

Function

What is Function Declarations:

```
function square(num) {  
  return num*num;  
}
```

What is Function Expression:

```
const square = function(num) {  
  return num * num;  
};  
  
console.log(square())  
  
//the function which does not have a name called an anonymous function
```

What are First Class Functions:

when a function can be treated like a variable these functions are called first-class functions

```
function square(num) { //params  
  return num * num;  
}  
  
function displaySquare(fn){  
  console.log("Square is " + fn(5)); //here fn(5) is first class function  
                                     //as it can be used as variable  
}  
  
displaySquare(square) //Arguments
```

IIFE

Immediately invoke function expressions

```

(function square(num){
    console.log(num*num);
}) (5);

(function (x){
    return (function(y){
        console.log(x);//output 5
    })(2);
}) (5);

```

Function Scope - o/p Based Question

```

for(var i = 0 ; i<5; i++){
    setTimeout(function(){
        console.log(i)
    },i * 1000)
}
//output 5 5 5 5 5
// var does not have block scope

for(let i = 0 ; i<5; i++){
    setTimeout(function(){
        console.log(i)
    },i * 1000)
}

// output 0 1 2 3 4

```

Hosting

function will work no matter where it is called

Spread vs rest

```

function multiply(...nums) { //rest operator and it will take the value as an array
    console.log(nums);
    console.log(nums[0]*nums[1])
}

var arr = [5,6]

multiply(...arr); //spread operator. it will destructure the array like 5,6 and pass the
// arguments

```

```
//when we use parameters in function along with the rest operator the rest operator should  
//be the last parameters. Otherwise it will give error
```

```
const fn = (a,x,y,...numbers)=>{  
  console.log(x,y,numbers)  
};  
fn(5,6,3,7,8,9)
```

Arguments in (function vs arrow function)

```
function fn(){  
  console.log(arguments)  
}  
fn(1,2,3)  
//output [1,2,3].we can access all the values as arguments array inside the function  
  
const fnArr=()=>{  
  console.log(arguments)  
};  
fnArr(1,2,3);  
  
//ReferenceError: arguments is not defined  
  
//so in the arrow function the arguments array is absent
```

This in (function vs arrow function)

```
let user = {  
  username: 'Hayat Hossain',  
  fn1:()=>{  
    console.log('Subscribe to ' + this.username)//output: Subscribe to undefined  
  },  
  fn2(){  
    console.log('Subscribe to ' + this.username)//output: Subscribe to Hayat Hossain  
  }  
}  
  
user.fn1();// (this) will point to the global object  
user.fn2();// (this) will point to the user
```