

BASICS OF

JS



PYTHONISTA PLANET

JavaScript is originally a client-side scripting language used to design responsive web pages. But now, it has evolved a lot and can do a lot of things.

These are some of the JS frameworks used widely nowadays:

- Node.js - An open source server environment that can handle everything on the server-side.
- React.js - React.js is a JavaScript library for building user interfaces.
- Angular.js - An open-source front-end web framework.
- Vue.js - A front-end framework for building user interfaces and single-page applications.
- Electron.js - To build desktop applications
- React.VR - To build VR websites
- Tensorflow.js - For machine learning

JS is a language that conforms to the ECMAScript specification. ECMAScript is a scripting-language specification that standardizes JS to help foster multiple independent implementations.

JavaScript in HTML

There are two different ways to embed JS in an HTML document: implicitly and explicitly. In explicit embedding, the JS code physically resides in the HTML document. This approach has many disadvantages.

Hence, JS is commonly placed in its own file, separate from the HTML document. This approach is called implicit embedding.

Link to the file from the HTML code using the `<script>` tag.

For example: `<script src="js/script.js"> </script>`

Variables

Variables are created using the **var** keyword. The preferred variable naming convention is camelCasing. Use semicolons at the end of every statement.

eg: `var firstName = "Ashwin";
var lastName = "Joy";
var age = 23;`

Comments

Single line comments can be created using two slashes.

eg: `//This is a comment`

Multi-line comments can be created using `/*` and `*/`

eg: `/* This is a
multi-line comment */`

Primitives

JavaScript has 5 primitive data types:

- Number
- String
- Boolean
- Undefined
- Null

All numeric literals are values of the type Number.

A string literal is a sequence of zero or more characters delimited by either single quotes (`'`) or double quotes (`"`).

The only values of type Boolean are true and false. These values are usually computed as the result of evaluating a relational or boolean expression.

The only value of type Undefined is 'undefined'. If the value of an undefined variable is displayed, the value 'undefined' will be printed.

The only value of the type Null is 'null', which indicates no value. A variable is null if it has not been explicitly declared or assigned a value.

Operations and Expressions

JavaScript has the typical collection of numeric operators.

Binary operators:

- + (add)
- - (subtract)
- * (multiply)
- / (divide)
- % (modulus)

Unary operators:

- + (plus)
- - (negate)
- ++ (increment)
- -- (decrement)

Assignment Operators

The statement `a+=8` means `a= a+8`

String concatenation

String concatenation is done using the operator denoted by a + sign.

typeof

We can find the type of a variable using the typeof function.

```
eg: var count = 5;  
     typeof(count) //outputs 'Number'
```

String Properties

```
var name = "Ashwin";
```

```
name.length //outputs 6  
name.charAt(2) //outputs h  
name.indexOf('h') //outputs 2  
name.substring(2,4) // outputs 'hwi'  
name.toLowerCase() //outputs 'ashwin'  
name.toUpperCase() // outputs 'ASHWIN'
```

Object Orientation in JavaScript

JavaScript is not an object-oriented programming (OOP) language. Rather, it is an object-based language.

JS doesn't have classes. Its objects serve both as objects and models of objects. JS does not have class-based inheritance and polymorphism, but it supports prototype-based inheritance.

Despite the fact that JS is not object-oriented, much of its design is rooted in the concepts and approaches used in OOP.

JS Objects

In JS. objects are collections of properties. Each property is either a data or a method or a function.

eg:

```
var website = {
  firstName: "Pythonista",
  secondName: "Planet",
  fullname: function(){
    return this.firstName+" "+
           this.secondName;
  }
}
console.log(website)

//outputs "Pythonista Planet"
```

```
/* The console.log method is used to print any messages  
or variables to the user. */
```

A JS object appears as a list of property-value pairs. The properties are names. The values are data values or functions. All functions are objects and are referenced through variables.

Date Object

There are occasions when information about the current date and time is useful in a program. JS has a *Date* object with rich collection of methods to use in such cases.

```
var checkingDate = new Date();
```

```
checkingDate.getDate(); //gives the day of the month  
checkingDate.getMonth(); //gives the current month  
checkingDate.getDay(); //gives the day of the week  
checkingDate.getFullYear(); //gives the current year
```

Type Conversions

We can convert one data type into another using implicit or explicit methods.

Implicit Conversion (Coercion)

When a value of one type is used in a position that requires a value of a different type, JS attempts to convert the value to the type that is required.

eg: "August" + 1997 gives "August 1997"
7 * "3" gives 21

Explicit Conversion

There are some methods in JS to explicitly convert one data type into another.

eg:

```
var num = 6;  
num.toString(); //gives "6"
```

Screen Output and Keyboard Input

Document object and Window object are two important objects in JS. JavaScript models HTML document with the Document object. The window in which the browser displays an HTML document is modeled with the Window object.

The Document object has many properties and methods. One of them is **write**, which is used to create script output, which is dynamically created HTML content.

eg: `document.write("The result is ", result, "
");`

Window includes 3 methods that create dialog boxes for 3 specific kinds of user interactions.

- alert
- confirm
- prompt

The **alert** method opens a dialog window and displays it's parameters in that window. It also displays an OK button.

eg: `alert("This is an alert message");`

The **confirm** method opens a dialog window in which the method displays it's string string parameter, along with two buttons, which are **OK** and **Cancel**. Confirm method return true for Ok and false for Cancel.

eg: `var question = confirm("Do you wanna continue downloading this e-book?");`

The prompt method creates a dialog window that contains a text box, which is used to collect a string of input from the user. The window also includes an OK and Cancel button.

eg: `name = prompt("What is your name?", " ");`

The prompt method takes 2 parameters: the string that prompts the user for input, and a default string in case the user doesn't type a string. In many cases, an empty string is used for the default input.

The alert, prompt, and confirm methods cause the browser to wait for a user response.



This JS basics guide is a resource from Pythonista Planet.

For more helpful tutorials on programming, go to <https://pythonistaplanet.com>