Lecture 21
10/2/23 [JS]

Date: 11
P. No:

Rutuja Patil

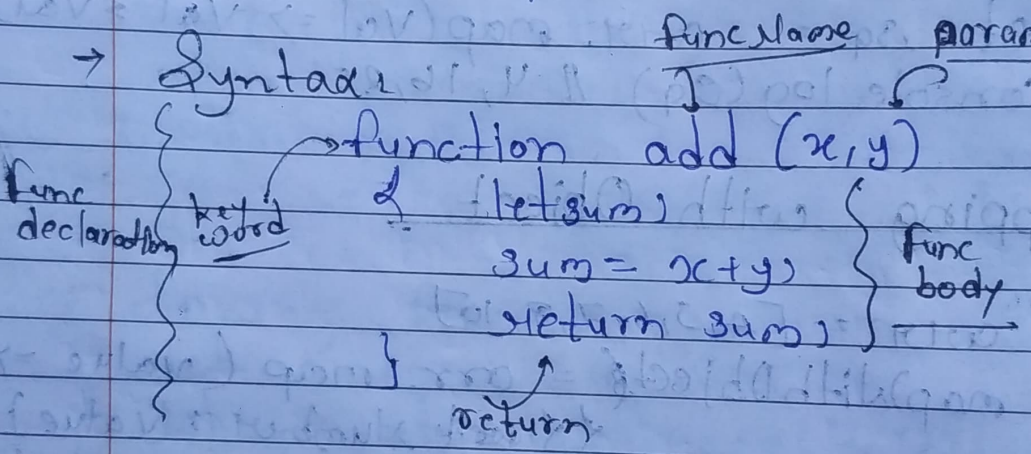
* Functions

→ Function is a block of code that fulfil specific task.

→ Increase Reusability.

→ Function is a subprogram which linked to specific task.

→ Syntax:



add(5,10); {invoked/Run func.

* Hoisting (Done by JS engine)

All function declaration comes at top of file is called "Hoisting".

* Function Assignment.

let stand = function walk()
{ console.log("walking"); }

* Function (Anonymous)

let walk = function(){
console.log("walking");
}

Date: 11

Priyansh Patel

* Arguments

→ all the arguments of a function stores in argument object.

```
function add (a,b)
{ let sum = 0;
  for(let i of argument)
    sum += i;
  return sum;
}
add (1,2,3,4,5,6,7,8); // 36
```

* Rest Operator (...)

→ Rest parameter must be last parameter
Ex function add (a,b,...args) ✓
function add (a,...arg,b) ✗
→ Array type.

* Default Parameter

```
function add (a, b=0)
{ return a+b; }

add (5, 10) // 15        add (5) // 5
```

→ when we give defaut parameter then remaining must be "

→ We can avoid it by:
Add (8,16, undefined, 12,14);
(undefined → default in func)

access property

change & mutate properties

Date: / /
P. No:

* Getter & Setter

```
let object = {
    firstName : "priyansh",
    lastName : "patel",
    get FullName () {
        return `${firstName} ${lastName}`
    }
    set FullName (value) {
        let path = value.split(' ')
        this.firstName = path[0]
        this.lastName = path[1]
    }
}
```

setter → `object.FullName = "Priyansh Patel";`

getter → `console.log(object.FullName);`

* Try & Catch

write code which may have error

If error present then executed

```
try {
    a = 10/0 ; }
catch (e) {
    console.log(e)
}
```

* Var → global Scope
let → local Scope
Const → local Scope