### super

Jump to: Syntax Description Example Specifications Browser compatibility See also

The **super** keyword is used to access and call functions on an object's parent.

The super.prop and super[expr] expressions are valid in any method definition in both classes and object literals.

## **Syntax**

```
super([arguments]); // calls the parent constructor.
super.functionOnParent([arguments]);
```

## **Description**

When used in a constructor, the super keyword appears alone and must be used before the this keyword is used. The super keyword can also be used to call functions on a parent object.

## Example

Using super in classes

This code snippet is taken from the  $\varnothing$  classes sample ( $\varnothing$  live demo). Here super() is called to avoid duplicating the constructor parts' that are common between Rectangle and Square.

```
class Rectangle {
      constructor(height, width) {
 2
        this.name = 'Rectangle';
 3
        this.height = height;
 4
        this.width = width;
 5
 6
      sayName() {
 7
        console.log('Hi, I am a ', this.name + '.');
 8
 9
10
      get area() {
        return this.height * this.width;
11
12
13
      set area(value) {
         this.height = this.width = Math.sqrt(value);
14
      }
15
16
    }
17
18
     class Square extends Rectangle {
      constructor(length) {
19
         this.height; // ReferenceError, super needs to be called first!
20
21
22
         // Here, it calls the parent class' constructor with lengths
```

```
// provided for the Rectangle's width and height
super(length, length);

// Note: In derived classes, super() must be called before you
// can use 'this'. Leaving this out will cause a reference error.
this.name = 'Square';
}

}
```

Super-calling static methods

You are also able to call super on static methods.

```
class Rectangle {
      constructor() {}
 2
      static logNbSides() {
 3
        return 'I have 4 sides';
 4
 5
    }
 6
 7
    class Square extends Rectangle {
 8
 9
      constructor() {}
     static logDescription() {
10
11
        return super.logNbSides() + ' which are all equal';
12
13
    Square.logDescription(); // 'I have 4 sides which are all equal'
14
```

Deleting super properties will throw an error

You cannot use the delete operator and super.prop or super[expr] to delete a parent class' property, it will throw a ReferenceError.

```
class Base {
     constructor() {}
2
      foo() {}
3
4
    }
    class Derived extends Base {
     constructor() {}
6
7
     delete() {
       delete super.foo; // this is bad
8
9
10
11
new Derived().delete(); // ReferenceError: invalid delete involving 'super'.
```

super.prop cannot overwrite non-writable properties

When defining non-writable properties with e.g. Object.defineProperty, super cannot overwrite the value of the property.

```
class X {
      constructor() {
 2
        Object.defineProperty(this, 'prop', {
 3
          configurable: true,
 4
          writable: false,
 6
          value: 1
 7
        });
 8
      }
 9
10
    class Y extends X {
11
12
     constructor() {
        super();
13
      foo() {
15
        super.prop = 2; // Cannot overwrite the value.
16
17
18
19
     var y = new Y();
20
    y.foo(); // TypeError: "prop" is read-only
21
     console.log(y.prop); // 1
```

#### Using super.prop in object literals

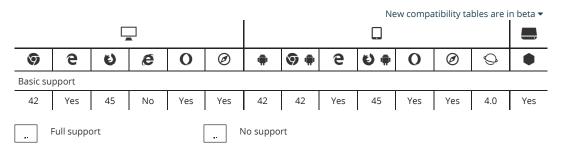
Super can also be used in the object initializer / literal notation. In this example, two objects define a method. In the second object, super calls the first object's method. This works with the help of Object.setPrototypeOf() with which we are able to set the prototype of obj2 to obj1, so that super is able to find method1 on obj1.

```
var obj1 = {
      method1() {
 2
 3
        console.log('method 1');
 4
 6
 7
     var obj2 = {
      method2() {
 8
       super.method1();
10
      }
11
12
    Object.setPrototypeOf(obj2, obj1);
13
    obj2.method2(); // logs "method 1"
```

# **Specifications**

Specification	Status	Comment
☑ ECMAScript 2015 (6th Edition, ECMA-262)   The definition of 'super' in that specification.	ST Standard	Initial definition.
☑ ECMAScript Latest Draft (ECMA-262)  The definition of 'super' in that specification.	D Draft	

### **Browser compatibility**



## See also

Classes