JS第一周第三天

`javascript`

1.运算符

1.逻辑运算符

- `或||` 表达式：如果前面是true就返回前面的内容，如果前面是false就返回后面的内容（只管前面内容，不管后面内容）
  输出的都是最终的结果，不是一个表达式（有运算符的是表达式）

```
console.log{false||1} //1
//如果前面不是布尔值，默认调用boolean方法将其变成布尔值
console.log{0||2} //2
```

- `与&&`
  前面是true，返回后面的内容；前面是false，返回前面的内容。（判断结果为false的，出来的是前面的内容，不是false）

2.++和- -

1.在原来的基础上加1减1
2.如果不是数字会默认转为数字（调用number方法）
`区别i++和++i`

- i++ 先调用再增加（调用完就会增加，后面的i值会改变）
- ++i 先增加再调用

3.等号

1.”=”为赋值

2.”==”为比较，判断两边是否相等，结果为布尔值 如果两边是不同数据类型，需要进行数据类型的转换

- 数据类型比较
  1.number中NaN和任何值比较都是false
  2.null和undefined 跟其他数据类型比较的时候都是false，但是他俩之间是相等的
  3.数字 == 布尔 将布尔值转为数字（number方法）
  4.数字 == 字符串 将字符串变为数字进行比较（number方法）
  5.字符串 == 布尔值 两边都转为数字进行比较
  6.数字 == 引用数据类型 先转为字符串（toString方法）再变成数字（number方法）；像对象、时间、正则、函数没有相等的，唯一有可能相等的只有数组，而且是空数组或者数组只有一项且为数字
  7.字符串 == 引用数据类型 将引用数据类型变为字符串再比较
  8.布尔值 == 引用数据类型 两边都转数字（引用数据类型先转为字符串再转为数字）
  9.引用数据类型 == 引用数据类型 比较的是地址值，看这两个是不是共用的一个地址

3.” === ” 为绝对相等，先判断数据类型是否一样，再去判断值或地址值是否一样，即数据类型不同结果肯定为false

4.”!=”为不等于，按照” == “的方式判断

5.” !== ” 按照 ” === “的方式判断

4.函数

- 函数定义：function 函数名（参数）{函数体}
- 函数执行：函数名（ ），如果不加括号就不是执行，而是得到整个函数的定义部分
  函数执行实际上是让函数体执行

一般我们会将实现一定功能的代码封装成一个函数，后期如果想实现这样的功能只需执行这个函数，提高代码的利用率。

- 形参：用来承接函数执行的时候传进来的参数
- 实参：确切的值，函数执行的时候传入

函数执行时给形参赋值，赋的值就是实参。
如果没有传实参，就是undefined

5.函数的arguments

```
function fn(){
    //实参集合
    //类数组（类似数组的对象），有length，有索引
    console.log(arguments);
```

```
}
```

6.函数的return

```
    function fn(){
        return 1
    }
    let res=fn()//给res赋值，赋的值就是fn执行的返回值，就看函数有没有retur
n，没有就是undefined，有就是return后面的值
    //函数中return后面的值就是函数的执行结果也叫返回值
```

return后面的代码不会执行

7.自执行函数

匿名函数，也可以加一个（ ）执行=>自执行函数 定义和执行一起

```
    (function(形参){
        console.log("自执行函数")
    })(实参)
```

8.函数的练习

当函数没有return，就没有返回值，console.log(函数)的执行结果为undefined，即使函数运行出一个值。

9.if判断

```
if(条件){
    条件成立(true)之后执行的代码
}
//如果条件不是布尔值，就会转为布尔值

if(条件){
    条件成立(true)之后执行的代码
}else{
    否则(条件不成立)执行的代码
}
```