

Introduction to JavaScript

One first essential thing to know about Javascript is that the browser reads the code from top to bottom. This means that the order of the code is important. For example, if you try to access an element that hasn't been created yet, you will get an error. To avoid this, you can use the `window.onload` event to make sure that the code runs only after the page has been fully loaded. You can also use the `defer` attribute in the script tag to make sure that the code runs only after the page has been fully loaded.

```
<head>
  <title>JavaScript</title>
  <script src="script.js" defer></script>
</head>
```

Components of JavaScript

Variables

Variables are containers for storing data values. You can use the `var`, `let`, or `const` keywords to declare a variable. We also use the `;` to end the statement. Watch out the JavaScript is case-sensitive as opposed to HTML. The `let` and `var` keywords are used to declare variables that can be reassigned, while the `const` keyword is used to declare variables that cannot be reassigned.

```
let myVar;
myVar = 5;
// Or this can be done in one single line
let myVar = 5;
```

Operators

Most of the operators are the same as in other programming languages. The main difference lies in the `==` and `===` operators. The `===` operator is used to compare both the value and the type of the operands.

Actions

Conditional statements

We can use the `if`, `else if`, and `else` statements to perform different actions based on different conditions.

```
if (condition1) {
  // code to be executed if condition1 is true
} else if (condition2) {
  // code to be executed if condition2 is true
} else {
```

```
// code to be executed if neither condition1 nor condition2 is true  
}
```

Functions

We can define a function using the `function` keyword. We can also use the `return` statement to return a value from a function.

```
function multiply(num1, num2) {  
  let result = num1 * num2;  
  return result;  
}
```

Querying elements

We can select elements using the `document.querySelector()` method. This method returns the first element that matches the specified selector. If no matches are found, it returns `null`.

```
const element = document.querySelector('selector');
```

Then, we can change the text content of the element using the `textContent` property.

```
element.textContent = 'new content';
```

Event listeners

We can use the `addEventListener()` method to add an event listener to an element. This method takes two arguments: the event type and the function to be executed when the event occurs.

```
element.addEventListener('click', () => {  
  // code to be executed when the element is clicked  
});
```

We can also use the `onclick` property to assign a function to be executed when the element is clicked.

```
element.onclick = () => {  
  // code to be executed when the element is clicked  
};
```

Or we could make the action assignment in the HTML file.

```
<script src="script.js" defer></script>  
...  
<button onclick="myFunction()">Click me</button>
```