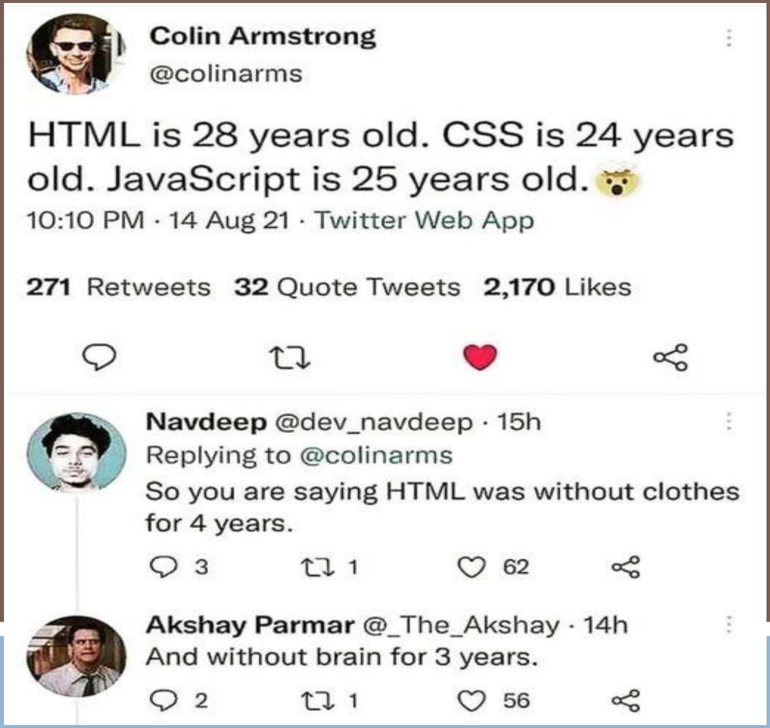
CHAPTER Four – JavaScript (JS)



Colin Armstrong
@colinarms

HTML is 28 years old. CSS is 24 years old. JavaScript is 25 years old. 😲

10:10 PM · 14 Aug 21 · Twitter Web App

271 Retweets 32 Quote Tweets 2,170 Likes

Navdeep @dev_navdeep · 15h
Replying to @colinarms
So you are saying HTML was without clothes for 4 years.

3 1 62

Akshay Parmar @_The_Akshay · 14h
And without brain for 3 years.

2 1 56

Outline

- ☐ **Introduction to Javascript**
  - ✓ Variables
  - ✓ Functions
  - ✓ Objects
- ☐ **DOM**
  - ✓ Accessing elements
  - ✓ Changing content
  - ✓ Changing attributes
- ☐ Events
  - ✓ HTML Event Handler
  - ✓ Dom Event handler (as a method)
  - ✓ Event Listener

Introduction to Java script

- Dynamic HTML (DHTML) is a grouping of technologies used to create **interactive** web pages, which are also referred to as dynamic web pages.
- Interaction implies that you can do something to a page and have it respond to your actions.
- For example, you can click button to change the appearance of an image on the page.
- DHTML is not a special version of HTML. Or even a version of HTML at all. Instead DHTML consists of a combination of the following three web technologies.
    - ✓ HTML
    - ✓ Cascading Style Sheet (CSS)
    - ✓ Web Scripting (like JavaScript, PHP scripting)

- With the combination of the above three, the web page is no longer static when you create the page.
- Using Web script, you can alter the style of the web content based upon user interaction such as clicking and dragging the mouse.
- In web there are two types of scripting: **Client** and **Server-side** scripting.
- Client-side script is **executed on the web browser** (client machine). Script written with JavaScript or jQuery library runs on the web browser machine.
- Server-side script **executed on the web server,** Script written with PHP, JAVA, C# runs on the web browser machine.

Adding Javascript to a page

- Embedded Script: to embed a script on a page, just add the code as the content of a script.

```
<script>
  … JavaScript code goes here
</script>
```

- External Script: The other method uses the src attribute to point to a script file (with a .js suffix) by its URL. In this case, the script element has no content:

```
<script src="my_script.js"></script>
```

- The script element can go anywhere in the document, but the most common places for scripts are in the head of the document and at the very end of the body.

JavaScript Basics

- A JavaScript program consists of *statements* and *expressions* formed from *tokens* of various categories.
- JavaScript is case-sensitive.
- An interpreted language
  - ❖ A JavaScript is simply lines of executable code
- Semicolon (;) at the end of a statement is optional if each statement is on a separate line
  - ✓ If more than one statement per line, separate them using(;)
- JavaScript ignores **spaces**, **tabs**, and **newlines** that appear between tokens in programs.
- Comments

```
// single line comment
/*  multi line comment */
```

- Gives the developer a means to:
  - ❖ Validate user input
  - ❖ Put dynamic text in an HTML page
  - ❖ React to events (interactive)
  - ❖ Read and write HTML elements in an HTML page
- A statement can be simple or compound.
  1) Simple

A *simple statement* is just simple, like so: x = 4;

  2) Compound

A *compound statement* combines multiple levels of logic. An *if/then/else* conditional

```
if (something == 1)
        {// some code here}
    else
        {// some other code here}
```

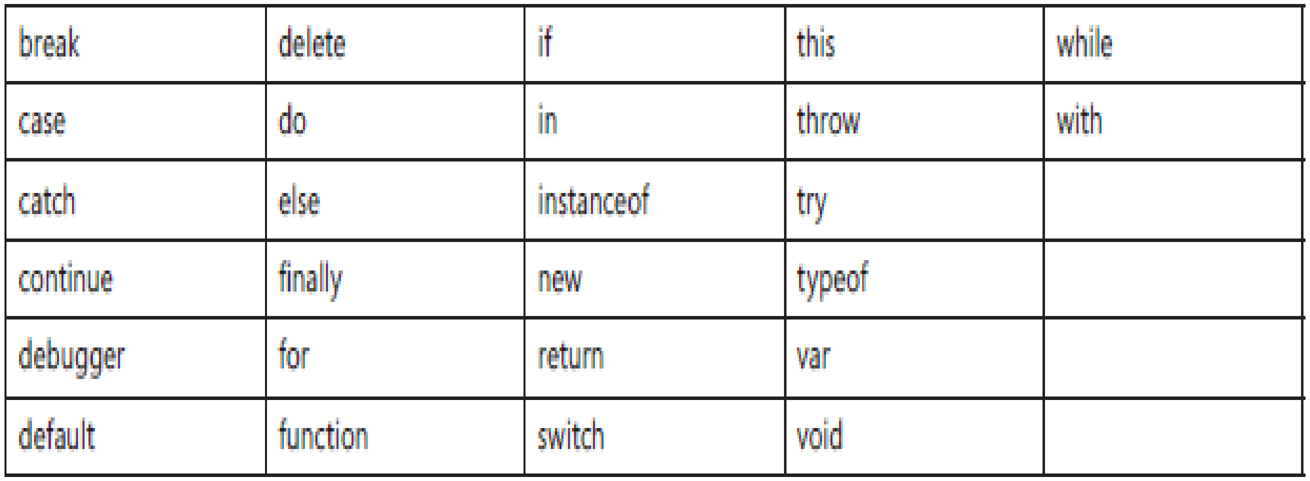
- ☐ Identifiers

  The first character must be a letter, an underscore (_), or a dollar sign ($). Subsequent characters can be a letter, a digit, an underscore, or a dollar sign.

Examples

```
i
distance_value
v13_dummy
$str
_some_variable
```

- ☐ Reserved Words
  - ✓ When we create a statement in JavaScript there are keywords that we don’t use them because they are reserved for JavaScript.



break	delete	if	this	while
case	do	in	throw	with
catch	else	instanceof	try	
continue	finally	new	typeof	
debugger	for	return	var	
default	function	switch	void	

Variable

- Variables are declared in JavaScript with the *var* keyword prior but now we can use let and const. JavaScript is loosely type so we don't have to specify type in declaration.
- The following are all valid variable declarations:

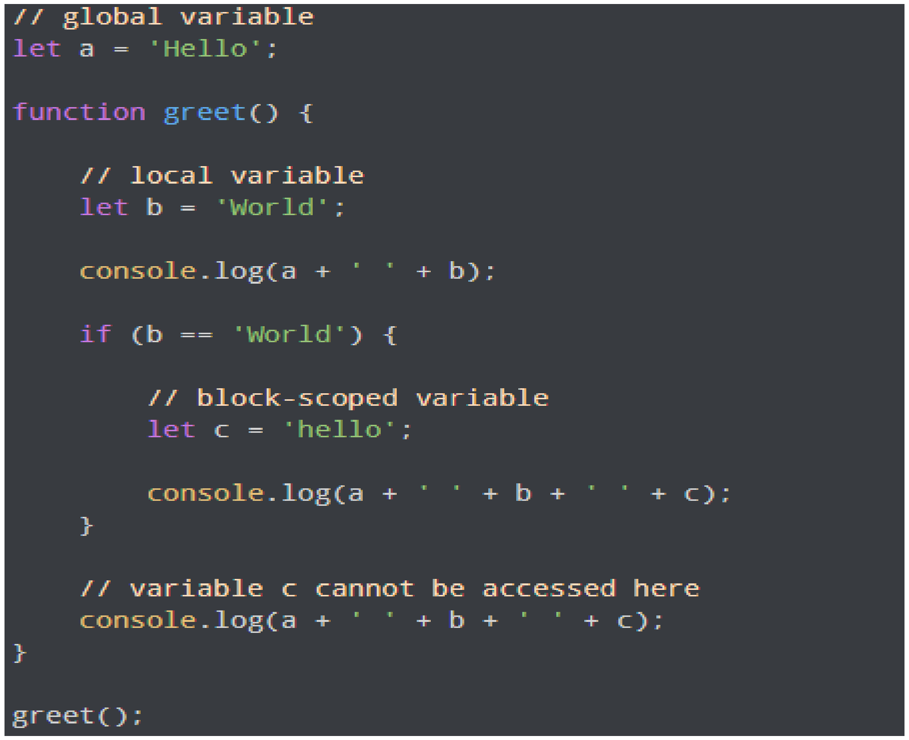
```
var x;
let myVar; // a variable defined by let can be changed later
```

const x=5// you cannot declare a constant without initializing it and the variable value can’t be changed.

- ☐ Variable names can contain uppercase and lowercase letters as well as numbers, but they cannot start with a number.

Variable Scope

- ☐ If a variable is defined inside a function using assignment (without var, let, const) then the variable is global.
- ☐ The local scope can be sub-divided into two: functions scoped and block scoped.
- ☐ If a variable is defined inside a function using ‘var’, then the variable is local to the function (function scoped). A variable defined using let keyword is block scoped (accessed only in the immediate block).



```javascript
// global variable
let a = 'Hello';

function greet() {

    // local variable
    let b = 'World';

    console.log(a + ' ' + b);

    if (b == 'World') {

        // block-scoped variable
        let c = 'hello';

        console.log(a + ' ' + b + ' ' + c);
    }

    // variable c cannot be accessed here
    console.log(a + ' ' + b + ' ' + c);
}

greet();
```

- ☐ Datatypes and Variables
  - ✓ Numbers: integer, floating point, hexadecimal, octal
  - ✓ Strings
  - ✓ Boolean
  - ✓ Objects
  - ✓ Arrays

Operators

- Arithmetic Operators

Operator	Description	Example	Result

+	Addition	x=2
		y=2
		x+y
			4

-	Subtraction	x=5
		y=2
		x-y
			3

*	Multiplication	x=5
		y=4
		x*y
			20

/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Assignment Operators

Operator	Example	Is The 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

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false

===	is equal to (checks for both value and type)	x=5
		y="5"
		
		x==y returns true
		x===y returns false

!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

- Logical Operators

Operator	Description	Example

&&	and	x=6
		y=3
		
		(x < 10 && y > 1) returns true

||	or	x=6
		y=3
		
		(x==5 || y==5) returns false

!	not	x=6
		y=3
		
		!(x==y) returns true

- Conditional statements
  - ✓ if, if else
  - ✓ switch
- looping
  - ✓ while
  - ✓ do … while

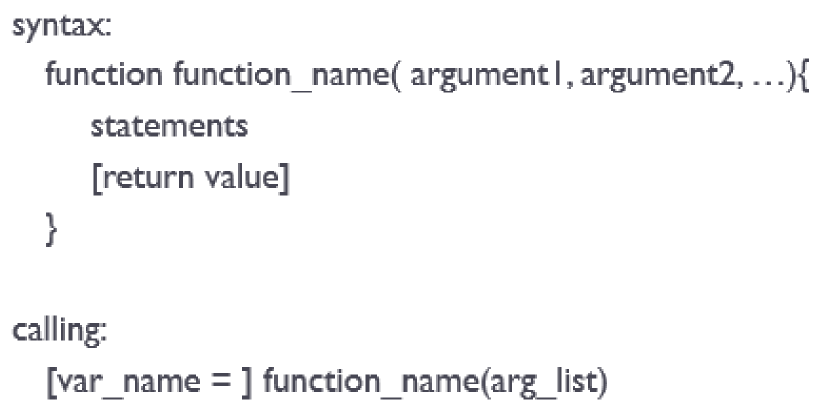
- ✓ for
- ☐ conditional operator
- ✓ var_name = (condition) ? true_value : false_value

Function

✓

- ☐ Function can be defined into two ways: named function or function expression.

Named function



```
syntax:
   function function_name( argument1, argument2, ...){
      statements
      [return value]
   }

calling:
   [var_name = ] function_name(arg_list)
```

Function Expression (Anonymous function)

```
    var x= function () {
      [return value]
}
```

Calling:

x()// calling the function is done by using variable name