**1. 了解jQuery**

**是什么: What?**

一个JS函数库: write less, do more

封装简化DOM操作(CRUD) / Ajax

**为什么用它: why?**

强大选择器: 方便快速查找DOM元素

隐式遍历(迭代): 一次操作多个元素

读写合一: 读数据/写数据用的是一个函数

链式调用: 可以通过.不断调用jQuery对象的方法

事件处理

DOM操作(CUD)

样式操作

动画

浏览器兼容

**如何使用: How?**

引入jQuery库

本地引入与CDN远程引入

测试版与生产版(压缩版)

使用jQuery

使用jQuery函数: $/jQuery

使用jQuery对象: $xxx(执行$()得到的)

**2. jQuery的2把利器**

**jQuery函数: $/jQuery**

jQuery向外暴露的就是jQuery函数, 可以直接使用

当成一般函数使用人: $(param)

param是function: 相当于window.onload = function(文档加载完成图片还没加载的监听)

<script>

//原生js，入口函数。页面上所有内容加载完毕，会执行。

//不仅文本加载完毕，而且图片也要加载完毕，在执行函数。

window.onload = function () {

alert(1);

}

//jquery的入口函数。 1.文档加载完毕，图片不加载的时候就可以执行这个函数。

$(document).ready(function () {

alert(1);

})

//jquery的入口函数。 2.文档加载完毕，图片不加载的时候就可以执行这个函数。

$(function () {

alert(1);

});

//jquery的入口函数。 3.文档加载完毕，图片加载完毕的时候在执行这个函数。

$(window).ready(function () {

alert(1);

})

</script>

param是选择器字符串: 查找所有匹配的DOM元素, 返回包含所有DOM元素的jQuery对象

param是DOM元素: 将DOM元素对象包装为jQuery对象返回 $(this)

param是标签字符串: 创建标签DOM元素对象并包装为jQuery对象返回

当成对象使用: $.xxx

each(obj/arr, function(key, value){})

trim(str)

**jQuery对象**

包含所有匹配的n个DOM元素的伪数组对象

执行$()返回的就是jQuery对象

基本行为:

length/size(): 得到dom元素的个数

[index]: 得到指定下标对应的dom元素

each(function(index, domEle){}): 遍历所有dom元素

index(): 得到当前dom元素在所有兄弟中的下标

**3. 选择器**

**是什么?**

有特定语法规则(css选择器)的字符串

用来查找某个/些DOM元素: $(selector)

**分类**

基本

#id

tagName/\*

.class

selector1,selector2,selector3: 并集

selector1selector2selector3: 交集

层次

找子孙后代, 兄弟元素

selector1>selector2: 子元素

selector1 selector2: 后代元素

// 后代

$("ul li").css("color", "red")

// 子代

$("ul>li").css("color", "blue")

// 下一个兄弟元素

$("#li1 + li").css("color", "green")

// 该元素往后的同级元素

$("#li1 ~ li").css("color", "pink")

过滤

在原有匹配元素中筛选出其中一些

:first

:last

:eq(index)

:lt

:gt

:odd

:even

:not(selector)

:empty

:hidden

:visible

[attrName]

[attrName=value]

                $("li:first").css("color", "red")

                $("li:last").css("color", "blue")

                $("li:lt(2)").css("color", "green")

                $("li:gt(2)").css("color", "blue")

                $("li:eq(2)").css("color", "yellow")

$("li:odd").css("color", "red")

                $("li:even").css("color", "pink")

                $("li:empty").css("color", "red")

                $("li:not(:empty)").css("color", "black")

                $("li:contains('4')").css("color", "green")

                $("li:nth-child(2)").css("color", "yellow")

表单

:input

:text

:checkbox

:radio

:checked: 选中的

**4. 属性/文本**

操作标签的属性, 标签体文本

attr(name) / attr(name, value): 读写非布尔值的标签属性

prop(name) / prop(name, value): 读写布尔值的标签属性

removeAttr(name)/removeProp(name): 删除属性

addClass(classValue): 添加class

removeClass(classValue): 移除指定class 如果不传参数则会移除所有类名

hasClass(classValue):返回布尔值 判断指定的类名是否存在

toggleClass(classValue):切换类样式，如果有该类 则移除，如果没有该类 则添加

val() / val(value): 读写标签的value

html() / html(htmlString): 读写标签体文本