[JavaScript笔记](#_Toc515191243)

[html笔记](#_Toc515191244)

[CSS笔记](#_Toc515191245)

[Angular笔记](#_Toc515191246)

[ng-file-upload](#_Toc515191247)

# JavaScript笔记

1. 语法
2. 功能提示

* JS正则表达式
* 表达式前后以/界定，如/........./；后面接/g表示查找所有匹配而非在找到第一个匹配后停止
* replace（）第一个参数（正则表达式）不能加引号
* 1. \*表示 0个或多个，+表示1个或多个，？表示0个或1个

2. ^表示起头，$表示结尾，匹配字符串（前后带空格的）

3. a|b表示或者满足a或者满足b

4.贪婪模式：a.\*b匹配最长的以a开始，以b结束的字符串

懒惰模式：a.\*?b匹配最短的以a开始，以b结束的字符串

# html笔记

1. 标签

|  |  |  |
| --- | --- | --- |
| <!- -> | 定义注释 |  |
| <! DOCTYPE> | 定义文档类型 | doctype: document type |
| <title> | 定义文档的标题 |  |
| <head> | 定义文档的头部 | 可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等 |
| <meta> | 标签位于文档的头部，不包含任何内容，提供有关页面的元信息 | <meta>标签的属性定义了与文档相关联的名称/值对（两个属性组成一对），可提供比如针对搜索引擎和更新频度的描述和关键词 |
| <hr> | 页面中创建一条水平分隔线 |  |
| <a href="URL"></a> | 超文本链接 | href：Hypertext Reference  <a>anchor 的缩写,['&aelig;ŋkə] n. 锚, 铁锚  URL全球资源定位器（UniformResource Locator） |
| <br> | 空标签，有换行的作用 | 无结束标签，因此这是错误的：<br></br>  Barter rabbet的缩写；barter：交易,rabbet槽 |
| <td> | 定义一个表格 |  |
| <caption> | 定义表格标题 |  |
| <tr> | 定义表格中的一行 |  |
| <th> | 定义单元格中的一个单元格 |  |
| <abbr> | 定义缩写 | abbr : abbreviation[ə,bri:vi'eiʃən] 缩写,省略 |
| <acronym> | 定义只取首字母的缩写 | acronym：['ækrənim],首母缩略字 |
| <address> | 定义文档作者 |  |
| <em>与<i> | 都是用斜体来显示文字 |  |
| <strong> | 粗大字体来显示文字 | 能针对搜索引擎优化的一个标签 |
| <strike> | 定义加删除线文本定义 | strike- 中划线 |
| <u> | 定义下划线文本 | u,underlined，下划线 |
| <small> | 小字体文本 |  |
| <base> | 标签为页面上的所有链接规定默认地址或默认目标 |  |
| <center> | 其所包括的文本进行水平居中 |  |
| <label> | 为 input元素定义标注（标记） |  |
| <ol>, | 有序列表 | ordered list |
| <li> | 列表项目 | listitem |
| <ul> | 无序列表 | unordered list |
| <select> | 可创建单选或多选菜单 |  |
| <optgroup> | 把相关的选项组合在一起 | 可放在<select>之下使用 |
| <span> | 用来组合文档中的行内元素 | 常用内联容器，定义文本内区块 |
| <textarea> | 定义多行的文本输入控件 | 本区中可容纳无限数量的文本，其中的文本的默认字体是等宽字体（通常是Courier） |

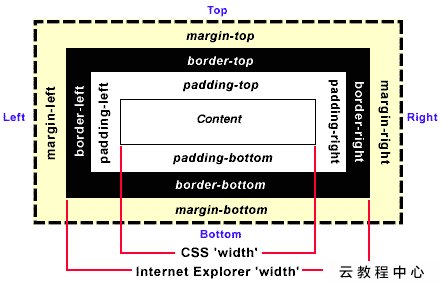
# CSS笔记

1. 代码解释

|  |  |  |
| --- | --- | --- |
| box-shadow:0px 0px  1px #000 [inset] | 盒子阴影 | 可选参数inset写入时表示内阴影 |
| text-align: center | 行内或类行内元素（比如文本和链接） |  |
| display: inline-block | div块的宽度随内容自适应 |  |
| margin:0 auto | 使div元素水平居中但需在父元素中设置 text-align:center |  |
| img { width: 100%; height: 100%;} | 图片自适应上层div大小 |  |
|  |  |  |
| a:link | 定义正常链接的样式 | eg: a:link { color:#3C3C3C;  text-decoration:underline; } |
| a:visited | 定义已访问过链接的样式 |  |
| a:hover | 定义鼠标悬浮在链接上时的样式 |  |
| a:active | 定义鼠标点击链接时的样式 |  |
|  |  |  |
|  |  |  |

2. 特别提示

* margin-top | margin-bottom | padding-top | padding-bottom 的百分比值参照的不是容器的高度，而是宽度
* margin，border，padding示意



# Angular笔记

1. 基本

* angular-resouse,angular-cookings版本必须要跟angualar版本一致
* 引用顺序
* 在html文件中，app.js的引用一定要放在最后（在sevice.js等之后）
* 在html文件中，引用JS脚本时，自己的一定要放在框架JS后面
* 模块注入原则

当模块a需要使用模块b时，在模块a的定义时进行注入；html中include时，模块b的js在前

* service和factory的使用是非常相似的，但是service是不能返回字符串的，而factory既可以返回对象也可以返回任意的字符串
* angular的异步传输
* 在服务里设定promise

upload: function (picFile) {

var deferred = $q.defer(); //设定promise

Upload.upload({

url: aliYunImageUrl,

data: {

file: picFile

},

}).then(function (response) {

deferred.resolve(response); //响应后成功后处理

}

, function (response) {

}, function (evt) {

});

return deferred.promise; //返回promise对象

* 在控制器中使用此服务时，使用then来获取异步响应后从服务器中返回的数据

$scope.uploadImage=function(picFile){

uploadImage.upload(picFile).then(function(response){ //使用then来获取异步响应后从服务器中返回的数据

$scope.artical.topicalimageUrl=response.data.url

});

}

1. 模块

* angular-ui-router
* 需在html 引入angular-ui-router.js，使用时注入ui.router服务
* **指令**

1. ui-view

注意：1. 在ui-view中引用的html模板，通过.state跳转时都会重新刷新在此state里的html模板，如不想刷新，此html模板通过ng-include放在主页面（例如导航栏）;

2. 子状态通过在(父状态名.子状态名)设置，其url会加到父url之后成为自己的url，并会继承父状态的view；

3. 每一个.state里包含的模板html，要想在跳转后起作用，必须在跳转前的html页面设置ui-view（此html跳转后必须依然存在）

4. 如果.state里有多个html模板（views），controller必须放在每个html模板下面才能生效

2. ui-sref

用法：ui-sref =“页面({参数1:参数1值,参数2:参数2值,…})”，解析时会查找$state里的名为“页面”的state，读取URL

* angular-resource
* 需在html 引入angular-resource.js，使用时注入ngResource服务
* $resource(url+’/:id’，{id:'@id'}，参数c)

{id:'@id'},表示url中的:id 将会使用资源的id属性值；c为自定义方法，例如{myGet{method：‘get’，isArray：True}}，其中myGet是方法名，isArray：True表示资源是数组不是对象。

如果第一个参数url+’/:id’没有’/:id’, 会对远程接口生成url?id=xxxxx的请求

**url的配置必须加上http://**

* $resource的方法

|  |  |  |
| --- | --- | --- |
| get([params],[success],[error]); |  | 通过resource对象来调用 |
| save([params],postData,[success],[error]); |  | 同上 |
| query([params],[success],[error]); | query向指定URL发送一个GET请求，并期望返回一个JSON格式的资源对象集合 | 同上 |
| remove([params],postData,[success],[error]); |  | 同上 |
| delete([params],postData,[success],[error]); |  | 同上 |
| $save([params],[success],[error]); |  | 通过需要处理的对象来调用，会把对象作为请求体发过去 |
| $remove([params],[success],[error]); |  | 同上 |
| $delete([params],[success],[error]); |  | 同上 |

* 以上的方法在resource的promise里都要求服务器**返回JSON 格式（响应消息类型是application/json，不能是text/plain）**
* angular-sanitize
* 需在html 引入angular-sanitize.js，使用时注入ngSanitize服务
* **指令**

1. ng-bind-html（属性指令）

在数据绑定的html标签中使用ng-bind-html属性，此属性依赖于$sanitize

* angular-summernote
* summernote框架在angular上的实现
* 需在html 引入summernote.css，summernote.js, angular-summernote.js，使用时注入summernote服务
* 在HTML文件里js的include顺序： jquery，bootstrap，angular，summernote，angular-summernote
* **指令**

1. summernote（元素指令）

2. height（属性指令）

config（属性指令），可绑定$scope里的配置，配置里有个callbacks项，可在此callbacks里添加一些动作发生时的回调函数，例如onImageUpload发生时传送图片到后台服务器

lang（属性指令）18n Support，中文为lang="zh-CN"，需在html中include相关语言js文件

1. 原生服务

* $sce
* 需在在使用时（例如定义controller）注入
* $sce.trustAsHtml(input)

将数据内容以html的形式进行解析并返回过滤器以提供给数据绑定

* $log会检查浏览器是否支持console.log，这样避免浏览器不支持console.log而报异常;
* 可以开关控制log: $logProvider.debugEnabled(false)。
* $log
* 需在在使用时（例如定义controller）注入
* $log.info('普通信息'); $log.warn('警告信息'); $log.error('错误信息'); $log.log('打印信息'); $log.debug('调试信息');
* $urlRouterProvider
* $urlRouterProvider.otherwise('/')，没有匹配的路由的时候，跳转到一个默认的路径
* filter
* limitTo（原生）

语法{{ limitTo\_expression | limitTo : limit : begin}}

limit正值从前往后，负值为从后往截取，begin参数用于控制从哪里开始截取

* $location





* ng-src

代替src属性，使用这个指令后，请求就只会在Angular执行这个绑定表达式之后才会发出，解决404 未找到的错误

# ng-file-upload

1. 参考网址

<https://github.com/danialfarid/ng-file-upload#features>

ngImgCrop是AngularJS的一个图片裁剪插件

ng-src可以接收引用的值是Base64 URL

ngf-src接受应用的file