补充:

1. 正则:

分组编号: 正则表达式中每个()，都会自动获得一个分组编号，从1开始，连续递增，不重复。

何时: 只要希望获得关键词的局部子字符串时

如何使用:

1. 在正则表达式内使用:

\n 匹配和第n个分组相同的内容

2. 在js函数中使用:

$n可自动获得第n个()分组匹配的子内容

正课:

1. 内置对象和包装类型

2. \*\*\*RegExp

3. Math

4. Date

1. 内置对象/类型:

什么是: ES标准中规定的，浏览器厂商已经实现的对象

为什么: 简化开发！

鄙视: 包括: 11个

String Number Boolean —— 包装类型

Array RegExp Math Date

Error

Function Object

Global: 是全局作用域对象: 在浏览器中被window代替

鄙视: 谈谈对包装类型的理解

什么是包装类型: 保存一个原始类型的值

提供操作原始类型值的API

为什么: 原始类型的值本身不具有任何功能

比如: var price=345.678; price.toFixed(2);

var unicode="张".charCodeAt();

何时: 只要试图对原始类型的值调用函数时，都会自动创建包装类型对象

如何: 自动使用:

过程:

试图对原始类型的值调用任何函数时:

1. 判断原始类型的类型名

2. 自动创建对应的包装类型对象,其中保存要操作的原始类型值

3. 调用包装类型对象，提起定义好的API

4. API调用后，包装类型对象自动释放！

鄙视: var str="hello";

str.len=10; //向str上添加一个新的属性，值为10

console.log(str.len);//输出str上的len属性值:?

原理: str.len=10; -> new String(str).len=10;

//执行后，new String()没人要，释放了！

console.log(str.len); -> console.log(new String(str).len);

另一种情况:

var str=new String("Hello");

str.money=10;

//因为new String()始终被str变量引用，不释放！

console.log(str.money);//10

2. \*\*\*RegExp:

什么是: 保存一个正则表达式

提供使用正则表达式执行验证和查找的API

何时: 2种:

1. 验证:

2. 查找: 第四种查找方式: 即查找内容，又查找位置

如何:

创建: 2种:

1. 创建一个固定不变的正则表达式:

var reg=/正则/ig;

何时: 如果正则表达式是固定不变的

2. 动态生成正则表达式:

var reg=new RegExp("正则","ig");

何时: 如果正则表达式不是固定，需要动态生成，就必须用new RegExp创建

API: 2个:

1. 验证:

var bool=reg.test(str) 验证str的格式是否符合reg的要求

问题: 默认只要部分匹配就返回true

解决: 只要验证，必须前加^，后加$

php中: preg\_match()

2. 查找: 补齐第四种查找方式：即查找内容，又查找位置

var arr=reg.exec(str);

在str中查找下\*一个\*符合reg要求的关键词的\*内容和位置\*

返回值: [

0: 关键词完整内容,

1: 第1个分组的子内容,

index:关键词的位置

]

如果找不到返回null

强调: 不用手动指定fromi，也可自动找下一个

因为: 每次都自动修改reg.lastIndex=index+关键词.length

php中: preg\_match\_all(

"正则",

"完整字符串",

保存查找结果的数组,

PREG\_OFFSET\_CAPTURE //包含关键词位置

)

3. Math:

什么是: 保存数学计算的常量，并提供数学计算常用API的对象

何时: 只要进行专门的数学计算时

如何:

创建: 不能new！

所有属性和API，直接通过大写Math调用！

包括:

（PHP中: 数学扩展->Math）

1. 取整: 3种:

上取整: 只要超过，就取下一个整数

Math.ceil(num)

下取整: 舍弃小数部分

Math.floor(num) 不能去单位

vs parseInt(str) 比floor聪明，能去单位——首选

四舍五入取整:

Math.round(num) vs num.toFixed(2)

缺: 只能取整 优: 可指定小数位数

优: 返回number 缺: 返回字符串

可直接用于计算 直接计算可能出错！

自定义round: 即可定义小数位数，又返回number

2. 乘方和开平方:

乘方: Math.pow(底数,幂)

比如: 10的2次方: Math.pow(10,2)

开平方: Math.sqrt(num)

3. 最大值，最小值:

Math.max(值1,值2,...) 获得多个值中的最大值

Math.min(值1,值2,...) 获得多个值中的最小值

问题: Math.max和min不支持从数组中获得最大值和最小值。

解决: Math.max/min(...arr)

...可打散数组为单个元素

4. 随机数:

Math.random() 随机生成0~1之间的一个随机小数

公式: 在任意min~max之间生成一个随机整数

parseInt(Math.random()\*(max-min+1)+min)