, parameter3) {
    // code to be executed
  }
  ```

Function Invocation

- The code inside the function will execute when "something" **invokes** (calls) the function:
- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Example 1

```
<p id="demo"></p>

<script>
var x = myFunction(4, 3);
document.getElementById("demo").innerHTML = x;

function myFunction(a, b) {
  return a * b;
}
</script>
```

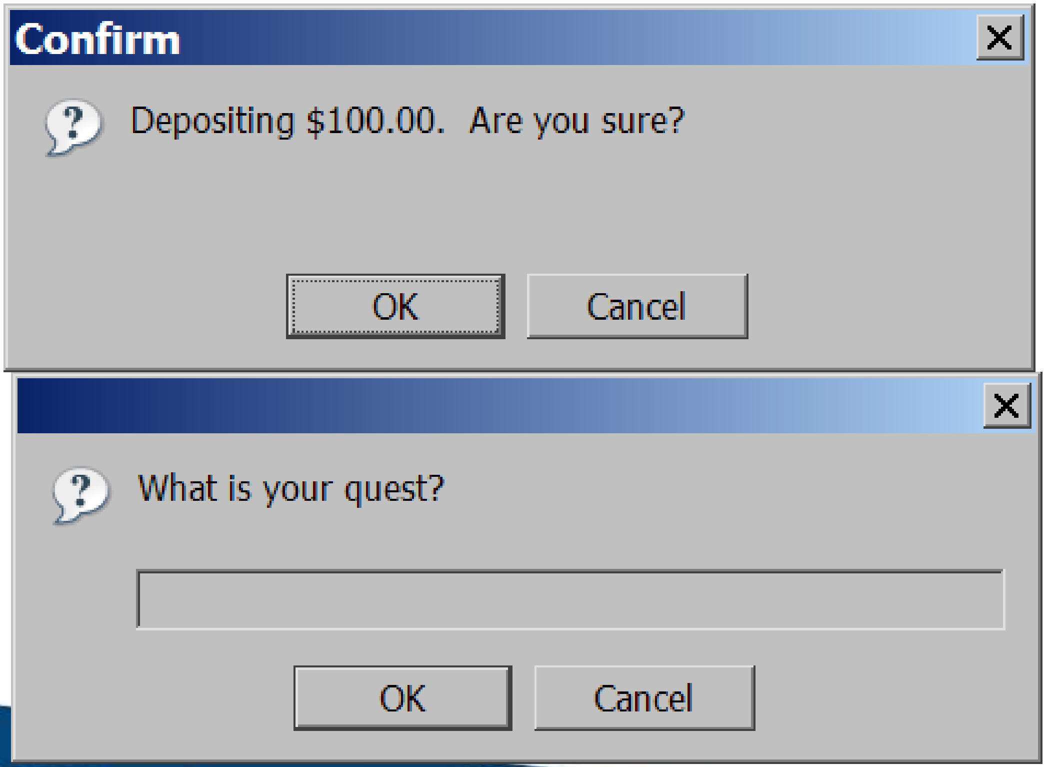
Example 2

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"The temperature is " + toCelsius(77) + " Celsius";

function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
</script>
```

Popup boxes

```
alert("message"); // message
confirm("message"); // returns true or false
prompt("message"); // returns user input string
```

JS



Depositing $100.00. Are you sure?

What is your quest?

JavaScript Objects

- Real Life Objects, Properties, and Methods
- In real life, a student is an object.
- Student can have properties name, roll_no.,marks,phone_num, age…
- Student can have methods to operate on properties like Calcualte_cgpa(), diplayInfo()….
- All student have same properties, values may change
- Assume car is an object

- `let car = "Fiat";`
- Objects are variables too. But objects can contain many values.
- `const car = {type:"Fiat", model:"500", color:"white"};`
- The values are written as **name:value** pairs (name and value separated by a colon).

```
<p id="demo"></p>
<script>
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
// Display some data from the object:
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
```

JavaScript Events

- HTML events are "things" that happen to HTML elements.
- When JavaScript is used in HTML pages, JavaScript can "react" on these events.

HTML Events

- An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

- HTML allows event handler attributes, **with JavaScript code**, to be added to HTML elements.

```
<element event='some JavaScript’>
```

```
<body>
<button
onclick="document.getElementById('demo').innerHTM
L=Date()">The time is?</button>
<p id="demo"></p>
</body>
```

PRESIDENCY UNIVERSITY

```
<body>
<h2>JavaScript HTML Events</h2>
<button onclick="this.innerHTML=Date()">The time
is?</button>
</body>
```

Try this

```
<button onclick="displayDate()">The time is?</button>
<script>
function displayDate() {
  document.getElementById("demo").innerHTML = Date();
} </script>
<p id="demo"></p>
```

JavaScript Arrays

Syntax:

- `const array_name = [item1, item2, ...];`

```
const cars = [
  "Saab",
  "Volvo",
  "BMW"
];
```

Accessing Array Elements

-
  ```
  const cars = ["Saab", "Volvo", "BMW"];
  let car = cars[0];  //Saab
  ```

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

- Objects use names to access its "members".
- In this example, person.firstName returns John:

```
<p id="demo"></p>
<script>
const person = {firstName:"John", lastName:"Doe",
age:46};
document.getElementById("demo").innerHTML =
person.firstName;
</script>
```

-
  ```
  const fruits = ["Banana", "Orange", "Apple", "Mango"];
  let length = fruits.length;   //4
  ```

Looping Array Elements

```
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = "<ul>";
fruits.forEach(myFunction);
text += "</ul>";
document.getElementById("demo").innerHTML = text;
function myFunction(value) {
  text += "<li>" + value + "</li>";
}  </script>
```

Adding elements to array

-
  ```
  const fruits = ["Banana", "Orange", "Apple"];
  fruits.push("Lemon");
  ```

  OR

-
  ```
  const fruits = ["Banana", "Orange", "Apple"];
  fruits[fruits.length] = "Lemon";
  ```

- JavaScript does not support associative arrays.
- You should use **objects** when you want the element names to be **strings (text)**.
- You should use **arrays** when you want the element names to be **numbers**.

Array methods

```
var a = ["Stef", "Jason"];          // Stef, Jason
a.push("Brian");             // Stef, Jason, Brian
a.unshift("Kelly");          // Kelly, Stef, Jason, Brian
a.pop();              // Kelly, Stef, Jason
a.shift();            // Stef, Jason
a.sort();             // Jason, Stef
```

- array serves as many data structures: list, queue, stack, ...
- methods: `concat, join, pop, push, reverse, shift, slice, sort, splice, toString, unshift`
  - push and pop add / remove from back
  - unshift and shift add / remove from front
  - shift and pop return the element that is removed

if/else statement (same as Java)

```
if (condition) {
      statements;
} else if (condition) {
      statements;
} else {
      statements;
}
                                                        JS
```

- identical structure to Java's if/else statement
- JavaScript allows almost anything as a condition

for loop (same as Java)

```
var sum = 0;
for (var i = 0; i < 100; i++) {
        sum = sum + i;
}                                                        JS
```

```
var s1 = "hello";
var s2 = "";
for (var i = 0; i < s.length; i++) {
        s2 += s1.charAt(i) + s1.charAt(i);
}
// s2 stores "hheelllloo"                                JS
```

```
<script>
const cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat",
"Audi"];
let text = "";
for (let i = 0; i < cars.length; i++) {
  text += cars[i] + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>                                  JS
```

The For In Loop

-
  ```
  for (key in object) {
    // code block to be executed
  }
  ```

-
  ```
  const person = {fname:"John", lname:"Doe",
  age:25};
  let text = "";
  for (let x in person) {
    text += person[x];
  }
  ```
- Output for text is John Doe 25

The For Of Loop

```
• for (variable of iterable) {
    // code block to be executed
  }
      • const cars = ["BMW", "Volvo", "Mini"];
        let text = "";
        for (let x of cars) {
          text += x;
        }  //  BMW  Volvo  Mini
```

```
let language = "JavaScript";
let text = "";
for (let x of language) {
 text += x ; }  // JavaScript
```

PRESIDENCY UNIVERSITY

while loops (same as Java)

```
while (condition) {
        statements;
}                                                          JS
```

```
do {
    statements;
} while (condition);
                                                           JS
```

- break and continue keywords also behave as in Java

String type

```
var s = "Connie Client";
var fName = s.substring(0, s.indexOf(" ")); // "Connie"
var len = s.length; // 13
var s2 = 'Melvin Merchant';
                                                     JS
```

- methods: `charAt`, `charCodeAt`, `fromCharCode`, `indexOf`, `lastIndexOf`, `replace`, `split`, `substring`, `toLowerCase`, `toUpperCase`
  - charAt returns a one-letter String (there is no char type)
- length property (not a method as in Java)
- Strings can be specified with "" or ''
- concatenation with + :
  - 1 + 1 is 2, but "1" + 1 is "11"

PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013

THANK YOU

PRESIDENCY UNIVERSITY

CSE3150 – Front-end Full Stack Development

Department of Computer Science Engineering

School of Engineering

Module 2 - Syllabus

[Lecture-5 Hrs, Practical-6 Hrs, Application]

BootStrap for Responsive Web Design;

JavaScript – Core syntax, HTML DOM, objects, classes, Async;

Ajax and jQuery Introduction.

Module 2 - Syllabus

BootStrap for Responsive Web Design

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Responsive Web Design

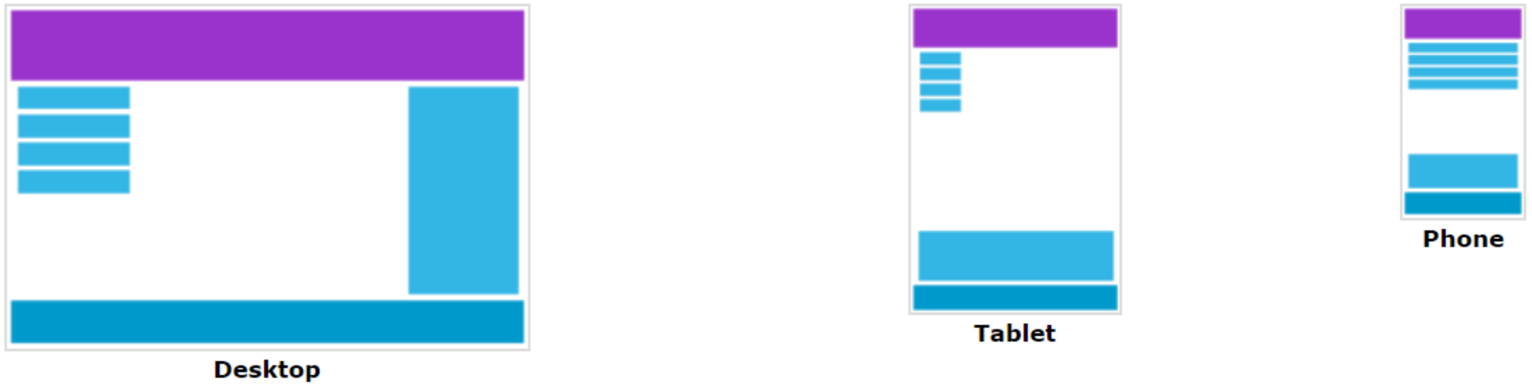
- The process of building websites & online portals with a stronger CX/UX (customer/user experience) optimal view solutions on a web page with the best browser compatibility that can run & operate in a variety of devices is known as **responsive web design**.

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.
- 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.

Responsive Web Design

- Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device.



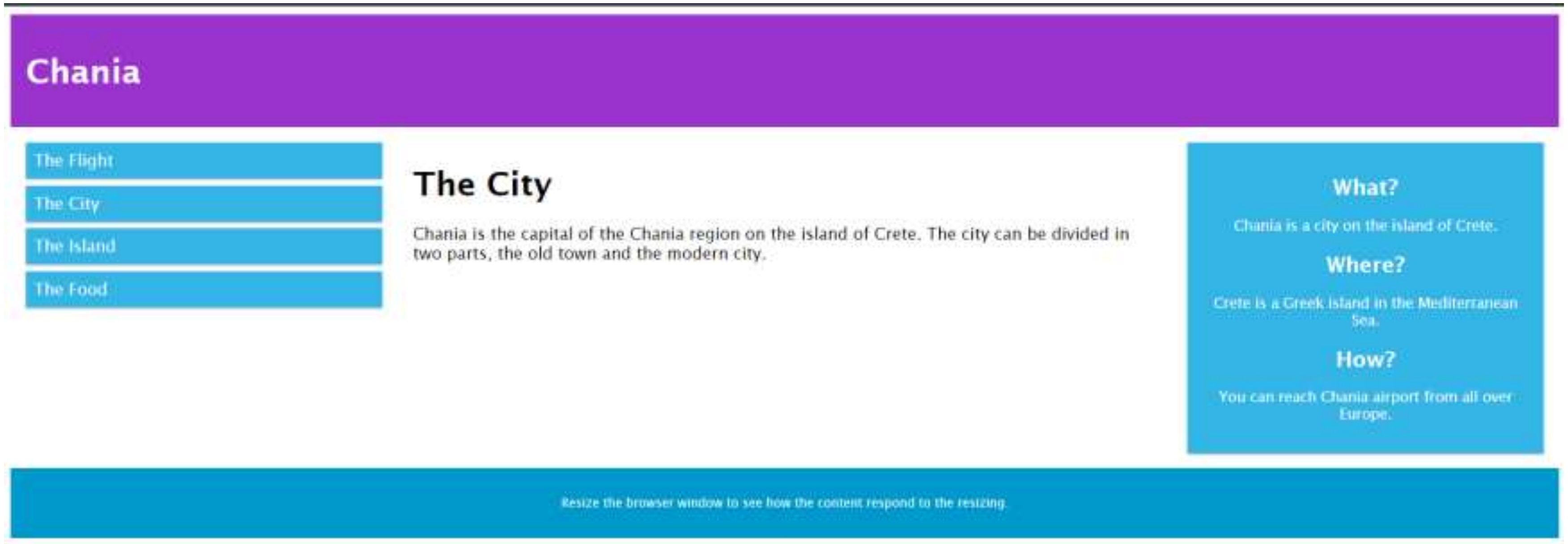
- It is called responsive web design when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

Responsive Web Design

- Responsive web design is a suitable, robust, & fast solution that enables lesser efforts from the developers' end.
- **Ethan Marcotte** first described responsive web design as responding to the needs of people and the devices they are utilizing.
- Depending on the size and capabilities of the gadget, the layout alters.
- E.g.: With a **phone**, consumers might see content presented in a single column perspective; on a **tablet**, the same content might be presented in two columns.

Responsive Web Design - Example

Refer Pgm1


Chania

The Flight

The City

The Island

The Food

The City

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.

What?

Chania is a city on the island of Crete.

Where?

Crete is a Greek island in the Mediterranean Sea.

How?

You can reach Chania airport from all over Europe.

Resize the browser window to see how the content respond to the resizing.

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Viewport

- The viewport is the user's visible area of a web page.

- The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

Setting The Viewport

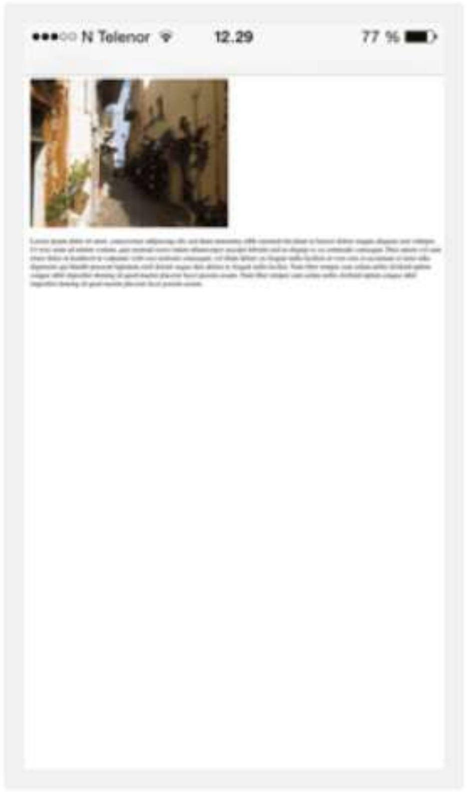
- HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.

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

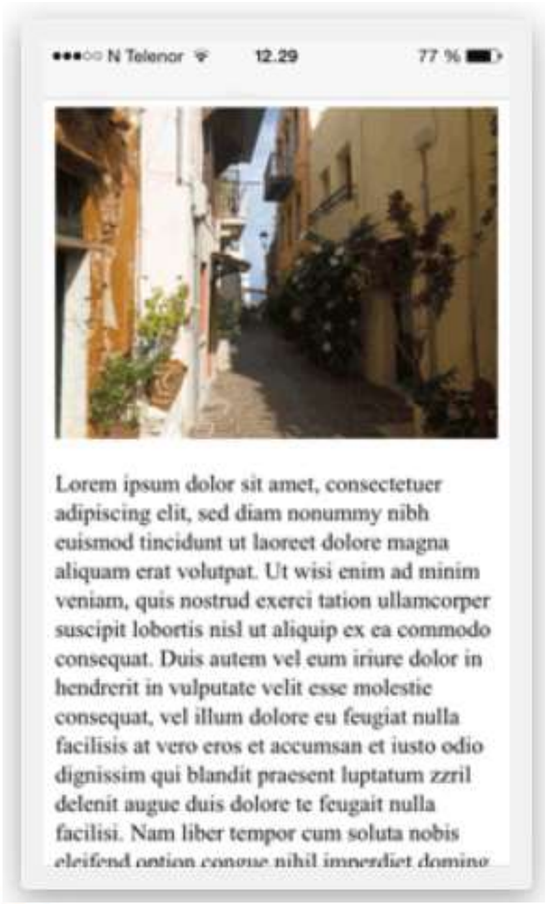
- This gives the browser instructions on how to control the page's dimensions & scaling.
- The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
- The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Setting The Viewport

Without the viewport meta tag



With the viewport meta tag



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

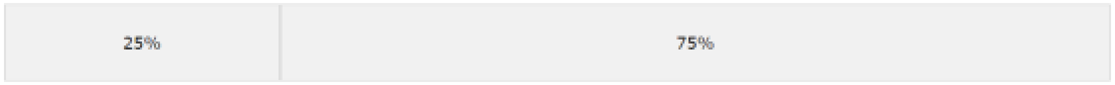
Grid View

- Many web pages are based on a grid-view, which means that the page is divided into columns
- Using a grid-view is very helpful when designing web pages.
- It makes it easier to place elements on the page.
- A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink & expand as you resize the browser window.

Building a Responsive Grid-View

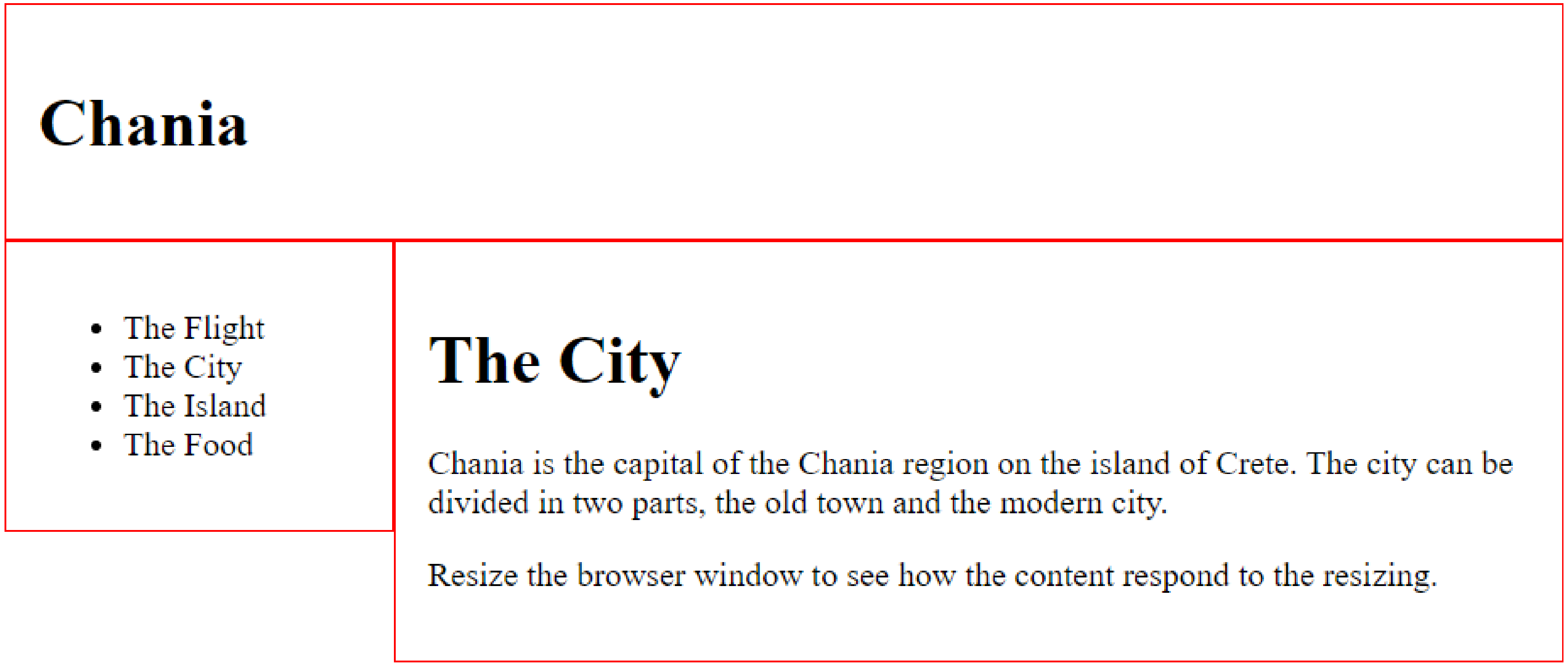
- All HTML elements have the box-sizing property set to border-box.
- This makes sure that the padding and border are included in the total width and height of the elements.

```
* {
  box-sizing: border-box;
}



.menu {
  width: 25%;
  float: left;
}
.main {
  width: 75%;
  float: left;
}
```



Building a Responsive Grid-View - Example

Refer Pgm2



Chania

- The Flight
- The City
- The Island
- The Food

The City

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.

Resize the browser window to see how the content respond to the resizing.

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Media Queries

- Media query is a CSS technique introduced in CSS3.
- It uses the @media rule to include a block of CSS properties only if a certain condition is true.
- E.g.: If the browser window is 600px or smaller, the background color will be lightblue

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

Media Queries

- Use a media query to add a breakpoint at 768px:

```
/* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
@media only screen and (max-width: 768px) {
  /* For mobile phones: */
  [class*="col-"] {
    width: 100%;
  }  }
```

Typical Device Breakpoints

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}

/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}

/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}

/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Images – Using the width & max-width Property

- If the **width property** is set to a percentage and the height property is set to "auto", the image will be responsive and scale up and down

```
img {
  width: 100%;
  height: auto;
}
```

- If the **max-width property** is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size

```
img {
  max-width: 100%;
  height: auto;
}
```

Images - Background Images

- Background images can also respond to resizing & scaling.
- 3 different methods

1. If the background-size property is set to "**contain**", the background image will scale, and try to fit the content area. However, the image will keep its aspect ratio (the proportional relationship between the image's width and height)
2. If the background-size property is set to "**100% 100%**", the background image will stretch to cover the entire content area.
3. If the background-size property is set to "**cover**", the background image will scale to cover the entire content area. Here, the "cover" value keeps the aspect ratio, and some part of the background image may be clipped.

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Videos - Add a Video

- When we want to add a video in our web page, the video will be resized to always take up all the available space

Responsive Web Design

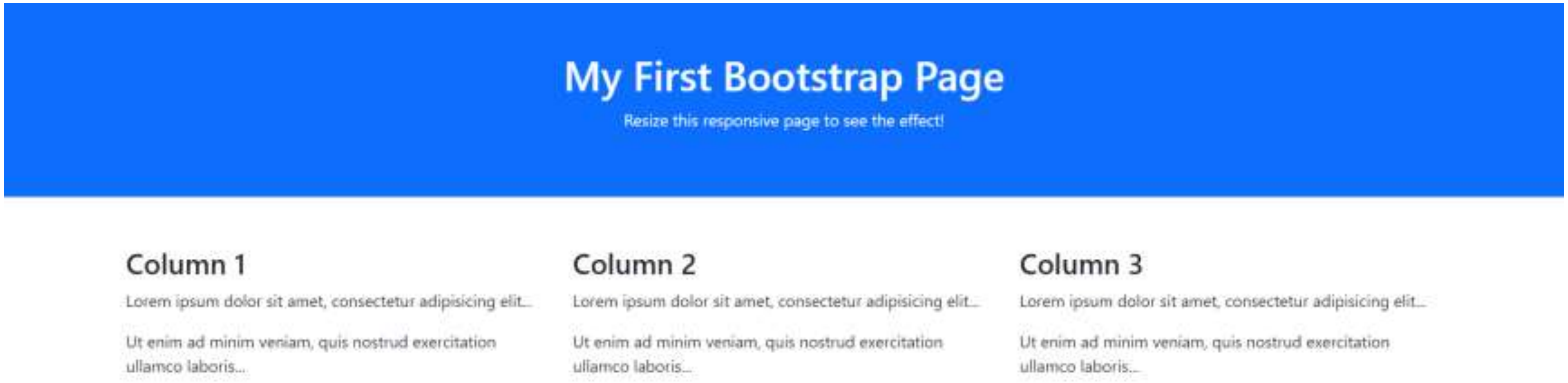
- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Frameworks

- There are many free CSS Frameworks that offer Responsive Design.
- A popular framework is **Bootstrap**. It uses HTML and CSS to make responsive web pages.
- Other Frameworks
  - **Tailwind CSS**
  - **Bulma**
  - **Materialize**
  - **Foundation by Zurb**
  - **Pure CSS**
  - **Element**
  - **Skeleton**
  - **Metro UI**
  - **Powertocss**

Frameworks – Bootstrap – Example

Refer Pgm3



My First Bootstrap Page

Resize this responsive page to see the effect!

Column 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...

Column 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...

Column 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...

Responsive Web Design

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Templates

- There are some responsive templates available with the CSS framework.
- You are free to modify, save, share, and use them in all your projects.
- E.g.:
  - Ecommerce
  - Education
  - Restaurant
  - Art Template
  - Architect Template
  - Blog Template
  - CV Template

Responsive Web Design (Completed)

- Introduction
- Viewport
- Grid View
- Media Queries
- Images
- Videos
- Frameworks
- Templates

Bootstrap

- Introduction
- Create Your First Web Page With Bootstrap 5
- Containers
- Grid System
- Colors
- Tables
- Images
- Buttons
- Elements
- Forms
- Select menu
- Validation
- Components

Bootstrap

- **Bootstrap 5** is the newest version of Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for creating responsive, mobile-first websites.
- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, & other interface components.
- Bootstrap is a free front-end framework for faster and easier web development

Bootstrap

- Bootstrap was developed by **Mark Otto** and **Jacob Thornton** at Twitter, and released as an open source product in August 2011 on GitHub.
- It contains pre-built components and design elements to style HTML content.
- Modern browsers such as Chrome, Firefox, Opera, Safari, & Internet Explorer support Bootstrap.
- How to Use :
  - Download Bootstrap from getbootstrap.com
  - Include Bootstrap from a CDN

Bootstrap

- Download Bootstrap from getbootstrap.com
  - If you want to download and host Bootstrap yourself, go to **getbootstrap.com**, and follow the instructions there.
- Include Bootstrap from a CDN
  - If you don't want to download and host Bootstrap yourself, you can include it from a **CDN** (Content Delivery Network).
  - MaxCDN provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery.

Bootstrap CDN

- You must include the following Bootstrap's CSS, JavaScript, and jQuery from MaxCDN into your web page.

```
<!-- Latest compiled and minified Bootstrap CSS -->
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<!-- Latest compiled Bootstrap JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<!-- latest jQuery library -->
<script src="https://code.jquery.com/jquerylatest.js"></script>
```

Advantages of Bootstrap

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)

Advantage of using the Bootstrap CDN

- Many users already have downloaded Bootstrap from MaxCDN when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time.
- Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

Create Your First Web Page With Bootstrap 5

- Bootstrap 5 uses HTML elements and CSS properties that require the HTML5 doctype.
- Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct title and character set

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Bootstrap 5 Example</title>
    <meta charset="utf-8">
  </head>
</html>
```

Bootstrap 5 is mobile-first

- Bootstrap 5 is designed to be responsive to mobile devices.
- Mobile-first styles are part of the core framework.
- To ensure proper rendering and touch zooming, add the following <meta> tag inside the <head> element.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

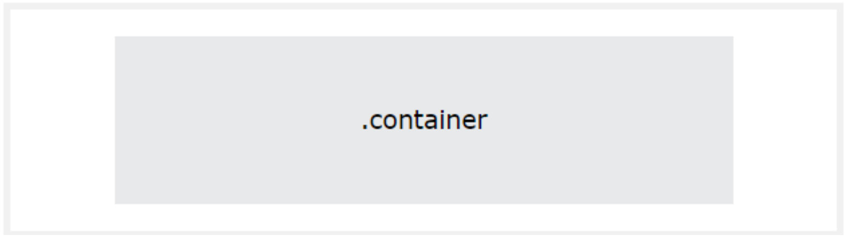
  - The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
  - The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.

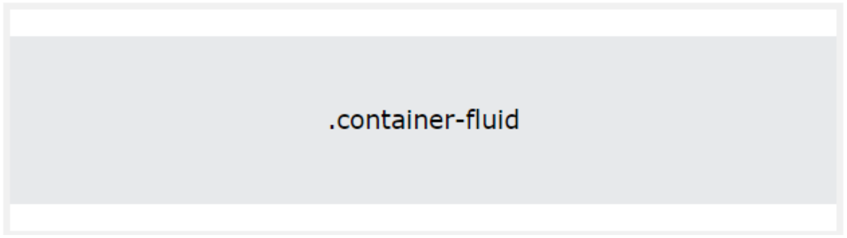
Containers

- Bootstrap 5 also requires a containing element to wrap site contents.
- There are 2 container classes to choose from:
  1. The **.container class** provides a responsive fixed width container
  2. The **.container-fluid class** provides a full width container, spanning the entire width of the viewport

Containers

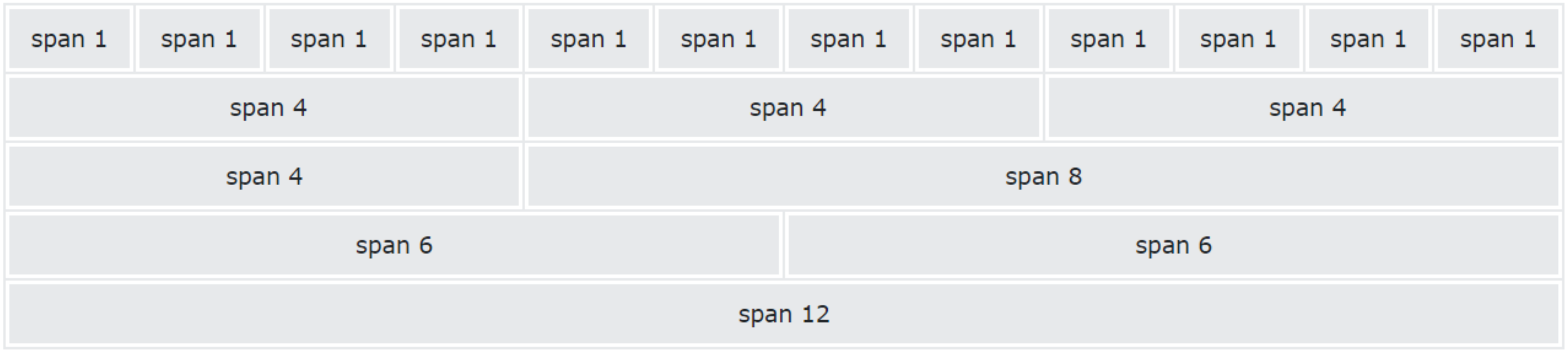
- Bootstrap 5 also requires a containing element to wrap site contents.
- There are 2 container classes to choose from:
  1. The **.container class** provides a responsive fixed width container
  2. The **.container-fluid class** provides a full width container, spanning the entire width of the viewport





Grid System

- Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.
- If you do not want to use all 12 columns individually, you can group the columns together to create wider columns


span 1 span 1 span 1 span 1 span 1 span 1 span 1 span 1 span 1 span 1 span 1 span 1
span 4 span 4 span 4
span 4 span 8
span 6 span 6
span 12

Grid Classes

- The Bootstrap 5 grid system has **six classes**:
  - **.col**- (extra small devices - screen width less than 576px)
  - **.col-sm**- (small devices - screen width equal to or greater than 576px)
  - **.col-md**- (medium devices - screen width equal to or greater than 768px)
  - **.col-lg**- (large devices - screen width equal to or greater than 992px)
  - **.col-xl**- (xlarge devices - screen width equal to or greater than 1200px)
  - **.col-xxl**- (xxlarge devices - screen width equal to or greater than 1400px)

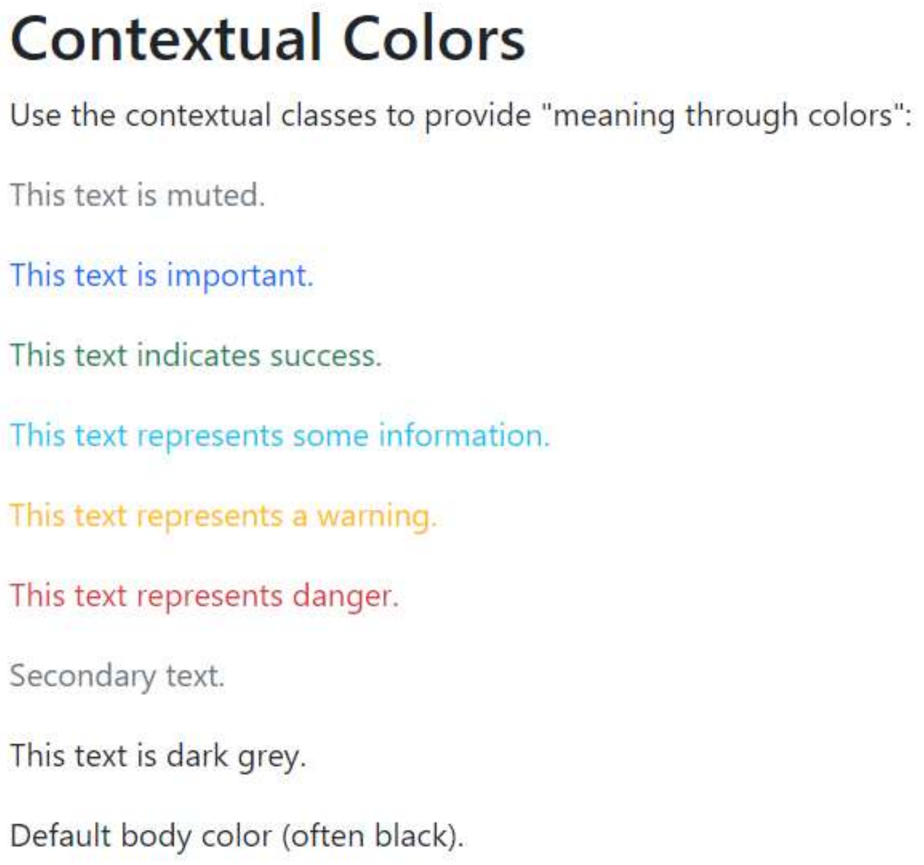
Colors

- Bootstrap 5 has some contextual classes that can be used to provide "meaning through colors".
- The classes for text colors are:

  .text-muted, .text-primary, .text-success, .text-info, .text-warning, .text-danger, .text-secondary, .text-white, .text-dark, .text-body (default body color/often black) and .text-light

Colors – Example

Refer Pgm4



Contextual Colors

Use the contextual classes to provide "meaning through colors":

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

This text is dark grey.

Default body color (often black).

Tables

- A basic Bootstrap 5 table has a light padding and horizontal dividers.
  - The **.table** class adds basic styling to a table
  - The **.table-striped** class adds zebra-stripes to a table
  - The **.table-bordered** class adds borders on all sides of the table and cells
  - The **.table-hover** class adds a hover effect (grey background color) on table rows
  - The **.table-dark** class adds a black background to the table
  - Combine **.table-dark** and **.table-striped** to create a dark, striped table
  - The .**table-borderless** class removes borders from the table
  - The **.table-sm** class makes the table smaller by cutting cell padding in half
  - The **.table-responsive** class adds a scrollbar to the table when needed (when it is too big horizontally)

Images

- **Rounded Corners**
  - The **.img-rounded** class adds rounded corners to an image (IE8 does not support rounded corners)
- **Circle**
  - The **.img-circle** class shapes the image to a circle (IE8 does not support rounded corners)
- **Thumbnail**
  - The **.img-thumbnail** class shapes the image to a thumbnail

Images

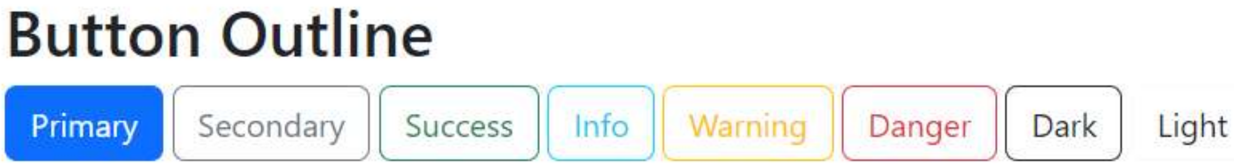
- **Responsive Images**
  - Images comes in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen.
  - Create responsive images by adding an .img-responsive class to the **<img>** tag. The image will then scale nicely to the parent element.
  - The **.img-responsive** class applies display: block; and max-width: 100%; and height: auto; to the image

Buttons

- Bootstrap provides seven styles of buttons with the following classes:

  .btn-default
  .btn-primary
  .btn-success
  .btn-info
  .btn-warning
  .btn-danger
  .btn-link

- **Refer Pgm5**



Button Sizes

- Bootstrap provides 4 button sizes with the following classes:

    .btn-lg

    .btn-md

    .btn-sm

    .btn-xs

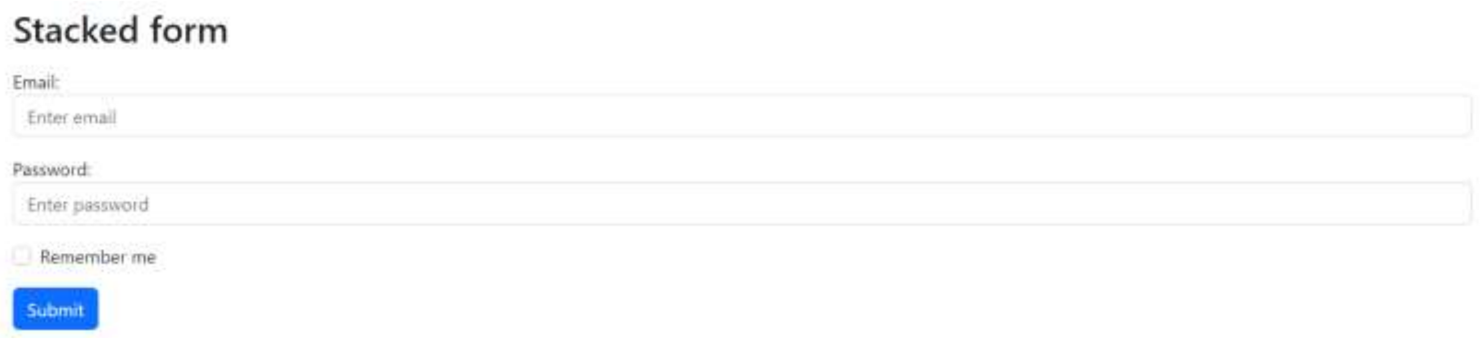
Elements

- The button classes can be used on the following elements:
  - **<a>**
  - **<button>**
  - **<input>**

Forms – Stacked Form

- All textual <input> and <textarea> elements with class **.form-control** get proper form styling

- **Refer Pgm6**



Stacked form

Email:

Enter email

Password:

Enter password

Remember me

Select - Select Menu

- Select menu (select one)
- Multiple select menu (hold ctrl or shift (or drag with the mouse) to select more than one)

- Select menus are used if you want to allow the user to pick from multiple options.
- To style a select menu in Bootstrap 5, add the **.form-select** class to the **<select>** element:

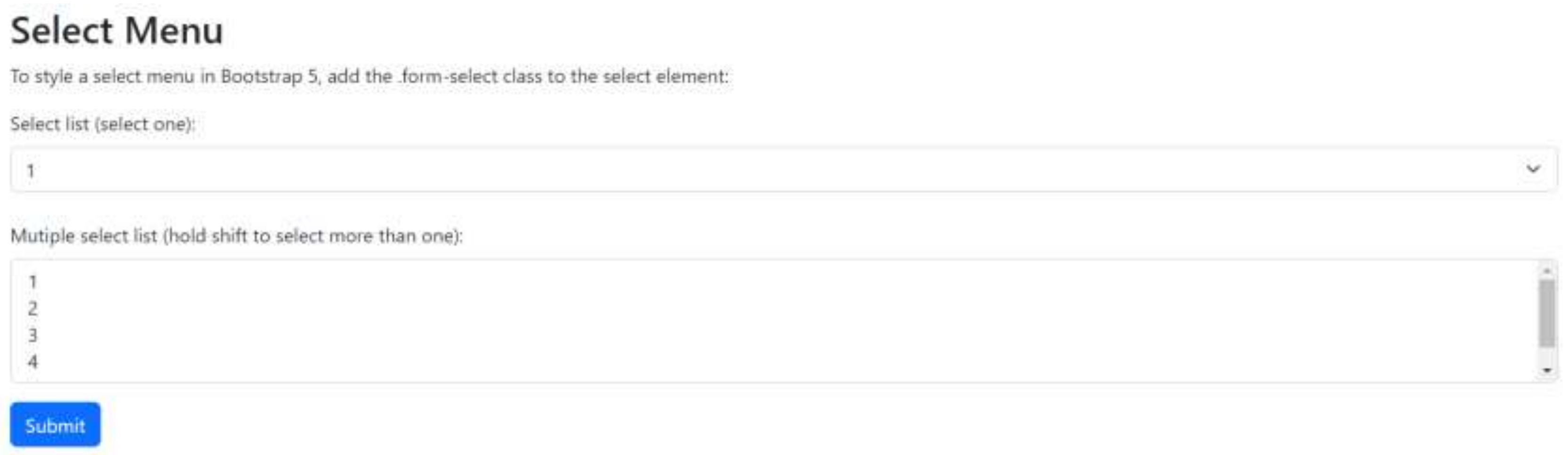
Select - Select Menu

- Select menu (select one)
- Multiple select menu (hold ctrl or shift (or drag with the mouse) to select more than one)

- Select menus are used if you want to allow the user to pick from multiple options.
- To style a select menu in Bootstrap 5, add the **.form-select** class to the **<select>** element:

Select - Select Menu – Example

- **Refer Pgm7**



Select Menu

To style a select menu in Bootstrap 5, add the .form-select class to the select element:

Select list (select one):

1

Mutiple select list (hold shift to select more than one):

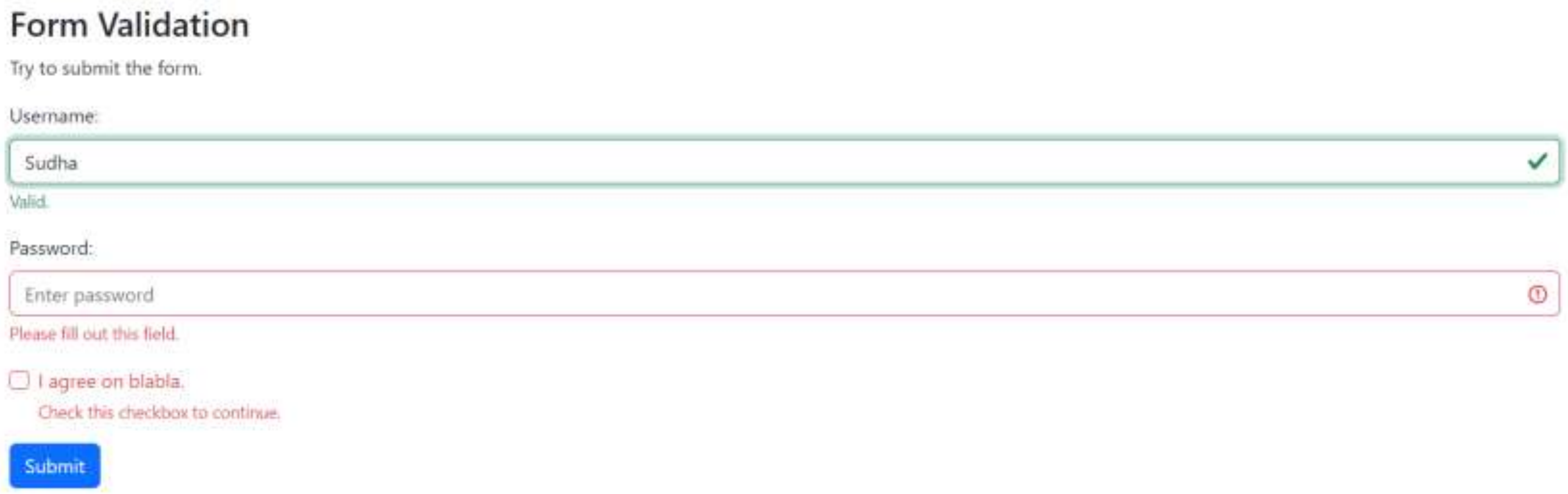
1
2
3
4

Validation - Form Validation

- You can use different validation classes to provide valuable feedback to users.
- Add either **.was-validated** or **.needs-validation** to the <form> element, depending on whether you want to provide validation feedback before or after submitting the form.
- The input fields will have a green (valid) or red (invalid) border to indicate what's missing in the form.
- You can also add a **.valid-feedback** or **.invalid-feedback** message to tell the user explicitly what's missing, or needs to be done before submitting the form.

Validation - Form Validation

- **Refer Pgm8**



Form Validation

Try to submit the form.

Username:

Sudha ✓

Valid.

Password:

Enter password

Please fill out this field.

☐ I agree on blabla.

Check this checkbox to continue.

Components (Self study topics)

- Accordion
- Alerts
- Badge
- Breadcrumb
- Buttons
- Button group
- Card
- Carousel
- Close button
- Collapse
- Dropdowns
- List group
- Modal
- Navs & tabs
- Navbar
- Offcanvas
- Pagination
- Popovers
- Progress
- Scrollspy
- Spinners
- Toasts
- Tooltips

Bootstrap (Completed)

- Introduction
- Create Your First Web Page With Bootstrap 5
- Containers
- Grid System
- Colors
- Tables
- Images
- Buttons
- Elements
- Forms
- Select menu
- Validation
- Components

Module 2 - Syllabus

[Lecture-5 Hrs, Practical-6 Hrs, Application]

BootStrap for Responsive Web Design; (Completed)

JavaScript – Core syntax, HTML DOM, objects, classes, Async;

Ajax and jQuery Introduction.

THANK YOU