

DHTML

UNIT-4

# Creating Dynamic and Interactive Web Page

Prof. Ankit Bhavsar

Prof. Nirav Suthar

Prof. Mansi Mehta

- ❑ Conditional Checking
- ❑ If – then -else
- ❑ For loop
- ❑ While loop
- ❑ Arrays
- ❑ String Object
- ❑ Math Object
- ❑ Date Object

# Conditional Checking

- Conditional statements are used to perform different actions based on different conditions.
- In JavaScript we have the following conditional statements:
  - Use **if** to specify a block of code to be executed, if a specified condition is true
  - Use **else** to specify a block of code to be executed, if the same condition is false
  - Use **else if** to specify a new condition to test, if the first condition is false
  - Use **switch** to specify many alternative blocks of code to be executed.

# The if Statement

- Use the **if statement** to specify a block of JavaScript code to be executed if a condition is true.
- Syntax:

*if (condition)*

*{*

*block of code to be executed if the condition is true*

*}*

# The else Statement

- Use the **else statement** to specify a block of code to be executed if the condition is false.

- Syntax:

*if (condition)*

*{*

*block of code to be executed if the condition is true*

*}*

*else*

*{*

*block of code to be executed if the condition is false*

*}*

# The else if Statement

- Use the **else if statement** to specify a new condition if the first condition is false.
- Syntax:

*if(condition1) {*

*block of code to be executed if condition1 is true*

*}*

*else if(condition2) {*

*block of code to be executed if the condition1 is false and  
    condition2 is true*

*} else {*

*block of code to be executed if the condition1 is false and  
    condition2 is false*

*}*

# The Switch Statement

- Use the **switch** statement to select one of many blocks of code to be executed.
- Syntax:

*switch(expression) {*

*case n:*

*code block*

*break;*

*case n:*

*code block*

*break;*

*default:*

*default code block*

*}*

# The loop Statement

- Loops can execute a block of code a number of times.
- Loops are handy, if you want to run the same code over and over again, each time with a different value.
- JavaScript supports different kinds of loops:
  - **for** - loops through a block of code a number of times
  - **while** - loops through a block of code while a specified condition is true
  - **do/while** - also loops through a block of code while a specified condition is true.
  - **for/in** - loops through the properties of an object



# The for loop Statement

- Loops can execute a block of code a number of times.
- The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```

- **Statement 1** is executed before the loop (the code block) starts.
- **Statement 2** defines the condition for running the loop (the code block).
- **Statement 3** is executed each time after the loop (the code block) has been executed.

# The While loop Statement

- The while loop loops through a block of code as long as a specified condition is true.
- The while loop has the following syntax:

```
while (condition) {  
    code block to be executed  
}
```

- **Statement 1** is executed before the loop (the code block) starts.
- **Statement 2** defines the condition for running the loop (the code block).
- **Statement 3** is executed each time after the loop (the code block) has been executed.

# Array

- Array are Java Script objects that are capable of storing a sequence of values.
- The length of an array is the number of elements that an array contains.
- The individual elements of an array are accessed by using the name of the array followed by the *index value*.

# Array

- Create an array, and assign values to it:
- Syntax:

*arrayName = new Array(Array length)*

*Or*

*arrayName = new Array()*

- Example:

*Cust\_name = new Array(2);*

*Cust\_name[0] = "Hitesh";*

*Cust\_name[1] = "Atul";*

*or*

*var cars = new Array("Saab", "Volvo", "BMW");*

# Arrays are Objects

- Arrays are a special type of objects. The `typeof` operator in JavaScript returns "object" for arrays.

## Array Properties and Methods

- The real strength of JavaScript arrays are the built-in array properties and methods:
  - Length
  - Sort

# Adding Array Elements

- The easiest way to add a new element to an array is using the push method.
- New element can also be added to an array using the length property.

# Looping Array Elements

- The best way to loop through an array, is using a "for" loop.