

具名函数

匿名函数

window.Function函数对象

具名函数赋值

箭头函数

总结

函数的name属性

具名函数

var
function ~~x~~(输入1, 输入2) {
 return undefined
}
console.log(x) // 字符串
x.toString() // 字符串
number
string
7 种

变量, 特别 function ~~x~~ — 函数.

返回什么和打印什么完全两码事:

可以什么都不返回, 但是肯定能打印出来东西!

```
< undefined
> console.log = function(a){
    alert(a)
    return undefined
}
< f (a){
    alert(a)
    return undefined
}
> console.log(1)
< undefined
```

即使你不写 return,函数里面也会自动添加 `return undefined`
具名函数

```
1 function f(x,y){
2     return x+y
3 }
4 f.name // 'f'
```

匿名函数

//匿名函数不能单独使用,必须将其赋给另外一个变量

```
>> //匿名函数
//匿名函数不能单独使用,必须将其赋给另外一个变量
var x = function(input1,input2){ return}
```

```
< undefined
```

```
>> function y(){}
< undefined
```

```
>> console.log(y)
< undefined
```

```
▼ y()
  arguments: null
  caller: null
  length: 0
  name: "y"
  ▶ prototype: Object { ... }
  ▶ <prototype>: function ()
```

```
>> function(input1,input2){ return}
```

❗ SyntaxError: function statement requires a name [\[详细了解\]](#)

```
>> |
```

对比下面这两个:

```
>> function y(){}  
< undefined  
  
>> console.log(y)  
< undefined  
  
...  
  
>> var x = function y(input1,input2){}  
< undefined  
  
>> console.log(y)  
! ▶ ReferenceError: y is not defined [详细了解]  
  
>>
```

其实上面这个是js的一个bug!(这是js的不一致现象!!!)

function y() { }

console.log(y) ✓

刷新页面

var x = function y() { }

}

console.log(y) ✗ 报错

不一致

垃圾

window.Function函数对象

这种方式很少有人用!

Window . Fuction 函数对象

new Fuction ('x' , 'y' , 'return x+y')

var n=1

new Fuction ('x' , 'y' , 'return x+ ' + n + ' +y')

```
> n = 1
< 1
> f = new Function('x','y','return x+'+n+'+y')
< f anonymous(x,y
  ) {
    return x+1+y
  }
> f(1,2)
< 4
>
```

注意: return后面x后面的加号不能省，不然执行f的时候会出错！

```
1 n = 1
2 1
3 f = new Function('x','y','return x+'+n+'+y')
4 f anonymous(x,y
5 ) {
6   return x+1+y
7 }
8 f(1,2)
9 4
```

具名函数赋值

```
1 var f
2 f = function f2(x,y){ return x+y }
3 f.name // 'f2'
4 console.log(f2) // undefined
5
6
```

箭头函数

注意其写法:

```
>> sum1=(x,y)=>{return x+y}
< ▶ function sum1()
>> sum1(1,2)
< 3
>> sum1(1,2)
< 3
>> sum1=(1,2)
< 2
>> sum2=(x,y)=> x+y
< ▶ function sum2()
>> sum2(1,2)
< 3
>> |
```

```
>> n2 = n=>n*n
< ▶ function n2()
>> n2(4)
< 16
>> |
```

总结

```
< undefined
> function f(x,y){return x+y}
< undefined
575825 f = function(){}
< f (){}
> f = function x(){}
< f x(){}
> f = new Function('x','y','return x+y')
< f anonymous(x,y
/*`*/) {
  return x+y
}
> f = (x,y) => x+y
< (x,y) => x+y
> |
```

函数的name属性

js的这里也有坑!

```
>> function f(){}
< undefined
>> f.name
< "f"
>> var f2 = function(){}
< undefined
>> f2.name
< "f2"
>> var f3 = function f4 (){}
< undefined
>> f3.name
< "f4"
>> |
```

具名函数

匿名函数

给具名函数赋值

注意这几者之间的区别

```
> f5 = new Function('x','y','return x+y')
< f anonymous(x,y) {
  return x+y
}
> f5.name
< "anonymous"
>
```

这是全局对象创建的函数

js真tm的坑!