复习

复杂查询

聚合查询/分组查询 GROUP BY

子查询

多表查询

JS

浏览器/服务器NodeJS

变量和常量

var a=1; a=2;

字母、数字、下划线、美元符号

var b,c;

const pi=3.14;

学习一门编程语言的基本步骤

(1)了解背景知识: 历史、现状、特点、应用场景

(2)搭建开发环境: 编写hello world

(3)变量和常量

(4)数据类型

(5)运算符

(6)逻辑结构

(7)通用小程序

(8)函数和对象

(9)第三方的库和框架

(10)实用的项目

1.数据类型

原始类型：数值型、字符串型、布尔型、未定义、空

(1)数值型

分为整型和浮点型

八进制 以0开头的数字 例如 010 -> 8

十六进制 以0X开头的数字 例如0XF ->15

A~F 代表10~15 不区分大小写

0XFF -> 255

15\*16+15=255

浮点型分为定点小数和指数型小数

3.14e3 3140

3.14e-3 0.00314

typeof 检测数据类型

(2)字符串型

数据被引号所包含就是字符串型，不区分单双引号

查看任意一个字符的Unicode码

'a'.charCodeAt() //97

(3)布尔型

true/false

在程序中表示真或者假

常用于一些是否结果，例如是否登录、是否注册；通常一些运算的结果返回布尔型的值

(4)未定义型

只有一个值undefined

声明了变量未赋值，返回undefined

(5)空

只有一个值 null

常和引用类型的数据一起使用

2.数据类型转换

(1)隐式转换

在运算的过程中，自动转换

1)数字+字符串 数字转成字符串

1+'2' // '12'

2)数字+布尔型 布尔型转成数字 true->1 false->0

3+true //4

3+false //3

3)字符串+布尔型 布尔型转成字符串

'5'+true //'5true'

JS中加号(+)的作用

执行加法运算

执行字符串的拼接，前提是加号的两侧至少有一个值为字符串

减法(-)、乘法(\*)、除法(/)隐式转换

如果运算符的两端不是数字，则会自动调用Number转为数值型

练习:查看以下程序运行结果

var num1=3,num2=true,num3='tedu';

console.log(num1+num2+num3);//'4tedu'

console.log(num2+num3+num1);//'truetedu3'

console.log(num3+num1+num2);//'tedu3true'

(2)强制转换

1)强制转为数值型

Number( 数据 )

|  |
| --- |
| Number('2') //2  Number('2a') //NaN  Number(true) //1  Number(undefined) //NaN  Number(null) //0 |

NaN -> Not a Number 不是一个数字，NaN和任何值执行加减乘除等返回的结果还是NaN

2)将数据转为整型

parseInt( 数据 )

|  |
| --- |
| parseInt(3.14) //3  parseInt('4a') //4  parseInt('a4') //NaN  parseInt(true) //NaN |

常用于将小数和字符串转为整型，如果要转换的数据中开头含有非数字，返回NaN

3)将数据转为浮点型

parseFloat( 数据 )

和parseInt的用法基本一样，只是转的数据为浮点型

4)将数值和布尔型转字符串

toString()

var n=1;

n.toString() // '1'

3.运算符

表达式: 由运算符连接的操作数据，组成的形式

运算符分为算术运算符、比较运算符、逻辑运算符、位运算符、赋值运算符、三目运算符

(1)算术运算符

+ - \* / % ++ --

% 取余/取模

++ 自增 在原来的基础之上加1

-- 自减 在原来的基础之上减1

console.log(a++) 先打印a的值，然后再执行自增

console.log(++a) 先执行自增，然后再打印a的值

练习：查看以下程序的执行结果

var num=3;

console.log( num++ + ++num );

(2)比较运算符

> < >= <= ==(等于) != ===(全等于) !==

返回一个布尔型的值

== 只是比较值是否相等

=== 不仅比较值，还会比较类型

|  |
| --- |
| 如果数据类型，会发生数据类型转换 |

3>'10' //字符串转为数字

'3'>'10' //比较的是首字符的Unicode码

3>51 1->49

3>'10a' //false

3<'10a' //false

3=='10a' //false

NaN==NaN //false

NaN和任何值比较(> < >= <= == ===)都返回false。

(3)逻辑运算符

并且 && 关联的两个条件都是true，结果是true，否则false

或者 || 关联的两个条件只需要有一个为true，结果是true，否则是false

非 ! 取反 !true ->false !false->true

**逻辑短路**

&& 当第一个条件为false，就不再执行第二个条件

|| 当第一个条件为true，就不再执行第二个条件

注意：逻辑短路无需关注最终结果是true还是false，重点是看第二个表达式是否会执行

练习: 以下程序是否会报错

|  |
| --- |
| var num=3;  num>5 && console.log(a);  num<1 || console.log(a); |

练习：声明变量保存年龄，如果满18岁，打印'成年人'

(4)位运算符(掌握)

在执行计算的时候，计算机把数据转成二进制进行运算

1 10 11 100 101 110 111 1000 1001 1010

2 4 8

100110=100000+100+10

32 4 2

110111=100000+10000+100+10+1

32 16 4 2 1

25=16+8+1

10000+1000+1=11001

按位与(&) 上下两位都是1，结果是1，否则是0

按位或(|) 上下两位含有1，结果是1，否则0

按位异或(^) 上下两位不同为1，相同为0

课后任务

(1)复习今天内容，整理思维导图

(2)练习

27&49 43^18 15|19

声明变量保存年份，判断这个年份是否为闰年，结合逻辑短路，如果是闰年，打印' 闰年 '

闰年：

(3)预习逻辑结构 if if-else if-else 嵌套