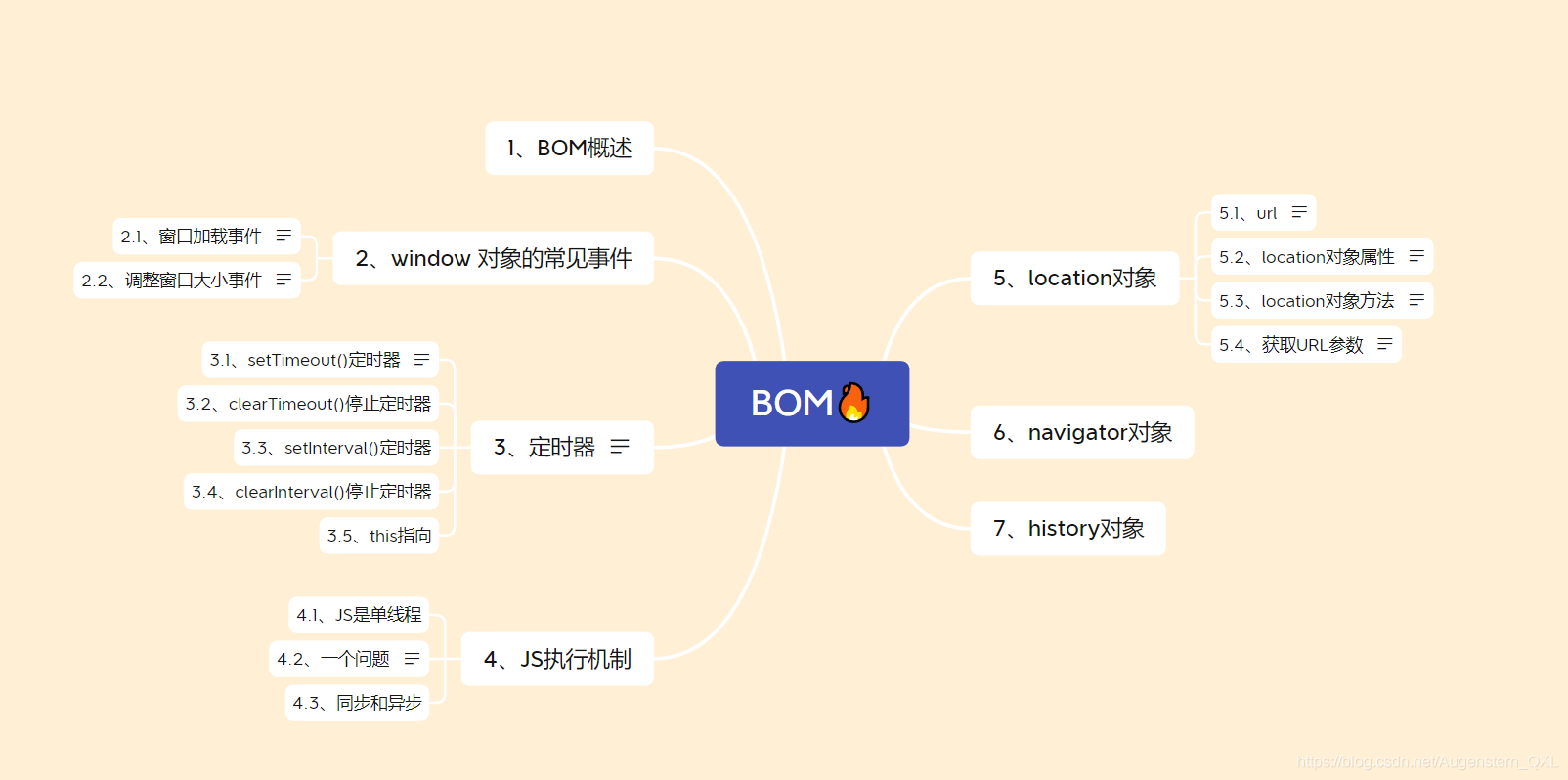
**目录总览：**



**5.location对象**

window 对象给我们提供了一个 location属性用于获取或者设置窗体的url，并且可以解析url。因为这个属性返回的是一个对象，所以我们将这个属性也称为 location 对象。

**5.1、url**

【统一资源定位符（uniform resouce locator）】是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的 URL，它包含的信息指出文件的位置以及浏览器应该怎么处理它。

url 的一般语法格式为：

protocol://host[:port]/path/[?query]#fragment

http://www.itcast.cn/index.html?name=andy&age=18#link

|  |  |
| --- | --- |
| **组成** | **说明** |
| protocol | 通信协议 常用的http,ftp,maito等 |
| host | 主机(域名) www.itheima.com |
| port | 端口号，可选，省略时使用方案的默认端口 如http的默认端口为80 |
| path | 路径 由零或多个'/'符号隔开的字符串，一般用来表示主机上的一个目录或者文件地址 |
| query | 参数 以键值对的形式，通过&符号分隔开来 |
| fragment | 片段 #后面内容 常见于链接 锚点 |

**5.2、location对象属性**

|  |  |
| --- | --- |
| **location对象属性** | **返回值** |
| location.href | 获取或者设置整个URL |
| location.host | 返回主机（域名）www.baidu.com |
| location.port | 返回端口号，如果未写返回空字符串 |
| location.pathname | 返回路径 |
| location.search | 返回参数 |
| location.hash | 返回片段 #后面内容常见于链接 锚点 |

重点记住：

**href和search**

**案例：5s之后跳转页面**

<body>

<div></div>

<script>

var div = document.querySelector('div');

var timer = 5;

setInterval(function() {

if (timer == 0) {

location.href = 'http://www.itcast.cn';//跳转页面！！

} else {

div.innerHTML = '您将在' + timer + '秒钟之后跳转到首页';

timer--;

}

}, 1000);

</script>

</body>

**案例：获取URL参数数据**

主要练习数据在不同页面中的传递

1.第一个登录页面里面有提交表单，action提交到index.html页面

2.第二个页面可以使用第一个页面的参数，这样实现了一个数据不同页面之间的传递效果

3.第二个页面之所以可以使用第一个页面的数据，是利用了URL里面的location.search参数

4.在第二个页面中，需要把这个参数提取。

5.第一步去掉？利用substr

6.第二步利用=号分割键和值 split（'='）

7.3.把数据写入div中

login.html:

<body>

<form action="index.html">

用户名：<input type="text" name="uname">

<input type="submit" value="登录">

</form>

</body>

index.html:

<body>

<div></div>

<script>

console.log(location.search);//获得?uname=andy

//1.先去掉？substr('起始位置'，[截取几个字符])；

var params = location.search.substr(1);//uname=andy

//2.利用=号分割为数组 split（'='）

var arr = params.split('=');

console.log(arr);//["uname","andy"]

var div = document.querySelector('div');

//3.把数据写入div中

div.innerHTML = arr[1] + '欢迎你';

</script>

</body>

**5.3、location对象方法**

location对象方法返回值

location.assign()跟href一样，可以跳转页面（也称为重定向页面）

location.replace()替换当前页面，因为不记录历史，所以不能后退页面

location.reload()重新加载页面，()内参数为空时相当于刷新按钮或者 f5 ，如果参数为true 强制刷新 ctrl+f5

（如果有个页面代码写对了却运行仍有问题，可以尝试强制刷新 ctrl+f5，因为有时页面缓存下来了效果看的不是很明显）

<body>

<button>点击</button>

<script>

var btn = document.querySelector('button');

btn.addEventListener('click', function() {

// 记录浏览历史，所以可以实现后退功能

// location.assign('http://www.itcast.cn');

// 不记录浏览历史，所以不可以实现后退功能

// location.replace('http://www.itcast.cn');

location.reload(true);

})

</script>

</body>

**6、navigator对象**

navigator 对象包含有关浏览器的信息，它有很多属性

我们常用的是userAgent,该属性可以返回由客户机发送服务器的user-agent头部的值

下面前端代码可以判断用户是用哪个终端打开页面的，如果是用 PC 打开的，我们就跳转到 PC 端的页面，如果是用手机打开的，就跳转到手机端（移动端）页面

if((navigator.userAgent.match(/(phone|pad|pod|iPhone|iPod|ios|iPad|Android|Mobile|BlackBerry|IEMobile|MQQBrowser|JUC|Fennec|wOSBrowser|BrowserNG|WebOS|Symbian|Windows Phone)/i))) {

window.location.href = ""; //手机

} else {

window.location.href = ""; //电脑

}

（作为了解）

**7、history对象**

window 对象给我们提供了一个 history 对象，与浏览器历史记录进行交互该对象包含用户（在浏览器窗口中）访问过的 URL。

history对象方法作用

back() 可以后退功能

forward() 前进功能

go(参数) 前进后退功能，参数如果是 1 前进1个页面 如果是 -1 后退1个页面

<body>

<a href="list.html">点击我去往列表页</a>

<button>前进</button>

<script>

var btn = document.querySelector('button');

btn.addEventListener('click', function() {

// history.forward();

history.go(1);

})

</script>

</body>

history对象一般在实际开发中比较少用，但是会在一些OA办公系统中见到。