



موسسه آموزش عالی غیردولتی غیرانتفاعی بصیر بکیر

INTERNET ENGINEERING

Basir University, 2020-2021

By: [Prof. Dr. Mohammad Hajarian](#)



موسسه آموزش عالی غیردولتی غیرانتفاعی بصیرتیک

- Session 10

JAVASCRIPT



موسسه آموزش عالی غیردولتی غیرانتفاعی بصیرتک

OVERVIEW OF JAVASCRIPT AND DOM

INTRODUCTION TO JAVASCRIPT



- What is JavaScript?
 - It is designed to add interactivity to HTML pages
 - It is a scripting language (a lightweight programming language)
 - It is an interpreted language (it executes without preliminary compilation)
 - Usually embedded directly into HTML pages
 - And, Java and JavaScript are different

WHAT CAN A JAVASCRIPT DO?



- JavaScript gives HTML designers a programming tool:
 - simple syntax
- JavaScript can put dynamic text into an HTML page
- JavaScript can react to events
- JavaScript can read and write HTML elements
- JavaScript can be used to validate data
- JavaScript can be used to detect the visitor's browser
- JavaScript can be used to create cookies
 - Store and retrieve information on the visitor's computer

JAVASCRIPT HOW TO



- The HTML `<script>` tag is used to insert a JavaScript into an HTML page

```
<script type="text/javascript">
document.write("Hello World!")
</script>
```
- Ending statements with a semicolon?
 - Optional; required when you want to put multiple statements on a single line
- JavaScript can be inserted within the head, the body, or use external JavaScript file
- How to handle older browsers?

```
<script type="text/javascript">
<!--
document.write("Hello World!")
// -->
</script>
```

JAVASCRIPT WHERE TO



- You can include JavaScripts in head, body, or simply use external JavaScript file (.js)
- JavaScripts in the body section will be executed while the page loads
- JavaScripts in the head section will be executed when called

Placing scripts at the bottom of the <body> element improves the display speed, because script interpretation slows down the display.

JAVASCRIPT BASICS



- Variables
- If ... Else
- Switch
- Operators
- Popup Boxes
- Functions
- Loops (for, while)
- Events
- Try ... Catch
- Throw
- onerror
- Special Text
- Guidelines

JAVA OBJECTS



- String
- Date
- Array
- Boolean
- Math
- RegExp
- HTML DOM

HTML DOM



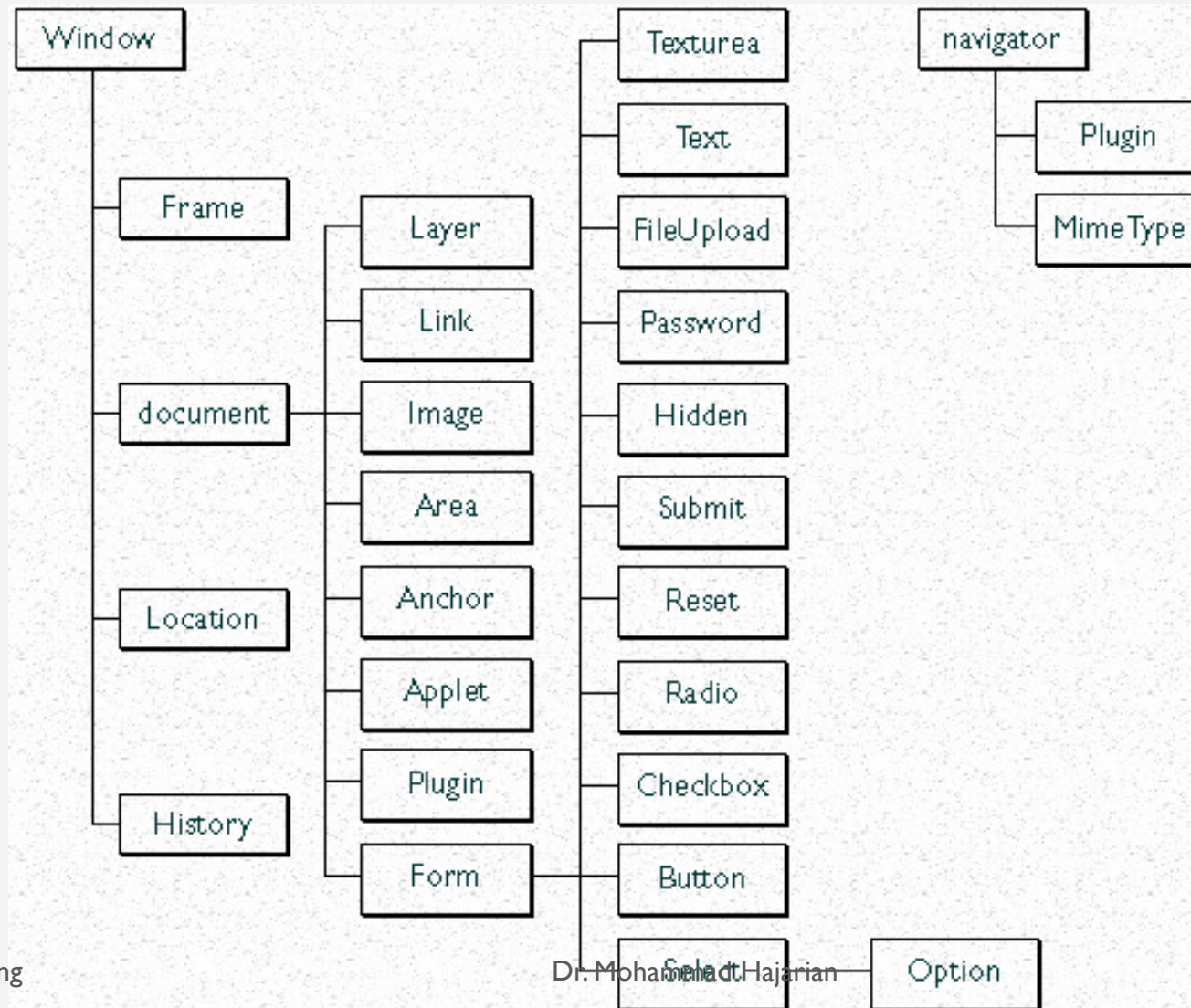
- What is the DOM?
 - It stands for **D**ocument **O**bject **M**odel
 - With JavaScript, we can restructure an entire HTML document by adding, removing, changing, or reordering items on a page
 - JavaScript gains access to all HTML elements through the DOM

USING JAVASCRIPT OBJECTS



- When you load a document in your web browser, it creates a number of JavaScript objects
- These objects exist in a hierarchy that reflects the structure of the HTML page

HTML DOM STRUCTURE





موسسه آموزش عالی غیردولتی غیرانتفاعی بصیرتک

PROGRAMMING CONCEPTS

VARIABLES



- A variable is a name associated with a piece of data
- Variables allow you to store and manipulate data in your programs
- Think of a variable as a mailbox which holds a specific piece of information

VARIABLES



- In JavaScript variables are created using the keyword `var`

- Example:

```
var x = 10;
```

```
var y = 17;
```

```
var color = "red";
```

```
var name = "Katie";
```

VARIABLES



- It is vitally important to distinguish between the *name* of the variable and the *value* of the variable
- For example, in the expression `var color="red"`, `color` is the name of the variable and `red` is the value. In other words, `color` is the name of the box while `red` is what is inside the box

DATA TYPES



- Primitive Data Types
 - Numbers
 - Strings
 - Boolean (True, False)
- Composite Data Types
 - Arrays
 - Objects

PRIMITIVE DATA TYPES



- **Numbers** - A number can be either an integer or a decimal
- **Strings** - A string is a sequence of letters or numbers enclosed in single or double quotes
- **Boolean** - True or False

VARIABLES & DATA TYPES



- JavaScript is *untyped*; It does not have explicit data types
- For instance, there is no way to specify that a particular variable represents an integer, string, or real number
- The same variable can have different data types in different contexts

IMPLICIT DATA TYPES



- Although JavaScript does not have explicit data types, it does have implicit data types
- If you have an expression which combines two numbers, it will evaluate to a number
- If you have an expression which combines a string and a number, it will evaluate to a string

EXAMPLE: VARIABLES



var x = 4;

Ans = x + y;

Ans => 15

var y = 11;

Ans = z + x;

Ans => cat4

var z = "cat";

Ans = x + q;

Ans => 417

var q = "17";

MORE EXAMPLES

var x = 4;

var y = 11;

var z = "cat";

var q = "17";

Ans = x + y + z;

Ans => 15cat

Ans = q + x + y;

Ans => 17411



ARRAYS



- An array is a compound data type that stores numbered pieces of data
- Each numbered datum is called an *element* of the array and the number assigned to it is called an *index*.
- The elements of an array may be of any type. A single array can even store elements of different type.

CREATING AN ARRAY



- There are several different ways to create an array in JavaScript
- Using the `Array()` constructor:
 - `var a = new Array(1, 2, 3, 4, 5);`
 - `var b = new Array(10);`
- Using array literals:
 - `var c = [1, 2, 3, 4, 5];`

ACCESSING ARRAY ELEMENTS



- Array elements are accessed using the [] operator
- Example:
 - var colors = [“red”, “green”, “blue”];
 - colors[0] => red
 - colors[1] => green

ADDING ELEMENTS



- To add a new element to an array, simply assign a value to it
- Example:

```
var a = new Array(10);  
a[50] = 17;
```

ARRAY LENGTH



- All arrays created in JavaScript have a special length property that specifies how many elements the array contains
- Example:
 - `var colors = ["red", "green", "blue"];`
 - `colors.length => 3`

PRIMITIVE DATA TYPES VERSUS COMPOSITE DATA TYPES



- Variables for primitive data types hold the actual value of the data
- Variables for composite types hold only references to the values of the composite type

VARIABLE NAMES



- JavaScript is **case sensitive**
- Variable names cannot contain spaces, punctuation, or start with a digit
- Variable names cannot be reserved words

PROGRAMMING TIPS



- It is bad practice to change the implicit type of a variable. If a variable is initialized as a number, it should always be used as a number.
- Choose meaningful variable names

STATEMENTS



- A statement is a section of JavaScript that can be evaluated by a Web browser
- A script is simply a collection of statements

Examples:

```
Last_name = "Dunn";
```

```
x = 10 ;
```

```
y = x*x ;
```

PROGRAMMING TIPS



- It is a good idea to end each program statement with a semi-colon; Although this is not necessary, it will prevent coding errors

- **Recommended:**

```
a = 3;
```

```
b = 4;
```

- **Acceptable:**

```
a = 3; b = 4;
```

- **Wrong:**

```
a =
```

```
3;
```


OPERATORS



+	Addition	= =	Equality
-	Subtraction	!=	Inequality
*	Multiplication	!	Logical NOT
/	Division	&&	Logical AND
%	Modulus		Logical OR
++	Increment	?	Conditional Selection
--	Decrement		

AGGREGATE ASSIGNMENTS



- Aggregate assignments provide a shortcut by combining the assignment operator with some other operation
- The += operator performs addition and assignment
- The expression $x = x + 7$ is equivalent to the expression $x += 7$

INCREMENT AND DECREMENT



- Both the increment (++) and decrement (- -) operator come in two forms: prefix and postfix
- These two forms yield different results

```
x = 10;      x = 10;  
y = ++ x;   z = x ++;
```

⇒ **y = 11**

⇒ **z = 10**

⇒ **x = 11 in both cases**

CONTROL STRUCTURES



- There are three basic types of control structures in JavaScript: the `if` statement, the `while` loop, and the `for` loop
- Each control structure manipulates a block of JavaScript expressions beginning with `{` and ending with `}`

THE IF STATEMENT



- The `if` statement allows JavaScript programmers to make a decision
- Use an `if` statement whenever you come to a “fork” in the program

```
if ( x == 10)
{
    y = x*x;
}
else
{
    x = 0;
}
```

REPEAT LOOPS



- A repeat loop is a group of statements that is repeated until a specified condition is met
- Repeat loops are very powerful programming tools; They allow for more efficient program design and are ideally suited for working with arrays

THE WHILE LOOP

- The while loop is used to execute a block of code while a certain **condition** is true

```
count = 0;
while (count <= 10) {
    document.write(count);
    count++;
}
```



THE FOR LOOP



- The for loop is used when there is a need to have a **counter** of some kind
- The counter is initialized before the loop starts, tested after each iteration to see if it is below a target value, and finally updated at the end of the loop

EXAMPLE: FOR LOOP



```
// Print the numbers 1 through  
10
```

```
for (i=1; i<= 10; i++)  
    document.write(i);
```

i=1 initializes the counter

i<=10 is the target
value

i++ updates the
counter at the end
of the loop

EXAMPLE: FOR LOOP



```
<SCRIPT
  LANGUAGE=
  "JavaScript">
document.write("1");
document.write("2");
document.write("3");
document.write("4");
document.write("5");
</SCRIPT>
```

```
<SCRIPT
  LANGUAGE=
  "JavaScript">

for (i=1; i<=5; i++)
  document.write(i);
```

FUNCTIONS



- Functions are a collection of JavaScript statement that performs a specified task
- Functions are used whenever it is necessary to repeat an operation

FUNCTIONS



- Functions have inputs and outputs
- The inputs are passed into the function and are known as **arguments** or **parameters**
- Think of a function as a “black box” which performs an operation

DEFINING FUNCTIONS



- The most common way to define a function is with the `function` statement.
- The function statement consists of the function keyword followed by the name of the function, a comma-separated list of parameter names in parentheses, and the statements which contain the body of the function enclosed in curly braces

EXAMPLE: FUNCTION



```
function square(x)
{return x*x;}
```

```
z = 3;
sqr_z = square(z);
```

Name of Function: square

Input/Argument: x

Output: $x*x$

EXAMPLE: FUNCTION



```
function sum_of_squares(num1,num2)
    {return (num1*num1) + (num2*num2);}
```

```
function sum_of_squares(num1,num2)
    {return (square(num1) + square(num2));}
```

FIND AN EXISTING ELEMENT



```
var inpVal;  
inpVal = document.getElementById("formID");
```

Display:

```
document.getElementById('formID').style.display = "block";  
document.getElementById('formID').style.display = "none";  
document.getElementById(formID').value= inpVal;
```


PRACTICE 4



تمرین چهارم - ۱ نمره

- برنامه ای بنویسید که ورودی فیلد تعداد محصول در وب سایت را بخواند و اگر کاربر عدد بزرگتر از ۱۰ وارد کرده بود به کاربر پیام دهد که قبول نیست، اگر زیر ۱۰ بود پیام بدهد که مورد قبول است و تعداد های منفی و بزرگتر از ۲۰ هم پیام بدهد که ورودی اشتباه است.
- با بکار بردن تابع برنامه ای بنویسید که قیمت دو جنس را از وب سایت بخواند، آنها را جمع ببیند و مجموع قیمت ها را در بخش قیمت نهایی نشان دهد. اگر قیمت محاسبه شده زیر ۱ میلیون تومان بود ۲۰ هزار تومان هزینه تحویل را به هزینه نهایی اضافه نماید. همچنین اگر قیمت نهایی محاسبه شده بیش از ۱ میلیون تومان بود عکسی را نمایش دهد که نشانگر تحویل رایگان و اگر زیر یک میلیون تومان بود عکسی را نمایش دهد که نشانگر تحویل پولی میباشد.

INSTRUCTIONS



برای هر تمرین مطابق فایل پرزنتیشن (دستور العمل) تکالیف خود را در قالب فایل **ZIP** به آدرس **MAINSYSTEM@GMAIL.COM** تا ۰.۹۰ نمره برای حل تمرین از این طریق بدست می آید. در جلسه حل تمرین آنها را در کلاس شرح دهید تا ۰.۱۰ نمره از این طریق بدست می آید.

- مهلت ارسال تا یک روز قبل از جلسه کلاس است
- ارسال با تاخیر و یا ارائه با تاخیر نمره ندارد
- از قراردادن مسائلی که تدریس نشده در پاسخ ها خود داری کنید (نمره منفی خواهد داشت).

Q/A

- End of Session 10



THANK YOU!