上课笔记

## JS的简单认识：

**一个页面有三个层：**

结构层：HTML标签

修饰层：CSS

逻辑层：JS 一无所知

JS负责页面逻辑，有个动作：事件 （比如：鼠标滑动或者点击事件，出现的一些页面效果，如果用css写不出来，就用js来写）

## 事件三要素：

事件源：你操作的对象（DOM节点），document.querySelectors(‘选择器’)

事件：click input focus keydown keyup

事件处理函数：要实现的效果需要一个函数

## 变量类型：

代码在运行的时候，需要对变量进行 变量解析；

1. Number：数字;
2. String：字符串，文本，用引号引起来的；
3. Boolean：true false；
4. Null：声明了一个变量，指向的存储空间里面没有东西 ；
5. Undefined：声明了变量，没有赋值；
6. Object：对象类型，后面你每天都会看到；
7. Function：函数，普通函数，构造函数，对象里面的方法
8. Array：数组，后面每天都会看到；

Js在运行的时候，会把变量转换成她需要的数据类型；

## 运算符：

+: 加，字符串连接

-:减号，用于数字运算；

\*:用于数字运算

/:用于数字运算，用于数据类型转换；

比较运算符：== ===（考虑数据类型） != !== > >= < <=

## 逻辑运算符：

&&:与，都返回true的时候才返回true

||:或，都返回false的时候才返回false

!:非，取反

熔断机制

## 逗号运算符：

总是返回最后一个表达式的结果；

## 变量作用域：

全局变量：定义在顶级代码里面的变量；

局部变量：在函数内部声明的变量；

作用域链：代码在运行的时候，会对变量进行解析==》要找到变量对应的值==》在函数内部找==》到全局里面去找

## 变量提升：声明提升

只提升声明，不提升赋值；

我们写代码的时候尽量规避变量提升；

## 函数：

对公用的功能的封装，一处定义，处处使用，方便维护；

声明函数：

Function fn(){

}

函数后面跟上小括号表示函数的执行；

#### 形参和实参：

**形参**：函数名后面的小括号里面指定函数的形参，形参的作用域是局部变量，只能在函数内部使用；

**实参**：在调用函数的时候，提供的具体的值就是实参；

函数内部用arguments这个类数组来接收所有的实参。

#### 函数返回值：

Return 负责返回值；

在函数里面才有return；

Return后面的代码不会执行了；

#### 自运行函数：

团队做一个项目：

王磊：index.js

罗鑫：common.js

自运行函数可以形成一个封闭的空间，不会污染全局变量；（即自己的文件里面写的变量不会影响污染其他同事写的文件，因为可能两个文件会用到相同的变量，合并时就会发生污染，使用自运行函数之后形成各自的封闭空间，就可以规避这种污染情况，即时有相同变量也无所谓了）

NaN：number



**引用错误：Username 没有 定义；**

This的指向：

谁调用这个方法，this就指向谁；

This指向的是距离this最近的function的调用者

#### 高阶函数：

使用函数作为实参；

函数本身可以直接作为实参；

## 闭包：

在函数内部返回函数；

形成一个镜像；

应用在，空间不灭；

内存消耗特别大；

## Undefined的情况：

1. 声明的变量没有赋值；
2. 函数的形参没有传递实参；
3. 使用对象未定义的属性；
4. 如果函数没有返回值，则返回值是undefined；

## 数组：

是一类数据的集合；（即一般不会把不同类型的数据写到一个数组中去）

常用的方法和属性：

Length：获取元素的数量；

Pop():删除数组的最后一个元素并返回删除的元素值；

Push():从数组的尾部开始添加元素；

Shift()：删除数组的第一个元素;

Unshift()：在数组的开始位置追加新的元素；

Sort()：对数组的元素进行排序；

Splice()：可以对数组的任意位置的元素进行增删改操作；

Concat:连接数组；