Object

Feb 20, 2024 03:02

Supported Clients

SmartClient WebClient NGClient MobileClient

Methods Summary

Boolean	hasOwnProperty(prop)	Determine whether the object has the specified property as its own property (as opposed to inheriting it).
Boolean	isPrototypeOf(object)	Checks if an object exists in another object's prototype chain.
Boolean	propertyIsEnumerable(prop)	Indicates whether the specified property is enumerable.
String	toLocaleString()	Returns a string representing the object.
String	toString()	Returns a string representing the specified object.
Object	valueOf()	Returns the primitive value of the specified object.

Methods Details

hasOwnProperty(prop)

Determine whether the object has the specified property as its own property (as opposed to inheriting it).

Parameters

String prop The name of the property to test.

Returns

Boolean A Boolean indicating whether or not the object has the specified property as own property.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
const object1 = new Object();
 object1.property1 = 42;
 application.output(object1.hasOwnProperty('property1')); // expected output: true
 application.output(object1.hasOwnProperty('toString')); // expected output: false
 application.output(object1.hasOwnProperty('hasOwnProperty')); // expected output: false
```

isPrototypeOf(object)

Checks if an object exists in another object's prototype chain.

Parameters

Object object The object whose prototype chain will be searched.

Returns

Boolean A Boolean indicating whether the calling object lies in the prototype chain of the specified object.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function object1() {}
function object2() {}
object1.prototype = Object.create(object2.prototype);
const object3 = new object1();
application.output(object1.prototype.isPrototypeOf(object3)); // expected output: true
application.output(object2.prototype.isPrototypeOf(object3)); // expected output: true
```

propertyIsEnumerable(prop)

Indicates whether the specified property is enumerable.

Parameters

Object prop The name or symbol of the property to test.

Returns

Boolean A Boolean indicating whether the specified property is enumerable.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
const array1 = [];
object1.property1 = 42;
array1[0] = 42;
application.output(object1.propertyIsEnumerable('property1')); // expected output: true
application.output(array1.propertyIsEnumerable(0)); // expected output: true
application.output(array1.propertyIsEnumerable('length')); // expected output: false
```

toLocaleString()

Returns a string representing the object. This method is meant to be overriden by derived objects for locale-specific purposes.

Returns

String A string representing the object.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
const number1 = 123456.789;
application.output(number1.toLocaleString()); // expected output: "123.456,789"
```

toString()

Returns a string representing the specified object.

Returns

String A string representing the object.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function Dog(name) {
  this.name = name;
}
var dog1 = new Dog('Spike');
Dog.prototype.toString = function dogToString() { return this.name; }

application.output(dog1.toString());
```

valueOf()

Returns the primitive value of the specified object. By default, the valueOf method is inherited by every object descended from Object.
Every built-in core object overrides this method to return an appropriate value.
If an object has no primitive value, valueOf returns the object itself.

Returns

Object The primitive value of the specified object.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function MyNumberType(n) {
 this.number = n;
}
MyNumberType.prototype.valueOf = function() { return this.number; };

const object1 = new MyNumberType(4);
application.output(object1 + 3); // expected output: 7
```