操作符

掌握各种操作符的用法

操作符

- 操作符，也叫运算符 operator ， 是 js 中发起运算最简单的方式。

  例如：5 + 6

- 表达式（expression） 的组成包含操作数和操作符，表达式会得到一个结果，然后用结果参与程序

操作符

掌握各种操作符的用法

- **算术运算符**
- 比较运算符
- 逻辑运算符
- 赋值运算符
- 一元运算符
- 运算优先级

算术运算符

- + - * / % ()
- %:取余，取模。 a / b = c 余 d，就说a % b = d。
- 运算顺序：先算乘除取余、再算加减，有小括号先算小括号。

正常情况

- 数字与数字之间的运算。

```
console.log(1 + 2);
console.log(1 * 2);
```

非正常情况 1

- 有特殊值字面量参与的运算。
- NaN参与的运算：得到的结果都是NaN
- Infinity参与的运算，视情况而定

- 工作中并不会使用特殊值运算，没有实际应用的意义，但是要了解，以防面试遇到。

非正常情况 2

- 其他类型的数据参与数学运算。
- 有字符串参与的 + 运算：+ 号变为连字符将前后连接成整体字符串。
- 隐式转换：除了字符串参与的 + 运算，其他情况下，所有其他数据类型参与数学运算时，计算机暗中将其他数据类型先自动转换成数字类型，再参与运算，这个过程中不需要使用 parseInt()、Number() 等方法，过程是暗中进行，这就是一个隐式转换的过程。

隐式转换

其他的数据类型会隐式转换为数字类型：

- 对应数字：纯数字字符串会转为对应的数字”123”→123
- 转换为1： true
- 转换为0：false、null、“”空字符串、空白字符串
- 转换为NaN：undefined、非空非纯数字字符串

操作符

掌握各种操作符的用法

- 算术运算符
- **比较运算符**
- 逻辑运算符
- 赋值运算符
- 一元运算符
- 运算优先级

比较运算符

- 也叫作关系运算符。一个 比较运算符 comparison operator 比较它的操作数并返回一个布尔类型值。运算结果要么是true，要么是false。

>	大于
<	小于
>=	大于等于
<=	小于等于
==	相等，只判断值大小是否相等，不判断数据类型
!=	不等，与相等完全相反
===	全等，不光判断值相等，还要判断数据类型相等
!==	不全等，与全等于完全相反

正常情况

- 数字与数字比较。

```
console.log(7 > 8);
console.log(7 < 8);
console.log(3 ⩽ 4);
console.log(3 ⩾ 4);
console.log(5 == 5);
console.log(5 ≠ 5);
console.log(5 === 5);
console.log(5 ≠= 5);
```

非正常情况 1

- 特殊值参与比较运算。
- NaN参与：不等于和不全等于结果是 true，其他的都得到 false。
- Infinity参与的运算，视情况而定

非正常情况 2

- 其他数据类型参与比较运算（排除字符串与字符串的比较）。
- 其他数据类型也会隐式转换为数字参与比较。

  “123”→123 true→1 false→0 null→0 undefined→NaN “”→0 “abc”→NaN
- null 的判断比较特殊：null 与 0 判断时，相等判断为 false，>= 和 <= 判断为 true
- null == undefined

非正常情况 3

- 字符串与字符串比较。
- 不会发生隐式转换为数字，而是比较两个字符串的 Unicode 编码顺序。
- 字符编码顺序：从前往后 0-9，A-Z，a-z，前面的小于后面的。
- 比较时，不关心两个字符串的长度，从第一个字符开始比较，依次往后顺延比较，直到比较出大小，就不再往后比较。

比较运算符运算顺序

- 从前往后比较，前面的结果与后面的进行比较。
- 3 > 2 > 1

操作符

掌握各种操作符的用法

- 算术运算符
- 比较运算符
- **逻辑运算符**
- 赋值运算符
- 一元运算符
- 运算优先级

逻辑运算符

- 逻辑运算符常用于布尔类型值之间; 当操作数都是布尔值时，返回值也是布尔值。

&&	逻辑与运算符 且
||	逻辑或运算符
!	逻辑非运算符

正常情况

- 布尔类型的值参与运算，返回值为布尔值。

```
console.log(true && true);
console.log(true && false);
console.log(false && true);
console.log(false && false);
console.log(true || true);
console.log(true || false);
console.log(false || true);
console.log(false || false);
1console.log(!true);
console.log(!false);
```

非正常情况

- 除了布尔类型的值之外，其他数据类型的值也可以参与逻辑运算。运算过程中需要将操作数隐式转换为布尔类型的值，参与判断计算，最终运算结果还是原来的某个位置的数据。
- 并不是所有逻辑运算返回结果都是布尔值，其他数据参与得到的就是数据本身。

隐式转换为布尔值的规律

- 转为false：NaN、0、“”空字符串、null、undefined
- 转为true：非0 非NaN数字、非空字符串
- 当它们用于非布尔值的时候，返回值就可能是非布尔值。其实这种运算非常简单，就两句话：
- (逻辑与 a&&b) 如果a能被转换为false，那么返回a；否则，返回b。
- (逻辑或 a ||b) 如果a能被转换为true，那么返回a；否则，返回b。

逻辑运算符运算顺序

- 同种运算符从前往后运算。
- 综合运算顺序：非、与、或。

操作符

掌握各种操作符的用法

- 算术运算符
- 比较运算符
- 逻辑运算符
- **赋值运算符**
- 一元运算符
- 运算优先级

赋值运算符

- 赋值运算符必须有变量参与运算，赋值运算符会做两件事情：
- 第一，将变量中原始值参与对应数学运算，与右侧的数据。
- 第二，将运算结果再重新赋值给变量。
- 变量位于操作符的左侧。

赋值运算符符号

=	等于
+=	加等于
-=	减等于
*=	乘等于
/=	除等于
%=	取余等于
++	递加
--	递减

```
var a = 5;
// 加等于
a += 3;    //等价于a = a + 3
console.log(a); //8
```

操作符

掌握各种操作符的用法

- 算术运算符
- 比较运算符
- 逻辑运算符
- 赋值运算符
- **一元运算符**
- 运算优先级

一元运算符

- ++ 和 -- 也叫一元运算符，只有一个操作数。
- ++ 或 -- 符号可以写在变量前和变量后面，位置不同可能导致程序运行结果不同。
- 以 ++ 为例：
- a++：++ 符号在变量之后，a++ 在参与程序过程中使用的原始没有加 1 的值，使用完后第二次用 a 变量时，a 用的就是加 1 后的新值。先参与，后自加。
- ++a：++ 符号在变量之前，++a 在参与过程中整体就使用 a 加 1 之后的新值，使用完后第二次用 a 变量时，a 用的也是加 1 的新值。先自加，后参与。

操作符

掌握各种操作符的用法

- 算术运算符
- 比较运算符
- 逻辑运算符
- 赋值运算符
- 一元运算符
- **运算优先级**

运算优先级

- 运算优先级也可以叫综合运算顺序。
- 优先级从高到底
    1. () 优先级最高
    2. 一元运算符 ++ -- !
    3. 算数运算符 先* / % 后 + -
    4. 关系运算符 > >= < <=
    5. 相等运算符 == != === !==
    6. 逻辑运算符 先&& 后||
    7. 赋值运算符

案例

```
var a = 4;

var num = 1 * (2 + 3) && ++a || 5 > 6 && 7 < 8 || !9;
```

拉勾教育

—互联网人实战大学—

下载「拉勾教育App」
获取更多内容