JavaScript第一天

作业提交： 20160808\_赵愉快\_ 查看作业 www.marrymin.com/Webs

交作业地址：<http://www.marrymin.com/UploadFile/WebPages/Upload.jsp>

协议 域名（IP地址） 子路径

建站流程

1. 选一台服务器设备——>运行操作系统（linux）——> web服务器（Apache，Tomcat）

2. 通过地址访问 地址组成：协议 + 域名（IP地址） + 端口 + 子路径

http [www.marrymin.com](http://www.marrymin.com) 80(默认)

3. 把你写的网站上传到服务器

回顾：HTML + CSS

1. Javascript （ECMAScript + DOM + BOM）

1、脚本（弱类型的语言）

var value; //value是一个变量（无类型）

var value1 = 1.1; //value是float类型变量

var value2 = ”字符串”； //value是字符串类型变量

强类型

Int value;

float value2;

char value3;

2、 如何使用JavaScript

IDE工具 ——> HBuilder ——> 介绍工具 ——> 新建第一个项目

Script­­标签 js代码片段就在此标签中

Script脚本会在浏览器解析文档内容时，顺次执行

Console：控制台 elements：元素

Js调试——>在浏览器中打开js控制台——> chrome/safari/firefox /ie

浏览器核心:webkit 网景修改 微软核心

——> log输出——> console.log(内容)

3、javascript的基本类型

undefined（未定义） ——>声明了变量（非引用）但未赋值

null ——>对象地址为空

Boolean

Number (int/float)

String

引用类型（object对象）

NaN not a Number (非数值类型)

转化功能：‘+’ toString() Number() parseInt()/parseFloat()

（+：字符串拼接 ）

Number 字符——> NaN 数值——>数值

布尔——>0和1

字符串（既有‘数值’又有‘字符’）——>NaN

——>0xf 十六进制字符串转为十进制数

对象——>先调用valueof再进行相应转化

**parseInt()/parseFloat()**

字符——>NaN

布尔——>0 1

数值——>数值

字符串（有数值和字符）

——>左起第一个数值类字符开始，到非数值类字符结束（忽略左起数值前的空格）

**明天**：字符串 查找 拼接 替换 大小写

Object对象

作用域+内存

**a.charCodeAt(0); a的ASCII**

**String.fromCharCode(97)； 根据ASCII返回字符**

**作业**：打印乘法表（log）

打印菱形（n行）

JavaScript第二天

1、JavaScript函数

本质是一个指针

1.函数的声明定义

function 函数名（参数）{

函数体

（如果有返回值，直接return）

}

2. 函数表达式

//函数表达式

//函数表达式定义的函数，不能提前执行

//console.log(sumPoint(55,45));

var sumPoint = function (num1,num2){

return num1 + num2;

}

//调用函数

console.log(sumPoint(55,45));

2、对象（Object）

类：将函数属性和数据属性封装在一起的数据结构

对象：属于某个类，具有唯一性的实例

Object 定义对象

a、使用new关键字定义对象（使用. 对特定对象添加数据属性和函数属性）

b、使用对象字面量定义（name:xxx, func: xxx）

**3、字符串常用方法**

3.1 **根据位置返回特定字符**（**单字符字符串**，无字符类型）

1.charAt（） 2.下标法（低版本的IE浏览器不支持）

**a.charCodeAt(0); a的ASCII**

**（2）根据位置返回特定字符**

slice（）/ substring() 相似 参数1：起始位置 参数2：索引前一位

subStr（） 参数1：起始位置 参数2：返回字符串的长度

区别：**对待负数（参数为负）**

参数1 参数2

**slice（）**  字符串的长度加参数1 字符串的长度加参数2

**substring()** 置0(第一位) 置0

**subStr（）** 字符串的长度加参数1 置0

**3.2根据字符串返回特定的位置**

**indexOf（）:查到返回索引，查不到返回-1**

参数1 参数2

indexOf（） 查找的字符串 从第几个索引查找

lastindexOf（） 查找的字符串 从第几个索引查找

区别：lastindexOf（）是从后往前查找

**3.3 字符串大小写转换**

转为大写：toUpperCase()

转为小写：toLowerCase()

**3.4字符串拼接**

**1. +**

**2.** concat（“参数”） **参数**：待拼接字符串

**3.5字符串比较**

**A.**localeCompare(**B**)

-1 A在前，B在后

0 AB相等

1. A在后，B在前

**a.charCodeAt(0); a的ASCII**

**String.fromCharCode(97)； 根据ASCII返回字符**

补充：

1、CSS Spite

对应一个类选择器

background-image

background-repeat

对应每一个具体的元素

background-position

2、display和 visibility

display: none； 元素不可见，不占文档空间

visibility：visible（默认）; 元素可见，占文档空间

hidden（隐藏）； 元素不可见，占文档空间

作业

1. 查找一个原始串中某目标串的个数
2. 打印出起始和结束位置

JavaScript第三天

昨天回顾：

1. 函数
2. 对象
3. 字符串常用方法
4. CSS Script 和 标签隐藏

1、javaScript的数据类型和内存管理

值类型：undefined , null, Boolean, Number, String

引用类型：Object， Array， Data var str = new String();

* 1. ==（相等） 和 ===（全等，等同）

==（相等） ——> 值相同就为真

===（全等，等同） ——> 值相同且类型相同为真

* 1. undefined 和 null

undefined 代表值类型

null 代表引用类型

**（2）内存管理**

1.标记法清除 大部分浏览器厂商

2.引用计数器 循环引用——>内存泄漏 解决：手动将对象A置为nullq

**2、javaScript函数、匿名函数和闭包**

2.1 script html文档流解析是按块解析

属性 defer 延迟脚本执行，等dom文档全部解析完

属性 async 脚本异步执行

javaScript解析器会先解析函数声明，使其在执行任何其他代码之前可以被使用；而函数表达式只会在解析器解析到其所在位置时，可以被执行使用。

2.2匿名函数和闭包

javaScript变量作用域：全局变量和局部变量（函数内）

<script type="text/javascript">

//匿名函数（无函数名）

var f = function(){

console.log("这是一个匿名函数");

}

**f();**  **//函数引用**

</script>

**闭包**：跨作用域访问变量，如局部函数中的变量在全局环境中被访问

3、javaScript函数的arguments对象和this对象

a．arguments对象可以通过下标访问函数参数

arguments[0], arguments[1]……

arguments对象属性length，传入参数个数

b. this对象

指向当前作用域对象

**javaScript第四天**

**昨天回顾**

1. javaScript数据类型和内存管理
2. javaScript匿名函数和闭包
3. javaScript函数的arguments对象和this对象

**今天课程**

1. Array对象
2. **初始化**
3. 字面量初始化
4. 构造方法初始化

读取数组元素使用**下标法（可读取数组元素和字符串元素）**

**2、length属性**

既可读又可写，可通过写，对数组扩充和删减

var arr5 = Array("a","b","c","d");

console.log(arr5);

arr5.length = **5**;

console.log(arr5[**4**]);

**3、数组和字符串的转化**

**1.数组——>字符串**

**toString（）**：使用 ，拼接字符串

**join（）** ：使用自定义的字符拼接字符串

2.**字符串——>数组**

参数1 参数2

**split()** **分割符** 元素个数上限

3.**数组对象常用方法**

**push()** 在数组的最后**压入**元素，返回数组的**元素个数**

**pop()** 在数组的最后**推出**元素，返回被**推出的元素**

**shift()** 在数组的首位**推出**元素，返回被**推出的元素**

**unshift()** 在数组的首位**压入**元素，返回数组的**元素个数**

**concat()** 在数组的**尾部拼接**新元素，返回拼接后的新数组，原数组不变

**splice(1,3，“a”,‘b’，c，9)**

**参数1**：起始位置(**1**)  **参数2**：元素个数(**3**)

**参数3**：插入的新元素 （**“a”，‘b’，c，9**）

**返回值：**被截取的元素构成的**新数组**

**4.根据元素查找位置，和字符串使用一致**

indexOf（） lastindexOf（）

**5.**

**参数1**   **参数2**   **参数3**

数组的项（数组元素） 数组项的索引（下标） 数组本身

**every()** 全真为真 （输出 true false）

**filter()**  过滤掉**不符条件的元素**，返回**符合条件的元素数组**

**some()** 有真则真 （输出 true false）

**foeEach()** 执行某种特定操作，无返回值

**参数1**：前一项 **参数2**：当前项 **参数3**：当前项索引 **参数4**：数组本身

**reduce()**  运行函数，执行特定操作，一次传入两个元素，**由左到右**

**reduceRight()** 运行函数，执行特定操作，一次传入两个元素，**由右到左**

1. **Date对象**
2. **初始化对象**

var nowDate = new Date()

1. **获取UTC时间 Date.UTC()/Date.parse(字符串)**

初始化时，传入UTC格式 Date(Date.UTC());

初始化时，传入字符串, Date(“2016/8/11 1:1:1”);

注意点：1、月份从0开始 2、年，月是必须参数（不可省）

3、年、月、日、时、分、秒、毫秒

**3、通过时间戳获取时间（从1970年1月1日起）**

初始化时，传入Number, Date(360\*24\*60\*60\*1000);

时间对象——>时间戳  **Date.parse**(dateObject);

**获取当前时间的时间戳：console.log(Date.parse(nowDate))**

**4、将时间对象转成字符串**

toString() / toDateString() / toTimeString() /

toLocaleDateString() / toLocaleTimeString()

1. **Math对象**

Math.**abs**()  **绝对值**

Math.**pow**(A,B)  **指数 A的B次幂**

Math.**min**()/max()  **取最小/最大值**

Math.**PI**  **π**

Math.**random**()  **随机值（0——1）**

1. **初识Dom（文档对象模型）Document Object Model**

HTML文档和JS交互，通过Dom

HTML

Head Body

Title …… link h1 …… p

Text text

**Dom对文档抽象出来的树形结构**，**Dom树**

DOM 是 W3C（万维网联盟）的标准。

DOM 定义了访问 HTML 和 XML 文档的标准：

明天：

Dom类型介绍，小测试

作业：

1. 字符串数组，选出最长的串
2. HelloWelcomeZhaoLanYuan 拆分成单词（首字母大写）
3. HelloWelcomeZhaoLanYuan 设计函数传入字符串，返回出现次数最多的字母以及次数组成的字符串

**javaScript第六天**

**上周回顾**

数据类型

对象 ——> 函数

**上周测试**

**今日课程**

**for in 遍历字符串，数组，对象属性**

**for（var value in Object）{}**

1. **类**（新版ECMAScript）

**Array，date... var arr = [];**

**Function function sum(){}**

* 1. 通过Object的create()方法，模拟父子对象，继承

prototype属性指向当前对象的原型（父对象）

* 1. **函数 （父子类、继承）**

**//类**

function Cat(age,eyes){

this.age = “age”;

this.eyes = “eyes”;

}

Cat.prototype.xxx = xxx;

//对象

Var cat = new Cat(xxx,xxx);

1. **DOM（文档对象类型） DOM树（动态）**

**HTML**

**Head Body**

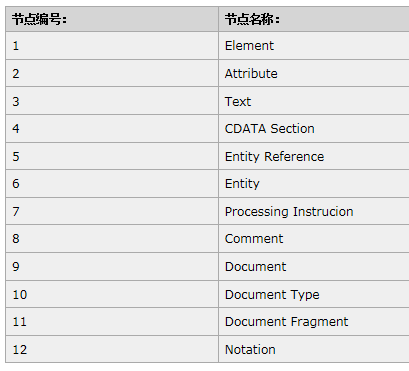
**Title …… link h1 …… p**

**Text text**

**<a href = “xxx”>百度</a>**

**a**——>标签名 **百度**——>标签值

**href**——>属性名  **xxx**——>属性值



**2.1、Node（节点）类型**

nodeType 节点类型

nodeName 节点名（标签名）

nodeValue 节点值

childNodes 子节点

**2.2、Document（文档）类型**

**document对象 ——> HTMLDocument类 ——> Document类**

（文档入口对象）

**getElementById(id)**; 返回Element对象

（**不支持链接查询**）

**getElementsByTagName(tagName)**; 返回HTML**Collection**对象

(**集合**)

对于HTMLCollection对象，使用[],传入数字时调用item（），

传入字符串时调用nameditem（）。nameditem（）先检索DOM树中有没有相同的id元素，有则返回此元素；没有则检索有没有相同name的元素，有则返回元素；如果有多个name相同的元素，先检索谁返回谁。如果没有符合条件的元素，返回undefined。

**2.3、Element（元素）类型**

**nodeName/tagName**(获取字符的大小写可能不同)

（**节点名**） （**标签名**）

**id**

**className**

**innerText**  元素文本（标签值）

**getAttribute（）** **获取属性**（不区分大小写）

**setAttribute（）**  **设置属性**

**2.4、TextNode类型**

文本

作业

1. 定义一个类，两个数据属性半径r，高h，一个函数，求v；
2. Div400\*400，放在屏幕中央，每次刷新，页面中央都会出现 1——6的不同数字

javaScript第七天

昨天回顾

1、类 使用create方法/使用函数构造方法

2、DOM Node/Document/Element/Text(TextNode)

今日课程

<noscript>当前环境不支持脚本 或者 您去设置打开脚本 </noscript>

**1.Node类型**

childNodes 子节点

hasChildNodes() 判断有无子节点，返回布尔值

**appendChild()**  添加节点，参数传入新节点

**removeChild（）**  移除节点，参数传入旧结点

**insertBefore()** 插入节点， 参数1：要插入的节点 参数2：位置参考节点

**replaceChild（）** 替换节点， 参数1：新节点 参数2：旧节点

**2.Document类型**

getElementsByClassName() 通过类名获取元素

getElementsByName() 通过name属性获取元素

**传入参数为选择器语法的选择方法**

querySelector() 符合条件的第一个元素

querySelectorAll() 符合条件的所有元素

createElement（参数） 创建节点元素，参数为标签名（如h1）

createTextNode() 创建文本节点，参数为文本内容

1. **Element类型**

使用Element元素继续检索元素，缩小了检索的范围；querySelectorXXX

方法支持链式查询。

以**query**开头的查找方法 query+选择器

querySelector() 符合条件的第一个元素

querySelectorAll() 符合条件的所有元素

**4.Text类型**

标签值为非空字符串即存在文本节点

**splitText**(分隔位置)  **分割文本节点**

**normalize**() **拼接文本节点**（父节点调用）

**appendData（**要添加的内容**）**  **追加文本内容**到当前节点

**deletedData**（参数1，参数2）  **删除文本内容**

参数1：**起始位置** 参数2：**元素个数**

**replaceData**（参数1，参数2，参数,3） 将特定位置的文本**替换**成传入的文本

参数1：**起始位置** 参数2：**元素个数**  参数3：**替换的字符串**

**insertData**（起始位置，插入的文本） 将文本**插入**到特定位置

**5、动态添加脚本**

创建script元素（Element类型），添加到文档即可。

script.setAttribute(”src”,”xxx”); xxx：脚本的相对路径或绝对路径

**6、事件**

鼠标单击事件（左键）

**click事件**就是**单击事件**，元素都有属性onclick可进行点击相应设置

作业

javaScript第八天

昨天回顾

DOM 创建 删除 添加 替换。。。

事件 click（单击事件）

作业

列表项

今日课程

**事件**

1. **事件和事件流**

**事件：文档或浏览器窗口发生的可以被监听相应的行为**

**观察者模式（**对事件的监听**）**

**事件流：**从页面中接收并传递事件的流程顺序

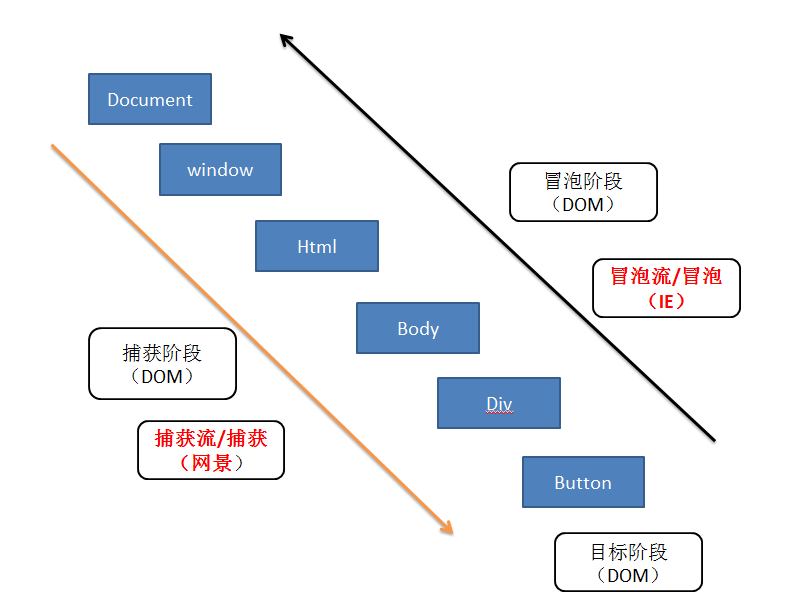
**网景：**从DOM树**最顶端**到DOM树**最深层**传递事件，叫**捕获**

**（window ——> 监听元素）**

**IE：** 从DOM树**最深层**到DOM树**最顶端**传递事件，叫**冒泡**

**（监听元素 ——> window）**

**DOM事件流：三个阶段. 捕获阶段、目标阶段、冒泡阶段**



1. **事件响应和事件对象**

**事件的响应添加：**

1.**使用事件类型同名的标签属性**(**on + 事件类型（click，dblclick）**)

onclick 标签**单击**属性

on**dbl**click 标签**双击**属性

使用函数调用或者JS代码片段对属性赋值

onclick = “function”；

onclick = “JS代码”；

onclick后跟的类似于函数调用（如 **buttonClick()**），而不是指向函数的变量，如buttonClick。

2.**通过javaScript访问事件响应属性（DOM1）**（onclick，ondblclick...）

使用形式：

**element.onclick = function(){}**

**element.onclick = null**; //删除响应，切断建立的链接

**3.使用DOM2定义的事件响应方法**

**addEventListenser**(参数1，参数2，参数3)

**参数1**： **事件类型** （**click**， **dblclick**......）

**参数2： 事件响应函数**

**参数3**： 是否**捕获**（true:**捕获**传递， false:**冒泡**传递）

**removeEventListenser**(参数1，参数2，参数3)

参数1： 事件类型 （click， dblclick......）

参数2： 事件响应函数

参数3： 是否**捕获**（true:**捕获**传递， false:**冒泡**传递）

**注**：使用**匿名函数**无法删除响应

IE添加、删除

attachEvent（）/detachEvent()

**事件对象：（Event）**

**事件是浏览器自动传入的**

**Type 事件类型**

**client/client 客户端坐标**

**pageX/pageY 网页坐标**

**currentTarget 响应函数绑定的元素**

**target 事件目标元素**

**button 鼠标按键**

**bubbles 是否为冒泡**

**mousedown 按下**

**mouseup 抬起**

**mousemove 在元素上移动**

**mouseover 在元素上，响应一次**

**mouseout 从元素内移到元素外**

**screenX**:鼠标位置相对于用户屏幕水平偏移量，而screenY也就是垂直方向的，此时的参照点也就是**原点是屏幕的左上角。**

**clientX:**跟screenX相比就是将**参照点改成了浏览器内容区域的左上角**，该**参照点会随之滚动条的移动而移动**。

**pageX：参照点也是浏览器内容区域的左上角**，但它**不会随着滚动条而变动**

作业

移动方块

**javaScript第九天**

昨天回顾

事件、事件流、事件响应、事件对象

作业

移动方块

原理：事件类型—>事件响应——>事件对象——>鼠标位置——>方块位置

1. **样式表**设置的top，left，在**DOM中**不能直接使用
2. 保证鼠标坐标点和元素左上角坐标点差值常量
3. 设置flag为false的临界点

**今日课程**

1. **事件对象和事件类型**

event 对象 ——>数据属性，函数属性

preventDefault(); 阻止默认行为

**事件类型：**

**1.1 鼠标事件：**

**mousedown 按下**

**mouseup 抬起**

**mousemove 在元素上移动**

**mouseover 在元素上，响应一次**

**mouseout 从元素内移到元素外**

**1.2 UI事件：**

**load 页面加载完毕**

**unload 页面卸载完毕**

**select 选择文本（input、 textarea）**

**blur 失去焦点**

**focus 获得焦点**

**1.3 键盘事件**

**keydown 按键按下**

**keyup 按键抬起**

**keypress 按键完成一次按键动作**

**altkey/controlkey/shiftkey.... 符合按键属性**

1. **BOM（浏览器对象模型）**

**window对象 BOM核心**

1. **全局变量**

添加到window对象中

1. **window.open**

参数1： URL地址

参数2： 打开属性（本页、新页、框架）

参数3： 特征字符串

参数4： 是否替换历史记录

1. **alert() / confirm() / prompt()**

**confirm true/false**

**prompt 返回文本**

1. **location对象**

href 链接地址

host 服务器地址和端口

Port 端口

hostname 服务器地址

protocol 协议

search URL查询字符串（Get方式的参数）

1. **history对象**

**go()**   **-1 返回 1 前进**

**back()**  **返回**

**forward()** 前进

setTimeout（） [clearTimeout()](http://www.w3school.com.cn/jsref/met_win_cleartimeout.asp) 取消由 setTimeout()设置的 timeout。

setInterval（） [clearInterval()](http://www.w3school.com.cn/jsref/met_win_clearinterval.asp) 取消由 setInterval() 设置的 timeout。

**setTimeout()** 方法用于在指定的[毫秒](https://www.baidu.com/s?wd=%E6%AF%AB%E7%A7%92&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHcvrjTdrH00T1YLmH-BrHTzPWI9PHR4uHc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjcLP164rHfv)数后调用函数或计算表达式。  
　　语法: **setTimeout**(**code,millisec**)  
　　参数：  
　　 **code** （必需）：要调用的**函数**后要执行的 JavaScript 代码串。  
　　 **millisec**（必需）：在执行代码前需等待的[**毫秒**](https://www.baidu.com/s?wd=%E6%AF%AB%E7%A7%92&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHcvrjTdrH00T1YLmH-BrHTzPWI9PHR4uHc40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjcLP164rHfv)**数**。

提示：  
　　 **setTimeout()** 只执行 code 一次。如果要多次调用，请使用 **setInterval()** 或者让 code 自身再次调用 setTimeout()。

作业

打飞机

javaScript第十天

昨天回顾

事件对象与事件类型：

BOM——>(核心)window

作业

打飞机

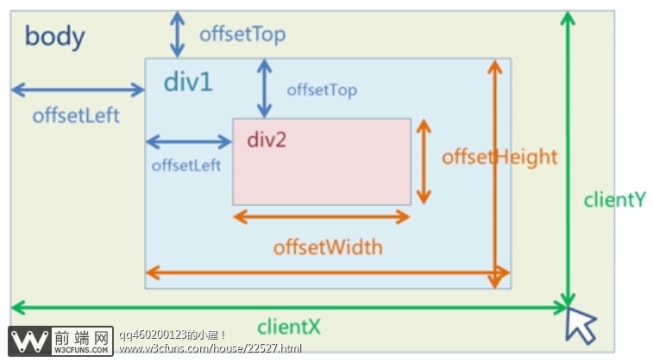
1. 游戏引擎——> 定时控制游戏流程
2. 我方战机，敌机，合金炮弹
3. 计分系统

断点调试（breakpoint）

执行下一个断点

一步步往下执行

一步步往下执行（进入函数内部）



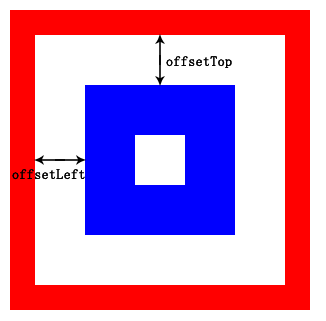
带单位长度，属性可读可写

style.top

style.left

style.width

style.height

纯数值，属性可读不可写

offsetLeft

offsetTop

offsetWidth

offsetHeight

//切换游戏场景

document.querySelector(".gamesence").style.backgroundImage = "url(img/background\_1.png)";

//隐藏按钮

**document.querySelector(".gamesence>button").style.display = "none";**

Window 对象表示浏览器中打开的窗口。

如果文档包含框架（frame 或 iframe 标签），浏览器会为 HTML 文档创建一个 window 对象，并为每个框架创建一个额外的 window 对象。

Javascript第十一天

上周回顾

DOM 事件 BOM

今日课程：

1. **this对象**

**代表对象**（1.执行对象 2.对象不明确，代表window对象）

**用到位置**：函数（对象，类）

**匿名函数对象不明确**（window）

1. **定时器**

setInterval（） **间隔**某个时间循环执行某个函数

参数1：执行函数（也可以是一段javascript代码）

参数2：时间间隔

参数3：传入的参数

返回值：定时器对象

clearInterval（） **终止**Interval定时器

参数1：定时器对象

setTimeout() **延迟**某个时间循环执行某个函数

参数1：执行函数

参数2：延迟时间

参数3：传入的参数

返回值：定时器对象

clearTimeout（） **终止**timeout定时器

参数1：定时器对象

//**使用setTimeout（）借助递归思想实现setInterval（）功能**

//1.函数调用本身

function customTimeout(){

setTimeout(customTimeout, 1);

}

//**递归** 1.调用本身函数 2.找好临界条件

function f(num){

if(num>1){

return num\*f(num-1);

}else{

return 1;

}

}

1. **正则表达式**

**3.1正则表达式语法：**

**元字符（字符串中使用，需要用转义字符）**

^ **代表字符串开始 $ 代表字符串结束**

**\*** **代表字符任意次数 + 代表字符至少一次**

**？ 代表字符最多一次**

**{} 代表字符重复次数的范围 {2}： 2次**

**{2,5}： 2—5次 ， {2,}: 2次以上**

**| 代表或，如a|b，指a或b**

**. 代表任意字符**

**[] 代表某些字符在字符串的任意位置出现**

**配合使用-（连接符）**

**3.2 正则表达式在javascript中使用**

**作业**

斐波那契数列

Html，倒计时

Javascript第十二天

昨天回顾：

this关键字

定时器（setTimeout， setInterval） （递归）

正则表达式（Regular Expression——> RegExp）

作业

1. 菲波那切数列
2. 倒计时

今日课程

1. **正则表达式**

**元字符：**^ $ \* + ? {} [] | .

[^] **相反**  [^0-9] **非数字字符**

\d **等效**[0-9]

\D **等效**[^0-9]

\w **匹配任何单词字符（数字，字母，下划线） 等价于[A-Za-z0-9\_]**

\W **匹配任何非单词字符 等价于[**^**A-Za-z0-9\_]**

\. **中的**\**表示转译，转成字符串**

**在javaScript中使用正则表达式：**

正则表达式对象（RegExp类）

1. 字面量形式

var reg = /表达式语法/模式；

模式：g **全局匹配** （默认**非全局**）

i **大小写不敏感（case insensitive）**

m **多行模式 （mutily line）**

b．RegExp对象构造方法

var reg = new RegExp(**参数1，参数2**)

**参数1：正则表达式语法字符串**

**参数2：模式字符串**

**双转译**

**test()**  **reg.test(str);** 正则对象方法，参数为字符串，返回为布尔值，匹配到返回true，否则返回false

**search()**   **str.search(reg);** 字符串方法，传入字符串或正则对象，返回值为第一次匹配到字符串的**起始索引**或**-1**（未找到）

**match()**  **str.match(reg);** 字符串方法，传入正则对

象，**全局匹配，返回数组。非全局匹配时，返回第一匹配项，index（匹配项开始位置索引），input（原串的引用），**length（数组长度）

**exec()**  **reg.exec(str)** 正则对象方法，在非全局匹

配模式下与match（）相同，在全局匹配模式下返回详细匹配信息。

**replace()**  可以传入正则对象

**split()**  可以传入正则对象

1. **bug调试**

**2.1 会打断点（断点**：程序执行到该位置，处于挂起，暂停**）**

**2.2** 断点时，将鼠标指针移到相应的变量上，会弹出变量的详细信息

**2.3** 断点时，代码中无此变量，想查看需要在控制台输入，再按回车

1. **滑动门效果**

enumerable 属性 true可枚举（for-in） false 不可枚举

mouseover(**在元素上，响应一次**) —> display —> none/block

1. **轮播图**

作业

邮箱：[xxx@xx.xxx.xx](mailto:xxx@xx.xxx.xx)(点的个数不定) 输入字符串，判断是不是邮箱，alert提示，input——>type text

动态判断输入的文本是不是有效数字。（计算器输入判断）

Javascript第十三天

昨天回顾

正则表达式的使用

bug调试

滑动门

作业

邮箱：[xxx@xx.xxx.xx](mailto:xxx@xx.xxx.xx)(点的个数不定) 输入字符串，判断是不是邮箱，alert提示，input——>type text

动态判断输入的文本是不是有效数字。（计算器输入判断）

今日课程

1. **正则表达式转译**

**原串 双引号内容 正则字面量 正则构造方法**

**abc abc abc abc**

**[ab]c [ab]c \[ab\]c \\[ab\\]c**

**abc/def abc/def abc\/def abc\\/def**

**\ab\cd \\ab\\cd \\ab\\cd \\\\ab\\\\cd**

1. **轮播图**

定时器—>自动完成轮播 setTimeout/setInterval

3张图—>左右排列 1—2—3—1

element.style.left (element.offsetLeft) 改变位置

1. **浏览器能力检测**

**3.1 浏览器检测**

BOM —> window —> navigator （用于检测浏览器类型）

userAgent（用户代理字符串）—> 可用于检测浏览器内核及浏览器名称

Chrome —> Webkit（开源）

Safari —> Webkit

IE —> Trident

Firefox —> Gecko

Opera —> Gecko

**设备检测**

iOS —> Webkit

Android —> Webkit

* 1. **能力检测**

检测属性 —> if（object.属性）{ }

检测函数 —> if（typeof object.方法 == function）{ }

throw new Error（异常信息）

try{ }catch（error）{ }

作业

按钮轮播图