

WMLScript *WirelessMarkupLanguageScript* is the client-side scripting language of WML *WirelessMarkupLanguage*. A scripting language is similar to a programming language, but is of lighter weight. With WMLScript, the wireless device can do some of the processing and computation. This reduces the number of requests and responses to/from the server.

This chapter will give brief description of all the important WML Script components.

WML Script Components:

WML Script is very similar to Java Script. Almost WML Script components have similar meaning as they have in Java Script. A WML Script program components are summarized as follows:

WML Script Operators:

WML Script supports following type of operators.

- Arithmetic Operators
- Comparison Operators
- Logical *or* Relational Operators
- Assignment Operators
- Conditional *or* ternary Operators

Check for complete detail of [The WML Operators](#).

WML Script Control Statements:

Control statements are used for controlling the sequence and iterations in a program.

Statement	Description
if-else	Conditional branching
for	Making self-incremented fixed iteration loop
while	Making variable iteration loop
break	Terminates a loop
continue	Quit the current iteration of a loop

Check for complete detail of [WML Script Control Statements](#).

WML Script Functions:

The user-defined functions are declared in a separate file having the extension .wmls. Functions are declared as follows:

```
function name (parameters)
{
    control statements;
    return var;
}
```

The functions used are stored in a separate file with the extension .wmls. The functions are called

as the filename followed by a hash, followed by the function name:

```
maths.wmls#suar()
```

WML Scripts Standard Libraries:

There are six standard libraries totally. Here is an overview of them:

- **Lang:** The Lang library provides functions related to the WMLScript language core.
Example Function: abs, abort, characterSet, float, isFloat, isInt, max, isMax, min, minInt, maxInt, parseFloat, parseInt, random, seed
- **Float:** The Float library contains functions that help us perform floating-point arithmetic operations.
Example Function: sqrt, round, pow, ceil, floor, int, maxFloat, minFloat
- **String:** The String library provides a number of functions that help us manipulate strings.
Example Function: length, charAt, find, replace, trim, compare, format, isEmpty, squeeze, toString, elementAt, elements, insertAt, removeAt, replaceAt
- **URL:** The URL library contains functions that help us manipulate URLs.
Example Function: getPath, getReferer, getHost, getBase, escapeString, isValid, loadString, resolve, unescapeString, getFragment
- **WMLBrowser:** The WMLBrowser library provides a group of functions to control the WML browser or to get information from it.
Example Function: go, prev, next, getCurrentCard, refresh, getVar, setVar
- **Dialogs:** The Dialogs library Contains the user interface functions.
Example Function: prompt, confirm, alert

WML Scripts Comments:

There are two types of comments in WMLScript:

- **Single-line comment:** To add a single-line comment, begin a line of text with the // characters.
- **Multi-line comment:** To add a multi-line comment, enclose the text within /* and */.

These rules are the same in WMLScript, JavaScript, Java, and C++. The WMLScript engine will ignore all comments. The following WMLScript example demonstrates the use of comments:

```
// This is a single-line comment.  
  
/* This is a  
   multi-line comment. */  
  
/* A multi-line comment can be placed on a single line. */
```

WML Script Case Sensitivity:

The WMLScript language is case-sensitive. For example, a WMLScript function with the name WMLScript Function is different from wmlscript function. So, be careful of the capitalization when defining or referring to a function or a variable in WMLScript.

Whitespaces in WMLScript:

Except in string literals, WMLScript ignores extra whitespaces like spaces, tabs and newlines. Hence, the code in the earlier "Hello World" example can be typed in the following way and the

result will remain the same:

WML Script Statement Termination by Semicolons:

A semicolon is required to end a statement in WMLScript. This is the same as C++ and Java. Note that JavaScript does not have such requirement but WML Script makes it mandatory.

Loading [MathJax]/jax/output/HTML-CSS/jax.js