

Statements

 Apr 03, 2024 13:32

Supported Clients

SmartClient WebClient NGClient MobileClient

Methods Summary		
void	<code>break()</code>	Break statement exits a loop.
void	<code>const()</code>	Constant declaration.
void	<code>continue()</code>	Continue statement, jumps to next iteration of the loop.
void	<code>do while()</code>	do while loop
void	<code>for()</code>	for loop
void	<code>for each in()</code>	foreach loop
void	<code>if()</code>	If statement
void	<code>if else()</code>	If/Else statement.
void	<code>label()</code>	Provides a statement with an identifier that you can refer to using a break or continue statement.
void	<code>switch()</code>	Switch statement.
void	<code>try catch()</code>	try/catch statement
void	<code>try catch finally()</code>	try/catch/finally statement
void	<code>var()</code>	Variable declaration
void	<code>while()</code>	while loop

Methods Details

break()
Break statement exits a loop.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
break
```

const()
Constant declaration.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
const #;
```

continue()
Continue statement, jumps to next iteration of the loop.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
continue
```

do while()
do while loop

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
do
{
}
while ( # )
```

for()

for loop

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
for ( var i = 0 ; i < # ; i++ )
{
}
```

for each in()

foreach loop

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
for ( var item in obj )
{
}
```

if()

If statement

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
if ( # )
{
}
```

if else()

If/Else statement.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
if ( # )
{
}
else
{
}
```

label()

Provides a statement with an identifier that you can refer to using a break or continue statement.

For example, you can use a label to identify a loop, and then use the break or continue statements to indicate whether a program should interrupt the loop or continue its execution.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```

var i = 0, j;
outer_loop: while (i < 10) {
    i++;
    j = 0;
    while (j < 10) {
        j++;
        if (j > i) continue outer_loop;
        application.output("i=" + i + ", j=" + j);
    }
}

```

switch()

Switch statement.

Supported Clients

SmartClient, WebClient, NGClient, MobileClient

Sample

```

switch( # )
{
case:
default:
}

```

try catch()

try/catch statement

Supported Clients

SmartClient, WebClient, NGClient, MobileClient

Sample

```

try
{
    #
}
catch( # )
{
    #
}

```

try catch finally()

try/catch/finally statement

Supported Clients

SmartClient, WebClient, NGClient, MobileClient

Sample

```

try
{
    #
}
catch( # )
{
    #
} finally
{
    #
}

```

var()

Variable declaration

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var #;
```

while()

while loop

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
while ( # )  
{  
    #  
}
```