# JavaScript Notes

Name: Yutika Prabhu

The Web's scripting language is JavaScript.

Millions of Web pages utilize JavaScript to enhance functionality, check forms, identify browsers, and do much more. It's used for design, validate forms, identify browsers, create cookies, and do much more.

The most widely used programming language on the Internet, JavaScript, is compatible with all the main browsers, including Internet Explorer, Mozilla Firefox, and Opera.

JavaScript was created to add interactivity to HTML pages JavaScript is a scripting language A scripting language is a simple form of programming JavaScript is typically embedded directly into HTML pages JavaScript is an interpreted language This means that scripts run immediately without first being compiled Anyone can use JavaScript without having to pay a license

Java and JavaScript are conceptually and aesthetically entirely distinct languages.

Java, created by Sun Microsystems, is a potent and significantly more complicated programming language, comparable to C and C++.

HTML authors are typically not programmers, but JavaScript is a scripting language with a relatively basic syntax, giving HTML designers a programming tool! Most people can add little "snippets" of code to their HTML websites.

JavaScript can insert dynamic text into HTML pages. For example:

JavaScript can react to events, such as when a page has finished loading or when a user clicks on an HTML element. JavaScript can read and write HTML elements. A JavaScript can read and change the content of an HTML element.

document.write(" " + name + " ") can write a variable text into an HTML page.

## Variables in JavaScript

The "containers" for storing information are variables.

Variables in JavaScript can store values or expressions.

A variable can be identified by a short name, such as x, or a longer name, such as carname.

Rules for variable naming in JavaScript:

Variable names must start with a letter or the underscore character. Variable names are case sensitive (y and Y are two separate variables).

Note that variable names are case-sensitive in JavaScript due to the language's case-sensitivity.

## Datatypes in JavaScript

Numbers are quantifiable values that may be analyzed and computed. They are not enclosed in quotation marks. There are two possible numbers: positive and negative.

A string is a group of characters that includes both letters and digits. JavaScript does not process the string; it simply uses it as is. Strings are used for text that needs to be displayed or for values that need to be transmitted.

Boolean (true/false) evaluations allow you to determine whether a condition satisfies or does not satisfy predetermined criteria.

An empty value is known as null. 0 is a real, calculable number, but null is the absence of any value. They are not the same thing.

## Operators in JavaScript

To assign values, use the operator =.

Value addition is done with the + operator.

To give JavaScript variables values, use the assignment operator =.

To add two values together, use the arithmetic operator +.

### JavaScript Arithmetic Operators

| Operator | Description | Example | Result |
|---|---|---|---|
| + | Addition | x=y+2 | x=7 |
| - | Subtraction | x=y-2 | x=3 |
| * | Multiplication | x=y*2 | x=10 |
| / | Division | x=y/2 | x=2.5 |
| % | Modulus (division remainder) | x=y%2 | x=1 |
| ++ | Increment | x=++y | x=6 |
| -- | Decrement | x=--y | x=4 |

**JavaScript Assignment Operators**

| Operator | Example | Same As | Result |
|---|---|---|---|
| = | x=y | | x=5 |
| += | x+=y | x=x+y | x=15 |
| -= | x-=y | x=x-y | x=5 |
| *= | x*=y | x=x*y | x=50 |
| /= | x/=y | x=x/y | x=2 |
| %= | x%=y | x=x%y | x=0 |

**Comparison Operators**

| Operator | Description | Example |
|---|---|---|
| == | is equal to | x==8 is false |
| === | is exactly equal to (value and type) | x===5 is true<br>x==="5" is false |
| != | is not equal | x!=8 is true |
| > | is greater than | x>8 is false |
| < | is less than | x<8 is true |
| >= | is greater than or equal to | x>=8 is false |
| <= | is less than or equal to | x<=8 is true |

**Logical Operators**

| Operator | Description | Example |
|---|---|---|
| && | and | (x < 10 && y > 1) is true |
\|\| | or | (x==5 \|\| y==5) is false |
| ! | not | !(x==y) is true |

**Conditional Operator**

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax: variablename=(condition)?value1:value2

## Conditional Statements

When writing code, you want to take different actions depending on the decision you make rather frequently. To accomplish this, you can utilize conditional statements in your code.

We have the following conditional statements in JavaScript:

If you want to only run certain code if a certain condition is true, use the if statement.

If you want to execute some code if a condition is true and another block of code if it is false, use an if...else statement. If you want to choose one of multiple blocks of code to be executed, use an if....else statement.

Use the switch statement to choose which of several blocks of code will be run.

## Looping Statements in JavaScript

JavaScript uses loops to repeat a block of code a certain number of times or only when a predetermined condition is true.

When writing code, you frequently want the same block of code to execute repeatedly in a row.

We can utilize loops to carry out a task like this rather than inserting several almost identical lines into a script.

There are two types of loops in JavaScript:

for: it loops through a code block a certain number of times.

while: if a condition is true, a block of code is looped through.

## JavaScript Break and Continue

There are two special statements that can be used inside loops: break and continue.

break: The break command will break the loop and continue executing the code that follows the loop.

continue: The continue command will break the current loop and continue with the next value.

## JavaScript Functions

A self-contained section of code that carries out a certain "function" is called a function (also known as a method). A function can be identified by its format, which consists of a section of descriptive text followed by open and close brackets. A function is a piece of reusable code that will run in response to an event or when it is invoked.

You can embed your script in a function to prevent the browser from running it when the page first loads.

A function includes code that will be run in response to an event or when the function is called.

Anywhere on the page—or, if the function is included in an external.js file, even on other pages—can call it.

Both the head and the body sections of a document can contain definitions of functions. However, it could be a good idea to put the function in the section to ensure that the browser reads/loads it before it is invoked.

**return Statement**

The return statement is used to specify the value that is returned from the function.

So, functions that are going to return a value must use the return statement.

## Event Handlers

JavaScript methods, or functions of objects, called event handlers let us, as JavaScript programmers, decide what occurs when certain events take place.

An Event is always the outcome of something a user performs, whether directly or indirectly. For instance, event handlers that react to mouse actions, such as onClick and onMouseOver, have already been demonstrated. An internal change-of-state to the page (the completion of loading or leaving the page) is another form of Event. You may think of a onLoad Event as the indirect outcome of a user action.

Even though we frequently use the terms interchangeably, it's crucial to understand the difference between events and event handlers. A simple event is just something that occurs and is started by an event handler (onClick, onMouseOver, etc.).

The objects that can cause events on a website are referred to as "targets" or "target elements," and a button that initiates a Click event is a target element for this event. Event Handlers, which are small pieces of script that instruct the browser what to do when a specific event occurs at a specific target, are typically used to define events. These Event Handlers are frequently expressed as HTML tag properties of the target element.

## Arrays in JavaScript

A script's database-like structure is built using an array object. Data points (array items) can be more easily accessed and used in scripts by being grouped together. Although there are ways to access actual databases (which are outside the purview of this series), we're only concerned with modest bits of data here.

An array can be thought of as a spreadsheet's column of data. The array's name would be the same as the column's name. In the array, each piece of information (or element) is identified by a number (or index), much like a row number in a column.

An object is an array. As I mentioned before, an object is an entity made up of a set of properties (in this example, an array of items).

The same naming conventions that apply to variables, functions, and objects also apply to arrays. Keep in mind these fundamental guidelines: There can be no numbers in the initial character, no reserved words, and no spaces.

Additionally, keep in mind that the name of the array object, such as Array, is capitalized.

The array's collection of elements, or the data, is accessed by the JavaScript interpreter using numerical values. Like an ID number, each index number (because it is the number of the data in the array's index) designates a particular piece of data in the array. It's crucial to keep in mind that the data's index numbering begins at "0." As a result, if there are 8 elements, the first element will be numbered "0" and the last element "7."

## JavaScript Objects

### Array Object

The Array object is used to store multiple values in a single variable.

### Date Object

The Date object is used to work with dates and times.

### Math Object

The Math object allows you to perform mathematical tasks.

The Math object includes several mathematical constants and methods.

**String object**

The String object is used to manipulate a stored piece of text.

**Window Object**

The Window object is the top-level object in the JavaScript hierarchy.

The Window object represents a browser window.

A Window object is created automatically with every instance of a <body> or <frameset> tag.

**Document Object**

The Document object represents the entire HTML document and can be used to access all elements in a page. It is part of the Window object and is accessed through the window.document property.

**History Object**

The History object is a JavaScript object, not an HTML DOM object.

The History object is automatically created by the JavaScript runtime engine and consists of an array of URLs. These URLs are the URLs the user has visited within a browser window.

The History object is part of the Window object and is accessed through the window.history property.

**Form Object**

The Form object represents an HTML form.

For each instance of a <form> tag in an HTML document, a Form object is created.

**Image Object**

The Image object represents an embedded image.

For each instance of an <img> tag in an HTML document, an Image object is created.

**Area Object**

The Area object represents an area of an image-map (An image-map is an image with clickable regions).

For each instance of an <area> tag in an HTML document, an area object is created.

**Navigator Object**

The Navigator object is a JavaScript object, not an HTML DOM object.

The Navigator object is automatically created by the JavaScript runtime engine and contains information about the client browser.