# JavaScript

# JS ( JavaScript ) is Case Sensitive

- All JS identifiers are **case sensitive**.
- Variables  age and Age, are two different variables.

# JS Character Set

- JS uses the **Unicode** character set.
- Unicode covers almost all the characters, punctuations, and symbols in the world.

# JS and Camel Case

- Developer have used different ways of joining multiple words into one name…

  first-name, last-name

  first_name, last_name

  FirstName, LastName

  firstName, lastName

# JavaScript Comments

- Comments can be used to explain JS code.

- Single Line Comments

  Single line comments start with //

- Multi-line Comments

  Multi-line comments start with /* and end with */.

# JS Variables

JavaScript variables are containers for storing data values.

Example :

```
var  i = 15;
var  j = 10;
```

# Example

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

<script>
var i = 15;
var j = 10;
var total = i + j;

document.getElementById("demo").innerHTML = "Total is " + total;
</script>
```

# JS Identifiers

- All JavaScript **variables** must be **identified** with **unique names**.
- These unique names are called **identifiers**.

# Rules for Declaring variables

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter.
- Names can also begin with $ and _ .
- Names are case sensitive (i and I are different variables).
- Reserved words cannot be used as variable name.

# Assignment Operator

- In JS (JavaScript), the equal sign (=) is an "assignment" operator, not an "equal to" operator.

- It is use to assign value to variables

```
var  i = 50;
```

# One Statement Code

- You can declare many variables in one single line.

```
var  user = “Felix ",  city= “Pune",  number = 987654;
```

or

```
var  user = “Felix ",
       city= “Pune",
       number = 987654;
```

# JavaScript Operators

- Operators is a symbol that used to perform some operation.

# Arithmetic Operators

| Operator | Description |
|---|---|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| ** | Exponentiation |
| / | Division |
| % | Modulus (Division Remainder) |
| ++ | Increment |
| -- | Decrement |

# JavaScript Assignment Operators

| Operator | Example | Same As |
|---|---|---|
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| *= | x *= y | x = x * y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |
| **= | x **= y | x = x ** y |

# Example

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

<script>
    var i = 5;
    var j = 2;
    var value = i * j;
    document.getElementById("demo").innerHTML = "Value is " + value;
</script>
```

# Output

```
Value is 10
```

# Example

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

<script>
    var x = 10;
    x += 5;
    document.getElementById("demo").innerHTML = x;
</script>
```

# Output

```
15
```

# JavaScript Comparison Operators

| Operator | Description |
|---|---|
| == | equal to |
| === | equal value and equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| ? | ternary operator |

# JavaScript Logical Operators

| Operator | Description |
| --- | --- |
| && | logical and |
| \|\| | logical or |
| ! | logical not |

# Example

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

<script>
    var x = 10;
    x++;
    var z = x;
    document.getElementById("demo").innerHTML = z;
</script>
```

# Output

```
11
```

# JS Data Types

JS variables can hold many **data types** like

numbers,

strings,

objects and more…

# Data Types in JS

```
<script>
    var i = 16;                                          // Number
    var name = "Felix";                                  // String
    var obj = {firstName: "John", lastName: "Doe"};      // Object
</script>
```

# Example

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

<script>
    var first = "Felix";
    var second = 'ITs';

    document.getElementById("demo").innerHTML =
            first + "<br>" +
            second;
</script>
```

# Output

```
Felix
ITs
```

# JavaScript Arrays

- JavaScript arrays are written with square brackets.
- Array items are separated by commas.

# Example

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

<script>
    var names = ["John", "Doe", "Andy"];

    document.getElementById("demo").innerHTML = names[1];
</script>
```

# Output

```
Doe
```

# JavaScript Objects

- JavaScript objects are written with curly braces {}.
- Object properties are written as name:value pairs, separated by commas.

# Example

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

<script>
    var person = {
        firstName: "John",
        lastName: "Doe",
        age: 50,
        eyeColor: "blue"
    };

    document.getElementById("demo").innerHTML =
            person.firstName + " is " + person.age + " years old.";
</script>
```

# Output

```
John is 50 years old.
```

# typeof Operator

You can use the JS  typeof operator to find the type of a JS variable.

# Example

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

 <script>

    var i = 10;
    var name = "Felix";
    var j = false;

    document.getElementById("demo").innerHTML =
          typeof i + "<br>" +
          typeof name + "<br>" +
          typeof j;
 </script>
```

# Output

```
number
string
boolean
```

**********