

```

//Functions in JS
//Q1 Function Declaration
function square(num) {
    return num * num;
}

//Q2 Function Expression
const squarefn = (num) => {
    return num * num;
};
console.log(squarefn(8));
//Anonymous function

//Q3 First Class Functions
//Functions can be treated as variables
function calcSquare(num) {
    return num * num;
}
function displaySquare(fn) {
    console.log("Square is : " + fn());
}
displaySquare(calcSquare(6));

//Q4 IIFE
(function displaynum(i) {
    console.log("Number is : " + i);
})(4);
//Q5 O/P based
(function (a) {
    return (function (b) {
        console.log("Value: " + a);
    })(5);
})(4);
//Q6 Function scope-MDN;
//*****FUNCTIONS***** */

//Q7 O/P based -Function scope
for (let index = 0; index < 5; index++) {
    setTimeout(() => {
        console.log(index);
    }, index * 1000);
}
for (var index = 0; index < 5; index++) {
    setTimeout(() => {
        console.log(index);
    }, index * 1000);
}

```

```

// Q8 Function hoisting
hoisted();
function hoisted() {
  console.log("Function is hoisted and called");
}

//Q9 O/P based
var x = 21;
var abc = function () {
  console.log(x);
  var x = 20;
};
abc();

//Q10 Params vs Arguments
function sq(x) {
  //Parameters
  return x * x;
}
sq(10); //Arguments
function multiply(...nums) {
  //Rest Operator
  return nums[0] * nums[1];
}
const ar = [1, 2, 3, 4, 5];
multiply(...ar); //Spread operator
//Rest and Spread Operator
// Q11 Parameter based o/p question
const fn = (a, x, y, ...z) => {
  console.log(a, x, y, z);
};
//Q12 Callback
// MDN
document.addEventListener("click", () => {});
//Q13 Arrow functions
function reg() {
  var num = 8;
  return num * num;
}
const arrow = () => {
  var num = 8;
  return num * num;
};
//Implicit return

const cube = (a) => a * a * a;
//Arguments objects
function f1() {
  console.log(arguments);
}

```

```
}  
const fn1 = () => {  
  console.log(arguments);  
};  
  
let user = {  
  name: "Vishal",  
  rc1: () => {  
    console.log("Hello " + this.name);  
  },  
  rc2: function () {  
    console.log("Hello " + this.name);  
  },  
};  
user.rc1();  
user.rc2();
```