jQuery

1. jQuery：2006年诞生，目的（1）简化DOM操作 （2）减少跨浏览器问题（兼容性问题）（3）它是一个JavaScript类库 （4）实现链式调用 （5）可以对外扩展封装自己的插件 （6）强大的选择器（sizzle）（7）无new构造
2. [www.jQuery123.com](http://www.jQuery123.com)是中文文档网站，[www.jQuery.com](http://www.jQuery.com)英文文档，下载无压缩jQuery（uncompressed），要多阅读源码提升自己
3. extend用来自己扩展插件
4. 先将jQuery 的js文件放到当前html,css,js文件夹中，jQuery 引入：<script src=’jquery3.3.3.js’></script>
5. $(‘#id’)相当于jQuery(‘#id’)选取id为id的元素，‘.class’选取类名为class的元素，jQuery选择器与CSS选择器类似
6. jQuery选择器：sizzle，如$(‘#list).css(‘position’,’absolute’).html(‘new content’)，常见问题是元素没有真正选到，注意关注浏览器返回length
7. 层级选择器：$(‘div ul li’),子级选择器:$(‘parent>child’)，$(‘prev+next’)选中后面第一个兄弟标签,$(‘prev~siblings’)选中prev后所有同级标签
8. 筛选选择器：$(‘li:first’)筛选list第一个，last筛选最后一个list，$(‘li:not(:first)’)筛选出除了第一个后的所有li元素，$(‘li’).eq(index)选取索引为index的li，$(‘li:even’)选中所有奇数的li，索引从零开始，odd选中偶数个li，索引从1开始
9. 内容选择器：$(‘li:contains(content)’)根据内容选取匹配的元素，选到li下有p标签的，$(‘li:has(selector)’)传p就能选出li下的p标签
10. var obj={
11. bindEvent:function(){
12. var that=this;
13. $('ul').on('click','li',function(){
14. console.log(this);
15. that.change(this);
16. })
17. },
18. change:function(dom){
19. console.log(dom);
20. $(dom).css('backgroundColor','red');
21. }
22. }
23. this问题,off清除事件，如$('ul li').off('click'),on绑定事件,可以自定义事件
24. obj.bindEvent();
25. $('ul li').on('click',function(){
26. $(this).css('backgroundColor','orange');
27. })
28. setTimeout(function(){
29. $('ul li').trigger('click');
30. },1000)
31. 主动给li绑定click事件,trigger主动触发事件(通常用于判断后触发事件)
33. $('body').on("playChange",function(e,data){
34. console.log(e);
35. console.log(data);
36. })
37. $('body').trigger('playChange','haha')
38. 自定义事件，注意e是第一个参数，系统传进去的事件对象
39. $('.box').hover(function(){
40. $(this).css('backgroundColor','black');
41. },function(){
42. $(this).css('backgroundColor','pink');
43. })
44. hover中第一个函数相当于js中的onmouseenter事件，第二个函数相当于js中onmouseleave事件
45. $('li').toggle();
46. toggle默认传参false,将选取元素的display变为none,传true则为block显式状态
48. $.type():判断当前数据类型，console.log($.type)
49. $.trim():去除字符串两端空格
50. $.noConflict():消除$与其他库的冲突,将$权限交出，重新自定义如my$=$.noConflict()
52. $.makeArray():将类数组转化为数组
53. $.inArray():$.inArray(value,array)返回查询值得索引值
54. $.fn==$.prototype
55. $.extend(true,{},obj1,obj2)深度克隆 $.fn.extend()
56. 多空格变单空格
57. $.extend({leftTrim:function(str){
58. var reg=/\s+/g;
59. return str.replace(reg,' ');
60. }})
61. $.fn.extend({
62. check:function(flag){
63. return this.each(function(value){
64. this.checked=flag;
65. })
66. }
67. })
68. jQyery源码手写：
69. (function(w,f){
70. f(w);
71. })(window,function(w){
72. function jQuery(str){
73. return new jQuery.fn.init(str);
74. }
75. jQuery.fn=jQuery.prototype={
76. init:function(str){
77. var obj=document.getElementsByTagName(str);
78. for(var i=0,len=obj.length;i<len;i++){
79. this[i]=obj[i];
80. }
81. this.length=obj.length;
82. return this;
83. },
84. css:function(){
85. console.log('css');
86. return this;
87. },
88. html:function(){
89. console.log('html');
90. return this;
91. }
92. }
93. jQuery.prototype.init.prototype=jQuery.prototype;
94. //改变入口函数this指向，让它也能调用jQuery原型上方法
95. window.$=window.jQuery=jQuery;
96. //暴露接口到全局
97. })