1、什么是jquery?

jquery 是js的一个类库      提高开发效率

jquery的好处 ：

    代码简洁

    提高开发效率

    兼容性好

2、jquery中的$

$  等价  jQuery

3、jquery和js的区别

jquery是对js的封装  是js的一个类库

能用jquery实现的用js一定能实现

能用js实现的jquery不一定能够实现

4、jquery的选择器（查找页面选择）

基本选择器 ： $("") （重要）

    $("#id名")  根据给定的id查找页面元素

    $(".class")根据给定的class名查找页面元素

    $("html元素") 根据给定的元素标签查找

    $("#id,.class")  组合选择器

后面的所有的选择器都是在基本选择器的基础上加一些条件限制

层级选择器 ：

空格  :   查找所有的后代元素

>  ：  查找所有的子代元素

~  ：  查找下面所有的兄弟元素

+  ：  查找下面紧邻的兄弟元素

简单选择器 ：

    :first  $("li:first")

    :last

    :eq()   第几个  下标从0开始   $("li:eq(1)")

    :not()  $("li:not(.pp)")

    :lt    小于  $("li:lt(3)")索引

    :gt     大于

    :even   下标为偶数的元素    奇数行

    :odd    下标为奇数的元素

查找过滤 （重要）：

    .first()

    .last()

    .eq()

    .find()  相当于  层级选择器中的空格含义    查找所有的后代元素

    .children()  相等于 层级选择器中的 >     children()   或   children("")

    .filter()

     $("p").filter(".pp").css("background","blue");

    $("p").filter(function(){

        return 表达式; 如果表达式的结果为真   就返回了满足某个条件的p元素

    })

    .not()   除了....  $("div").not( "" )

    .parent()       查找父级....

    .next()    查找下面紧邻的兄弟元素  如果带参数，查找指定的兄弟      $("div+p")

    .nextAll() 查找下面所有的兄弟元素    如果有参数  执行查找的兄弟  否则就查找所有的兄弟 相当于 ~

    .prev()    前一个紧邻的兄弟元素 用法同next()

    .prevAll()

    .siblings()  查找除了自身之外的所有兄弟元素   如果有参数 就指定兄弟元素

6、js的选择器和jquery选择器的使用问题

如果是通过js方式查找到的元素，不能通过jq方法操作样式、属性、内容...  例如 ： document.getElementById().css()   错误

如果是通过jq的方式查找到元素，不能通过js方法操作样式....    例如 ： $().style.color = ""   错误

7、获取jquery中元素的个数

length  属性

size()  方法

8、jquery中的样式操作

获取样式值

    $().css("样式")

设置样式值

    .css("属性","值")  设置一个样式

    .css({"属性":"值",...})   设置多个样式

9、jquery中下标的获取方式

index()

    如果没有参数  ，  表示获取该元素在同辈元素中的位置

10、jquery元素和dom元素互换

jq元素转成dom元素 ：

     $().get(下标)     第几个元素

     $()[下标]

dom元素转jq元素 ：

    obj 是dom元素 转成jq元素     $(obj)    $(document)

11、样式类操作

addClass()  添加样式类

removeClass()删除样式类

12、jquery的属性操作

获取属性值 ：

    attr("属性")

设置属性 ：

    attr("属性","值")

    attr({ "属性":"值",... })

为某个元素添加自定义属性

    $().attr("level","1")

注意 ： attr 不能操作属性值为布尔值的属性  比如 ： checked   selected

使用  prop() 操作值为布尔类型的属性

13、属性选择器  [    ]

$("img[title]")   找到含有title属性的img

$("img[title=aa]")   找到title值为aa的img

$("img[title ^= aa]")   title以aa开始

$("img[title $= aa]")   title以aa结束

$("img[title \*= aa]")  包含....

$("img[title ！= aa]")   不包含...

14、内容操作

获取普通标签的内容

    html()   只能获取一个元素的内容

    text()    获取找到的多个元素的内容

设置操作 ：

    $().html("<b>内容</b>")   /  text("内容")

获取表单内容

val()

15、表单选择器  :

:text  :password   :type属性值

:checked  表示查找到被选中的单选或复选按钮

:radio:checked  获取被选中的单选按钮

:checkbox:checked

:selected  获取被选中的下拉列表

三级联动数据：

  var provArr = ["辽宁","山西","河北"];

    var cityArr = [["沈阳","大连","铁岭"],["大同","太原"],["邯郸","石家庄","唐山","雄安"]];

    var countyArr = [[["s1","s2"],["d1"],["昌图","莲花乡"]],[["t1","t2"],["y1"]],[["h1"],["s1","s2"],["tangshan1"],["xiongan1"]]];

16、动画

基本动画 ：

show() / hide() / toggle()

可以有三个参数 ：

    第一个参数 ： 动画执行的时间

    第二个参数 ： 动画执行方式 swing  linear

    第三个参数 ： 动画完成后的回调函数  function(){}

下拉/上拉 ：

slideDown()    通过高度的变化 动态的显示某个元素

slideUp()      通过高度的变化 动态的隐藏某个元素

slideToggle()  切换

淡入和淡出 ：

fadeIn(时间,function(){  })   通过透明度的改变动态的显示某个元素

fadeOut()  通过透明度的改变动态的隐藏某个元素

fadeToggle() 切换透明度

fadeTo(时间,透明度值)   调整透明度  注意 ： 时间不能省  如果时间是0，也要写0

自定义动画  animate

第一个参数 ： json

第二个参数 ： 动画执行时间

第三个参数 ： 回调函数

停止动画 ：

stop()  没有参数 ，表示停止当前正在运行的动画  （其余动画正常运动）

stop(true) 一个参数true  表示停止该元素的所有动画

stop(true,true) 两个参数 true  表示停止该元素的所有动画，但当前正在运动的动画会强制执行达到目标值

17、jquery遍历

each()

$().each(function(index,ele){

    index 遍历的元素的下标

    ele 遍历的元素

})

18 jquery中的事件

普通事件 ： $().事件（function(){}）

mouseover 和 mouseenter 区别 ：

前者会产生向上冒泡  频繁触发事件

mouseenter 不会冒泡  只触发一次

19 、jquery的事件绑定

bind() 绑定:只能绑定静态元素，不能绑定动态元素

用法：

$().bind("事件",function(){ } )   为一个对象绑定一个事件

$().bind("事件1 事件2",function(){  })  位一个对象绑定多个事件 但只有一个处理程序  （了解）

$().bind({

    事件  : function(){},

    事件  ： function(){},

    ....

})

为一个对象绑定多个事件 多个处理程序

$().unbind("事件")   为某个对象解除某个事件

delegate()事件委托在jQuery1.7中 .delegate()已被.on()取代

用法 ：

父级元素.delegate("子元素","事件",function(){

})

$().undelegate("事件")

on() 事件绑定或委托

用法 ： 实现绑定 用法和bind

       实现委托 ：

            $(父级元素).on("事件","子元素",function(){

            })

off()  解除事件绑定$("ul li").off("click")

one()  为某个对象绑定一次事件的方法

　jQuery 提供的.on()方法，绑定事件后并不是自动移除事件的，需要通过.off()来手工移除。

　　jQuery 提供的.one()方法，绑定元素执行完毕后自动移除事件，可以方法仅触发一次的事件，不能绑定动态元素

hover()  事件切换

$().hover(function(){

    鼠标移入enter

},function(){

    鼠标移出leaver

})

20 、jquery插件

插件就是一个工具

插件实现1 ：

    $.extend({

       方法名 : function(){},

       方法名 : function(){}

      });

    方法调用 ：  $.方法名()    全局调用

插件实现2 ：

    $.fn.extend({

        方法名 ： function(){},

        方法名 ： function(){},

        ....

     })

    方法调用：    选择器.方法名()   局部调用

插件中的this指向方法的调用者

21、ajax的异步请求之 load

load() 方法是一个局部的方法，通过jquery的选择器作为开始调用load方法

load()方法是异步的，加载的信息中如果需要事件响应，必须通过回调函数来实现，也就是说所有load加载的数据中的事件都要写在回调函数中

load()方法如果想要请求不同结构的内容，可以把这些内容写到一个文件中，请求数据时，在url后加一个请求容器的选择器名称即可（这里选择器建议使用基本选择器）

load()方法的参数有三个：

第一个：请求的url

第二个： 发送的数据  json格式  {}

第三个参数 ：回调函数

回调函数中有三个参数，分别是：

function(res,type,xhr){

       console.log(res);    //请求数据

       console.log( type );//请求的状态

        console.log(xhr);  //deffered 对象  （类似于js中promise对象）

}

22、ajax请求之  $.get 或 $.post

以 $. 开始的ajax请求方式是全局方法

$.get(三个参数)   以get方式请求数据

第一个参数：请求的路径

第二个参数：请求数据　　json对象   ｛｝

第三个参数： 回调函数   （回调函数中也有三个参数，同load）

$.post()  以post方式请求数据  用法同$.get()

23、ajax请求之  $.getJson()      $.getScript()

$.getJson("xxx.json"，data, 回调函数) 通过这种方式请求json数据   三个参数  ： url    data     function(){}

$.getScript("xxx.js"，回调) 通过这种方式请求js脚本数据 ，立即执行请求的脚本代码

24、$.ajax

jquery的ajax请求服务器传递过来的数据一般都是object对象（请求的数据是object类型）

用法：

     $.ajax({

          type:"get",

          url:"[http://127.0.0.1/jqAjx1706/data.json"](http://127.0.0.1/jqAjx1706/data.json%22);,

          success:function(){

              console.log("成功");

          }

     });

常用方法：

     $.ajax({

          type:"get",

          url:"[http://127.0.0.1/jqAjax1706/data.json"](http://127.0.0.1/jqAjax1706/data.json%22);,

          datatype:"json",//指定请求数据的类型

          data:{"name":"admin"},//向服务器发送数据

          success:function(res){

              //alert(typeof res);

              //此处处理服务器返回数据的业务逻辑

          }

     });

ajax跨域：

[https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su?wd="+txt+"&cb=fn](https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su?wd=%22+txt+%22&cb=fn)

$.ajax({

        type:"get",

        url:"[https://api.douban.com/v2/book/search?q=css&callback=fn&start=0&count=10"](https://api.douban.com/v2/book/search?q=css&callback=fn&start=0&count=10%22);,

        dataType:"jsonp",

        jsonCallback:"fn"   //设置回调函数

});

function  fn(msg){

    alert( msg );

}