

# 构造方法

构造方法是创建和初始化使用类创建的一个对象的一种特殊方法。

## 语法

```
constructor([arguments]) { ... }
```

## 描述

一个 class 中只能有一个指定的 "constructor"（构造）方法。如果 class 定义的时候包含多个构造方法，程序将会抛出 `SyntaxError` 错误。

在构造方法中可以使用 `super` 关键字来调用父类的构造方法。

如果没有显式指定构造方法，则会添加默认的 `constructor` 方法。

## 实例

### 使用 构造方法

以下代码来自 [类](#) 的实例 ([在线 demo](#))。

```
1  class Square extends Polygon {  
2    constructor(length) {  
3      // 在这里调用父类的"length",赋值给矩形的"width"和"height".  
4      super(length, length);  
5      // 注意:子类必须在constructor方法中调用super方法,否则新建实例时会报错。  
6      this.name = 'Square';  
7    }  
8  
9    get area() {  
10     return this.height * this.width;  
11   }  
12  
13   set area(value) {  
14     this.area = value;
```

```
15 | }
16 | }
```

## 默认 构造方法

如果没有显式定义，会默认添加一个空的constructor方法。对于基类"Base classes"，默认构造方法如下：

```
1 | constructor() {}
```

对于派生类"Derived classes"，默认构造方法如下：

```
1 | constructor(...args) {
2 |     super(...args);
3 | }
```

## 规格

规格	状态	注释
<a href="#">ECMAScript 2015 (6th Edition, ECMA-262)</a> Constructor Method	<div>ST</div> Standard	初始定义.
<a href="#">ECMAScript Latest Draft (ECMA-262)</a> Constructor Method	<div>LS</div> Living Standard	

## 浏览器兼容

	Desktop	Mobile				
特性	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari	
Basic support	42.0	45 (45)	?	?	?	
Default constructors	?	45 (45)	?	?	?	

## 参阅

- [super\(\)](#)

- `class expression`
- `class declaration`
- `Classes`