

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

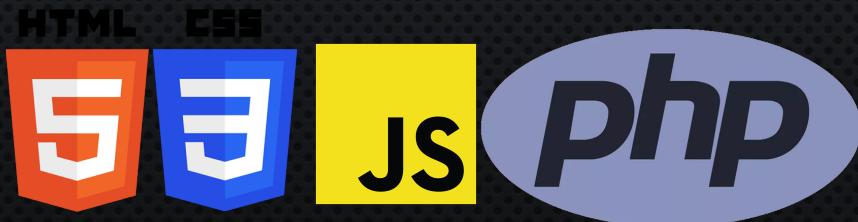


# Web Programming

## Front-End Development

Instructor: **Anas Tawalbeh**

JavaScript (JS)



# OUTLINE

- INTRODUCTION
- HTML AND WEBPAGES PARTS
- CSS AND DIVIDING WEBPAGE PARTS

# INTRODUCTION

- THREE LANGUAGES USED FOR FRONT-END DEVELOPMENT
  - HTML
  - CSS
  - JS
- 98% OF WEBSITES USE JAVASCRIPT ON THE CLIENT SIDE
  - FOR WEB PAGE BEHAVIOUR



# JAVASCRIPT OVERVIEW

- JAVASCRIPT IS A SCRIPTING LANGUAGE
  - PROGRAMMING LANGUAGE WRITTEN FOR A SPECIAL RUNTIME ENVIRONMENT
- JAVASCRIPT ENGINE EMBEDDED IN WEB BROWSER TO EXECUTE THE CODE ON CLIENT

# HISTORY

- NETSCAPE
  - NETSCAPE NAVIGATOR
- INVENT NEW LANGUAGE, SIMILAR SYNTAX TO JAVA
  - LIVESCRIPT (BETA RELEASE)
  - JAVASCRIPT (OFFICIAL RELEASE)
- MICROSOFT
  - INTERNET EXPLORER (JSCRIPT)
- SUBMIT JAVASCRIPT TO ECMA
  - EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

**ES6**  
**ECMA Script**

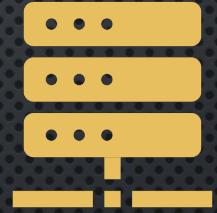
# JAVASCRIPT COMPONENTS



Core (operators,  
expressions, statements, &  
subprograms)



Client-side (objects control  
user interface)



Server-side (object useful  
for server)

# JS CORE

- 25 RESERVED WORDS

<b>If</b>	<b>else</b>	<b>switch</b>	<b>for</b>	<b>while</b>	<b>function</b>	<b>var</b>	<b>this</b>	<b>switch</b>
try	catch	throw	in	return	typeof	case	break	Void
continue	finally	instanceof	new	with	default	do	delete	NULL

- JS COMMENTS DENOTED BY
  - // SINGLE LINE COMMENT
  - /\* SINGLE AND MULTIPLE LINE COMMENT(S) \*/
- JS DATA TYPE, OPERATIONS, AND EXPRESSIONS ARE LIKE THOSE OF OTHER PROGRAMMING LANGUAGES

# JS DATA TYPE

- NUMBER
  - 33, 3.3, -3
- STRING
  - "MY NAME IS ANAS"
- BOOLEAN
  - TRUE/FALSE
- UNDEFINED
- NULL

# JS OPERATIONS AND EXPRESSIONS

- THE ASSIGNMENT OPERATOR `=, +=, -=, ETC.`
- ARITHMETIC OPERATORS `(*, /, %, +, -)`
- UNARY OPERATORS `(+, -)`
- INCREMENT AND DECREMENT OPERATORS `(++,--)`
- PRECEDENCE RULE IS APPLIED
- COMPARISON OPERATORS
- LOGICAL OPERATORS

# DECLARING VARIABLES IN JS

- VAR VARIABLENAME;
  - VAR GPA;
  - VAR STUDNETNAME= “ANAS”;
  - VAR MAJOROFSTUDY=“IT”, LISTOFCOURSES;
- VAR VS LET

# ARRAY DECLARATION

```
const arrayName = [element1, element2, etc];
```

```
var arrayName;  
arrayName = [element1, element2, etc];
```

# SELECTION STATEMENT

- IF
- ELSE IF
- ELSE
- SWITCH

# SELECTION STATEMENT

```
if (time < 10) {  
    greeting = "Good morning";  
} else if (time < 20) {  
    greeting = "Good day";  
} else {  
    greeting = "Good evening";  
}
```

```
var x = "0";  
switch (x) {  
    case 0:  
        text = "Off";  
        break;  
    case 1:  
        text = "On";  
        break;  
    default:  
        text = "No value found";  
}
```

# LOOP STATEMENTS

- WHILE (CONDITION){//STATEMENTS}
- DO { //STATEMENTS } WHILE (CONDITION)
- FOR LOOP

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

# INPUT AND OUTPUT METHODS

- THERE ARE THREE METHODS IN THE WINDOW OBJECT
- APPEARED AS POPUP BOXES
  - ALERT
  - CONFIRM
  - PROMPT

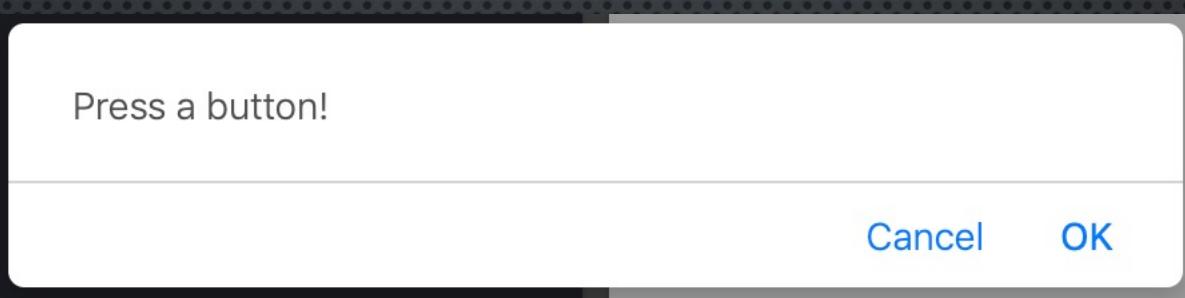
# ALERT

It's between 0 and 10

[Close](#)

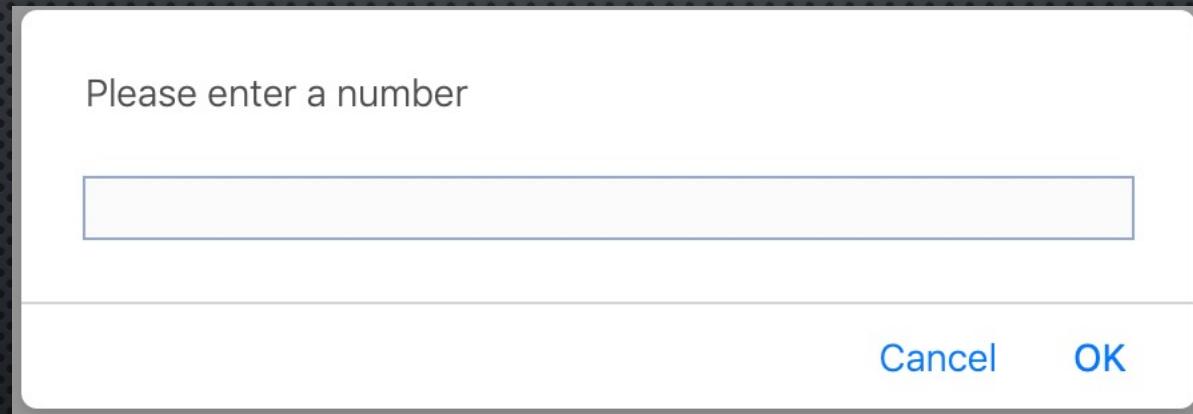
```
window.alert("It's between 0 and 10");
```

# CONFIRM



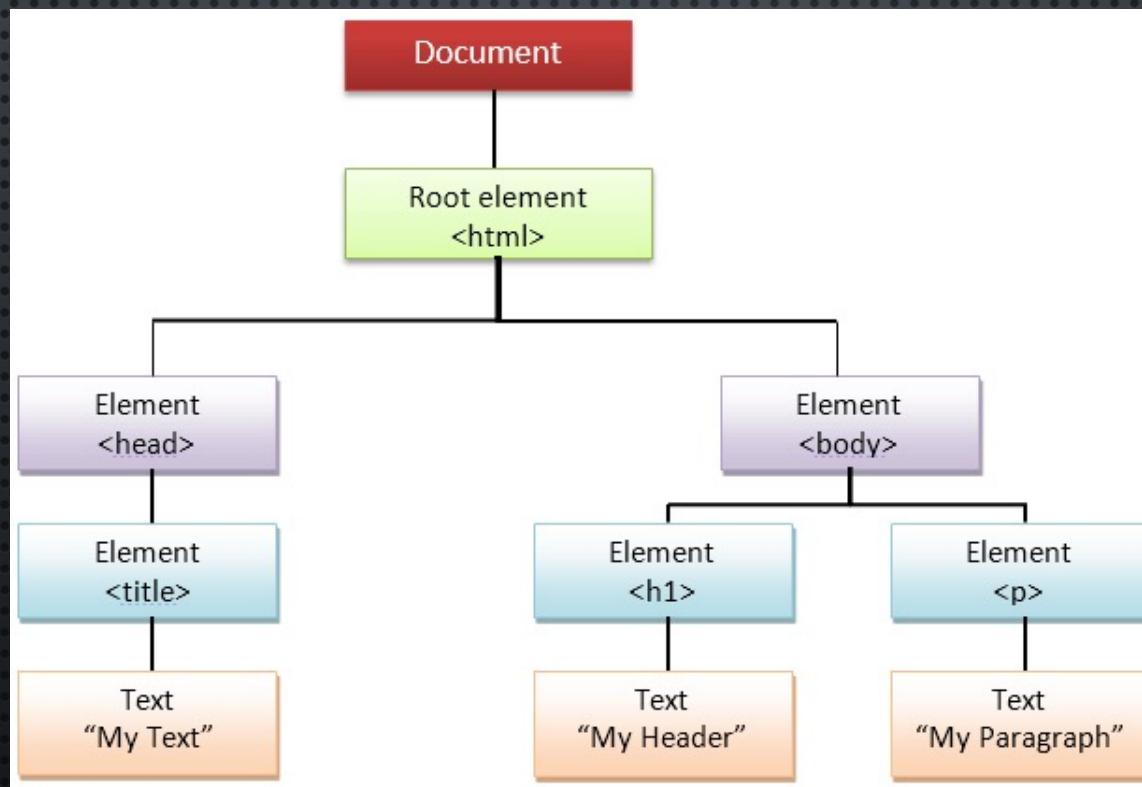
```
window.confirm("Horaaay ! !\nYou guessed it.");
```

# PROMPT



```
prompt("Please enter a number between 1-10");
```

# THE HTML DOM (DOCUMENT OBJECT MODEL)



# HTML DOM

- THE HTML DOM IS A STANDARD OBJECT MODEL AND PROGRAMMING INTERFACE FOR HTML.
- HTML DOM MODEL IS CONSTRUCTED AS A TREE OF OBJECTS
- HTML DOM DEFINES:
  - THE HTML ELEMENTS AS OBJECTS
  - THE PROPERTIES OF ALL HTML ELEMENTS
  - THE METHODS TO ACCESS ALL HTML ELEMENTS
  - THE EVENTS FOR ALL HTML ELEMENTS
- IN OTHER WORDS: THE HTML DOM IS A STANDARD FOR HOW TO GET, CHANGE, ADD, OR DELETE HTML ELEMENTS.

# THE DOM PROGRAMMING INTERFACE

- THE HTML DOM CAN BE ACCESSED WITH JAVASCRIPT (AND WITH OTHER PROGRAMMING LANGUAGES).
- IN THE DOM, ALL HTML ELEMENTS ARE DEFINED AS OBJECTS.
- THE PROGRAMMING INTERFACE IS THE PROPERTIES AND METHODS OF EACH OBJECT.
- A PROPERTY IS A VALUE THAT YOU CAN GET OR SET (LIKE CHANGING THE CONTENT OF AN HTML ELEMENT).
- A METHOD IS AN ACTION YOU CAN DO (LIKE ADD OR DELETING AN HTML ELEMENT).

# THE HTML DOM DOCUMENT OBJECT

- THE DOCUMENT OBJECT REPRESENTS YOUR WEB PAGE.
- ACCESS THE DOCUMENT OBJECT IS ESSENTIAL TO ACCESS ANY ELEMENT IN AN HTML PAGE

# JS HTML DOM

JS USE HTML DOM TO CREATE DYNAMIC WEBPAGE

- CHANGE TML ELEMENTS
- CHANGE HTML ATTRIBUTES
- CHANGE CSS STYLES
- REMOVE EXISTING HTML ELEMENTS AND ATTRIBUTES
- ADD NEW HTML ELEMENTS AND ATTRIBUTES
- REACT TO ALL EXISTING HTML EVENTS IN THE PAGE
- CREATE NEW HTML EVENTS IN THE PAGE

# JS HTML DOM METHODS

HTML DOM METHODS ARE IN THE DOM INTERFACE  
IT IS USED BY JS

Get an HTML Element by Id

```
document.getElementById("intro");
```

Get an HTML Elements by Tag Name

```
document.getElementsByTagName("p");
```

Get an HTML Elements by Class Name

```
document.getElementsByClassName("intro");
```

Get an HTML Elements by CSS Selectors

```
document.querySelectorAll("p.intro");
```

# EVENT HANDLING

- CLICK
- FOCUS
- MOUSEOVER
- SELECT
- SUBMIT
- RESET

# FORM VALIDATION

- CHECK THE FORM CONTENTS
- LET THE USER INSERT CORRECT & REAL INPUT

# FORM VALIDATION

```
<form name="myForm" action="/action_page.php" onsubmit="return validateForm()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
```

```
function validateForm() {
  let x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
```

# OPEN AND CLOSE MENU

```
function myFunction() {  
  var x = document.getElementById("myLinks");  
  if (x.style.display === "block") {  
    x.style.display = "none";  
  } else {  
    x.style.display = "block";  
  }  
}
```



# WORKSHOP

- ADD “CONTACT ME” FORM TO YOUR WEB PAGE
- USER IS REQUIRED TO ENTER THE FOLLOWING
  - NAME
  - EMAIL
  - SUBJECT
  - MESSAGE
- USER MIGHT ENTER
  - PHONE NUMBER
- VALIDATE THE USER INPUTS USING JAVASCRIPT