**1. 什么是函数?**

\* 实现特定功能的n条语句的封装体

\* 只有函数是可以执行的, 其它类型的数据不能执行

**2. 为什么要用函数?**

\* 提高代码复用

\* 便于阅读交流

**3. 如何定义函数?**

\* 函数声明

\* 表达式

**4. 如何调用(执行)函数?**

\* test(): 直接调用

\* obj.test(): 通过对象调用

\* new test(): new调用

\* test.call/apply(obj): 临时让test成为obj的方法进行调用

//让一个函数成为任意指定的对象的方法进行调用

**5. 什么函数才是回调函数?**

  1). 你定义的

  2). 你没有调

  3). 但最终它执行了(在某个时刻或某个条件下)

**6. 常见的回调函数?**

  \* dom事件回调函数 ==>发生事件的dom元素

  \* 定时器回调函数 ===>window

  \* ajax请求回调函数

  \* 生命周期回调函数(后面讲)

**7.回调函数的好处**

1) 解决异步

2) 对函数的功能进行拓展

**8.立即执行函数表达式**

①理解

\* 全称: Immediately-Invoked Function Expression

②作用

\* 隐藏实现

\* 不会污染外部(全局)命名空间

在全局中命名的变量内存在页面关闭才会被释放

而在局部作用域命名的变量 在作用域销毁时释放内存

\* 用它来编码js模块

**9. this是什么?**

\* 任何函数本质上都是通过某个对象来调用的,如果没有直接指定就是window

\* 所有函数内部都有一个变量this

\* 它的值是调用函数的当前对象

**10. 如何确定this的值?**

\* test(): window

\* p.test(): p

\* new test(): 新创建的对象

\* p.call(obj): obj