js第二天

复习第一天

获取元素

- 通过ID获取元素
  - `document.getElementById('元素ID名称')`
- 通过类名获取元素
  - `document.getElementsByClassName('元素的ClassName')[0]`
  - 通过类名获取的元素，我们得到的是一个集合，想要获取某个元素，需要用对应的索引`[0]`来获取
  - 索引都是从0开始的；
- 通过标签名获取元素
  - `document.getElementsByTagName('元素标签名称')[0]`
  - 通过标签名获取元素和通过类名一样，得到的是一个集合，也需要`[0]`索引来获取；

变量（var let）

可变的量

- 定义变量
  - 语法：`var oDiv = document.getElementById('元素ID名称')`；或
  - `let oDiv = document.getElementById('元素ID名称')`

常量（const）

不可变的量

- 定义常量
  - 语法：`const oDiv = document.getElementById('元素ID名称')`

点的用法

点的作用

- 点 从属关系 翻译成汉语就是“的”；
- 用点的地方都可以用[]；但是中括号里必须写成字符串`''`.
- 但是能用[]的地方，不一定能用点.。
  `debugger`断点
  用来停顿系统运行，来检查代码；

数据类型

数据类型包括基本数据类型和引用数据类型

> 二者的区别在于基本数据类型操作“值”，引用数据类型操作“地址”；

一、基本数据类型
string 字符串
定义：用单引号、双引号、反引号包含的部分都叫字符串
注意双引号内不能包含双引号，单引号内不能包含单引号
比如：`var str2 = "as\"df";`
用`\"`来转义里边的双引号只是双引号本身。
举例：

- `oDiv.innerHTML = "str3"` 给的值就是字符串”str3”
- `oDiv.innerHTML = str3` 给的值是变量名str3对应的值

字符串的拼接
1. `oDiv.innerHTML = str+str3+str2;` 就是把各个变量名或者字符串用加号`+`链接起来。
2.模板字符串

```
oDiv.innerHTML = `<h1>
    // //          <div>
    // //              珠峰
    // //          </div>
    // //      </h1>
    // // `
```

模板字符串就是给要加的内容用反引号（`）包住。里边如果需要用变量名可以加(
)字符串拼接一种是直接＋；若是反引号还可以通过{}

number 数字

number 数字包括:
整数、小数、NaN(not a namber)

```
var n2 = '1' + 2 ; // 字符串的 12
    var n3 = 1 + 2 ; // 数字的 3
    console.log(n2,n3);// 12   3

    var n4 = '1' + 2 + 3; // '12' + 3 '123'
    var n5 = 1 + 2 + 3; // 3 + 3 6
```

以上这时各个变量输出的值：可以看出：**对于`+`两边只要有一个是字符串类型的，那么输出的值肯定是字符串拼接；**

- Number（'数字'）字符串转义成数字，可以把数字字符串转义成数字类型，然后再去运算 但是**Number 转化的时候， 字符串中有非数字(不含小数点)， 结果就是 NaN**
  比如 Number（'1234qq'） 转化的值就是NaN
- **// 四则运算 + - * / % 除了+ ;其他的都会自动把符号两边转成数字再去运算**相当于使用Number一样，字符串中有非数字，结果就是NaN。
- =号
  - 一个 `=` 是赋值； 俩 `==` 是相对比较； 仨 `===` 是绝对比较

```
// '1' == 1;/// true
    // '1' === 1;// false
    // NaN == NaN    false   NaN 和谁 都不相等
    // 'NaN' == NaN  false
    // 'NaN' == 'NaN'  true
```

NaN和任何数字做运算结果都是NaN

- parstInt 整数

```
 parseInt('12.3') // 12
    parseInt('12.3q')// 12
```

后边属于输出的值，只输出整数；
// 从左向右查看 遇到非数字(包含小数点)即停，获取到的是前边的数字部分
// 若第一位就是非数字，则结果是 NaN

- parsfloat

```
// parseFloat('12.3q')
    // parseFloat('1q2.3q')
```

// 从左向右查看 遇到非数字(不包含小数点)即停，获取到的是前边的数字部分
// 若第一位就是非数字，则结果是 NaN

isNaN 判断数字是不是NaN

`isNaN('12.3q') //先用Number处理 括号内的参数`

Boolean 布尔值

Boolean 只有两个值：// true false
Number():将其他数据类型转化成数字类型
Boolean():将其他数据类型转化成布尔类型（true或false）

- 数字类型中，只有0和NaN转换成布尔值是false，其他都是true
- 字符串中，只有`''`空字符串转换成布尔值是false 其他都是true；
- 两个特殊的，null和undefined转换成布尔是false；
  布尔值小总结，——除了 0 NaN 空字符串” null undefined这五个值，是false 其他都是true
- 取反（!）
  - 语法 !值
  - 原理：取非、取反。先把!后边的值转换成Boolean值。然后再去取反；ture或者false
  - 取反再取反（!!）
  - 原理和`!`一样。结果相当于用了一个Boolean（ ）；

-null 空指针对象 undefined 未定义对象

举例：

```
var a = document.getElementById('oDiv');
```

例如以上代码；在body中。未找到oDiv这个ID；返回的只就是null 空指针对象

```
var b;
```

例如以上代码：在stript中，为找到B变量名，表示从未定义过这个变量；返回值是undefined;

小知识（查看数据类型）

typeof值：查看该值的数据类型

```
 console.log(typeof 1) // 'number'
```

```
    console.log(typeof '1') // 'string'
    console.log(typeof true) // 'boolean'
    console.log(typeof undefined) // 'undefined'
    console.log(typeof null) // 'object'
    console.log(typeof typeof 1);// 'string'
```

以上代码则是各个基本数据类型查看后的返回值；
特例：null类型的返回值不是null；则是object；

引用数据类型

- 对象
  - 普通对象
  - 定义普通对象 `var obj = { }`

```
var obj = {
        name:"小红",
        age:9,
        1:2,
        _:2,
        $:3,
    };
```

例如以上代码

- 大括号里边的值叫做（键值对–>属性名：属性值）其中属性值可以是任何数据类型，但是属性名则统一都是字符串类型；
- 获取属性值有两种方式
  - 1，打点的方式
  - 2， [ ]的方式
    打点的方式完全等同于 [ ]的方式；**如果打点的前后有数字类型则只能用[]**
    对象中， 若没有对应的属性时，获取到的就是一个 undefined