**EXTENDS**

La palabra clave **extends** es usada en la [declaración](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/class) o [expresión](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/class) de clases, para crear una clase hija de otra.

[**Sintaxis**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/extends#sintaxis)

class ChildClass extends ParentClass { ... }

[**Descripción**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/extends#descripci%C3%B3n)

La palabra clave extends se puede usar para crear una subclase a partir de clases personalizadas, así como sus objetos incorporados.

La propiedad .prototype de la nueva subclase debe ser un [Object](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Object) o [null](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/null).

[**Ejemplos**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/extends#ejemplos)

[**Como usar extends**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/extends#como_usar_extends)

El primer ejemplo crea una clase con el nombre Square a partir de una clase llamada Polygon. Este ejemplo ha sido extraido del siguiente [live demo](https://googlechrome.github.io/samples/classes-es6/index.html) [(código fuente)](https://github.com/GoogleChrome/samples/blob/gh-pages/classes-es6/index.html).

class Square extends Polygon {

constructor(length) {

// Aquí se invoca el constructor de la clase padre con longitud

// proporcionada por el ancho y alto de Polygon

super(length, length);

// Nota: En las clases extendidas, se debe llamar a super()

// antes de poder usar 'this'. El no hacerlo provocará un reference error.

this.name = 'Square';

}

get area() {

return this.height \* this.width;

}

set area(value) {

this.area = value;

}

}

Copy to Clipboard

[**Como usar extends con objetos incorporados**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/extends#como_usar_extends_con_objetos_incorporados)

Este ejemplo extiende el objeto incorporado [Date](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Date). Este ejemplo ha sido extraido del siguiente [live demo](https://googlechrome.github.io/samples/classes-es6/index.html) [(código fuente)](https://github.com/GoogleChrome/samples/blob/gh-pages/classes-es6/index.html).

class myDate extends Date {

constructor() {

super();

}

getFormattedDate() {

var months = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'];

return this.getDate() + "-" + months[this.getMonth()] + "-" + this.getFullYear();

}

}

Copy to Clipboard

[**Extendiendo de null**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/extends#extendiendo_de_null)

Extender de [null](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/null) es como hacerlo de una clase normal, excepto que el objeto prototype no hereda de [Object.prototype (en-US)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object" \o "Currently only available in English (US)).

class nullExtends extends null {

constructor() {}

}

Object.getPrototypeOf(nullExtends); // Function.prototype

Object.getPrototypeOf(nullExtends.prototype) // null