https://shorturl.at/kAHNV

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Javascript Intro**

**Client side script**

**Variables**

**var let const**

**Problems with var**

**Operators**

**Control statements**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

===============================================

Javascript intro

===============================================

- Javascript is a scripting language.

- Javascript was implemented by Brendan in 1995, at Netscape.

- It was initially named Livescript.

- It is an interpreted language.

- Latest version is ES14, released in Jun 2023.

(ES -> ECMA Script)

- Use of Javascript

- Scripting

- Making Page as dynamic.

- Client side validation.

- Modern UI framework applications

- AngularJS

- ReactJS

- VeuJS

- NodeJS

- ...

- etc

- Javascript programs(scripts) can be executed by

either web browsers or Node Engines.

- Javascript is an Object Based Scripting language.

- Javascript supports predefined objects

Eg Object

window,

document,

console, etc

- Javascript supports Primitives and nonprimitives

primitives

- string

- number

- boolean

- undefined

- null

- symbol ???

- bigInt

nonprimitives

- object

- list

- arrays

- class ???

- Set

- Map

- JSON

===============================================

Environmental Setup

===============================================

- VS Code

Website :- https://code.visualstudio.com/#alt-downloads

- NodeJS

Website :- https://nodejs.org/en/download/

Note:- Javascript programs must have extension '.js'

===============================================

Execution of Javascript programs

===============================================

demo.js

console.log("Hello world")

1. Command prompt

>node demo.js

Note:- 'document' object not supported here

2. HTML

index.html

...

<script src = "demo.js">

...

- open this file in web browser

- inspect console

3. HTML with script

...

<script>

console.log("Hello World")

</script>

...

- open this file in web browser

- inspect console

===============================================

Variables:-

===============================================

- Variables are used to store data.

- by using variables we can store any kind of data,

Eg number, sting, date, audio, video, file, etc.

- Variables can be declared using 'var', 'let', 'const'

- let and const introduced in ES5.

Strings:-

- Strings can be declared with '', "", ``

- `` introduced in ES6.

- `` is used to define multiline strings

var wish = 'Good Morning'

console.log(wish)

var sub = `Javascript`

console.log(sub)

var myWish = `Welcome to ${sub}`

console.log(myWish)

Numbers

- Javascript supports decimal, double, bigInt, hexadecimal, octal and binary

/\*

var wish = 'Good Morning'

console.log(wish)

var sub = `Javascript`

console.log(sub)

var myWish = `Welcome to ${sub}`

console.log(myWish)

\*/

let decimal = 100

console.log(decimal)

console.log(typeof(decimal))

let double = 12.34

console.log(double)

console.log(typeof(double))

let hexadecimal = 0x12ab

console.log(hexadecimal)

let octal = 0o127

console.log(octal)

let bin = 0B11

console.log(bin)

//Output will be in base 10 form

//all alphabates are case insensitive

let bigInt = 5n

console.log(bigInt)

console.log(typeof(bigInt))

let largest = 1.7 \* 10\*\*308

console.log(largest)

let anotherMax = Number.MAX\_VALUE //is it really maximum number ?

console.log(anotherMax)

anotherMax += 1

console.log(anotherMax)

console.log(typeof(anotherMax))

//Boolean

var flag = true

console.log(flag)

var flag = false

console.log(flag)

//var v/s let

//Eg01

for(var i = 0; i<10; i++)

{}

console.log(i)

for(let j = 0; j< 10; j++)

{}

console.log(j) //ReferenceError: j is not defined

//Eg02

var data = 100

var data = 200

console.log(data)

let num = 100

let num = 200 //SyntaxError: Identifier 'num' has already been declared

console.log(num)

//Eg03

var data = 100

{

var data = 200

}

console.log(data) //200

let num = 100

{

let num = 200

}

console.log(num) //100

//global polluting issue:- local variables affecting global variables.

//Eg04

console.log(data)

var data = 100

console.log(num) //ReferenceError: Cannot access 'num' before initialization

let num = 200

//const

const data = {

value : 100

}

console.log(data.value)

data.value = 200

console.log(data.value)

data = {} //TypeError: Assignment to constant variable.

===============================================

Control statements:-

===============================================

- Changing flow of program

Types

i) Decision control

ii)Looping control

iii)Case control

iv)Jump control

i) Decision control (if statements)

/\*Simple if\*/

if(no < 0)

negative

/\*if - else\*/

if(no < 0)

negative

else

positive

/\*if - else ladder\*/

if(no < 0)

negative

else if (no > 0)

positive

else

Zero

Short Assignment

Find biggest number among 3

a

b

c

if a > b && a > c

a

else if b > c

b

else

c

ii)Looping control

important

Initialyzation

Termination

Incrementing Factor

i) for loop

ii)while loop

iii)do while loop

i) for loop

Syntax for(Initialyzation; Termination; Incrementing Factor)

ii)while

Syntax

Initialyzation

while(Termination)

{

//body

Incrementing Factor

}

iii) do while

Syntax

Initialyzation

do

{

//body

Incrementing Factor

}

while(Termination)

1. Factorial of a number

4! = 24 4 x 3 x 2

5! = 120 5 x 4 x 3 x 2

6! = 720 6 x 5 x 4 x 3 x 2

2. Check number is prime or not

iii)Case control

switch(choice):

{

case 1:

//

break

case 2:

//

break

...

default:

break /\*optional for last case\*/

}

Menu driven program for various options

1. Addition

2. Subtraction

3. ...

0. Exit

Assignment

Case study of grade calculation

per < 40 -> failed

41 - 50 -> pass

51 - 55 -> SC

56 - 60 -> HSC

61 - 65 -> FC

65 > -> FC with distiction

let per = 54

switch (true) {

case (per > 0 && per <= 40):

console.log("Failed")

break

case (per > 40 && per <= 50):

console.log("Pass")

break

case (per > 50 && per < 55):

console.log("Second class")

break

case (per >= 55 && per < 60):

console.log("Higher Second class")

break

case (per >= 60 && per < 66):

console.log("First class")

break

case (per>= 66 && per < 100):

console.log("First class with distiction")

break

default :

console.log('Invalid percent')

}

iv)Jump Control

break :- it terminates complete block

continue :- it terminates current iteration only

;(semicolon) :- used as empty block

for (let i = 1; i <= 10; i++)

process.stdout.write(i + " ")

console.log()

for (let i = 1; i <= 10; i++) {

if (i == 5)

break

process.stdout.write(i + " ")

}

console.log()

for (let i = 1; i <= 10; i++) {

if (i == 3 || i == 6)

continue

process.stdout.write(i + " ")

}

console.log()

for (let i = 1; i <= 10; i++) {

if (i == 3 || i == 6)

;

else

continue

process.stdout.write(i + " ")

}

===============================================

Operators

===============================================

i) Arithmetic Operators -> eg +, -, \*, /, %, ++, --, \*\*

ii)Relational Operators -> eg ==, !=, >, <, >=, <=,

iii)Logical Operators -> eg &&, ||, !

iv)Bitwise Operators -> eg &, |, ^, ~, <<, >>

v) Assignment Operators -> eg =, +=, -=, \*=, /=,...,<<=,>>=,...

Strictly equal to -> ===

i)Arithmetic Operators:-

Consider A = 25, B = 7, Calculate C

C = A + B it adds A & B, C = 32

C = A - B it substracts B from A C = 18

C = A \* B it multiplies A and B C = 175

C = A / B it devides A by B C = 3.5714285714285716

C = A % B it stores reminder of division of A by B, C = 4

A++ it increments A by 1 A = 26

B-- it decrements B by 1 B = 6

3 \*\* 4 = 81

ii)Relational Operators:-

Consider A = 5, B = 10, C = 5

d = '5'

A == B False

A == C True

A == d True

A === d False

A != B True

A > C False

A >= C True

B >= A True

C < B True

B < C False

iii)Logical Operators

Consider A = 1, B = 0

(A && B) -> 0

(A || B) -> 1

!(A && B) ->true

iv)Bitwise Operators

Truth Table

a b a&b a|b a^b ~a ~b

0 0 0 0 0 -1 -1

0 1 0 1 1 -1 -2

1 0 0 1 1 -2 -1

1 1 1 1 0 -2 -2

2's Complement form = 1's Complement form + 1

1's Complement form = replace 0's with 1's and vice versa

Dec bin 1’s 2’s -Dec

10 1010 0101 0110 -10

11 1011 0100 0101 -11

12 1100 0011 0100 -12

13 1101 0010 0011 -13

14 1110 0001 0010 -14

15 1111 0000 0001 -15

Shifting

<< left shift -> pad 0's from right

>> right shift -> bit discard from right

Consider C = 4 bin format 0100

bin shift dec

C << 1 0100 01000 8

C >> 1 0100 010 2

5<<2

7<<2

120>>2

5<<2 0101

7<<2 111

120>>2 1111000

5<<2 0101 10100

7<<2 111 11100

120>>2 1111000 11110

5<<2 0101 10100 20

7<<2 111 11100 28

120>>2 1111000 11110 30

v) Assignment Operators

Consider A = 5

Expression Result

C = A 5

C += A C = C + A 10

C -= A C = C - A 5

C \*= A C = C \* A 25

C /= A C = C / A 5

C <<=2 C = C << 2 20

Task2

POC

- Design e-commerse website

- options are (two or more)

- Medicins

- Clothing

- Workout

- Mobiles

- Mobile accessories

- Computer equipments

- Computer accessories

- Books

- Groceries

- Car accessories

- Jewelleries

- watches

- Sun glasses

- Website should have rich user interface

- There should be card layout for each product

- on mouse hover on product image that product should be enlarged

- There should be 'learn more', 'add to cart' and 'buy now' options

- There should be login page for sigining in

- There should be cart page where we will get number of added items

- In buy now page show complete cost of purchase

Enhancements (Optional)

- Try user interface without using Bootstrap CDN

- Use different GST rates for various products

================================================================

================================================================