JS

functions :- set of statements to perform the perticular task.

Define :-

1) Function Declaration

Syntax:- function functionname (parameterlist)
```
{
    // body

}
```

e.g.

declr + defn

```
let a = 10;
let b = 20;
function add (n_1, n_2)
{
    let result = n_1 + n_2;
    console.log (result);
}
```

Calling the function

$\underline{add}$ (a, b)

② Function Expression   let a = 10
                             let b = 20

```
let addition = function add (n₁, n₂)
                    {
                        let result = n₁ + n₂;
                        c.log (result);
                    }

          addition (a,b)    // calling statement
```
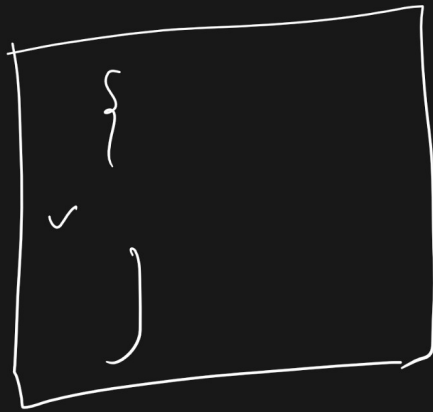
③ Anonymous Function

```
let a = 10
let b = 20
let addition = function (n₁, n₂)
                    {
                        let result = n₁ + n₂
                        c.log (result)   ✓ 30
                    }

          addition (a,b)   ⟹
```

```
┌─────────────┐
│     {       │        ← when it is run
│     ✓       │
│     }       │          when i called this
└─────────────┘               fun                    funˆname()
```

④ IIFE (Immediate Invoked Function Expression)

```
let a = 10 ─
let b = 20 ─;
     start
  └→ ( function (a,b)
IIFE      {
              let result = a+b
              C.log(result)
          }) (a,b) ; end
```

⑤ Arrow function:-

→ Same as a function expression

eg.
```
let a = 10
let b = 20

let addition = (a, b) =>
    {
        let result = a+b
        c.log (result)
    }
```
arrow fun⁰

addition (a,b)

Hoisting

→ c.log (a)   undefined

var a = 10

Variable of var

→ ✓ let a = 10
→ let b = 20

→ ✓ add (a,b)
function add (a, b)
{
    let result = a+b
    c.log (result)
}

gcl →

MAP	exp. phase
a: undefined —	a: 10 —
b: undefined —	b: 20 —
add: [body]	