

JavaScript

- 什么是JavaScript

JavaScript,简称js, 是一种运行于js解释器/引擎中的脚本语言。

js运行环境：

1. 独立安装的js解释器(node)

<https://nodejs.org>

2. 嵌入在浏览器内核中的js解释器

- js发展史

- 1992年Nombas公司开了一款脚本语言 (ScriptEase) , 可以运行在浏览器中
- 1995年网景,LiveScript,后期更名javascript
- 1996年microsoft在IE3.0发了javascript的克隆版本JScript
- 1997年JavaScript提价给ECMA (欧洲计算器联合制造商) , 定义了ECMAScript(简称ES5 , ES6).

- js的组成

- 核心 (ECMAScript)
- DOM (w3c万维网联盟) Document Object Model 文档对象模型
提供了一组允许操作页面元素的函数和属性
- BOM Browser Object Model 浏览器对象模型
提供一组允许操作浏览器的函数和属性

- js的特点

- 语法类似c , java
- 无需编译 , 由js解释直接运行
- 弱类型语言
- 面向对象

- 变量的命名规范

- 不允许使用js的关键字和保留的关键字
var,string,boolean,number,if,else,for,while...
- 不能以数字开头
- 语义化
var a;
var uname;
- 尽量使用小驼峰命名法
var uanme;
var userName; //小驼峰
var UserName; //大驼峰
var user_name; //下划线

- 允许包含数字，字母，下划线，\$

数据类型

- 数据类型的作用

规定了数据在内存中所占的空间

20 20.1

4个字节 8个字节

bit:位

byte : 字节

8bit=1byte

1024byte=1KB 小的网页

1024KB=1MB 兆 一首歌3-4MB

1024MB=1GB 一个高清电影 1G

电脑的存储空间500G

1024G=1T

- 数据类型分类

- 原始类型（基本类型）

分类：数字类型，字符串类型，布尔类型，undefined,null

- 引用类型（复合类型）

分类：数组，函数，对象

Number

作用：可以表示32位整数，也可以表示64位浮点数（俗称小数）

整数：

1. 十进制；由0-9十个数字组成，逢十进一

ex:var age=25;

2. 八进制

由0-7八个数字组成，逢八进一

ex:8->010 始终以0开头

3. 十六进制

由0-9和A-F组成，逢十六进一；以0x开头

ex:0x10 ; 10 : A ; 11 : B ; 12 : C ; 13 : D ; 14 : E ; 15 : F

浮点数：又称小数

ex：小数算法：23.5；指数算法：3.4e3 3.4*10的3次方

String

作用：表示一系列的文本字符数据，如：姓名，地址，性别，信息等

每个字符在计算机中都有一个唯一的表示该字符的编码，该码称为unicode（十六进制）码

查找字符的unicode码：charCodeAt();

ex:

"李".charCodeAt().toString(16); 输出16进制的unicode码

"李".charCodeAt().toString(8); 输出8进制的unicode码

"李".charCodeAt().toString(2); 输出的2进制的unicode码

李 ==>674e

如何将unicode码转换成字符？

使用转义字符\u,将unicode码转换成字符。

var str="\u674e";

js中提供了转义字符：

\n:换行；\t:制表符（缩进）；\"：\"；\'：\'；\\: \

Boolean

作用：在程序中表示真或假的结果

取值：true 或 false；参与数字运算时，true可以当做1运算，false可以当做0运算

ex:

```
1 var result = 25 + true; // 26
```

undefined

作用：用于表示使用的数据不存在

undefined类型只有一个值，即undefined。当声明变量未赋值时，该变量默认值就是undefined

null

作用：用于表示尚不存在的对象。

null类型只有一个值即null；如果函数或方法返回的是对象，找不到对象时，返回值就是null

数据类型转换

- 隐式转换（自动转换）

不同类型的数据在计算的过程中自动进行转换

1.数字+字符串：将数字转换为字符串

var num = 15, str = "hello";

```
var result = num+str; // 15hello
```

2.数字+布尔：将布尔转换为数字类型

```
var num = 15, isSun = true;
```

```
var res = num + isSun; // 16
```

3.字符串+布尔：将布尔类型转换为字符串

```
var str = "Hello", isSun = true;
```

```
var res = str + isSun; // Hellotrue
```

4.布尔+布尔：将布尔类型转换为数字

```
var isSun = true; // 1
```

```
var isBig = false; // 0
```

```
var res = isSun + isBig; // 1
```

• 强制转换 —— 转换函数

1. toString()

将任意类型数据转换为字符串

语法：var result=变量.toString();

ex：

```
1 var num=15;
2 var str=num.toString();
3 console.log(typeof(str)); //结果:string
```

2. parseInt()

将任意类型的数据转换为整数

语法：var result=parseInt(数据);

注意：如果转换不成功，结果为：NaN (Not a Number)

ex:

```
1 var num1="123abc"; // 123
2 var num2="abc123"; // NaN
3 var num3="23.87"; // 23
```

3. parseFloat()

将任意数据类型转换为小数

语法：var result=parseFloat(数据);

4. Number()

将任意类型的数据转换为number类型。

注意：如果数据包含非法字符，则返回NaN

运算符和表达式

位运算符

`<<`, `>>`, `&`, `|`, `^`(异或)

- `<<`, `>>` 往左移在右边补 0，往右移在左边补 0

左移是把数字变大，右移是把数字变小

ex:

1000

10000 0100

- `&` 按位与 判断数字奇偶性

任意数字与1做按位与，结果是1，则为奇数，结果是0，则为偶数。

ex:

```
var num=23562;
```

```
var result=num & 1;
```

```
console.log(result); //结果：0
```

- `|` 按位或，对小数取整

将任意小数与0做按位或，结果则取整数部分

ex:

```
var num = 123.456;
```

```
var res = num | 0;
```

```
console.log(res); // 123
```

- `^` 按位异或 用于交换两个数字

二进制位数，逐位比较，不同则为1，相同则为0

3 : 011; 5 : 101

ex: `var a = 3, b = 5;`

```
a = a ^ b; // 6
```

```
b = b ^ a; // 3
```

```
a = a ^ b; // 5
```

```
console.log(a); // 5
```

```
console.log(b); // 3
```

ECMAScript提供的全局函数

parseInt()

parseFloat();

Number();

isNaN();

encodeURIComponent()

URL:Uniform Resource Locator统一资源定位器

URI:Uniform Resource Identifier(标识符)

作用：对统一资源标识符进行编码，并返回编码后的字符串

所谓的编码，就是将多字节的文字编译成单字节的文字（汉字：2-3字节）

decodeURI()

作用：对已编码的URI进行解码，并返回解码后的字符串

encodeURIComponent()

在encodeURIComponent的基础上，允许对特殊字符也进行编码

decodeURIComponent()

在decodeURI的基础上，还可以解码特殊字符

eval()

作用：执行以字符串表示的js代码