JavaScript If Else – Syntax & Examples

JavaScript If Else

JavaScript If Else is used to implement conditional programming. In this tutorial, we shall learn following statements related to JavaScript If Else.

- If statement
- If-else statement
- if-else-if statement
- Nested If-else

JavaScript If

It is used to conditionally execute a set of statements.

Syntax of if statement

```
if(expression){
    // set of statements
  }
```

Explanation : If expression is true, then set of statements are executed. Else execution continues with the statements after if-statement.

Example

```
var today='Sunday';
  if(today=='Sunday'){
    document.getElementById("message").innerHTML = "Today is Sunday.";
  }
```

Try Online

JavaScript If-Else

It is an extension to Javascript If statement. When the condition is false, another set of statements are executed.

It is an extension to Javascript if statement. When the condition is false, another set of statements are executed.

Syntax of if-else statement

```
if(expression){
    // set of statements
  } else{
    // another set of statements
  }
```

Explanation : If expression is true, then set of statements are executed. Else another set of statements are executed. Either of the two sets of statements shall be executed for sure based on the condition. Execution continues with the statements after **if-else** statement.

Example

```
var today='Monday';
  if(today=='Sunday'){
    document.getElementById("message").innerHTML = "Today is Sunday.";
  } else{
    document.getElementById("message").innerHTML = "Today is not Sunday. You have to go
  }
```

Try Online

JavaScript If-Else-If

It is an extension to Javascript If-Else statement. Instead of a single condition, there are multiple conditions.

Syntax of if-else-if statement

```
if(expression){
    // set of statements
  } else if(expression_2){
    // another set of statements
  } else if(expression_3){
    // another set of statements
  } else{
    // default set of statements
  }
```

Explanation : In a sequential order, from top to bottom, the block of statements for which the condition is true is executed. Once the control comes across a condition that is true, rest of the **else if** blocks are skipped.

There could be as many number of **else if** blocks as required.

There could be as many number of else if blocks as required.

else block is optional. When no condition is satisfied, **else** is executed.

Example

```
<script>
  var today='Tuesday';
  if(today=='Sunday'){
    document.getElementById("message").innerHTML = "Today is Sunday.";
  } else if(today=='Monday'){
    document.getElementById("message").innerHTML = "Today is Monday. You had a wonderful
  } else if(today=='Tuesday'){
    document.getElementById("message").innerHTML = "Today is Tuesday. Four more days for
  } else{
    document.getElementById("message").innerHTML = "Hang tight in there. You are about t
  }
</script>
```

Try Online

JavaScript Nested If

JavaScript If statements could be nested. If statement is like any other JavaScript statement. So it could be one of the 'set of statements' inside another if block.

Syntax of nested if statement

```
if(expression_1){
    // set of statements
    if(expression_2){
      // another set of statements
    }
  }
```

Explanation : If expression_1 is true, control enters the first if block and executes the set of statements. Next if condition is evaluated.

Example

```
var today='Monday';
  var holiday = true;
  if(today=='Monday'){
    document.getElementById("message1").innerHTML = "Today is Monday.";
    if(holiday){
      document.getElementById("message2").innerHTML = "Its long weekend. There is more f
```

```
    }
}
```

Try Online

Conclusion

In this JavaScript Tutorial, we have learnt about JavaScript If, JavaScript If-Else, JavaScript If-Else-If, JavaScript Nested If with Syntax and Examples.

JavaScript

- JavaScript Tutorial
- JavaScript Primitive Datatypes
- **JavaScript - If Else**
- JavaScript - Switch
- JavaScript - Arrays
- JavaScript - forEach

JavaScript - HTML Element

- JavaScript - Access HTML Element by ID
- JavaScript - Change style of HTML Element
- JavaScript - Date
- JavaScript - Date Methods

JavaScript - Fixing Errors

- JavaScript SyntaxError: Invalid shorthand property initializer