JSON

Method Summary

Object	parse(text) Parses a string as JSON and returns the parsed value.
Object	parse(text, reviver) Parses a string as JSON and returns the parsed value.
String	stringify(value) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified
String	stringify(value, replacer) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer, space) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer, space) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer, space) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer, space) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer, space) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
String	stringify(value, replacer, space) Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.

Method Details

parse

Object **parse** (text)

Parses a string as JSON and returns the parsed value.

Parameters

{String} text - The string to parse as JSON. See the JSON object for a description of JSON syntax.

Returns

Object

Sample

```
JSON.parse('[1, 5, "false"]');
```

parse

Object **parse** (text, reviver)

Parses a string as JSON and returns the parsed value.

Parameters

{String} text - The string to parse as JSON. See the JSON object for a description of JSON syntax.
{Function} reviver - A function, prescribes how the value originally produced by parsing is transformed, before being returned.

Returns

Object

Sample

```
var transformed = JSON.parse('{"p": 5}', function(k, v) { if (k === "") return v; return v * 2; });
```

stringify

String **stringify** (value)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified

Parameters

{Object} value - The value to convert to a JSON string.

Returns

String

Sample

```
JSON.stringify([1, "false", false])
```

stringify

String **stringify** (value, replacer)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{Function} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.

Returns

String

Sample

```
function censor(key, value) {
 if (typeof(value) == "string") {
   return undefined;
 }
 return value;
}

var foo = {foundation: "Mozilla", model: "box", week: 45, transport: "car", month: 7};
var jsonString = JSON.stringify(foo, censor);
```

stringify

String **stringify** (value, replacer, space)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{Function} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.
{Number} space - The space argument may be used to control spacing in the final string (causes the resulting string to be pretty-printed). If it is a number, successive levels in the stringification will each be indented by this many space characters (up to 10). If it is a string, successive levels will indented by this string (or the first ten characters of it).

Returns

String

Sample

```
JSON.stringify({ uno: 1, dos : 2 }, null, '\t')
```

stringify

String **stringify** (value, replacer, space)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{Function} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.
{String} space - The space argument may be used to control spacing in the final string (causes the resulting string to be pretty-printed). If it is a number, successive levels in the stringification will each be indented by this many space characters (up to 10). If it is a string, successive levels will indented by this string (or the first ten characters of it).

Returns

String

Sample

```
JSON.stringify({ uno: 1, dos : 2 }, null, '\t')
```

stringify

String **stringify** (value, replacer)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{Number[]} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.

Returns

String

Sample

```
function censor(key, value) {
 if (typeof(value) == "string") {
   return undefined;
 }
 return value;
}

var foo = {foundation: "Mozilla", model: "box", week: 45, transport: "car", month: 7};
var jsonString = JSON.stringify(foo, censor);
```

stringify

String **stringify** (value, replacer, space)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{Number[]} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.
{Number} space - The space argument may be used to control spacing in the final string (causes the resulting string to be pretty-printed). If it is a number, successive levels in the stringification will each be indented by this many space characters (up to 10). If it is a string, successive levels will indented by this string (or the first ten characters of it).

Returns

String

Sample

```
JSON.stringify({ uno: 1, dos : 2 }, null, '\t')
```

stringify

String **stringify** (value, replacer, space)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{Number[]} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.
{String} space - The space argument may be used to control spacing in the final string (causes the resulting string to be pretty-printed). If it is a number, successive levels in the stringification will each be indented by this many space characters (up to 10). If it is a string, successive levels will indented by this string (or the first ten characters of it).

Returns

String

Sample

```
JSON.stringify({ uno: 1, dos : 2 }, null, '\t')
```

stringify

String **stringify** (value, replacer)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{String[]} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.

Returns

String

Sample

```
function censor(key, value) {
 if (typeof(value) == "string") {
   return undefined;
 }
 return value;
}

var foo = {foundation: "Mozilla", model: "box", week: 45, transport: "car", month: 7};
var jsonString = JSON.stringify(foo, censor);
```

stringify

String **stringify** (value, replacer, space)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{String[]} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.
{Number} space - The space argument may be used to control spacing in the final string (causes the resulting string to be pretty-printed). If it is a number, successive levels in the stringification will each be indented by this many space characters (up to 10). If it is a string, successive levels will indented by this string (or the first ten characters of it).

Returns

String

Sample

```
JSON.stringify({ uno: 1, dos : 2 }, null, '\t')
```

stringify

String **stringify** (value, replacer, space)

Convert a value to JSON, optionally replacing values if a replacer function is specified, or optionally including only the specified properties if a replacer array is specified.
As a function, the replacer takes two parameters, the key and the value being stringified. Initially it gets called with an empty key representing the object being stringified,
and it then gets called for each property on the object or array being stringified.

Parameters

{Object} value - The value to convert to a JSON string.
{String[]} replacer - If a function, transforms values and properties encountered while stringifying; if an array (of String or Number), specifies the set of properties included in objects in the final string.
{String} space - The space argument may be used to control spacing in the final string (causes the resulting string to be pretty-printed). If it is a number, successive levels in the stringification will each be indented by this many space characters (up to 10). If it is a string, successive levels will indented by this string (or the first ten characters of it).

Returns

String

Sample

```
JSON.stringify({ uno: 1, dos : 2 }, null, '\t')
```