A JavaScript **closure**

**Example :**  
for (**var** i = 0; i < 3; i++) { setTimeout(function() { alert(i); }, 1000 + i); }

**Output:3 3 3**

**Example :**  
for (**let** i = 0; i < 3; i++) { setTimeout(function() { alert(i); }, 1000 + i); }

**Output : 0 1 2**

**Example :**

for (var i = 0; i < 3; i++) {

var abc=(i)=>{

setTimeout(function () { console.log(i); }, 1000 + i);

}

abc(i);

}

**Output : 0 1 2**

**Example :**

function createBase(baseNumber) {

return function(N) {

// we are referencing baseNumber here even though it was declared

// outside of this function. Closures allow us to do this in JavaScript

return baseNumber + N;

}

}

var addSix = createBase(6);

addSix(10); // returns 16

addSix(21); // returns 27

**Factory function**

**Example**

// Facroty function

function Fun(name) {

return {

name: name,

display() {

console.log(`Hi! --, ${this.name}`)

}

}

}

var a = Fun("suraj")

a.display()// Hi!, suraj

a.display=function(){

console.log(`bye ${this.name}`)

}

a.display()// bye!, suraj

var b = Fun('Harini')

b.display()// Hi!, Harini

console.log(a.display)

console.log(b.display)

**pollifill of map**

// Custom Map function

const a = [1, 2, 4, 8, 4, 6, 9, 4, 7, 9]

console.log(a.map((x) => {

    return x;

  }));

  const Cmap = (x, func) => {

    let NewArray = []

    for (let i = 0; i < x.length; i++) {

      NewArray.push(func(x[i]))

    }

    return NewArray;

  }

  console.log(Cmap(a,(x) => {

    return x;

  }));