

lets break the concepts into two parts

- basics
- advanced

1st learn this..

- word vs keyword

↓
have no meaning
in js.

e.g. kiran, meow,
bhaiya etc.

↓
they've some meaning
in JavaScript

e.g. for, if, else, true

↓? ↓?
loop condition ...

- variables and constants, let

↓
The one whose
value keep changing

e.g. score = 5

!
after sometime

var score = 10;

↓
always remain's same

e.g. const pie = 3.14

let is also used to create
variables, same like var

e.g. let name = "meow"

but, difference betn
var & let is -- topic

- hoisting

↳ The functions and variables are hoisted which
means their declaration is moved on top of code.

e.g. console.log(a);
var a = 10

o/p: undefined

but, 1st we are printing & then
declaring, so it should give error?

This is when hoisting comes into picture -

var a = 10; } → var a; declaration
a = 10;

classmate

Date _____
Page _____

var a;

console.log(a);
a = 10;

console.log(a);
var a = 10;

#hoisting

undefined vs not-defined

↓
we are defining 'a'
later in program

↓
do not exists
eg. console.log(b)

- types in js

- Primitives (number, string, boolean, ...)
- Reference -- [], {}, () all brackets.

↓
when we try to copy these values, then it passes
their reference @ address.

if no brackets → primitives → copy directly..
if yes brackets → reference

eg. var a = [1, 2, 3, 4]; a having his own 1, 2, 3, 4
var b = a; b having a's 1, 2, 3, 4
// b = [1, 2, 3, 4]
b.pop();

∴ if you make some changes
to 'b' they'll reflect
in a

o/p: a = [1, 2, 3]
b = [1, 2, 3]

Same values are shared among
each other

then how? -- topic

- conditionals

→ if
→ if else
→ else-if

```
if ( )  
{  
}  
else if ( )  
{  
}  
else if ( )  
{  
}  
else  
{  
}
```

आर मार की बातें

- with great power comes great responsibility

if (true / false)
{
 execute;
}
else {
}

true → 1, 12 > 10 ✓, a == a
false → 0, 12 < 10, etc..

but, what if someone gives string in if () condition..

↓
-- truthy & falsy values switch

transhine

- loops

→ while
→ for loop

The one which repeats

Syntax :

For (var a = 1; a <= 10; a++)
 console.log(a);

① initialize; ② condition; ③ update..

var a = 1;

while (a < 10)

{
 print(a);

}

when the condition gets false
it stops working..

-- for each, for in, for of

- functions

→ write some code give it a name & use that name as many times you want.

- whenever you call it, it runs

- reusability

Function maggiBanao()

```
{  
  clg..  
  clg..  
  !  
}
```

Day1: maggiBanao();

Day2: maggiBanao();

↑
function calling..

→ parameter's..

eg. function abcd (var a, var b)

```
{
```

```
}
```

abcd(3, 4);

→ actual values

- arrays

→ you cant store multiple values in one variable
∴ use array

eg.

var a = 12, 13;

×

var a = [12, 13];

✓

operations

- pop()

a.pop(); removes last ∴ a = [12]

- push()

a.push(10); add to last ∴ a = [12, 10]

- shift()

a.shift(); removes from start ∴ a = [10]

- unshift()

a.unshift(15); add from start ∴ a = [15, 10]

- splice()

a.splice(1, 1); removes from middle

index of element → how many elements

- objects

→ storing details of a particular thing in key-value

1) blank object

var a = {}

2) filled object

var a = {
 name: "Kiran",
 age: 22
}

accessing

⇒ a.age

→ 22

- props and methods

classmate

Date _____
Page _____

props

var a = {

name: "Kiran",

class: "BTech",

number: 123

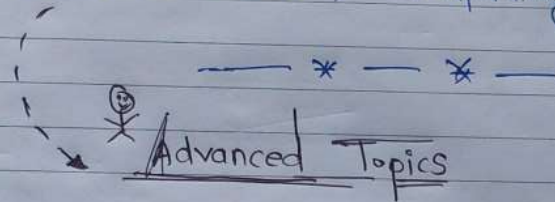
tasks: function() { } }

Kiranshinde

method

when a value of a property is a function, then it is method.

a.name = "Meow"; → updating..



-- topic