# 语言基础

## 1. 语法

- 区分大小写
- 标识符
  1.第一个字符必须是一个字母、下划线（_）或美元符号（$）
  2.剩下的其他字符可以是字母、下划线、美元符号或数字
  3.字母可以是Extended ASCII中的字母，也可以是Unicode字母字符
- 注释
  i. // 单行注释
  ii. /多行注释/
- 严格模式
  ES5增加了严格模式，脚本开头加上'use strict';
  也可以单独对一个函数使用，放在函数体内开头
- 语句
  ；非必须，但是推荐加
  {}代码块

## 2. 关键字保留字

```
break       do          in          typeof
case        else        instanceof  var
catch       export      new         void
class       extends     return      while
const       finally     super       with
continue    for         switch      yield
debugger    function    this
default     if          throw
delete      import      try
```

规范中也描述了一组**未来的保留字**，同样不能用作标识符或属性名。虽然保留字在语言中没有特定用途，但它们是保留给将来做关键字用的。

以下是 ECMA-262 第 6 版为将来保留的所有词汇。

**始终保留：**

```
enum
```

**严格模式下保留：**

```
implements  package     public
interface   protected   static
let         private
```

**模块代码中保留：**

```
await
```

关键字保留字不能做标识符，但是可以做对象的属性名，最好还是不要做属性名，以确保兼容过去和未来的ES版本

# 3. 变量

ES变量是松散类型，意思是变量可以用于保存任何类型的数据。
3个关键词声明变量：var，const，let
var所有版本都能用，const和let只有ES6之后的版本能用

## var

作用域：

- 函数内声明成为该函数的局部变量，函数退出时销毁
- 函数内定义变量省略var，则创建全局变量！，严格模式报错

```
function test() {
 message = "hi"; // 全局变量
}
test();
console.log(message); // "hi"
```

但是得运行过函数外部才能访问到
**var 声明提升**
自动提升到函数作用域/全局顶部

```
function foo() {
 console.log(age);
 var age = 26;
}
foo(); // undefined
```

## let

let是块作用域，{} 块
冗余报错：同一个块中使用不同关键词声明

```
var name;
let name; // SyntaxError

var name;
var name; // 不报错
```

暂时性死区
因为let没有声明提升，在未声明之前不能以任何方式引用
在 let 声明之前的执行瞬间被称为“暂时性死区”（temporal dead zone）

```
// name 会被提升
console.log(name); // undefined
var name = 'Matt';
// age 不会被提升
console.log(age); // ReferenceError：age 没有定义
let age = 26;
```

全局声明
var，全局作用域声明的变量会成为window对象的属性；let不能

不能条件声明
因为是块作用域，所以不能条件声明

for循环中的let声明
let位每一次循环都声明一个新的迭代变量，var则是同一个变量

```
for (let i = 0; i < 5; ++i) {
 setTimeout(() => console.log(i), 0)
}
// 会输出 0、1、2、3、4

for (var i = 0; i < 5; ++i) {
 setTimeout(() => console.log(i), 0)
}
// 你可能以为会输出 0、1、2、3、4
// 实际上会输出 5、5、5、5、5
```

## const

与let基本一直，区别就是必须初始化，然后是常量。

# 总结

1. 关键字、保留字不做标识符，但是可以当属性名
2. var声明提升，const，let没有声明提升
3. var在for循环中一直是同一个变量；const，let每一次循环都声明一个新的迭代变量（包括for...in/of）
4. var函数/全局作用域，let/const块作用域
5. 函数中没有用操作符声明变量，则为全局变量