

JavaScript 第二章

本章主要内容

1. 什么是运算符
2. 按照用途进行分类
3. 按照操作值个数分类
4. 运算符优先级

一、运算符

概念：

用于操作数据或者变量的符号，特定的符号有特定的用途！

二、按照用途进行分类

算数运算符

+ 加
- 减
***** 乘
/ 除
% 求余/取余/求模/取模
++ 递加
前置型：
对于表达式，先执行自身++运算再参与表达式运算

后置型：
对于表达式，先参与表达式的运算，再执行自身++运算

不管前置还是后置，对于自身来说都是+1
-- 递减
前置型：
对于表达式，先执行自身--运算再参与表达式运算

后置型：
对于表达式，先参与表达式的运算，再执行自身--运算

不管前置还是后置，对于自身来说都是-1

赋值运算符

=	赋值运算符
+=	先加再赋值
-=	先减再赋值
*=	先乘再赋值
/=	先除再赋值
%=	先求余再赋值

比较运算符

==	判断值是否相等
===	判断值和类型是否同时相等
>	大于
<	小于
>=	大于等于
<=	小于等于
!=	不等于 (值不相等)
!==	不全等 (值和类型不同时相等)

执行规则：

1. 比较运算符最终返回的结果肯定是布尔值，要么真 `true`，要么假 `false`
2. 符合运算符的条件，则返回真；不符合，则返回假

逻辑运算符

! 逻辑非, 取反
原先为假, 现在为真
原先为真, 现在为假

&& 逻辑与

1. 左右两侧表达式都是布尔值
两侧同时为真, 则返回值
只要有一侧为假, 就返回假
2. 两侧表达式其中至少有一个不是布尔值时:
左侧为真, 返回右侧的值
左侧为假, 返回左侧的值
3. 逻辑与有逻辑断点(短路)操作
当左侧表达式为假, 右侧表达式不再执行

|| 逻辑或

1. 左右两侧表达式是布尔值
其中有一个为真, 就返回真
两侧表达式同时为假, 才返回假
2. 当两侧表达式其中至少有一个不是布尔值时:
左侧为真, 返回左侧的值
左侧为假, 返回右侧的值
3. 逻辑或有逻辑断点(短路)操作
当左侧表达式为真时, 右侧表达式不再执行

字符串连接运算符

在JS中+号是一个特别的符号
如果左右两侧都是数值, 执行算术加法运算
如果其中有一个是字符串, 执行字符串连接

猜结果:

```
var a = 100+100;           // a的值  
var a = 100 + '100';      // a的值  
var a = 100+ '100' + 100; // a的值
```

位运算符

&	按位与
	按位或
~	按位非

注：位运算时要转成对应的二进制进行相应运算，目前了解即可

按照操作值个数分类

一元运算符

++ -- !

二元运算符

+ - * / +=

三元运算符

也称之为三目运算符

格式：

表达式?真区间:假区间

执行规则：

当表达式为真时，执行真区间；

当表达式为假时，执行假区间

运算符优先级

运算符	描述
. [] ()	字段访问、数组下标、函数调用以及表达式分组
++ -- ~ ! delete new typeof void	一元运算符、返回数据类型、对象创建、未定义值
* / %	乘法、除法、取模
+ - +	加法、减法、字符串连接
<< >> >>>	移位
< <= > >= instanceof	小于、小于等于、大于、大于等于、instanceof
== != === !==	等于、不等于、严格相等、非严格相等
&	按位与
^	按位异或
	按位或
&&	逻辑与
	逻辑或
?:	条件
= oP=	赋值、运算赋值
,	多重求值