本文章综合了很多地方对global和widow区别的解释，我加以挑选再结合我自己的理解以及遇到的问题写出来的。可能比较乱，但是全部看完或许对global和window对象的理解还是有帮助的。一个关于window和global区别的链接，可以更方便去理解：https://segmentfault.com/q/101000000305437、关于定义和声明的区别：http://blog.csdn.net/xiaoyusmile/article/details/5420252

## 1.Global

1) 所有在全局作用域内定义的属性和方法，都是Global对象的属性。

2) Global对象不能直接使用，也不能用new运算符创建。

3) Global对象在JavaScript引擎被初始化时创建，并初始化其方法和属性。

4) 浏览器把Global对象作为window对象的一部分实现了，因此，所有的全局属性和函数都是window对象的属性和方法。

global对象是单体内置对象，即不依赖宿主环境的对象。

ECMAScript 中的Global 对象在某种意义上是作为一个终极的“兜底儿对象”来定义的。换句话说，不属于任何其他对象的属性和方法，最终都是它的属性和方法。事实上，没有全局变量或全局函数；所有在全局作用域中定义的属性和函数，都是Global 对象的属性。本书前面介绍过的那些函数，诸如isNaN()、isFinite()、parseInt()以及parseFloat()等看起来都像独立的函数，实际上全都是Global对象的方法。还有就是常见的一些特殊值，如：NaN、undefined等都是它的属性。

global的属性和方法：http://blog.csdn.net/qq\_32758013/article/details/74781788

2.window(浏览器下的一个web api)

在全局作用域中声明的变量、函数都是window对象的属性和方法。

window对象是相对于web浏览器而言的，依赖于浏览器。在一定程度上，window对象，是global的一个子对象。浏览器环境下，global对象指的就是window对象，可以说window对象是global在浏览器中的具体表现。（自我理解：如果把global对象比作java中的接口，那么window就相当于在浏览器环境中创建的一个具体实现类，因此global是标准是接口，window是具体应用实现类）

ECMAScript 虽然没有指出如何直接访问Global 对象，但Web 浏览器都是将这个全局对象作为window 对象的一部分加以实现的。因此，在全局作用域中声明的所有变量和函数，就都成为了window对象的属性。来看下面的例子：

var color = "red";

function sayColor(){

alert(window.color);

}

window.sayColor(); //"red"

这里定义了一个名为color 的全局变量和一个名为sayColor()的全局函数。在sayColor()内部，我们通过window.color 来访问color 变量，说明全局变量是window 对象的属性。然后又使用window.sayColor()来直接通过window 对象调用这个函数，说明全局作用域中声明的函数是window对象的方法。

window对象是宿主对象也就是在一定的环境中才会生成的对象（这里也就是指浏览器），而global对象是在任何环境中都存在的，window对象具体也就是指浏览器打开的那个窗口。

window对象的属性和方法（window的属性和方法比较多，这个链接可能说的不全仅供参考，具体的请在chrome后台中，输入window查看）：http://www.cnblogs.com/zengwei/archive/2007/11/02/946520.html

## 3.document

document对象是window对象的一个属性，是显示于窗口内的一个文档。而window对象则是一个顶层对象，它不是另一个对象的属性。document可以理解为文档，就是你的网页，而window是你的窗口，就是你的浏览器包含的。它们俩在没有框架的情况下可以说是等同的，在有框架的情况下就要区别对待了。

区别:1、window 指窗体。document指页面。document是window的一个子对象。

        2、用户不能改变 document.location(因为这是当前显示文档的位置)。但是,可以改变window.location (用其它文档取代当前文档)window.location本身也是一个对象,而document.location不是对象

3、Window对象：http://www.w3school.com.cn/htmldom/dom\_obj\_window.asp

     Document对象：http://www.w3school.com.cn/htmldom/dom\_obj\_document.asp

## 4.this

this对象是在运行时基于函数的执行环境绑定的：在全局函数中，this等于window；当函数被作为某个对象的方法调用时，this等于那个对象。

特别注意：匿名函数的执行环境具有全局性，因此匿名函数中的this对象通常指向window对象!!!

总结：

在node.js中运行下面代码

function pet(words){

this.words = words;

console.log(this.words);

console.log(this === global);

}

pet('wang');

输出：wang

  true

而在chrome后台中运行

function pet(words){

this.words = words;

console.log(this.words);

console.log(this === window); //注意，在浏览器中，没有global，会显示global is not defined

}

pet('wang');

输出：wang

  true

Global是js语言核心的东西，而window只是浏览器引擎上实现的一个对Global对象的封装。JS不仅仅局限于浏览器，还有其他地方。window对象是宿主对象也就是在一定的环境中才会生成的对象，这个环境就是浏览器。而global对象是在任何环境中都存在的。

---------------------

作者：o菜鸟猿o

来源：CSDN

原文：https://blog.csdn.net/lzh\_15617663141/article/details/78081229

版权声明：本文为博主原创文章，转载请附上博文链接！