Day16-Variables (var, let, const), Data Types, Operators

Class notes:

Heap + Stack memory

HTML + CSS + Boostrap

2 types of js

Math & Funtional process - like Your Logical things and code perfomance and understand of your js.  - 20%

DOM - 50%:

Project expreance - 70%

20% Communication - to tell what you learn in the js topics not code.

React - JS - React - Node - java, python

10% - git , laoptop useage, city , unnoda gender , unnoda communication .

thinking process, attitute, calm , time management , documents ,

detication and intersert .

Veriables:

var    - old es1 es5

let    - es6

const  - es6

Declaration Phases (memory creation phases) -

variable keyword variable name;  - memory allocation creation

variable name - camel case, upper camel case pascel case , snake case,

oneVariable -   camel case

OneVariable -   upper camel case pascel case

One\_variable   - snake case

variable keyword variable name

var headingProcess;  declaration part;

type of using js in web application.

internal - <script></script>

external - filename.js

Access phase:

Printing statement -

console.log()

document.write()

alert()

Execution phase -

initialization - part

variable keyword variable name = data

what are all the data we have in js :

Primitive data types - single values

string       -   "two 2345 this is"

number       -   1234567890123456789012345667890

boolean      -   true or false

null         -   No Values. but in js its showing the type is object

undefined    -   We give any declaration with out data is undifined the js engines shows the error

bigint       -

symbol       -

NaN          - NaN

Non Primitive Data Types:

Referance type strore - we use like address pointer.

collection of datas or Process in single varible.

Array

Array of object

object

functions

Arithmatic Operators

+,-,\*,/,%,\*\*

Assignment Operators

=,+=,\*=,

+=

x=x+1

Comparison Operators:

==             - less equal comparison or validation =

===            - strict equal -

!=

>

<

<=

>=

Logical Operators:

&& (AND)  - t-f =f f-f=f,f-t=f,t-t=t

|| (OR)   - f-t -t t-t-t ,t-f=t,f-f=f

! Not     - t-f,f-t

es6

Ternary operator

constion?"":""

Variable -

variable key words variable name = 10

Printing statement