

The Concerto Script Quick Reference Guide

Version 4.0

Script Tag

```
<script language="Javascript" runat="server">
.....
</script>
```

Object Model Legend

<name>	indicates that an object has a collection of named properties.
[index]	indicates that the properties of an object can be retrieved using array index notation.
{name}	indicates that the property is only present under certain conditions.
name[]	indicates that a property is an array.
name{}	indicates an object model property that holds properties of no pre defined object structure.
name<>	indicates an object model property that holds a structured collection of properties.
name()	indicates a function.
<name>()	indicates that an object has a collection of named functions.
(name)	indicates and optional argument.
([..])	indicates a number of additional optional arguments.
(num)	indicates a numeric argument.
(str)	indicates a string argument.
(bool)	indicates a boolean argument.
(obj)	indicates an object argument.
(func)	indicates a function argument.
(prop)	indicates a property name argument.
(jobj)	indicates a java object argument

A line linking a property or a function to an object indicates that the property is an object of that type or that the function returns an object of that type.

Prototype properties that do not have a link to an object refer to the Object Prototype.

Operators		Statements
<p>Arithmetic Operators</p> <p>+ Addition</p> <p>++ Increment</p> <p>- Subtraction, Unary negation</p> <p>-- Decrement</p> <p>* Multiplication</p> <p>/ Division</p> <p>% Modulus</p> <p>String Operators</p> <p>+ Concatenation</p> <p>+= Concatenate to value</p> <p>Logical Operators</p> <p>&& AND</p> <p> OR</p> <p>! Negation</p> <p>Bitwise Operators</p> <p>& AND</p> <p>^ XOR</p> <p> OR</p> <p>~ NOT</p> <p><< Left shift</p> <p>>> Sign-propagating right shift</p> <p>>>> Zero-fill right shift</p>	<p>Assignment Operators</p> <p>= Equate</p> <p>+= Add to value</p> <p>-= Subtract to value</p> <p>*= Multiply to value</p> <p>/= Divide to value</p> <p>%= Modulus to value</p> <p>&= Bitwise AND to value</p> <p>^= Bitwise XOR to value</p> <p> = Bitwise OR to value</p> <p><<= Left shift to value</p> <p>>>= Sign-propagating right shift to value</p> <p>>>>= Zero-fill right shift to value</p> <p>Comparison Operators</p> <p>== Equals</p> <p>!= Does not equal</p> <p>> Is greater than</p> <p>>= Is greater than or equal to</p> <p>< Is less than</p> <p><= Is less than or equal to</p> <p>Special Operators</p> <p>var Variable definition</p> <p>. Variable property reference</p> <p>[num] Array reference</p> <p>[name] Indirect reference</p> <p>new Object creation</p> <p>this Current object reference</p>	<p>;</p> <p>Defines the end of a statement (new line is also defined as the end of a statement).</p> <p>{a;b...}</p> <p>Defines a block of zero or more statements.</p> <p>break</p> <p>Terminate the current loop.</p> <p>continue</p> <p>Terminate the current loop iteration.</p> <p>for(a;b;c) {x; y..}</p> <p>Loop, a is the initial statement, b is the stop condition, c is the end iteration statement. Supports single statement or statement block.</p> <p>for(a in b) {x; y..}</p> <p>Loop for all properties of b, a is set to the property name at the start of each iteration. Supports single statement or statement block.</p> <p>if(a) {d; e..} else {f; g..}</p> <p>Conditional execution. Execute d/e statements if a is true or f/g statements if a is false. Supports single statements or statement blocks.</p> <p>if(a) {d; e..} else if(b) {f; g..} else {h; i..}</p> <p>Conditional execution. Execute d/e statements if a is true or f/g statements if c is true or h/i statements. Can have multiple 'else if' clauses. Supports single statements or statement blocks.</p> <p>x = a?b:c</p> <p>Conditional assignment. If a is true set x to b, if a is false set x to c.</p> <p>while(a) {x; y..}</p> <p>Loop, a is the stop condition. Supports single statement or statement block.</p> <p>function(a..) {x; y..}</p> <p>Defines a statement block as a function, a is zero or more parameters.</p> <p>return a</p> <p>Return from a function, a is zero or one return parameter.</p>

Built In Script Objects

Global

Infinity NaN undefined	createSession() decodeURI(str) decodeURIComponent(str) encodeURI(str) encodeURIComponent(str) exit() getTmpDir() isFinite(num) isNaN(num) parseInt(str[, num]) parseFloat(str) parseQuery(str) redirect(string[, bool]) forward(string) toArray(obj) write(str) writeln(str)
------------------------------	--

Math Prototype

E LN10 LN2 LOG10E LOG2E PI SQRT1_2 SQRT2 prototype	abs(num) asin(num) atan2(num, num) cos(num) floor(num) max(num[, num[...]]) pow(num, num) round(num) sqrt(num)	acos(num) atan(num) ceil(num) exp(num) log(num) min(num[, num[...]]) random() sin(num) tan(num)
--	--	---

response

contentType binaryMode isCommitted cookies<> Prototype setBinaryMode() getBinaryMode() isCommitted()

Math

prototype

cookie

domain expires path prototype secure
--

request

cookies<> country http{} jsession language prototype queryString scriptName serverName <name>
--

Locking Prototype

prototype	lock() unlock()
-----------	--------------------

project

name path prototype <name>	expire(prop, num) reload()
-------------------------------------	-------------------------------

encoding

currentValue {error} includeCharset prototype supportedEncodings[]
--

Basic Script Objects

String Object

length prototype

String Constructor

prototype	fromCharCode(num[...])
-----------	------------------------

String Prototype

prototype	charAt(num) charCodeAt(num) compareTo(str) concat([str[...]]) indexOf(string[, num]) lastIndexOf(string[, num]) localeCompare(str) slice(num[, num]) split(string[, num]) substring(num[, num]) toLocaleLowerCase() toLowerCase() toLocaleUpperCase() toUpperCase() valueOf()
-----------	---

Number Object

prototype

Number Constructor

MAX_VALUE MIN_VALUE NaN NEGATIVE_INFINITY POSITIVE_INFINITY prototype
--

Number Prototype

prototype	toFixed([num]) toExponential([num]) toLocaleString() toPrecision([num]) toString() valueOf()
-----------	---

Boolean Object

prototype

Boolean Constructor

prototype

Boolean Prototype

prototype	toString() valueOf()
-----------	-------------------------

Object Object

prototype

Object Constructor

prototype

Object Prototype

getAttributes() hasOwnProperty(str) isPrototypeOf(obj) propertyIsEnumerable(str) setAttributes(obj) toLocaleString() toString() valueOf()
--

Basic Script Objects

Date Object

prototype

Date Constructor

prototype parse(str[, str[, str]])
UTC(num[, num[, num[, num[, num[, num[, num]]]]]])

Date

prototype getDate() getDay()
getFullYear() getHours()
getMilliseconds() getMinutes()
getMonth() getSeconds()
getTime() getTimezoneOffset()
getYear() getUTCDate()
getUTCDay() getUTCFullYear()
getUTCHours() getUTCMilliseconds()
getUTCMinutes() getUTCMonth()
getUTCSeconds()
setDate(num)
setFullYear(num[, num[, num]])
setHours(num[, num[, num[, num]]])
setMilliseconds(num)
setMinutes(num[, num[, num]])
setMonth(num[, num])
setSeconds(num[, num])
setTime(num)
setYear(num[, num[, num]])
setUTCDate(num)
setUTCFullYear(num[, num[, num]])
setUTCHours(num[, num[, num[, num]]])
setUTCMilliseconds(num)
setUTCMinutes(num[, num[, num]])
setUTCMonth(num[, num])
setUTCSeconds(num[, num])
toDateString()
toGMTString()
toLocaleDateString(null[, str[, str]])
toLocaleString(null[, str[, str]])
toLocaleTimeString(null[, str[, str]])
toInternetTime() toTimeString()
toString() toUTCString()
valueOf()

Array Object

isArray
length
prototype

Array Constructor

prototype

Array Prototype

prototype concat(obj[, obj[...]])
join(str)
pop()
push(obj[, obj[...]])
reverse()
shift()
slice(num, num)
sort(func[, args[...]])
splice(num, num[, obj[...]])
toLocaleString()
toString()
unshift(obj[, obj[...]])

Email

bcc[]
cc[]
contentType
{error}
from
headerText
message
port
prototype
server
subject
timeout
to
attachments[]

Email Constructor

prototype

Script Document Script Objects

Library Object (Deprecated)

name init()
prototype <name>
<name>

Library Constructor

getPrototype(str)
setPrototype(str, obj)

Library Prototype

prototype toString()

Fragment Object

<name> execute()
prototype <name>()

Fragment Constructor

getPrototype(str)
setPrototype(str, obj)
setPrototypeProperty(str, str, obj)

Fragment Prototype

prototype toString()

InternationalFragment Constructor

getPrototype(str)
setPrototype(str, obj)
setPrototypeProperty(str, str, obj)

Function Object

length name>()
prototype

Function Prototype

prototype toString()

File Script Objects

File Constructor

prototype	getRootPath()
-----------	---------------

File Object

prototype	canRead()	canWrite()
	del()	exists()
	getJavaPath()	getName()
	getParent()	getParentFile()
	getPath()	isDirectory()
	isFile()	lastModified()
	length()	list()
	listFiles([func])	
	setLastModified(num)	
	setReadOnly()	

FileReader Constructor

prototype

FileReader Object

{error}	close()
prototype	exists()
	read(num)
	readln()

FileWriter Constructor

prototype

FileWriter Object

{error}	close()
prototype	write(str)
	writeln(str)

BinaryFile Constructor

prototype

BinaryFile Object

{error}	close()
prototype	del(string)
	eof()
	exists()
	getLength()
	getPosition()
	lastModified()
	readByte()
	setPosition(num[, num])
	writeByte(num)

ByteString Prototype

prototype	byteToString(num)
	stringToByte(str)

ObjectFile Constructor

prototype

ObjectFile Object

{error}	copyTo(str)
<name>	merge(obj)
prototype	remove(str[.])
	write()

Locking prototype

prototype	lock()
	unlock()

InternationalTokens Constructor

prototype

I18nTokens Prototype

prototype	read([prop[, str]])
	format(obj, obj)

Properties Constructor

prototype

Properties Object

{error}	fileName
prototype	<name>

Properties Prototype

prototype	getFileName()
	read()
	remove(str)
	setFileName(str)
	write([str])

XML Objects

XMLWriter Constructor

prototype

XMLWriter Object

{error}	addHandler(str, func[, arg[.]])
prototype	toFile(str, obj)
	toString(obj)
	removeHandler(str)

XMLParser Constructor

prototype

XMLParser Object

{error}	addHandler(str, func[, arg[.]])
ignoreWhiteSpace	parseFile(str[, str])
prototype	parseURL(str[, str])
om	parseString(str[, str])
	removeHandler(str)

XML Template Object

{error}	newTransform()
prototype	

XML Transform Object

{error}	transformFile(file)
prototype	transformString(string)
	transformURL(url)

Servlet Script Objects

Servlet Object

info context params accessCount loadErrorCount runtimeErrorCount notFoundCount	log(str)
--	----------

Context Object

info params version	getAttribute(str) getAttributeNames() getMimeType(str) getRealPath(str) removeAttribute(str) setAttribute(str, jobj)
---------------------------	---

Session Object

creationTime id isNew	getAttribute(str) getAttributeNames() getLastAccessedTime() getMaxInactiveInterval() removeAttribute(str) setAttribute(str, jobj) setMaxInactiveInterval(num) invalidate()
-----------------------------	---

Enterprise Objects

Task Constructor

prototype

Task Object

disableOutput collectReports delimiter delimiterSplit {error} parameters{ prototype reportOutput<> taskName	execute() nextReport()
---	---------------------------

Reality Constructor

prototype	fromArray(obj) toArray(str)
-----------	--------------------------------

Reality Object

{error}	<name()>
---------	----------

Context Environment Object

initial objectFactories stateFactories urlPkgs providerUrl dnsUrl authoritative batchsize referral secProtocol secAuthentication secPrincipal secCredentials language applet [index]	
---	--

Directory Object

{error}	lookup(name) close()
---------	-------------------------

Database Objects

Database Constructor

prototype

Database Object

{error} prototype	beginTransaction() close() cursor(str) endTransaction() execute(str) rollback() preparedStatement()
----------------------	---

PreparedStatement Object

{error}	close() execute() getMoreCursors() getCursor() getUpdateCount() clearParams() setDate() setFloat() setLong() setString()	setAsciiStream() setBigDecimal() setBlob() setBoolean() setCharacterStream() setClob() setDouble() setInt() setShort() setTime() setTimestamp()
---------	---	---

Cursor Object

<name> [index] prototype	close() columnCount() columnName(num) moveTo(num) next() setRange(num[, num])
--------------------------------	--



Clob / Blob Objects

length	
--------	--

