**Pre-Notes on Functions**

* **Definition:** Functions are reusable blocks of code that perform specific tasks. They allow you to encapsulate code logic, making it possible to execute it multiple times throughout your program, which aids in organising and managing code more effectively.
* **Characteristics of Functions:**
  + **Reusable:** Functions can be invoked multiple times, which helps avoid code duplication. This makes maintenance easier because you only need to update the function in one place.
  + **Parameters:** Functions can accept inputs, known as parameters, which affect how the function behaves or what output it produces.
  + **Return Values:** Functions can return a result after execution, which can then be used in other computations or operations.
* **Creating Functions:**

**Function Declaration:** This is the most common way to define a function. It starts with the function keyword, followed by the function name, parameters in parentheses, and the function body enclosed in curly braces.  
  
function greet(name) {

return `Hello, ${name}!`;

}

**Explanation:** The greet function takes a single parameter name and returns a greeting message. The ${name} syntax is used to interpolate the value of name into the string.

**Function Expression:** This method involves defining a function and assigning it to a variable. It can be anonymous (without a name) or named.

const greet = function(name) {

return `Hello, ${name}!`;

};

**Explanation:** Here, the function is assigned to the variable greet. This function behaves the same way as the function declaration but is often used for situations where functions are used as arguments or need to be assigned to variables.

* **Common Function Methods:**

**Calling Functions:** To execute a function, you use its name followed by parentheses. You can pass arguments inside the parentheses if the function requires them.  
console.log(greet('Alice')); // Output: Hello, Alice!

**Explanation:** The greet function is called with 'Alice' as an argument, so it returns "Hello, Alice!" which is then logged to the console.

**Returning Values:** Functions can return a value which can be used in further operations or computations.  
  
function add(a, b) {

return a + b;

}

console.log(add(5, 3)); // Output: 8

**Explanation:** The add function takes two parameters, a and b, adds them together, and returns the result. When add(5, 3) is called, it returns 8.

**Default Parameters:** Functions can have default values for parameters, which are used if no argument is provided for that parameter.  
javascript  
Copy code  
function multiply(a, b = 1) {

return a \* b;

}

console.log(multiply(5)); // Output: 5

**Explanation:** The multiply function has a default value of 1 for the parameter b. If only one argument is passed (5), b defaults to 1, and the result is 5 \* 1 = 5.