正课:

1. \*\*函数:

\*\*\*作用域, \*\*\*声明提前, \*\*\*按值传递

2. 全局函数

3. \*分支结构

1. \*\*函数:

什么是: 内存中存储一段代码段的存储空间。

代码段专门用于执行一项任务

为什么: 重用一段代码段!

何时: 只要可能反复使用的代码段，都要先保存在一个函数中，再反复使用函数！

如何: 2步:

1. 声明函数: 创建一个函数，保存一段代码

function 函数名(参数变量列表){

步骤清单;

return 返回值;

}

参数变量: 专门定义一项任务必须的数据的变量

何时: 只要一项任务的执行，必须某些数据时，就必须用参数定义.

如何:

定义: 在函数名后的()中，每个参数变量用逗号分隔

不用var！

函数内: 参数变量的用法和普通变量完全一样！

总结: 参数也是变量！

作用: 1. 接收传入函数中的数据

2. 可让函数变得更灵活！

返回值: 专门返回一个函数执行结果的一个数据

何时: 如果函数使用者，需要获得函数的执行结果时

2. 调用函数: 让引擎按照函数记录的步骤，多次执行。

为什么: 任何函数，不调用，绝不会自动执行！

只有调用，才执行，调用几次，执行几次.

如何: 变量=函数名(参数值列表)

强调: 参数值的顺序和个数要和函数定义时的参数变量列表保持一致！

\*\*\*作用域(scope):

什么是: 一个变量的可用范围

为什么: 避免两个范围的数据，互相干扰

包括: 2种:

1. 全局作用域: window

什么是: 不属于任何函数的，所有函数之外的范围，默认都是全局作用域

保存着: 全局变量!

什么是: 在函数外声明，不属于任何函数的变量

何时: 希望随处都可用的变量，就要在全局创建

2. 函数作用域:

什么是: 函数内部的范围

保存着: 局部变量:

什么是: 在函数内创建的, 仅属于当前函数的变量

何时: 希望一个变量仅在函数内可用时，都要在函数内创建。

包含: 2种: 1. 在函数内var的变量; 2. 参数变量

变量的使用顺序: 先用局部。局部没有，才用全局

\*\*\*声明提前(hoist):

什么是: 在开始正式执行程序前，引擎会将var声明的变量和function声明的函数，提前到\*当前作用域\*的顶部，集中创建。

将赋值留在原地！

问题: 破坏了程序的正常执行顺序

变量的解决办法: ES6 let 代替 var

let声明的变量, 不允许提前使用！

函数的解决办法: 第二种声明函数:

let 函数名=function(...){

...

}

总结: 今后，所有的变量都要用let声明

\*\*\*按值传递:

什么是: 两变量间传值时，或将变量传递给函数的参数时，其实只是将原变量中的值复制一个副本给对方:

原始类型的值: 修改新变量的值，不影响原变量

2. 全局函数:

什么是: ES标准中规定的，浏览器厂商已经实现的，不用.就可直接调用的函数!

已经学过: Number() String() Boolean()

parseInt/Float() isNaN()

反例: alert() prompt() - BOM

document.write() - DOM

console.log()

其他:

1. 编码解码:

什么是编码: 将url中的多字节字符，转化为单字节字符

为什么: url中不支持多字节字符!

何时: 只要url中包含多字节字符，都要先编码为单字节字符，再发送请求！

如何: var code=encodeURI(kword)

什么是解码: 将编码后的url中的单字节字符再转回多字节原文

如何: var kword=decodeURI(code);

问题: url中还不能包含保留字: 比如: 冒号: 斜线/

解决: 用encodeURIComponent()和

decodeURICOmponent() 代替

encodeURI和decodeURI

2. eval: 可执行字符串格式的程序代码

3. \*分支结构:

什么是: 让程序根据不同的条件执行不同的操作!

如何:

1. 1个条件1件事，满足条件才执行，不满足就不执行

当操作简单时, 可用短路: 条件&&(操作)

如果操作复杂, 就用if结构: if(条件){

...

}

2. 1个条件2件事，二选一执行:

如果操作复杂，就用if...else结构:

if(条件){//如果满足条件

操作1 //就执行操作1

}else{//否则

操作2 //执行操作2

}

如果操作简单, 用三目/三元/条件:

条件?操作1:操作2;

其实就是用?代替了if, 用:代替了else

更简化: 如果只是在两个值之间选一个:

条件?值1:值2

3. 多个条件，多件事，多选一执行:

如果操作复杂, 可用if...else if...结构:

if(条件1){//如果满足条件1

操作1 //就执行操作1

}else if(条件2){//否则 如果满足条件2

操作2 //就执行操作2

}else if(...){

...

}[else{//否则,就执行默认操作

默认操作

}]——可省略

强调: 如果前一个条件满足，则后续条件不再执行！

如果程序走到后一个条件，暗示前一个条件不满足!

最后一个else可省略: 表示，如果所有条件都不满足，则什么也不做！

如果操作简单，可用三目:

条件1?操作1:

条件2?操作2:

... ? ... :

默认操作; ——不能省略！

更简化: 如果只是多个值选一个

条件1?值1:

条件2?值2:

... ? ... :

默认值

特殊: 如果所有条件都是等于比较:

switch(表达式){ //先计算表达式的值

//再用表达式的值和以下每个case的值做全等===比较

case 值1: //如果表达式的值===值1

操作1; //就执行操作1

break;

case 值2: //如果表达式的值===值2

操作2; //就执行操作2

break;

case ... :

... ;

break;

default: //否则,就执行默认操作

默认操作

}

问题: 只要前一个case满足，则之后所有case都会触发！

解决: 希望每次只有一个case触发

在每个case之间添加break, 中断执行

作业: 100题: 2,4,5