* 函数的概念及作用

函数对任何语言来说都是一个核心的概念。 通过函数可以封装任意多条语句， 而且可以在任何地方、任何时候调用执行，即函数是定义一次但却可以调用或执行任意多次的一段 JS 代码。

JS中的函数也是对象，所以JS函数可以像其它对象那样操作和传递，我们也常叫JS中的函数 为函数对象。

* 函数的编写

ECMAScript 中的函数使用 function 关键字来声明，后跟一组参数以及函数体。

普通函数声明

函数表达式:函数赋值给变量（匿名函数）

匿名函数：自执行的函数。

事件处理函数：就是普通的函数，函数给事件专用的。

* 事件的概念种类及作用
* 事件，就是文档或浏览器窗口中发生的一些特定的交互瞬间。可以使用侦听器（或处理程序）来预订事件，以便事件发生时执行相应的代码。

焦点事件，当元素获得或失去焦点时触发；

鼠标事件，当用户通过鼠标在页面上执行操作时触发；

键盘事件，当用户通过键盘在页面上执行操作时触发;

* + onclick/ondblclick/onmouseover/onmouseout/onkeydown/onkeyup......
* 事件和函数的关系，以及综合运用
* 函数和循环结合打印九九乘法表
* 函数的好处（参考概念）

函数是用来帮助我们封装、调用代码的最方便的工具！

* 参数的声明（形参和实参）

ECMAScript 函数不介意传递进来多少参数，也不会因为参数不统一而错误。实际上，函数体内可以通过 arguments 对象来接收传递进来的参数（实参）

* 参数的传递
* 应用

1. 编写一个函数，计算两个数字的和、差、积、商（要求：使用传参的形式）
2. 编写一个函数，计算三个数字的大小，按从小到大顺序输出。
3. 使用函数完成任意数字阶乘的计算（传参）

* 函数的两种创建方式：声明式和赋值式；
* JS的编译和执行

JS的解析过程分为两个阶段：预编译期(预处理)与执行期。

预编译期JS会对本代码块(script)中的所有声明的变量和函数进行处理（类似与C语言的编译）， 但需要注意的是此时处理函数的只是声明式函数，而且变量也只是进行了声明但未进行初始化 以及赋值。执行期就是在编译后的基础上开始从上到下执行脚本，遇到错误时中断。

* 变量的声明提升
* return关键字

函数都有一条return语句，return语句导致函数停止执行，并返回它的表达式的值给调用者。 如果return语句没有相关的表达式，则返回undefined。

* 应用

1. 编写任意个数字，求和、差、积、商的函数。

* 综合应用

1. 某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：每位数字都加上5,然后用除以10的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换，请编写一个函数，传入原文，输出密文。