Notes: Function

```
// 1. Function Declaration
// A function declared with the function keyword.

function greet(name) {
  return "Hello, " + name;
}

console.log(greet("Niraj")); // Hello, Niraj
// Can be hoisted (called before it is declared).

//  2. Function Expression
// A function stored in a variable. It is not hoisted.

const greet = function(name) {
  return "Hi, " + name;
};

console.log(greet("Niraj")); // Hi, Niraj


// 3. Arrow Functions
// A shorter syntax for writing function expressions. Introduced in
ES6.
const add = (a, b) => {
  return a + b; // 5 + 3
};
console.log(add(5, 3)); // 8

// Shorter if only one line
const square = x => x * x; // 4 * 4
console.log(square(4)); // 16
// Note: Arrow functions do not have their own this, which makes
them useful in callbacks or inside classes.

// 4. Parameters vs Arguments
// Parameters are variables listed in function definition.
// Arguments are actual values passed when calling the function.
function greet(name) {  // 'name' is a parameter
  console.log("Hi " + name);
}
```

```
greet("Niraj");  // 'Niraj' is an argument

// 5. Return Values
//Functions can return a value using the return keyword.
function multiply(a, b) {
  return a * b;
}
const result = multiply(4, 5);
console.log(result); // 20

// 6. IIFE (Immediately Invoked Function Expression)
//An IIFE is a function that runs immediately after it is defined.
// Syntax:
(function() {
  console.log("IIFE executed!");
})();
//Example with Parameters:
(function(name) {
  console.log("Welcome, " + name);
})("Niraj");

//Why use IIFE?
// To avoid polluting the global scope.
// Create private variables or scope.
// Useful in modular code or one-time initialization.

// Example with private variable:

const counter = (function() {
  let count = 0;
    return function () {
    count++;
    console.log("Count: " + count);
  };
})();

counter(); // Count: 1
counter(); // Count: 2
```

Task : Function

1. Function Declaration Questions

Q1. Write a function that takes a number and returns its cube.

Q2. Create a function that checks whether a number is even or odd.

Q3. Write a function that takes your name and returns a greeting message.

Q4. Write a function that returns the sum of numbers from 1 to N.

---

2. Function Expression Questions

Q5. Convert your Q1 (cube function) into a function expression.

Q6. Write a function expression that calculates the area of a rectangle (length × width).

Q7. Create a function expression that returns the factorial of a number.

---

3. Arrow Function Questions

Q8. Write an arrow function to calculate the square of a number.

Q9. Create an arrow function to return the larger of two numbers.

Q10. Make an arrow function that returns true if a number is divisible by 3 and 5.

4. Parameters vs Arguments

Q11. Write a function greetUser(name, language) that prints:

- “Hello [name]” if language is English,
- “Namaste [name]” if Hindi,
- “Bonjour [name]” if French.

Q12. Create a function that takes a student's marks in 3 subjects as parameters and returns the average. Pass actual marks as arguments.

---

5. Return Values

Q13. Create a function that takes two numbers and returns an object with sum, difference, product, and quotient.

```
// Example return:
{
  sum: 10,
  difference: 2,
  product: 24,
  quotient: 1.5
}
```

---

6. IIFE (Immediately Invoked Function Expression)

Q14. Write an IIFE that prints your name.

Q15. Create an IIFE that calculates the square root of a number.

Q16. Write an IIFE that accepts two numbers and logs their sum.

Q17. Create an IIFE to calculate the factorial of 5 and log it.

Q18. Use IIFE to create a private variable counter and a function to increment it each time it's called (simulate private scope).