# CLIENT SIDE SCRIPTING (22519)

**Practical No. 04:** Develop JavaScript to implement functions
**Roll No.:** 220447

---

**1) Write a program to check whether the number is Palindrome or not using function.**

```
function isPalindrome(num){
let  rev=0,temp=num;
while(Math.floor(temp)>0){
        a=temp%10;
        temp= Math.floor(temp/10);
        rev = rev*10 + a;
    }
    rev==num? console.log("A palindrome
number") : console.log("Not a palindrome
number");
}
```

```
> isPalindrome(121);
  A palindrome number
< undefined
> isPalindrome(01881);
  A palindrome number
< undefined
> isPalindrome(12320);
  Not a palindrome number
```

**2) Write a program to print the reverse of n digit number using function.**

```
function reverse(num){
let  rev=0,temp=num;
while(Math.floor(temp)>0){
        a=temp%10;
        temp= Math.floor(temp/10);
        rev = rev*10 + a;
    }
console.log('Reverse : ' + rev);
}
```

```
reverse(2006)
Reverse : 6002
undefined
reverse(0823)
Reverse : 328
undefined
reverse(08230)
Reverse : 328
```

**3) Write a program to print the sum of n digit number using method.**

```
function sumOfDigits(num){
    let temp =num, sum=0,a;
while(Math.floor(temp)>0){
        a=temp%10;
        temp= Math.floor(temp/10);
        sum+=a;
    }
console.log("The sum of digits : " + sum);
}
```

```
sumOfDigits(12)
The sum of digits : 3
undefined
sumOfDigits(2006)
The sum of digits : 8
undefined
sumOfDigits(12345)
The sum of digits : 15
```

# CLIENT SIDE SCRIPTING (22519)

**Practical No. 04:** Develop JavaScript to implement functions
**Roll No.:** 220447

**4) Write a program to check whether number is an Armstrong or not using method.**

```
function isArmstrong(num){
    let temp=num, digitCount=0, sum=0,a;
    while(Math.floor(temp)>0){
        digitCount++;
        temp = Math.floor(temp/10);
    }
    temp = num;
    while(Math.floor(temp)>0){
        a=temp%10;
        a**= digitCount;
        sum+=a;
        temp = Math.floor(temp/10);
    }
    sum==num? console.log("An Armstrong number") : console.log("Not an Armstrong number");
}
```

```
> isArmstrong(153)
  An Armstrong number
< undefined
> isArmstrong(88593477)
  An Armstrong number
< undefined
> isArmstrong(9127)
  Not an Armstrong number
```

**5) Write a program to print the factorial series using constructor.**

```
function factorialSeries(size){
   let series = new Array(size);
   let j=1;
   for(let i=1;i<=size;i++){
    j*=i;
    series[i-1] = j;
   }
   return series;
}
```

```
> factorialSeries(6);
< ▶ (6) [1, 2, 6, 24, 120, 720]
> factorialSeries(10);
< ▶ (10) [1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800]
```

# CLIENT SIDE SCRIPTING (22519)

**Practical No. 04:** Develop JavaScript to implement functions
**Roll No.:** 220447

**6) Write a program to print the Fibonacci series using constructor.**

```
function fibonacciSeries(size){
    let series = new Array(size);
    let f1=0,f2=1,f3,i=3;
    series[0] = f1;
    series[1] = f2;
    f3=f2+f1;
    while(i<=size){
        f3=f2+f1;
        series[i-1] = f3;
        f1 = f2;
        f2 = f3;
        i++;
    }
    return series;
}
```

```
fibonacciSeries(10)
▶ (10) [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
fibonacciSeries(20)
▶ (20) [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181]
fibonacciSeries()
▶ (2) [0, 1]
```