

The background features abstract, overlapping geometric shapes in various shades of purple and blue, creating a modern, layered effect.

Course Module 1: Course Unit 6: JavaScript (Objects and Functions)

Week 8

JavaScript Functions

JavaScript Functions

- is a group of statements that perform specific tasks and can be kept and maintained separately from main program
- a block of code designed to perform a particular task
- executed when "something" invokes it (calls it)

Advantages of using functions

- Functions reduces the repetition of code within a program
- Functions makes the code much easier to maintain
- Functions makes it easier to eliminate the errors

Defining and Calling a Function

- The declaration of a function start with the *function* keyword, followed by the name of the function you want to create, followed by parentheses i.e. () and finally place your function's code between curly brackets {}

```
function functionName() {  
    // Code to be executed  
}
```

Defining and Calling a Function

Example

```
1 // Defining function
2 function sayHello() {
3     alert("Hello, welcome to this website!");
4 }
5
6 // Calling function
7 sayHello(); // Outputs: Hello, welcome to this website!
```

Note: A function name must start with a letter or underscore character not with a number, optionally followed by the more letters, numbers, or underscore characters. Function names are case sensitive, just like variable names.

Adding Parameters to Functions

- You can specify parameters when you define your function to accept input values at run time. The parameters work like placeholder variables within a function; they're replaced at run time by the values (known as argument) provided to the function at the time of invocation.
- Parameters are set on the first line of the function inside the set of parentheses

```
function functionName(parameter1, parameter2, parameter3) {  
    // Code to be executed  
}
```

Adding Parameters to Functions

Example

```
1  // Defining function
2  function displaySum(num1, num2) {
3      var total = num1 + num2;
4      alert(total);
5  }
6
7  // Calling function
8  displaySum(6, 20); // Outputs: 26
9  displaySum(-5, 17); // Outputs: 12
```


Adding Parameters to Functions

Example

```
1 // Defining function
2 function showFullname(firstName, lastName) {
3     alert(firstName + " " + lastName);
4 }
5
6 // Calling function
7 showFullname("Clark", "Kent"); // Outputs: Clark Kent
8 showFullname("John"); // Outputs: John undefined
```

Returning Values from a Function

- A function can return a value back to the script that called the function as a result using the return statement. The value may be of any type, including arrays and objects.
- The return statement usually placed as the last line of the function before the closing curly bracket and ends it with a semicolon

Returning Values from a Function

Example

```
1 // Defining function
2 function getSum(num1, num2) {
3     var total = num1 + num2;
4     return total;
5 }
6
7 // Displaying returned value
8 alert(getSum(6, 20)); // Outputs: 26
9 alert(getSum(-5, 17)); // Outputs: 12
```

Applications???

Any questions???

REFERENCES:

- JavaScript basics. (2022). [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/JavaScript basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics)
- JavaScript Tutorial. (2022). <https://www.javascripttutorial.net/>
- JavaScript Tutorial. (n.d.). <https://www.javatpoint.com/javascript-tutorial>
- JavaScript Tutorial. (n.d.). <https://www.tutorialrepublic.com/javascript-tutorial/>
- JavaScript Tutorial. (n.d.). <https://www.tutorialspoint.com/javascript/index.htm>
- JavaScript Tutorial. (n.d.). <https://www.w3schools.com/js/default.asp>