Console

The Console module provides you with an interface to log JavaScript actions. Logs are generated in Log4J-compliant TTCC format using the Firebug API.

Examples:

```
console.log("The %s jumped over %d tall buildings", animal, count);

console.log("The", animal, "jumped over", count, "tall buildings");
```

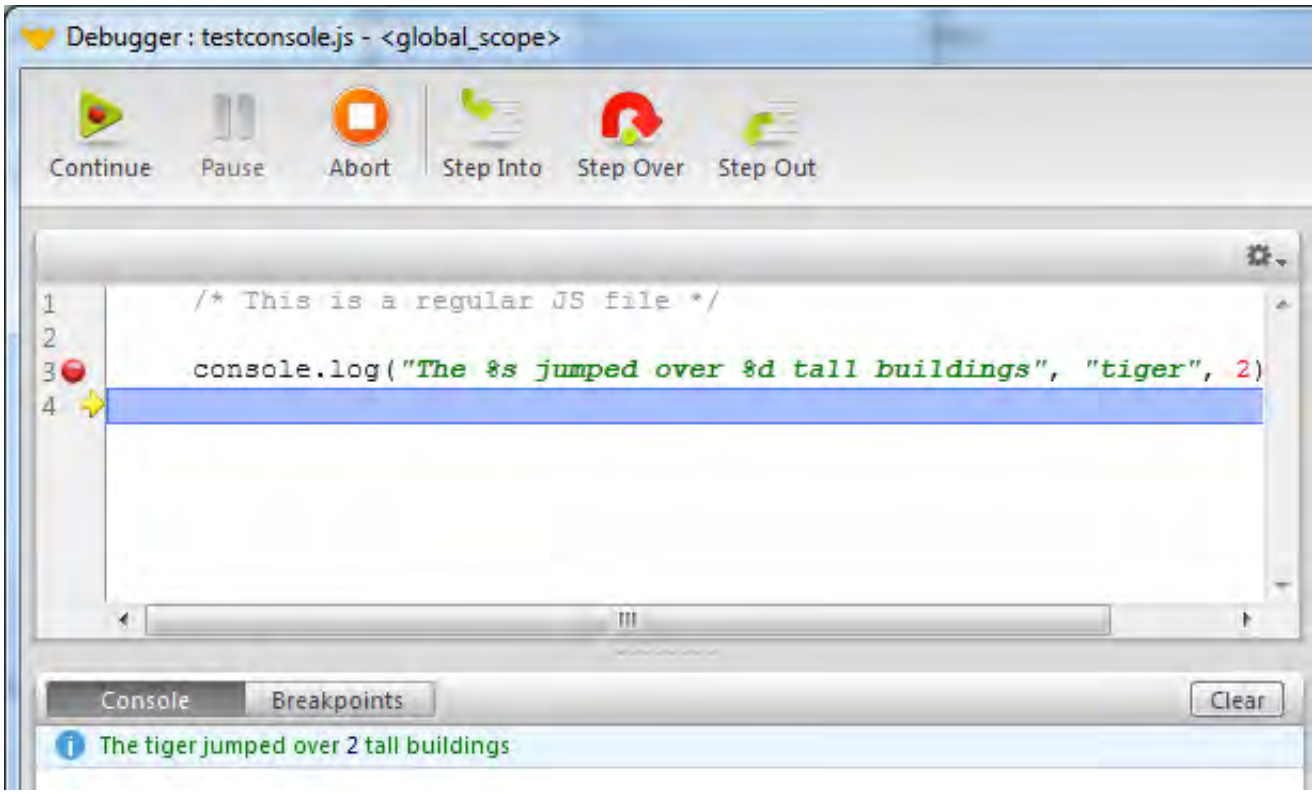
Here is the complete set of patterns that you may use for string substitution:

String Substitution Patterns

%s	String
%d, %i	Integer
%f	Floating point number
%o	Object hyperlink

The **console** statement is a property in the Application global object. For more information, refer to the **console** property in the **Application** section.

Logs are recorded in the solution's log file and also displayed in the Debugger's Console area:



Console Class

content

Description

The **content** property returns the last logged messages in an *Array* of strings.

Example

```
var lastLogs = console.content.join('\n');
```

error( )

void **error**(Object *message*)

Parameter	Type	Description
message	Object	The message or value to log

Description

The **error( )** method writes *message* to the log file and Wakanda Studio Debugger's Console with the visual "ERROR" label.

Example

```
console.error("Action failed:", action);
```

This code would display the following line in the debugger's console:

info( )

void **info**(Object *message*)

Parameter	Type	Description
message	Object	The message or value to log

Description

The **info( )** method writes *message* to the log and Wakanda Studio Debugger's Console with the visual "INFO" label.

Example

```
console.info("Product updated:", product.name);
```

This example would display the following line in the debugger's console:

log( )

void **log**(Object *message*)

Parameter	Type	Description

message	Object	The message or value to log

Description

The **log( )** method writes *message* to the log file and Wakanda Studio Debugger's console with the visual "TRACE" label.

This method accepts any number of arguments that will be joined by a space-delimited line. The first argument to log may be a string containing printf-like string substitution patterns.

Example

```
console.log("The %s jumped over %d tall buildings", animal, count);
```

The example above can be re-written without string substitution to achieve the same result:

```
console.log("The", animal, "jumped over", count, "tall buildings");
```

This example could display the following line in the debugger's console:

warn( )

void **warn**(Object *message*)

Parameter	Type	Description
message	Object	The message or value to log

Description

The **warn( )** method writes *message* to the log file and Wakanda Studio's Debugger console with the visual "WARNING" label.

Example

```
console.warn("File not found:", path);
```

This example could display the following line in the debugger's console: