# 编程语言

* 1. 什么是编程语言

编程语言（programming language）**（计算机语言）是人和计算机交流的一种语言**。编程语言主要是用来向计算机发出指令（cmd控制台：shutdown -s，关机指令）, 计算机会严格按照指令一步一步完成相应的数据处理。

虽然计算机被称为电脑,但是电脑是最笨的,需要人为操作,而且严格按照指令完成数据处理（终结者中的机器人，接收到杀人指令后，就会执行相应操作）。所以说编程语言实际上是告诉计算机完成一个数据处理的过程.

人和人交流:语言(汉语,英语等)

人和计算机交流:编程语言(Javascript,Java等).



强调:

我们应该站在主人的角度使用计算机编程语言来命令计算机帮我实现我想要的数据处理.简单理解：程序员通过程序代码（指令）让计算机帮我们完成的数据处理工作；

* 1. 为什么要使用编程语言

使用**编程语言可以命令计算机帮我们处理日常数据**的.而**编程语言正是如何处理数据的一个过程**。

处理数据的过程可以很简单，也可以很复杂，比如：某个软件公司要给全体员工涨工资，我们只需要获取到每一个员工的工资+500即可。 但是如果公司规定，只有入职满一年的员工才涨工资，那么这个处理数据的过程比前者就稍麻烦一点，我们就可以“涨工资”这个过程交给编程语言来完成。

* 1. 编程语言被用在哪里

编程语言在生活中无处不在.

ATM机上: 当用户输入卡号和密码时,编程语言会告诉计算机如何检查用户的卡号和密码是否正确.

QQ: QQ如何发送信息给好友以及接收好友信息.

所有电子设备都用到了编程语言；

* 1. 如何学习编程

前面已经说过编程语言是和计算机交流的一种语言,所以编程语言也有自己的语法,语言结构. 学习编程语言的步骤和学习汉语的步骤是一样的.

汉语: 字词🡺造句🡺小学作文

编程: 变量🡺语言结构🡺功能

不同之处:

学习汉语目的是将一个事物描述清楚. 使你拥有很好的文采.

学习编程目的是命令计算机按照你的步骤执行. 所以说就像写小学作文一样.

学习编程主要是如何将现实生活中的过程使用计算机语言描述出来.

1. Javascript简介
   1. 什么是Javascript

**JavaScript是一门强大的编程语言，它既是一门非常简单的语言，又是一门非常复杂的语言**。 说它简单是因为学会它就一杯茶的功夫，说的复杂是因为要掌握他的精髓需要我们花很多功夫，而且还需要有为师带领大家避坑。

JavaScript诞生于1995年。当时，它主要是来处理页面中的表单验证工作，是否输入了有效值。在上个世纪90年代，大家都是通过**拨号上网**，网速极差，而要进行表单值的验证，需要把数据从一台电脑（客户端）提交到另一台电脑（服务端）上，漫长的提交过程，让用户无比的焦灼，用户体验极差。Netscape Navigator(网景公司)浏览器希望通过JavaScript来做解决这个问题，用户提交表单之前对表单的值进行验证，如果无效就提示给用户，如果有效就提交给服务端，大大提高了用户体验。

虽然JavaScript是一门为表单验证而创造的一门编程语言，但JavaScript能做的事情岂止是验证，现在网页的交互(验证，动态，幻灯片，动效)操作都是由JavaScript完成。

* 1. JavaScript和Java语言的区别

Javascript和Java没有任何关系，这两个语言是不同的两种语言，只是名字上都有一个Java而已。

了解：之所以叫JavaScript，因为当时Java团队的人参与前期的设计和开发，而且Java很火。

* 1. 为什么要学习Javascript

公司要求会使用Javascript

Javascript可以我们的开发的网站用户体验性更好，现在的网站离不开Javascript。、

一个完整的网页是三位一体：html+css+javascript

结构 + 表现 + 行为

* 1. Javascript用在哪里（了解）

**1.表单数据验证**：表单数据验证是JavaScript最基本也是最能体现效率的功能。

**2.动态HTML（即DHTML）**：动态HTML指不需要服务器介入而动态变化的网页效果，包括动态内容、动态样式、动态布局等。

**3.用户交互**：用户交互指根据用户的不同操作进行的响应处理。例如：联动菜单等。

**4.数据绑定**：HTML中表单和表格能够以.txt文件定义的数据源，通过对位于服务器端的数据源文件的访问，便可以将数据源中的数据传送到客户端，并将这些数据保存在客户端。

**5.少量数据查找**：能够实现在当前网页中进行字符串的查找和替换。

**6.AJAX核心技术**：AJAX即异步JavaScript+XML。该对象提供一种支持异步请求的技术，使客户端可以使用JavaScript向服务器提出请求并处理响应，但并不影响用户在客户端的浏览。

哪些公司在使用Javascript。

淘宝，百度，京东 只要有网站的公司都要使用Javascript。 所以我们必须学好Javascript.

* 1. 开发工具

Web前端开发工具：

记事本（装逼可行）

editplus

notepad++

dreamweaver

**webstorm**

**sublime**

**hbuilder**等等。

1. Hello World（重点）
   1. 天才第一步(无脑入门）

先看看什么是javascript代码，可以将以下代码写在一个html文件的任何位置。

<script type="text/javascript">

alert('hello world!');

</script>

alert()是告诉浏览器输出里面的内容。

* 1. Javascript写在哪里

**Javascript代码一般写在<head>标签的<script></script>标签中**

* + 1. Javascript可以写在script标签内（内嵌方式）

<script type="text/javascript">

alert('这是Javascript程序');

</script>

通过type属性告知浏览器标签内容是Javascript代码。

* + 1. Javascript可以写在单独的文件中（外联方式）

1.创建以.js结尾的文件并且写入Javascript代码

2.hello.js(文件名任意):

alert('这是Javascript程序');

3.在html文件中使用Javascript代码文件

<script type="text/javascript" src=’hello.js’></script>

* + 1. JavaScript可以写在HTML标签的事件属性中

onclick点击事件（用户点击），属性值：点击时，执行的js代码。

<button onclick=” alert(250); ”>点击我</button>

用户点击，button的时候会提示250；

* + 1. Javascript可以A标签的href属性中

a标签：超链接，主要用于跳转到href属性设置连接地址；

<a href=”http://www.itsource.cn”>源码时代官网</a>

a可以设置href值前缀以“javascript:”（前缀其实叫协议,例如：thunder://xxxfasdfasd,迅雷的协议，点击这样的连接打开的不是页面，而是迅雷）开头，当用户点击的时候就会告诉浏览器这不是一个连接地址，而是需要执行一段javascript代码。

<a href=”javascript: alert(‘搞啥子？？？’);”>点我</a>

* + 1. script标签的规范位置

虽然<script>标签可以放在<html>标签的任何位置，但是为了网页的整体结构和开发人员多数的契约，我们应该放在如下两个位置：

* 1. 所有的<script>标签都应该放在页面的<head>标签中；

<html>

<head>

<title>文档标题</title>

<script src=”xxx” type=”xxx”></script>

</head>

</html>

* 1. 为了让内容优先加载显示，提高用户体验，也可以把<script>标签放在</body>结束标签之前。

<html>

<head>

<title>文档标题</title>

</head>

<body>

网页内容...........

网页内容...........

网页内容...........

<script src=”xxx” type=”xxx”></script>

</body>

</html>

* + 1. 四种方式的区别和用法

**区别：**

1. 内嵌方式的Javascript只能够在当前html文件中使用；
2. 外联方式的js文件可以被多个html页面使用；
3. onclick是一个事件（发现什么什么后）属性，只有发现后才执行；
4. a标签本身是超链接，但是我们一旦指定javascript:协议后，点击这个超链接，就告诉浏览器需要执行一段javascript代码。与onclick一样都是点击后执行。

**<script>标签选择：**如果一个Javascript代码被多个html文件使用，请选择外联方式， 或者可以选择内嵌方式,策略与css一样。

* + 1. 注意点

1. **script标签的type属性可以不设置，因为默认就是为text/javascript**,如果你设置了请保证你type属性值为text/javascript；请勿选择text/vbscript等。

2、script标签中的内容为js代码，如果指定了src属性，那么万万不能在script标签中编写js代码，因为浏览器会加载src属性对应的文件内容到script内容中。

hello.js 文件

alert(“hello world”);

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

<script type=”text/javascript” src=”hello.js”>

alert(“hehehehehe”);//代码不会被执行，因为的内容被src对应的文件内容替换掉。

</script>

3、<script>标签中只能写JavaScript代码，其他的东西都不可以，比如标签、样式。

* 1. Javascript的输出（死记住）

计算机编程语言是用来向计算机发出指令,计算机会严格按照指令一步一步完成相应的功能，但是需要将完成功能的结果输出给我们。 可以通过一下三种方式：

* + 1. 网页输出
       1. 输出纯文本：

document.write("hello world");

* + - 1. 输出html代码：

document.write("<h1>hello world</h1>");

* + 1. 弹出框

alert('hello word');

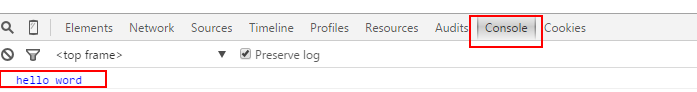
* + 1. 开发者工具控制台输出

现代浏览器都具有“开发者工具” ，浏览器中快捷键为F12；

console.log ('hello word');

如何打开控制台：

谷歌google浏览器🡺按F12（或shift+ctrl+i）🡺console



* + 1. 为什么先要学习这几个输出

编程语言是命令计算机运行的过程，需要通过上面的输出查看运行过程和结果是否正确。

* 1. 第一个复杂的JS程序（现实生活与编程语言）

将我们现实生活中实际完成的步骤通过计算机语言描述出来,告诉计算机就会按照你的命令完成 （程序是解决我们现实生活中的问题，所有程序来源于生活）.

当你学会了编程之后你就会感觉写代码就像写小学作文一样.

大家跟着我学习小学作文.

如何将长颈鹿先放入冰箱,再将大象放入冰箱中?

* + 1. ****现实生活中使用中文描述****

把长颈鹿放到冰箱里

第一步：把冰箱门打开

第二步：把长颈鹿放进去

第三步：把门关上

把大象放到冰箱里

第一步：把冰箱门打开

第二步：把长颈鹿拿出来

第三步：把大象放进去

第四步：把门关上

步骤都非常简单，小学生都能明白，不过也需要逻辑思维。

* + 1. ****计算机如何完成具体的步骤****

通过编程语言告知计算机如何完成上面的步骤.

<script type="text/javascript">

//>>1.我们不可能把真现实生活中的东西放到电脑中，所有使用编程语言模拟两个动物和冰箱，

var dongwu1 = '长颈鹿';

var dongwu2 = '大象';

var hezi = '冰箱';

//>>2.把长颈鹿放到冰箱里

document.write("把"+hezi+"门打开");

document.write("把"+dongwu1+"放进去");

document.write("把门关上");

//>>3.把大象放到冰箱里

document.write("把"+hezi+"门打开");

document.write("把"+dongwu2+"放进去");

document.write("把门关上");

</script>

上面的代码大致知道什么意思就可以了, 如果不知道也没有关系,也是我们几个月要学习的内容.

**比对现实生活和计算机代码的目的是:计算机很笨,必须所有的流程必须通过编程语言告诉他才行.**

* + 1. 程序的结构

1. 程序由若干语句组成，语句是组成程序的最基本单位。
2. 每条语句以分号“;”结束。
3. 一行中可以放置多条语句。
   * 1. 初学者容易犯的错误

标点符号错误：编程语言中最常见的错误，要明白编程语言是西方传入，并不能识别中文的标点符号。必须使用**英文标签**（设置输入法）

单词拼写错误。 ：document –不要写为- dcument

语法结构错误。 ：必须按照语法结构，少些符号或者是符号不对应。

* + 1. 初学者解决错误的方法

1. 通过chorme浏览器的控制台查看错误。
2. 通过alert或者document.write()或者是console.log()查看执行过程和结构
3. 通过排除法逐步排除错误， 从大的范围到小的范围。
   1. 学习编程的难点和重点

难点：

如何将现实生活中的思想转化到计算机中，使用计算机语言将这种思想通过代码表现出来。 一定要注意计算机很笨，它是没有任何思想，都需要我们通过程序代码告诉它。

重点：

编写代码中的思想，感觉。 如果写程序像写小学作文一样，那么你就成功了。整个基础班的重点也是培养大家的编程想象。

有的**语句记不住不要紧，熟能生巧,注意代码提示**。

1. Javascript基础语法

前面说过学习编程语言就会学习汉语一样，只要是语言都有他们的语法。

* 1. 注释

注释是不需要执行，主要是对程序进行文字说明和解释。

单行注释：

// 注释的内容

多行注释：

/\*

注释的内容。

注释的内容

\*/

* 1. JavaScript区分大小写

html标签名和属性名，不区分大小写，比如<a href=””>与<A HREF=””>效果一样，但是 JavaScript 是一门区分大小写到语言。因此，就需要在编程的时候注意这个细节，username与userName所代表的是不同的两个东西。

* 1. 关键字和保留字

**关键字：编程语言中有特殊含义的单词**，后面学习中慢慢讲解；

break else new var case finally return void

catch for switch while continue function this with

default if throwdelete in try do instanceof typeof

保留字：现在**JavaScript版本中没有使用这些单词，供以后版本使用**，但是也不准我们使用。

abstract enum int short boolean export interface static

byte extends long super char final native synchronized

class float package throws const goto private transient

debugger implements protected volatile double import public

**提示：关键字和保留字不需要特别记忆，慢慢来积累和记忆，老师在讲课的时候会提醒各位；**

* 1. 标识符

标示符（IDentifier）是指用来标识某个实体的一个符号。其实就是**命名**；

编程语言中，标识符就是程序员自己规定的具有特定含义的词，比如类名称，属性名称，变量名等。

**标识符可以由任意顺序的大小写字母、数字、下划线(\_)和美元符号($)组成，但标识符不能以数字开头，不能是JavaScript中的保留字或关键字。**

合法标识符：userName

user\_name

$

\_

$0

\_0

非法标识符：

250

abc&

helloworld

Mini-Copue

a-zA-z0-9\_$ ： **不能以数字开头**

简单理解：就是对某个事物取一个名字，但是名字有一点的规范要求，就比如我们中国人取名字，只能是汉语，不能有日文是一个道理。

* 1. 变量

程序运行过程中，会产生很多很多的数据，比如以下情况：



消消乐游戏中的数据：

金币

步数

得分

..........



LOL游戏中数据：

经验

等级

杀猪数据

..........

数据对于一个程序非常的重要，数据与程序的执行也是息息相关的；

问题1：数据存在什么地方？

数据可以存储在电脑的**内存中**或**硬盘**（可断电设备，比如U盘，光盘，软盘等...）中；

**内存中的数据特点：仅在程序运行的时候存在，一旦关闭程序或关闭电脑，那么数据就不在了；（临时存储）**

**硬盘中的数据的数据特点：存储后就一直存在，如果需要进行删除，则需人为的进行删除；（永久存储）**

问题2：数据存储的意义？

1、程序对数据进行运算，根据数据进行合理的执行；

2、获取数据展现给用户；

问题3：我们应该把数据存储在什么地方？

**对于那些需要运算的数据，直接把数据存储在内存中，使用方便，因为内存的读写速度是最快的；（目前现阶段重点关注）**

对于那些需要永久保存的数据，我们就可以存储到硬盘中，硬盘可断电，可以用来存档；

* + 1. 什么是变量

变：变化 量：数据， 变量用于存储程序中变化的数据；

变量就是让我们告诉计算机我们需要在内存中开辟一块存储空间，用于数据的保存。

**变量存储在计算机内存的“小格子”中，内存是由N多个“小格子”构成，每个“小格子”为它起一个名字然后将数据存放进去。通过这个格子名（变量名）我们就可以操作格子中的数据。**

定义变量的语法结构：

var 变量名 = “变量值”;-

例如：

//>>1.使用编程语言定义两个动物和冰箱以及在是否在冰箱中的状态;

var dongwu1 = '长颈鹿';

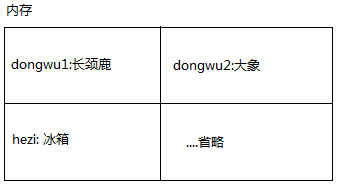
var dongwu2 = '大象';

var hezi = '冰箱';

var dongwu1InHezi = false;

var dongwu2InHezi = false;

图解：



* + 1. 变量命名规范

变量由字母、数字、字符组成（\_，$）,首字母不能以数字开头；

变量一般首字母小写；

变量名尽量使用有意义的单词；

var name;//变量名体现存储的内容

var aaaa; //无意义

变量如果是多个单词，一般采用**驼峰命名**法或使用\_分割

var userName;

var user\_name;

var backgroundColor;

* + 1. 为什么要使用变量

**变量可以在内存中通过名字保存一个数据，然后通过变量的名称来多次使用该数据**。 也可以通过变量的名字让变量的值传递给其他的程序使用。

* + 1. 思想转化过程

现实生活中有 “长颈鹿”， “大象”，‘冰箱’ 和 是否在冰箱中存在的状态， 但是在计算机的世界中概念。 我们就应该通过变量的方式告诉给计算机。

通常会将现实生活的名词和数据通过变量的形式告诉计算机。

* + 1. 实例

将凤姐的信息保存告诉给计算机。

<script type="text/javascript">

var name = '凤姐';

var age = 21;

var sex = '怪物';

var boyFirend = '奥巴马';

var isMarry = true; //是否结婚

alert(name+"的男朋友是"+boyFirend);

name = '春春';

age = 28;

alert(name+"的年龄为"+age);

</script>

* + 1. 语法细节
       1. 变量可以单独定义之后再赋值

//>>1.定义变量

var name;

//>>2.给变量赋值

name = '张三';

“=” 会将右边的赋值给左边。

* + - 1. 常见的语法形式

var v1=1;//定义变量的同时进行赋值

var v2,v3,v4;//同时定义多个变量

var v5,v6=1,v7=2;//同时定义多个变量，并对其中的某些变量赋值

var v7=v8=0;//同时定义多个变量，并设置相同值

* 1. Javascript中的数据类型
     1. 什么是数据类型

人以类聚,物以群分，其实就是现实版的分类，编程语言中数据类型其实就是给数据分类，比如，数字250，123和字符串“呵呵”、“哈哈”,在JavaScript中将数据的值划分成了不同的类型，目的是为了让计算机可以进行合理的分配内存和数据运算。

* + 1. 为什么要划分为不同的类型呢？

1.计算机自动为不同类型的数据分配不同的内存空间。合理使用数据类型可以节约内存资源。

2.程序在执行过程中一般都是同种类型的数据之间运算。

* + 1. 数据类型的分类

Javascript中类型分为两大类，基本数据类型与引用数据类型(后面课程)。

Javascript支持5种基本数据类型： number(数字)、string(字符串)、boolean(布尔型)、undefined(未定义)和null(空)

例如：

var name = '凤姐'; // string（字符串）

var age = 21; // number(数字)

var isMarry = true; // boolean（布尔型）

* + 1. 数字（number）

日常生活中使用的数字和进制表示法都是数字类型的数据，都可以使用一个数字类型的变量保存该数据。

数字类型又分为两种： 整数（int）和小数(float)

* **整数（int）**

整数可以使用以下四种表示形式：

十进制：日常生活使用的数字。 （常用的方式）

八进制：以0开头的数字

十六进制：以0x开头的数字

二进制：是以0或者1组成的数字

var number\_1 = 15; //十进制表示法

var number\_2 = 017;//八进制表示法

var number\_3 = 0xF;//十六进制表示法

//但是程序的输出都会将上面的转换为十进制表示法输出

console.debug(number\_1,number\_2,number\_3)

Javascript使用变量无法表示二进制。

* **小数（float）**

带有小数点的数字使用float类型的值。

var height = 1.75;

* **特殊值**

特殊数值：NaN(计算错误，类型转换失败)

Infinity(除数为零)

* + 1. 字符串(string)

使用“”或者是‘’表示的数据是字符串类型的数据。

var name = '张三'; //姓名

var sex = "男"; //性别

var intro = "他是一个很好很好的孩子！"; //简介

特殊的用法：

如果字符串的数据中出现特殊的符号需求使用转义字符（\）额外处理。

|  |  |  |  |
| --- | --- | --- | --- |
| **转义符** | **含义** | **转义符** | **含义** |
| \’ | 表示单引号 | \” | 表示双引号 |
| \n | 表示换行符 | \\ | 表示反斜线 \ 号本身 |

* + 1. 布尔类型（boolean）

boolena类型的值只有两个： true或者false，代表真假性。

var isMarry = true;

* + 1. 未定义(undefined)

变量定义后从未赋值时，该变量的默认值为undefined。

数据类型undefined只有一个值undefined。

注意：undefined代表，一个变量已经定义但是未复制，默认值是undefined.

var address;

console.log(address);//undefined

如果我们直接使用未定义的变量将会报错！

//var age; 代码被注释了

console.log(age); //age not defined , age这个变量没有定义；

* + 1. 空(null)

数据类型null只有一个值null。

可以通过给一个变量赋 null 值来清除变量的内容。

* + 1. null与undefined比较（了解）

面试题：

相同点：null,undefined均代表无值；

不同点：undefined代表一个变量如果没有赋值，那么值是undefined,变量的默认值;null代表一个变量如果是存储的值是**引用类型**，但是目前对象还没有创建，那么就可以设置为null，另外如果一个应该函数有返回值，如果最终没有数据作为返回值，那么也会返回null。

* + 1. 确定变量的数据类型

变量的数据类型会随着变量值的变化而变化。 说明变量的数据类型由值来决定。

* 1. 通过下面的代码可以获取变量在执行过程中的类型：

typeof （变量名）

var value = '凤姐'; // string（字符串）

console.debug(typeof value);

var value = 21; // number(数字)

console.debug(typeof value);

var value = true; // boolean（布尔型）

console.debug(typeof value);

* 1. Javascript中的类型转换
     1. 什么是类型转换

将一种类型转为另外一种数据类型.例如: 将string类型的’1’转换成数字类型的1.

* + 1. 为什么要进行类型转换

因为数据之间的运算只有同类型之间的运算, 如果不同类型之间存在运算的话,需要将其转换为同一种类型后再进行运算.

类型转换分为两种:

强制类型转换, 自动类型转换

* + 1. 强制类型转换

使用parseInt()和parseFloat()将一种类型的数据到数字类型.

parseInt（变量名）：将变量转换为整数 把小数和不是 数字的部分去掉；字符串 必须前面是数字；

parseFloat(变量名)：将字符串转为小数

Number (变量名)：将变量转换为数字，只能数字构成字符串才可以转换

Boolean(变量名)：转换为布尔值，true/false;

String(变量名):转换为字符串；

Number(true) 1

Number(false) 0

* + 1. 自动类型转换

在运行过程中根据编程语言的运算的语意环境,Javascript会自动进行类型转换.

<script type="text/javascript">

var value1 = "2";

var value2 = "1"

//减法的语意： 将会按照数字进行运算

var value3 = value1 - value2;

//加法： 将会按照数字进行运算

var value4 = true;

var value5 = value1 - value4;

console.debug(value5);

</script>

在运行过程中根据上下文环境,Javascript会自动进行类型转换.

最经典的实例为:if语句中的条件.

if(条件表达式){

}

**因为条件表达式的期望结果为boolean类型. 所以放在条件表达式的任何值都会被自动转换成boolean类型.**

//0 就会根据语意转化为false

if(0){

//if成立之后要执行的代码

}

1. 运算符
   1. 有哪些运算符
2. 算术运算符： + - \* / % ++ --
3. 比较运算符：== != > >= < <= ===(全等于) !==(不全等于)
4. 条件运算符：(expr1) ? (expr2) : (expr3)，

例：

$v1 = $fs>60 ? "及格" : "补考" ; //则v1根据fs的值有两个情况

1. 逻辑运算符：&&（与） ||（或） ! (非)
2. 字符串运算符：+
3. 赋值运算符： = += -= \*= /= %=
   1. 算术运算符

算术运算符： + - \* / % ++ --

%取模运算符(取余)的符号有 被除数决定了正负性。

**++,--分为前,后.**

**放在前面: 先自身+1运算,再和其他的运算数计算.**

**放在后面: 先和其他的运算数计算,再自身+1运算**

* 1. 比较运算符

比较运算符：== != > >= < <= ===(全等于) !==(不全等于)

==: 只需要变量值相等

===(全等于): 变量值相等,并类型也需要相等.

* 1. 三元(目)运算符

条件运算符：(expr1) ? (expr2) : (expr3)，

条件？表达式1：表达式2🡪三目运算符 有一个结果 当条件成立时 结果为表达式1结果 否则为表达2的结果；

var num1 = 10;

var num2 = 5;

var max = (num1>num2) ? num1 : num2;

console.log(max);//10

* 1. 逻辑运算符

变量：

var x = 6;

var y = 3;

&& 逻辑与，当左右两边操作数都为true时，返回值为true，否则返回false。

语法： 表达式1 && 表达式2；

(x < 10 && y > 1) 为 true

| | 逻辑或，当左右两边操作数都为false时，返回其中第一个不为false的值或者最后一个值。

语法： 表达式1 || 表达式2；

(x==5 || y==5) 为 false

!逻辑非，当操作数为true时，返回值为false，否则返回true。

语法： ！表达式；

!(x==y) 为 true

技巧：

1. && 逻辑与，只有所有值为true,才可能返回最后一表达式的值；
2. | | 逻辑或，从左向右运算表达式，如果表达式为true，就返回表达式的值，如果为false,继续向后寻找表达式为true的值；
3. ！逻辑非； !!表达式可以将任何数据类型转换为布尔值;

JavaScript中所有的值都具有真假性，0、""、false、null、undefined、NaN为false，其他的数据均为true。

* 1. 字符串连接符号（+）

一个表达式中，字符串与任何变量进行“+”运算，都是进行字符串连接，运算结果为一个新的字符串。

var s = “二狗”;

var s1 = s+”250”;

var s2 = s+null;

* 1. 赋值运算符

=：赋值运算符

+=： a+=b 🡺 a=a+b;

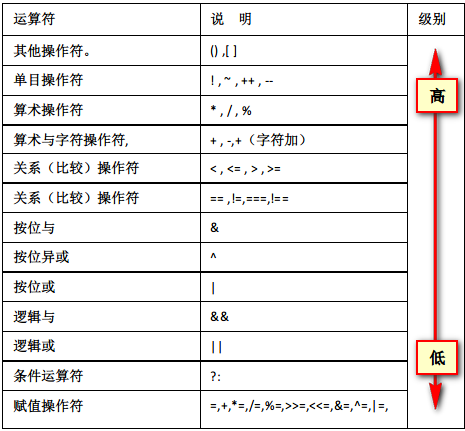
-=: a-=b 🡺 a=a-b;

\*=: a\*=b 🡺 a=a\*b;

/=： a/=b 🡺 a=a/b;

%=: a%=b 🡺 a=a%b;

* 1. 运算符优先级（运算符结合性）



优先级越高的优先运算，不用死记该图片，可以使用（）提升优先级

赋值优先级最低

先乘除后加减

优先级相同情况下,从左到右依次计算.

1. 位运算符（了解）
   1. 什么是位运算

位运算是针对十进制整数在二进制的基础上进行运算的. 即需要将十进制数转换成二进制之后才进行按对应的每位上值的运算.

二进制是计算技术中广泛采用的一种数制。二进制数据是用0和1两个数码来表示的数。

二进制就是等于2时就要进位。

0=00000000

1=00000001

2=00000010

3=00000011

4=00000100

5=00000101

6=00000110

7=00000111

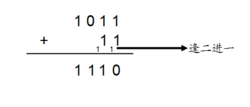
8=00001000

9=00001001

10=00001010

……

求二进制数 1011+11的和：

[](http://baike.baidu.com/pic/äºè¿å¶/361457/0/d1a20cf431adcbef74d1243eaeaf2edda3cc9f24?fr=lemma&ct=single)

二进制的按位运算符：

&(与) 两个运算数相应的位都为1，结果为1，否则为0；

|(或) 两个运算数相应的位某一个为1，结果为1，否则为0；

^(异或) 两个运算数相应的位某一个为1，结果为1,但不能同时都为1；

~(非) 一元操作，按运算数所有位取反。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 二进制位a | 二进制位b | a与（&）b | a或（|）b | a异或（^）b | ~(非)a |
| 0 | 0 | 0 | **0** | **0** | **1** |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | **1** | 1 | **0** | **0** |

**var a=0,b=1;**

**console.log(a & b);**

**console.log(a | b);**

**console.log(a ^ b);**

**console.log(~a);**

将0看作假(false)

将1看作真(true)

* 1. 为什么使用位运算

程序中的所有数在计算机内存中都是以二进制的形式储存的。位运算说穿了，就是直接对整数在内存中的二进制位进行操作. 这样效率就高了，但可读行不高，不够直观，实际开发中使用的并不多。

* 1. 位运算在哪里使用呢?

记录开关类事物的状态: 位运算是基于二进制运算的,而二进制中的每一位只有0,1的两个值. 可以通过二进制的位上值记录多个不同的开关状态. 这样不仅提高了效率也节约了空间.

例如记录1—32盏灯的开关状态.

c = a++ + b;

c = a-- + b;

a = c++ +b;

b = a-- + c++;