

**BOOTCAMP '20**

**SESIONI #14**

**JS**

# What will be covered?

- What is JavaScript?
- Where do we put JavaScript code?
- `document.write` vs. `console.log` vs. `alert`
- Code, Statements, and Expressions
- Operators
- Values & Types
- Code Comments
- Variables
- Blocks
- Conditionals



# What is JavaScript?

- JavaScript është gjuhë programuese e nivelit të lartë
- Është gjuhë e interpretueshme (interpretohet përgjatë ekzekutimit)
- Përkrah disa paradigma
  - imperative/procedural
  - OOP (Object-Oriented)
  - functional
- E influencuar nga disa gjuhë programuese: AWK, C, HyperTalk, Java, Lua, Perl, Python, Scheme, Self
- E krijuar në vitin 1995 nga Brendan Eich
- Përputhet me specifikimin [ECMAScript](#)



# Where do we put JavaScript code?

JavaScript mund të shkruhet brenda tagjeve `<script>` dhe `</script>` në vetë HTML dokumentin apo në dokument të posatshëm (që ka prapashtesën .js) dhe mandej të lidhet me HTML dokumentin duhe u specifikuar lokacioni i skedës përmes atributit `src` në tagun `<script>`.

# Where do we put JavaScript code?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    alert("Hello, world!");
  </script>
</body>
</html>
```

# Where do we put JavaScript code?

```
JS hello.js
```

```
1 alert("Hello, world!");
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <script type="text/javascript" src="hello.js"></script>
</body>
</html>
```

# document.write vs. console.log vs. alert

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    document.write("Hello, world!");

    console.log("Hello, world!");

    alert("Hello, world!");
  </script>
</body>
</html>
```

# Code, Statements, and Expressions

- Kodi - është set i instruksioneve të posatshme të cilat përdoren për të i treguar kompjuterit se cilën punë dhe si duhet ta kryej.
- Deklaratë - një grup i fjalëve, numrave, dhe operatorëve të cilat kryejnë ndonjë punë të caktuar.
  - Shembull: `a = 5;`
- Shprehje – përbëhen prej disa deklaratave.
  - Shembull: `a = b * 5;`



# Variables

Deklarimi i variablave (ndryshoreve) në JavaScript bëhet përmes çelësfjalës var, mandej pason një emër valid për ndryshoren.

```
var x;
```

## ME RËNDËSI

- JavaScript është Weak typing (apo dynamic typing) language
- Tipi (lloji) i variables përcaktohet përgjatë kohës të ekzekutimit

# Values & Types

- Vlerat janë të dhëna të cilat i ndahen variablave.
- Reprezentimet e ndryshme të vlerave quhen tipe (lloje) të të dhënave.
- Tipet bazike të të dhënave quhen tipe primitive (primitive types).
- Tipet primitive janë
  - String
  - Number
  - Bigint
  - Boolean
  - Undefined
  - Symbol

# Value assignment operator

- Operatori për ndarje të vlerës është simboli =
- Në anën e majtë të simbolit shkruhet emri i variables
- Në anën e djathtë shkruhet vlera që dëshirojmë të ia ndajmë asaj variable

```
<script type="text/javascript">  
    // Variable x ka vlerën 10  
    var x = 10;  
</script>
```

# Operators

- Operatorët aritmetik (+, -, \*, /, %, ++, --)
- Operatorët për krahasim (<, >, <=, >=, ==, !=)
- Operatorët logjik (&&, ||, !)
- Operatorët për bita (&, |, ^, ~, <<, >>, >>>)
- Operatorët për ndarje të vlerës (=, +=, -=, \*=, /=, %=)
- Operatori kusht                   ?:
- Operatori typeof               typeof

# Operatorët aritmetik

Sr.No.	Operator & Description
1	<b>+ (Addition)</b> Adds two operands <b>Ex:</b> A + B will give 30
2	<b>- (Subtraction)</b> Subtracts the second operand from the first <b>Ex:</b> A - B will give -10
3	<b>* (Multiplication)</b> Multiply both operands <b>Ex:</b> A * B will give 200
4	<b>/ (Division)</b> Divide the numerator by the denominator <b>Ex:</b> B / A will give 2
5	<b>% (Modulus)</b> Outputs the remainder of an integer division <b>Ex:</b> B % A will give 0
6	<b>++ (Increment)</b> Increases an integer value by one <b>Ex:</b> A++ will give 11
7	<b>-- (Decrement)</b> Decreases an integer value by one <b>Ex:</b> A-- will give 9

# Operatorët për krahasim

Sr.No.	Operator & Description
1	<b>== (Equal)</b> Checks if the value of two operands are equal or not, if yes, then the condition becomes true. <b>Ex:</b> (A == B) is not true.
2	<b>!= (Not Equal)</b> Checks if the value of two operands are equal or not, if the values are not equal, then the condition becomes true. <b>Ex:</b> (A != B) is true.
3	<b>&gt; (Greater than)</b> Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true. <b>Ex:</b> (A > B) is not true.
4	<b>&lt; (Less than)</b> Checks if the value of the left operand is less than the value of the right operand, if yes, then the condition becomes true. <b>Ex:</b> (A < B) is true.
5	<b>&gt;= (Greater than or Equal to)</b> Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true. <b>Ex:</b> (A >= B) is not true.
6	<b>&lt;= (Less than or Equal to)</b> Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true. <b>Ex:</b> (A <= B) is true.

# Operatorët logjik

Sr.No.	Operator & Description
1	<b>&amp;&amp; (Logical AND)</b> If both the operands are non-zero, then the condition becomes true. <b>Ex:</b> (A && B) is true.
2	<b>   (Logical OR)</b> If any of the two operands are non-zero, then the condition becomes true. <b>Ex:</b> (A    B) is true.
3	<b>! (Logical NOT)</b> Reverses the logical state of its operand. If a condition is true, then the Logical NOT operator will make it false. <b>Ex:</b> ! (A && B) is false.

# Operatorët për bita

Sr.No.	Operator & Description
1	<b>&amp; (Bitwise AND)</b> It performs a Boolean AND operation on each bit of its integer arguments. <b>Ex:</b> (A & B) is 2.
2	<b>  (BitWise OR)</b> It performs a Boolean OR operation on each bit of its integer arguments. <b>Ex:</b> (A   B) is 3.
3	<b>^ (Bitwise XOR)</b> It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both. <b>Ex:</b> (A ^ B) is 1.
4	<b>~ (Bitwise Not)</b> It is a unary operator and operates by reversing all the bits in the operand. <b>Ex:</b> (~B) is -4.
5	<b>&lt;&lt; (Left Shift)</b> It moves all the bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros. Shifting a value left by one position is equivalent to multiplying it by 2, shifting two positions is equivalent to multiplying by 4, and so on. <b>Ex:</b> (A << 1) is 4.
6	<b>&gt;&gt; (Right Shift)</b> Binary Right Shift Operator. The left operand's value is moved right by the number of bits specified by the right operand. <b>Ex:</b> (A >> 1) is 1.
7	<b>&gt;&gt;&gt; (Right shift with Zero)</b> This operator is just like the >> operator, except that the bits shifted in on the left are always zero. <b>Ex:</b> (A >>> 1) is 1.



# Operatorët për ndarje të vlerës

Sr.No.	Operator & Description
1	<b>+</b> (Addition) Adds two operands <b>Ex:</b> A + B will give 30
2	<b>-</b> (Subtraction) Subtracts the second operand from the first <b>Ex:</b> A - B will give -10
3	<b>*</b> (Multiplication) Multiply both operands <b>Ex:</b> A * B will give 200
4	<b>/</b> (Division) Divide the numerator by the denominator <b>Ex:</b> B / A will give 2
5	<b>%</b> (Modulus) Outputs the remainder of an integer division <b>Ex:</b> B % A will give 0
6	<b>++</b> (Increment) Increases an integer value by one <b>Ex:</b> A++ will give 11
7	<b>--</b> (Decrement) Decreases an integer value by one <b>Ex:</b> A-- will give 9

# Operatori kusht

Sr.No.	Operator and Description
1	<b>? : (Conditional )</b> If Condition is true? Then value X : Otherwise value Y

# Operator typeof

The **typeof** operator is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

The *typeof* operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

Here is a list of the return values for the **typeof** Operator.

Type	String Returned by typeof
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"
Function	"function"
Undefined	"undefined"
Null	"object"

# Code Comments

- Koment në një rresht

```
<script type="text/javascript">  
    // This function will print "Hello, world!"  
    alert("Hello, world!");  
</script>
```

- Koment në shumë rreshta

```
<script type="text/javascript">  
    /* This function  
       will print  
       "Hello, world!"  
    */  
    alert("Hello, world!");  
</script>
```

# Blocks

Blok i kodit llogaritet kodi që gjendet brenda kllapave gjarpërore

```
<script type="text/javascript">  
  var x = 10;  
  if(x > 5)  
  {  
    // bllok i kodit  
    var y = 20;  
    x += y;  
  }  
</script>
```

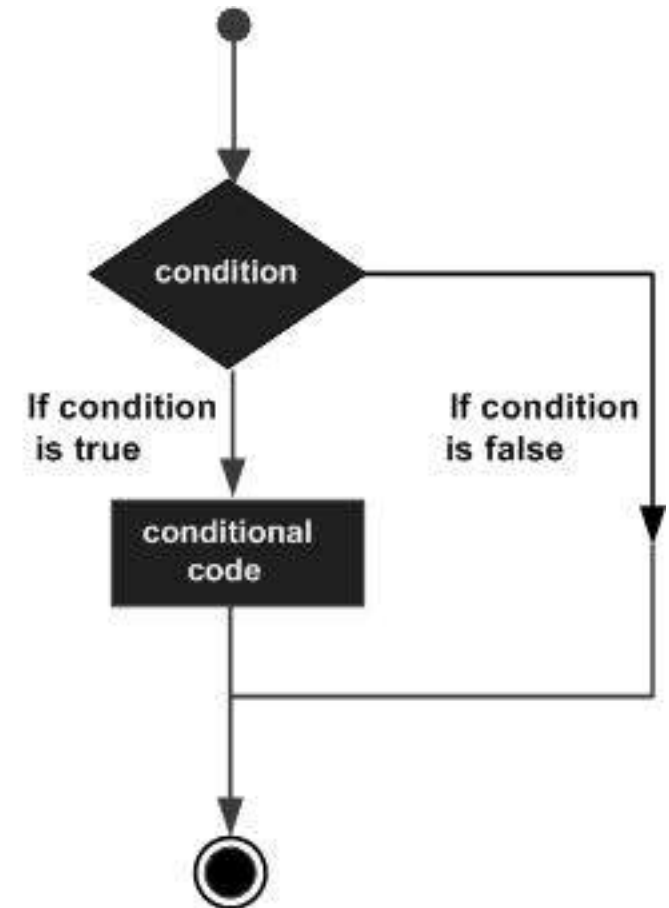
# Conditionals

- If-else

```
if (expression) {  
    Statement(s) to be executed if expression is true  
}
```

```
if (expression) {  
    Statement(s) to be executed if expression is true  
} else {  
    Statement(s) to be executed if expression is false  
}
```

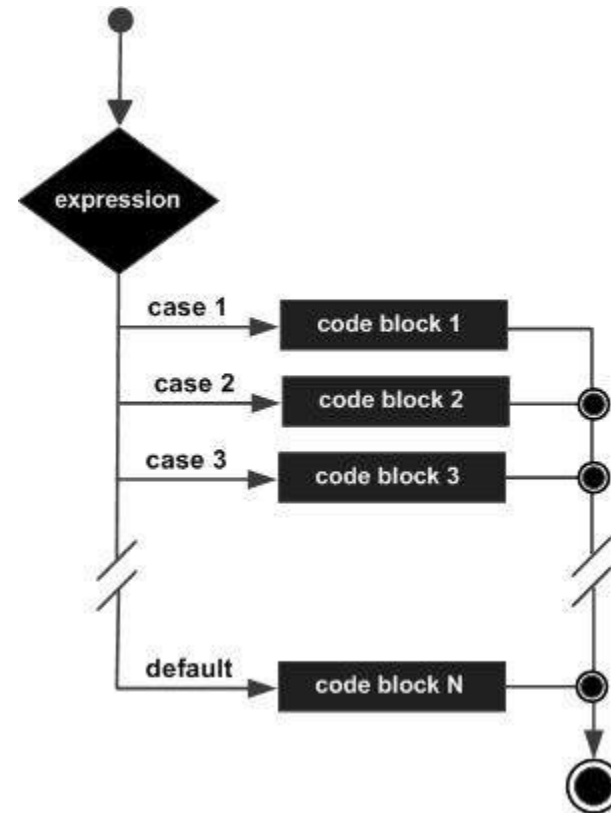
```
if (expression 1) {  
    Statement(s) to be executed if expression 1 is true  
} else if (expression 2) {  
    Statement(s) to be executed if expression 2 is true  
} else if (expression 3) {  
    Statement(s) to be executed if expression 3 is true  
} else {  
    Statement(s) to be executed if no expression is true  
}
```



# Conditionals

- Switch

```
switch (expression) {  
  case condition 1: statement(s)  
  break;  
  
  case condition 2: statement(s)  
  break;  
  ...  
  
  case condition n: statement(s)  
  break;  
  
  default: statement(s)  
}
```



# QUESTIONS

