

# "JavaScript"

Definition of JavaScript:

JavaScript is a scripting language that makes web pages interactive. It allows developers to add dynamic behavior to websites, such as animations, user input validation, and fetching data. JavaScript enables client-side scripting and it is also widely used in server-side e.g (Node.js) mobile app development and gaming development.

History of JavaScript:

Creation (1995):

JavaScript was created in "1995" by Brendan Eich while he was working at Netscape Communications Corporation.

It was originally called

Date: \_\_\_ / \_\_\_ / 20\_\_\_

"Mocha" and then renamed

"LiveScript" before being

officially named "JavaScript".

JavaScript was developed

in just "10 days" as a

lightweight scripting language

for the Netscape Navigator

Web browser.

Difference b/w Java/JavaScript:

- Java is a compiled language typically used for building complex applications.
- While JavaScript is an interpreted scripting language primarily used for web development adding interactivity and dynamic behavior to website.

What is Node.js?

Node.js is a Javascript runtime environment that allows developers to execute Javascript

code outside of web-browser..

Creation:

It was created by Ryan Dahl in 2009, It enables server-side programming with JavaScript, making it possible to build web servers backend systems and command-line tools.

Difference b/w Server-side and Client-side

Client-side Javascript refers to code that user interface handles events, and manipulates the DOM (document object model).

Server-Side Javascript using Node.js runs on a server and handles tasks such as interacting with databases, managing user authentication, and serving web pages.

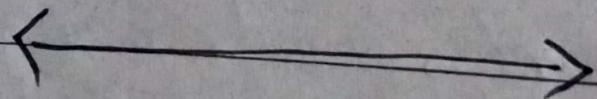
Date: 1 / 20

M T W

## Browser Engines :

Browser engine are software components responsible for rendering web pages. They interpret HTML, CSS and JavaScript code to display website correctly.

Popular browser Engine include chrome's V8, Firefox's Spider Monkey, and Safari's Javascript core.



## Chapter No. 1 "Alerts"

The `alert()` function in Javascript is used to display a pop-up message to users. It is commonly used for displaying informational message or warnings.

Say Syntax:

```
alert("This is my name.");
```

`alert` is a keyword that has special meaning for Javascript.

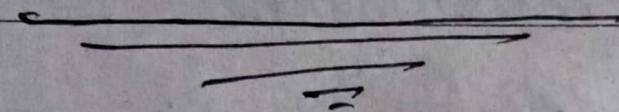
- The `alert` was not capitalized If you capitalize the script will be stop.
- The parentheses are a special requirements of Javascript.
- The quoted text is called a text string or simply string.

Date: \_\_\_ / \_\_\_ / 20\_\_\_

M T W

• Carefully writes ends every declarative sentence with a period, and ends every statement with a semicolon.

alert("Hello, World");



## Chapter No. 2 "Variables for Strings"

What is variables?

A variable is a named storage for data

In JavaScript, variables can hold different type of values including strings.

What is String?

A string is a sequence of characters

enclosed in quotes. It can contain numbers, letters, spaces and special characters.

A variables is created when you write var (for variable)

followed by the that you choose it to give it.

It takes a particular value when you assign the

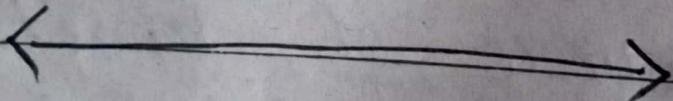
Date: \_\_\_ / \_\_\_ / 20

M T W T

value to it. This is a  
JavaScript statement that creates  
the variables name and  
assigns the value "A Hello" to  
it.

var name = "Hello";

You can use var, let or  
const, to declare variable  
for strings.



# Chapter No. 3

## Variable for Numbers -

In JavaScript numbers are one of the basic datatypes and they represent both integers (whole) numbers and floating-point (decimal) numbers. The way we declare and use variables for numbers in JavaScript is influenced by the concept of data-types, variable-scope, and memory-management.

In JavaScript variable are used to store values.

You can declare variables for numbers using . var, let const.

But then since it's not enclosed in "quotes."

Date: / / 20

If you enclosed a number in quotation marks. It's string. JavaScript can't do addition on it.

e.g. var num = "1";

var num1 = 2;

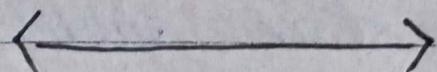
var num2 = 3;

~~console.log~~

var num3 = 2 + 3;

out-put :-

5



Date: 1/20

# Chapter No. 4

## Variable Names legal and illegal

You have already learned three rules about naming. You can't enclose it in quotation marks. The name can't be a number or start with a number.

- A variable name cannot contain any space.
- A variable name can contain only letters, numbers, dollar signs and underscores.
- e.g userAlert, myVar legal.
- Variable names are case sensitive
- I teach a camelCase naming convention.  
e.g: \$userName;
- Make your variable names

M T W

Date: 1 / 20

descriptive.

"e.g favBreed is better  
than favBrd":



# Chapter No. 5

## Math Expressions:

### Familiar Operators:

When you want use a number, you can use a math expressions.

For example you familiar with this kind of statement.

```
var popularNumber = 2+2;
```

This is displays the message "4" in an alert box-

When it sees a math expression, JavaScript always does the the math and delivers the result.

Here's a statement that subtract 24 from 12.

```
var num = 12 - 24;
```

This is display the message "12" in the alert box.

You can also use nothing but variables.

Var num = 2 ;

Var num2 = 3 ;

Var num3 = num + num2 ;

The arithmetic operators

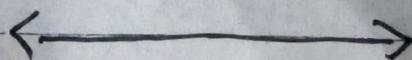
I've been using, +, -, \* x, /.

% is the modulus operators.

It doesn't give you the result of dividing one number by another - It gives you the remainder when the division is executed.

If one number divides evenly into another, the modulus operation return 0.  
in the following statement -

Var num = 9 % 3 ;



# Chapter No. 6

## Math Expressions:

### Unfamiliar Operators:

`num++;`

`num--;`

You can use these expressions in an assignment. This is a short way of writing:

`var num = 1;`

`var numNew = num++;`

The original value of "num" is assigned "newnum" and "num" is incremented afterward. If "num" is originally assigned 1.

In the first statement, the second statement boots its value to "2".

If place the plus before the variable, you get a different result.

`var = 1`

`var newNum = +num`

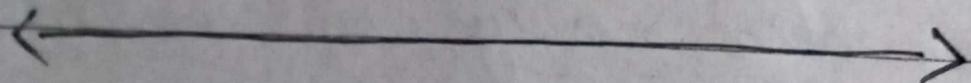
Date: 1 / 20

[M] [T] [W] [T]

In this statement both num and newNum windup with a value of 2.

Note: You decrement using minuses instead of pluses.

num-- ;



Date: \_\_\_\_\_

1 / 20 -

## Chapter No. 7

### Math Expressions

#### Eliminating ambiguity:

Complex arithmetic expressions can pose a problem, one that you may remember from high school algebra.

In JavaScript as in algebra, the ambiguity is cleared up by precedence rules. As in algebra the rule that applies here is that multiplications operation are completed before addition operators.

You can finesse the issue by using parentheses to eliminate ambiguity.

Parens override all the built-in precedence rules.

Date: / / 20

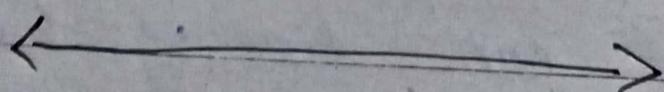
M T W T

They force JavaScript's  
~~complex precedence rules to~~  
complete operations enclosed  
by parens before completing  
any other operations.

In this statement the  
parentheses tell JavaScript  
to first multiply -3 by 4,  
then add 1.

Var totalCost = 1 + (3 \* 4);

By placing the first  
multiplication operation inside  
parentheses You've told  
JavaScript to do that operation  
first.



## Chapter No. 8 Concatenating text Strings:

Suppose you wanted to personalize a message. In another part of your code you've asked the user for her name and assigned the name that he entered to a variable.

When the user provided her name, we assigned it to the variable.

Using the plus operator the code combines-concatenates-these elements into the message.

You can concatenate any combination of strings and variables or all strings or all variables.

Date: \_\_\_ / \_\_\_ / 20

For example:

```
var message = "Thanks";
```

```
var bangs = "!";
```

```
var alert(message + userName + bangs)
```

If you put numbers in quotes, JavaScript concatenate them as strings rather than adding them.

For example:

```
alert("2" + "2");
```

Javascript automatically converts the number to strings.

