Charpter-5

JavaScript Structure

JavaScript Structures:

JavaScript has three basic structures. The three structures are as

1. Sequences

2. Branches

3. Loops (iterative)

Compound Structure: A compound structure is simply a structure that uses two or more of the three basic structures.

Statement: A statement is any line of code in a JavaScript program is a statement. The keywords and combination of expressions or even expressions themselves can be statements.

Sequential Structure: Sequential Structures refer to the order in which the code is entered into a JavaScript program. In one sense, all code is sequential.

Variable Declaration: A simple statement that declares a variable using var is a variable declaration.

Function

A JavaScript function is a block of code designed to perform a particular task. A JavaScript function is executed when "something" invokes it (calls it).

Example:

function myFunction(p1, p2) {

return p1 \* p2;

}

Syntax

function name(parameter1, parameter2, parameter3) {

code to be executed

}

Function Invocation (Calls)

The code inside the function will execute when "something" invokes (calls) the function:

• When an event occurs (when a user clicks a button)

• When it is invoked (called) from JavaScript code

• Automatically (self-invoked)

Function Return

When JavaScript reaches a return statement, the function will stop executing. If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement. Functions often compute a return value. The return value is "returned" back to the "caller":

Example:

Calculate the product of two numbers, and return the result:

var x = myFunction(4, 3);

function myFunction(a, b) {

return a \* b

}

JavaScript If...Else Statements

Conditional statements are used to perform different actions based on different conditions.

Conditional Statements

Very often when you write code, you want to perform different actions for different decisions.

You can use conditional statements in your code to do this.

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

}

Example:

Make a "Good day" greeting if the hour is less than 18:00:

if (hour < 18) {

greeting = "Good day";

}

The else Statement

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

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

}

Example:

If the hour is less than 18, create a "Good day" greeting, otherwise "Good evening":

if (hour < 18) {

greeting = "Good day";

} else {

greeting = "Good evening";

}

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

}

Example:

If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening":

if (time < 10) {

greeting = "Good morning";

} else if (time < 20) {

greeting = "Good day";

} else {

greeting = "Good evening";

}

JavaScript Switch Statement

The switch statement is used to perform different actions based on different conditions.

The JavaScript Switch Statement:

Use the switch statement to select one of many code blocks to be executed.

Syntax

switch(expression) {

case n:

code block

break;

case n:

code block

break;

default:

code block

}

This is how it works:

• The switch expression is evaluated once.

• The value of the expression is compared with the values of each case.

• If there is a match, the associated block of code is executed.

Example:

The getDay() method returns the weekday as a number between 0 and 6.

(Sunday=0, Monday=1, Tuesday=2 ..)

This example uses the weekday number to calculate the weekday name:

switch (new Date().getDay()) {

case 0:

day = "Sunday";

break;

case 1:

day = "Monday";

break;

case 2:

day = "Tuesday";

break;

case 3:

day = "Wednesday";

break;

case 4:

day = "Thursday";

break;

case 5:

day = "Friday";

break;

case 6:

day = "Saturday";

}

The result of day will be:

Friday

Ternary Shortcut

The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement.

Different Kinds of Loops

JavaScript supports different kinds of loops:

• for - loops through a block of code a number of times

• for/in - loops through the properties of an object

• 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

Looping Statement in

Set of instructions given to the compiler to execute set of statements until condition becomes false is called loops. The basic purpose of loop is code repetition.

The way of the repetition will be forms a circle that's why repetition statements are called loops. Some loops are available In JavaScript which are given below.

• while loop

• for loop

• do-while

While loop

When we are working with while loop always pre-checking process will be occurred. Pre-checking process means before evolution of statement block condition part will be executed. While loop will be repeats in clock wise direction.

Syntax:

while (condition)

{

code block to be executed

}

Example of while loop

<script>

var i=10;

while (i<=13)

{

document.write(i + "<br/>");

i++;

}

</script>

Result

10

11

12

13

Do-while loop

In implementation when we need to repeat the statement block at least 1 then go for do-while. In do-while loop post checking of the statement block condition part will be executed.

syntax

do

{

code to be executed

increment/decrement

}

while (condition);

Example of do-while loop

Example

<script>

var i=11;

do{

document.write(i + "<br/>");

i++;

}while (i<=15);

</script>

Result

11

12

13

14

15

For Loop

For loop is a simplest loop first we initialized the value then check condition and then increment and decrements occurred.

Steps of for loop

Syntax

for (initialization; condition; increment/decrement)

{

code block to be executed

}

Example of for loop

Example

<script>

for (i=1; i<=5; i++)

{

document.write(i + "<br/>")

}

</script>

Result

1

2

3

4

5