# HTML

http://www.w3school.com.cn/

页面跳转的几种方式：

# CSS

# JavaScript

## My97DatePicker时间控件详细使用说明

**onpicking事件**

实现日期选择联动：

<label class="date-title">就诊日期：</label>

<input id="d5221" class="Wdate" type="text" onFocus="var d5222=$dp.$('d5222');WdatePicker({onpicked:function(){d5222.focus();},minDate:'%y-%M-%d',maxDate:'#F{$dp.$D(\'d5222\')}'})"/>

至

<input id="d5222" class="Wdate" type="text" onFocus="WdatePicker({onpicked:doDateSearch, minDate:'#F{$dp.$D(\'d5221\')}'})"/>

选中日期后触发事件，调用doDateSearch方法：

<input id="d5222" class="Wdate" type="text" onFocus="WdatePicker({onpicked:doDateSearch, minDate:'#F{$dp.$D(\'d5221\')}'})"/>

# angularjs

# jquery

## jQuery 语法

jQuery 语法是为 HTML 元素的选取编制的，可以对元素执行某些操作。

基础语法是：$(selector).action()

美元符号定义 jQuery

选择符（selector）“查询”和“查找” HTML 元素

jQuery 的 action() 执行对元素的操作

示例

$(this).hide() - 隐藏当前元素

$("p").hide() - 隐藏所有段落

$(".test").hide() - 隐藏所有 class="test" 的所有元素（注意selector中不能有空格）

$("#test").hide() - 隐藏所有 id="test" 的元素

提示：jQuery 使用的语法是 XPath 与 CSS 选择器语法的组合。

**文档就绪函数：**

您也许已经注意到在我们的实例中的所有 jQuery 函数位于一个 document ready 函数中：

$(document).ready(function(){

--- jQuery functions go here ----

});

这是为了防止文档在完全加载（就绪）之前运行 jQuery 代码。

如果在文档没有完全加载之前就运行函数，操作可能失败。

## [jQuery取得select选中的值](http://www.cnblogs.com/greatverve/archive/2010/02/03/1662565.html)

获取选中的值：$("#select1").val();

获取选中的文本：$("#select1  option:selected").text();

## jQuery更改超链接的href

$(document).ready(function(){

$(".js-btn").click(function(){

var hospitalId = $(".js-hospital").val();

if(hospitalId == "" ){

alert("请选择医院");

return false;

}

this.href = this.href + "?hospitalId=" + hospitalId;

});

});

提交超链接是HTML本身就会自己完成的事情，JavaScript只是在提交之前更改一些属性的值，如href的值。另外，JavaScript也可以通过return false 来中断提交。

## jquery获取li列表中当前点击的元素下的各种值

**列表的HTML代码如下：**

<ul>

<li>

<a href="#1" class="case-info J\_CaseInfo js-meeting" >

<div class="expert">

<div class="date J\_Date">

<p>2015-01-26</p>

<p>星期一</p>

<input type="hidden" name="deptId" value="2001">

<input type="hidden" name="doctorId" value="5001">

</div>

<i class="icon pm">aa</i>

</div>

</a>

</li>

<li>

<a href="#2" class="case-info J\_CaseInfo js-meeting" deptid="aaa">

<div class="expert">

<div class="date J\_Date">

<p>2015-01-27</p>

<p>星期二</p>

<input type="hidden" name="deptId" value="2002">

<input type="hidden" name="doctorId" value="5002">

</div>

<i class="icon pm">bb</i>

</div>

</a>

</li>

</ul>

**jquery获取各个元素的代码示例如下：**

<script type="text/javascript">

$(document).ready(function(){

$(".js-meeting").click(function(){

alert($(this).attr('href')); //#1 (获取某个属性的值)

alert(this.href); //http://localhost:8080/opc/hospital/hospitalschedule?hospitalId=2000001#1

alert($(this).html()); //返回html文档： <div class="expert">…… </div>

alert($(this).find('div i').text()); //aa

alert($(this).find('div div p').text()); //2015-01-26星期一

alert($(this).find('div div p').eq(0).text()); //2015-01-26

alert($(this).find('div div p').eq(1).text()); //星期一

alert($(this).find("input[name='doctorId']").val()); //5001

});

});

</script>



## jquery与动态弹出层

## jquery与Json

$(document).ready(function(){

$(".js-meeting").click(function(){

var hisPlanId = $(this).find("div div input").val();

var url = "${urlPath}" + "/opc/resource/getResourceByPlanId?hospitalId=${hospitalId}&hisPlanId=" + hisPlanId;

$.get(url, function(data, status){

//接收到的数据转化为JQuery对象，由JQuery为我们处理

var jsonData = eval(data);

if(jsonData.success == true){

$("tbody").html("<tr><th width=\"40%\">时间段</th><th width=\"20%\">就诊序号</th><th width=\"20%\">剩余号源</th><th width=\"20%\">操作</th></tr>");

//遍历对象数组，index是数组的索引号，objVal是遍历的一个对象。

$.each(jsonData.obj, function(index, objVal) {

//在table上追加一行

$("tbody").append("<tr><th height=\"40px\">"+objVal.startTime+"</th><th>"+objVal.orderNum+"</th><th>"+objVal.leaveNums+"</th><th><b>"+state+"</b></th></tr>");

});

}

});

});

});

public class Json {

private boolean success = false;

private String msg = "";

private Object obj = null; //实体列表，如 List<ResourceDto>

public boolean isSuccess() {

return success;

}

public void setSuccess(boolean success) {

this.success = success;

}

public String getMsg() {

return msg;

}

public void setMsg(String msg) {

this.msg = msg;

}

public Object getObj() {

return obj;

}

public void setObj(Object obj) {

this.obj = obj;

}

}

# ajax