

JS 语法

从入门到工作：JS 全解

版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究 responsibility。

联系方式

如果你想要购买本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

如果你发现有人盗用本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

JS 版本

ES 6 是最低要求

JS 版本

- 历史版本

- ✓ ES 3, IE 6 支持, 总体评价: 垃圾
- ✓ ES 5, 总体评价: 还是垃圾
- ✓ ES 6, 大部分浏览器支持, 总体评价: 一半垃圾一半好
- ✓ ES 2019 与 ES 6 差别不大

- 为什么说 ES 6 一半垃圾

- ✓ 因为 ES 不能删除以前的特性, 要兼容旧网站
- ✓ 也就是说以前能运行的网站, 以后都要能运行
- ✓ 对比 Python 3 你就能知道兼容的好处: 稳定

取其精华，去其糟粕

对待 JS，这是我们的态度

我会直言 JS 的糟粕

不会为了维护 JS 留面子

那我们为什么还要学习
这么垃圾的语言

为什么不去学更优美的语言

一门语言的价值

是由其产生的价值决定

JS 是世界上使用最广的语言

JS 是门槛极低的语言（只要你不学糟粕）

JS 是一门能产生价值的语言（虽然不美）

JS 之父对 JS 的评价

它的优秀之处并非原创，它的原创之处并不优秀。

一些人对 JS 的批评

- 王垠

- ✓ 很多JavaScript程序员也盲目地鄙视Java，而其实JavaScript比Python和Ruby还要差。
- ✓ JavaScript的社区以幼稚和愚昧著称。你经常发现一些非常基本的常识，被JavaScript“专家”们当成了不起的发现似的，在大会上宣讲。

- 道格拉斯（JSON之父）

- ✓ JavaScript脚本语言一定会消失，最终被抛弃！
- ✓ 上面这句话我并没有找到原话

一些人对 JS 的辩护

- ✓ 世界上只有两种编程语言，一种是天天挨骂的,一种是没人用的
- ✓ JavaScript: 世界上最被误解的语言

JS 语法

开始学习

表达式与语句

- 表达式

- ✓ `1 + 2` 表达式的值为 3
- ✓ `add(1,2)` 表达式的值为函数的返回值
- ✓ `console.log` 表达式的值为函数本身
- ✓ `console.log(3)` 表达式的值为多少? `undefined`

- 语句

- ✓ `var a = 1` 是一个语句

- 二者的区别

- ✓ 表达式一般都有值，语句可能有也可能没有
- ✓ 语句一般会改变环境（声明、赋值）
- ✓ 上面两句话并不是绝对的

大小写敏感

- 不要写错

- ✓ var a 和 var A 是不同的
- ✓ object 和 Object 是不同的
- ✓ function 和 Function 是不同的
- ✓ 具体含义后面说

空格

- 大部分空格没有实际意义

- ✓ var a = 1 和 var a=1 没有区别

- ✓ 加回车大部分时候也不影响

- ✓ 只有一个地方不能加回车，那就是 return 后面 这就是JS的原创之处。

```
function fn(){  
    return  
    3  
}
```

//如果return后不接东西，会自动补充
undefined。所以这个函数返回值是undefined。

标识符

- 规则

希腊字母、中文、法语字母、德语字母等等

- ✓ 第一个字符，可以是 Unicode 字母或 \$ 或 _ 或中文
- ✓ 后面的字符，除了上面所说，还可以有数字

- 变量名是标识符

- ✓ `var _ = 1`
- ✓ `var $ = 2`
- ✓ `var _____ = 6`
- ✓ `var 你好 = 'hi'`

- 其他标识符用到再说

注释

写代码要多写注释——放屁

注释的分类

- 不好的注释

- ✓ 把代码翻译成中文
- ✓ 过时的注释
- ✓ 发泄不满的注释

- 好的注释

- ✓ 踩坑注解
- ✓ 为什么代码会写得这么奇怪，遇到什么 bug

区块 block

- 把代码包在一起

```
{  
  let a = 1  
  let b = 2  
}
```

- 常常与 if / for / while 合用

if 语句

如果 ... 那么 ...

if 语句

• 语法

- ✓ `if (表达式) {语句1} else {语句2}`
- ✓ `}` 在语句只有一句的时候可以省略，不建议这样做

• 变态情况

- ✓ 表达式里可以非常变态，如 `a = 1`
- ✓ 语句1里可以非常变态，如嵌套的 `if else`
- ✓ 语句2里可以非常变态，如嵌套的 `if else`
- ✓ 缩进也可以很变态，如面试题常常下套

```
a = 1
if(a === 2)
  console.log('a')
  console.log('a等于2')
```

最后打出a等于2。如果省略花括号，if只会管第一句。
JS里没有行的概念。

```
a=1
if(a===2)
  console.log('a');console.log('a等于2')
//打出a等于2。分号表示这句话完了。
```

```
a=1
if(a===2)
  console.log('a'),console.log('a等于2')
//什么都不打。逗号表示这句话没完。
```

使用最没有歧义的写法

程序员戒律

最推荐使用的写法

```
if (表达式) {  
    语句  
} else if (表达式) {  
    语句  
} else {  
    语句  
}
```


次推荐使用的写法

```
function fn(){  
  if (表达式) {  
    return 表达式  
  }  
  if (表达式) {  
    return 表达式  
  }  
  return 表达式  
}
```

switch 语句

不推荐

if ... else ... 升级版

switch 语句

- 语法

```
switch (fruit) {  
    case "banana":  
        // ...  
        break;  
    case "apple":  
        // ...  
        break;  
    default:  
        // ...  
}
```

- break

- ✓ 大部分时候，省略 break 你就完了
- ✓ 少部分时候，可以利用 break

问号冒号表达式

表达式1 ? 表达式2 : 表达式3

&& 短路逻辑

A && B && C && D 取第一个假值或 D

并不会取 true / false

如: `console&&console.log&&console.log()`
//IE没有console

|| 短路逻辑

A || B || C || D 取第一个真值或 D

并不会取 true / false

如: a=a||100

//如果a存在就等于a, a不存在就等于100

总结

- 条件语句

- ✓ if ... else ...
- ✓ switch
- ✓ A ? B : C
- ✓ A && B
- ✓ fn && fn()
- ✓ A || B
- ✓ A = A || B

while 循环

当 ... 时

while

- 语法

- ✓ while (表达式) { 语句 }
- ✓ 判断表达式的真假
- ✓ 当表达式为真，执行语句，执行完再判断表达式的真假
- ✓ 当表达式为假，执行后面的语句

- 其他

- ✓ do...while 用得不多，自行了解

```
while(a!==1){  
  console.log(a)  
  a=a+0.1  
}
```

//死循环。浮点数不精确。

for 循环

• 语法糖

- ✓ for 是 while 循环的方便写法
- ✓ 方便在哪?

```
for(var i=0;i<5;i++){  
  console.log(i)  
}  
//最后i等于5
```

```
for(var i=0;i<5;i++){  
  setTimeout(()=>{  
    console.log(i+'随机数'+Math.random())  
  },0)  
}  
//打出五个5。
```

• 语法

- ✓ for(语句1; 表达式2; 语句3){
- ✓ 循环体
- ✓ }
- ✓ 先执行语句1
- ✓ 然后判断表达式2
- ✓ 如果为真, 执行循环体, 然后执行语句3
- ✓ 如果为假, 直接退出循环, 执行后面的语句

```
for(let i=0;i<5;i++){  
  setTimeout(()=>{  
    console.log(i+'随机数'+Math.random())  
  },0)  
}  
//打出1, 2, 3, 4, 5。for里的let有特殊处理。之后再讲。
```

break 和 continue

退出所有循环 V.S. 退出当前一次循环

退出离它最近的for

label 语句

用的很少，面试会考（概率 5%）

label

- 语法

```
foo: {  
    console.log(1);  
    break foo;  
    console.log('本行不会输出');  
}  
console.log(2);
```

- 面试

```
{  
    foo: 1  
}
```

```
var a={  
    foo:1  
}  
//这就是对象。Object{foo:1}
```

- ✓ 上面的东西是什么

有一个代码块，里面有一个标签，标签名是foo，foo的内容就是一句话，1。
即foo是label，语句是1。

推荐书籍

- 阮一峰的免费教程
 - ✓ 入门篇
- 你不知道的 JavaScript
 - ✓ 先买上卷，适合进阶