Section 9 → Introduction to JavaScript.

① Introduction to Javascript.

⇒ In the 90s Netscape (N1) was the only browsing Application, later it was overpowered by Internet Explorer.

⇒ In 90s, if you wanted functionality, the request was sent to the server and it return the data.

⇒ A website can't load ads without Javascript. Youtube & Netflix don't work.

⇒ Name formation ⇒ Live script → JScript → ECMAScript → Javascript.

What it does

⇒ Like in case of actors, in their script they have func like, Adam appears after 1 sec, Adam gets angry & raises his voice.

⇒ Simi Javascript applies script to web.

Java vs JavaScript

⇒ As much in common as that in Car & Carpet.

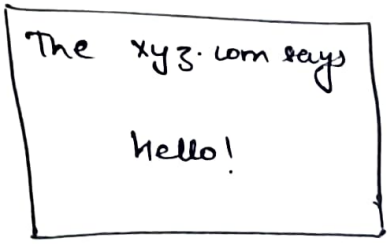
⇒ JS → interpreted (like python, Ruby)
Java → compiled (C++, C).

---

② JavaS Alerts.

⇒ alert("Hello!"); ⟶

(a pop up).



⇒ [" " ≠ " "]

3) Data types and Variables

→ alert ("Hello"); → Data type → String, Bool, Integer
- "Hello" → string
- alert → func (like printf).
- Data type ↳ common initiz var.
- string → combine string using + ["a" + "b"] = (ab).

→ **Prompt** → Like alert but allows user to type something.

→ var myName = "Angela"; & alert (myName);
- var → initis.
- "Angela" → data.
- myName ↳ Name of fn.

∴ In prompt → var YourName = prompt ("what is your name");

4) Naming Conventions

→ Give your var meaningful names.

→ Can't use keywords (func) as name.

→ Can't begin with number, and spaces.

→ Lower case then upper case → " myName (called camel Casing).

5) String length

→ fnName.length;

→ Used for counting letters before accepting.

For eg:- In an essay only 140 characters are req then if (fnName.length) > 140
{ Cutoff = true;
}

⑥ Slice & Extract

⇒ var myName = "Tyagi";

myName.slice(0,1); → i.e starting from 0 & going all the way to 1 (1 not included)

output → T ∴ pos(0) = T.

→ Extract means to take what you need from slice or word.

Extra → word.toUpperCase() → changes word to uppercase.

⑦ Basic Arithematic

Ctrl + F → change same word repeated many times.

→ var a+b; (same for −, +, *, /)

% → remainder / modular.

→ Pranthesis → (3+5)*2 & 3+(5*2).

→ Increment → x++ ⇒ x = x+1

Decrement → x−− = x = x−1

−−x

⑧ Functions.

⇒ A package with instruction to perform a specific task.

→ Syntx → calling → getMilk();

Declaring → fn getMilk(){

....

}

» console.log → Like alert, but not visible to user, just the developer.

↳ Only visible on console, not a pop up.

Standford Karel. → A simple IDE.

Functions Part 2

» Parameters & Arguments → Calling → getMilk(2);

Create → function. getMilk(bottles)

↳ not needed

```
{ console.log("buy" + bottles + "bottles of milk");
}
```

Math.floor to round value

Math.round.

Functions Part 3

» Return & outputs → x = getMilk(5);

```
func getMilk(bottles){
    Return bottles*1.5;
}
```

now x = 7.5.