正课:

1. Date:

2. Error:

3. \*\*\*Function:

1. Date:

API:

1. 8个单位:

FullYear Month Date Day

Hours Minutes Seconds Milliseconds

2. 每個單位上都有一對兒get/set方法

其中: .getXXX() 用于获取单位的数值

.setXXX(值) 用于修改单位的数值

特例: Day，没有set

3. 取值范围:

Month: 0~11 计算机中的月份数，总是比现实中小1

需要修正

Date: 1~31

Day: 0~6

Hours: 0~23

Minutes/Seconds: 0~59

计算: 2种:

1. 两日期对象可相减，结果是毫秒差

2. 计算任意一个单位的值:

3步:

1. 取出单位的值: var n=date.getXXX();

2. 做加减

3. 修改单位的值: date.setXXX(n)

强调: setXXX() 可自动调整时间进制

简写: now.setXXX(now.getXXX()+n);

问题: 直接修改原日期

解决: **先复制原日期，再用副本修改新日期**

日期格式化:

date.toString() 转为当地时间的完整格式

date.toLocaleString() 转为当地时间的简化版格式

date.toLocaleDateString() 仅保留日期部分

date.toLocaleTimeString() 仅保留时分秒部分

date.toGMTString() 转为**国际标准时间**(0时区)

2. Error:

什么是: 保存错误信息

错误: 程序执行过程中，导致程序无法正常执行的异常情况

后果: 程序都会强行退出

错误处理: 即使出错，也保证程序不中断退出的一种机制

何时: 只要希望，即使出错，也保证程序不中断退出

如何错误处理:

try{

可能出错的正确代码

}catch(err){

//err: 自动获得错误的信息

只有发生错误，才执行的错误处理代码:

1. 提示, 2. 记录日志, 3. 释放资源

}

优化: 尽量用if...else 代替try catch

Error: 鄙视: js中共有几种错误类型: 6种:

1. SyntaxError: 语法错误

2. ReferenceError: 要用的变量没找到

3. TypeError: 类型错误: 错误的使用了类型或类型的方法。错误使用对象或函数（函数名 写错）

4. RangeError: 范围错误: 参数超范围:

强调: js中数组下标超范围不报错！

5. URIError 6. EvalError

主动抛出错误:   
 何时: 团队协作开发中，程序的作者向程序的调用者提示错误的使用了程序。

3. Function:

什么是函数: 封装一段代码段的对象

何时: 如果一段代码可能被反复使用

如何: 3种:

1. 声明:

function 函数名(参数列表){

函数体;

return 返回值

}

问题: 被声明提前hoist



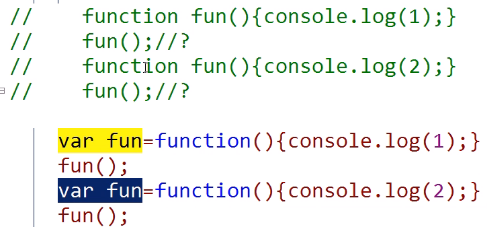
解决: 用赋值方式创建函数

2. 赋值:

var函数名=function (参数列表){

函数体;

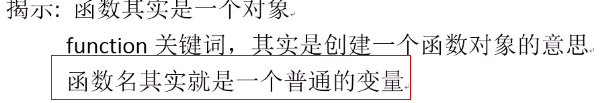
return 返回值

}

揭示: 函数其实是一个对象

function关键词，其实是创建一个函数对象的意思

函数名其实就是一个普通的变量



3. 用**new:**

**var fun=new Function("a","b","return a-b");**

鄙视: 以下定义函数正确的是:

function fun(a,b){return a-b}

var fun=function(a,b){return a-b}

var fun=new Function(a,b,"return a-b") X

var fun=new Function("a","b","return a-b")

重载overload:

什么是: 相同函数名，不同参数列表的多个函数，在调用时可根据传入参数的不同，自动选择匹配的函数调用。

为什么: 减少API的数量，减轻调用者的负担

何时: 如果一项任务，可能根据传入参数的不同，执行不同的逻辑时

比如: arr.slice(starti,endi+1)

arr.slice(starti)

arr.slice()

arr.splice(starti,n)

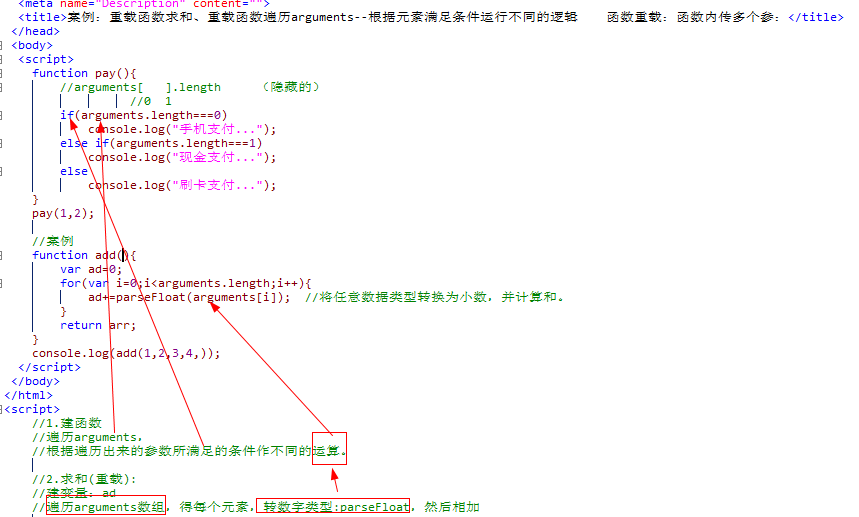
arr.splice(starti,0,值1,值2,....)

arr.splice(starti,n,值1,值2,....)

如何:

问题: js默认不支持重载!

为什么: js不允许多个同名函数同时存在





解决: arguments

每个函数内，默认都会自动创建一个arguments对象

arguments对象可自动接收所有传入函数的参数值

arguemnts是: 保存所有参数值的类数组对象

相同: 1. 下标, 2. .length, 3. for遍历

不同: 类型不同, API不通用

强调: 如果同时定义了参数变量和arguments

则arguments中的值和参数变量中的值始终是共用同一个值。

垃圾回收: 鄙视: 谈谈你对垃圾回收的理解：**避免内存泄露/全局污染，**一旦一个函数使用后，不会再使用，立刻释放（创建一个匿名函数包起来）

垃圾: **不再被任何变量引用的对象**

为什么: 内存空间都是有限的

垃圾回收: js引擎会自动释放不再被使用的对象的内存

垃圾回收器: 专门回收不再被使用的对象的小程序

随主程序启动，并伴随主程序并行执行

随时记录每个对象的引用次数

一旦发现一个对象不再被引用，则释放该对象

匿名函数:

什么是: 创建函数时，不指定函数名

为什么: 节约内存

何时: 希望一旦一个函数使用后，不会再使用，立刻释放

如何: 2种:

1. 回调:

比如: arr.sort(function(a,b){return a-b})

str.replace(正则, function(kw){ return 新值 })

setInterval(function(){ ... }, 间隔ms)

2. 自调: 定义函数后，立刻调用自己

为什么: 避免使用全局变量，避免内存泄露/全局污染

何时: 今后所有js代码，都应该放在匿名函数自调中，不再使用全局变量

如何: (function(){ ... })()

作业: function add(){

}

console.log(add(1,2));//3

console.log(add(1,2,3));//6

console.log(add(1,2,3,4,5));//15

作业：