

Reflect

Reflect

Reflect 是一个内置的对象，它提供拦截 JavaScript 操作的方法，这些方法与处理器对象的方法相同。Reflect不是一个函数对象，因此它是不可变的。

[TIP]
与大多数全局对象不同，Reflect没有构造函数。你不能将其与一个new运算符一起使用，或者将Reflect对象作为一个函数来调用。Reflect的所有属性和方法都是静态的（就像Math对象）

Reflect.apply()

语法

```
Reflect.apply(target, thisArgument, argumentsList)
```

解释

参数	含义	必选
target	目标函数	Y
thisArgument	target函数调用时绑定的this对象	N
argumentsList	target函数调用时传入的实参列表，该参数应该是一个类数组的对象	N

示例

```
Reflect.apply(Math.floor, undefined, [1.75])  
// 1;  
  
Reflect.apply(String.fromCharCode, undefined, [104, 101, 108, 108, 111])  
// "hello"  
  
Reflect.apply(RegExp.prototype.exec, /ab/, ['confabulation']).index  
// 4  
  
Reflect.apply('').charAt, 'ponies', [3])  
// "i"
```

ES5 对比

该方法与ES5中Function.prototype.apply()方法类似：调用一个方法并且显式地指定this变量和参数列表(arguments)，参数列表可以是数组，或类数组的对象。

```
Function.prototype.apply.call(Math.floor, undefined, [1.75]);
```

Reflect.construct()

Reflect.construct() 方法的行为有点像 new 操作符 构造函数， 相当于运行 new target(...args).

语法

```
Reflect.construct(target, argumentsList[, newTarget])
```

解释