**Week1:**

1. **Var, let & const**

**Var:** 全局变量，在全局范围或者整个函数本地定义变量，不管范围如何。

**Let:** let 允许声明使用它的块，语句，表达式变量的范围。

**Const:** 常量。块范围的，非常类似let语句但是不能重新分配而更改也不能重新声明。

**2）Function**

函数定义由function关键字组成，后面跟：

函数名称

函数的参数列表，用括号括起用，分割

定义函数的js 语句用大括号{}括起来

基本参数（数字）按值发送给函数；值将传达给函数，函数如果更改了参数的值则此更改不会反映在全局或者调用函数中

如果将对象（非原始值，Array或者用户定义的对象）作为参数传达并且函数更改了对象的属性则该更改在函数外部可见。

三种方式创建function

1）Functions

function myFunc(theObject) {

theObject.make = 'Toyota';

}

2）Function expression

let square = function(number) { return number \* number; };

3）Arrow functions

const power = (base, exponent) => {

let result = 1;

**for** (let count = 0; count < exponent; count++) {

result \*= base;

}

return result;

};

**3）Closures**

闭包，允许在函数内嵌套函数，内部函数授予对外部函数的所有变量和函数的完全访问权限但是外部函数无法访问内部函数的内部定义变量和函数。

**4）Class**

Class 是对象的蓝图，他包含用于储存数据的变量和用于对这些数据执行操作的功能。

为什么需要class？

1. Encapsulation:封装，将数据和在一个单元中处理该数据的方法分组
2. Inheritance:将数据项和函数排列到层次结构中
3. Reusability:可重用性
4. Maintainability:课维护性

Class 实际上是一个特殊函数：分为类表达式和类声明。

**Week2:**