

# Console

---

## Console Class

---

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.

This module contains a unique class, named `Console`.

### content

---

#### Description

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

#### Example

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

### debug()

---

void **debug**( Object *message* )

Parameter	Type	Description
message	Object	The message or value to log

#### Description

The `debug()` method writes a message to the log and Web admin console with the visual "TRACE" label.

#### Example

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

### error()

---

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

Parameter	Type	Description
message	Object	The message or value to log

#### Description

The `error()` method writes a message to the log and Web admin console with the visual "ERROR" label.

#### Example

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

### info()

---

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

Parameter	Type	Description
message	Object	The message or value to log

## Description

The `info()` method writes a message to the log and Web admin console with the visual "INFO" label.

## Example

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

## log()

---

```
void log( Object message )
```

Parameter	Type	Description
message	Object	The message or value to log

## Description

The `log()` method writes a *message* to the log and Web admin 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");
```

## warn()

---

```
void warn( Object message )
```

Parameter	Type	Description
message	Object	The message or value to log

## Description

The `warn()` method writes a *message* to the log and Web admin console with the visual "WARNING" label.

## Example

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