

Understanding JavaScript Functions

19/9/24

What Are Functions?

- Functions are reusable blocks of code that perform a specific task.
- They can take inputs (parameters) and can return outputs (return value).

```
function greet() {  
  console.log("Hello, world!");  
}
```

Function Declaration

- Syntax: function keyword followed by the function name and parentheses.
- The function body contains the code to be executed.

```
function sum(a, b) {  
    return a + b;  
}
```

Function Invocation (Calling a Function)

- A function is executed when it is called or invoked.
- xample: Invoking the sum function.

```
function sum(a, b) {  
    return a + b;  
}
```

```
let result = sum(5, 10);
```

Understanding Parameters and Arguments

- Parameters: Variables listed inside the function's parentheses.
- Arguments: Values passed to the function when it is invoked.

```
function multiply(x, y) {  
    return x * y;  
}  
  
let product = multiply(4, 5);
```

Return Statement

- A function can return a value using the return statement.
- -After the return statement, the function stops executing.

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

```
let result = square(4); // Output: 16
```

Arrow Functions

Arrow functions are anonymous and can be stored in variables.

```
const add = (a, b) => a + b;
```

```
console.log(add(3, 7)); // Output: 10
```

Function Scope

- Variables declared inside a function are in local scope and cannot be accessed outside.
- Global scope variables can be accessed from anywhere in the code.

```
function localScope() {  
    let message = "Hello";  
    console.log(message);  
}
```

```
// console.log(message); // Error: message is not defined
```


Class Task

- Write an arrow function that takes two numbers and returns their difference.
- Write a function that takes a string and returns the number of characters in the string. (**length**)

Let userInput = 'Hello World'

let lengthStr = userInput.**length**