

## 一.dom元素的获取

- 1.id获取元素: `$('#name');` `$('#name .className');` `$('#name elementName');`
- 2.元素名称获取元素: `$('name');`
- 3.class名称获取元素: `$('.name');`

## 二.查找页面元素

- 1.`$('#div').find('li').eq(n);`//div jquery对象下第n个li元素
- 2.`$('div').first();``$('div').last();`//第一个和最后一个元素
- 3.`$('li').parent();`//p元素往上第一个父元素
- 4.`$('li').parents();`//p元素所有父级元素

## 三.事件的触发

- 1.点击事件: `document.addEventListener('click',fnName);`
- 2.按键事件: `document.addEventListener('keydown',fnName);`//keyup
- 3.bind事件: `$('#id').bind('click',function(){...});`

## 四.事件的解除

- 1.`document.removeEventListener('click',fnName);`
- 2.`$(' #id ').unbind('click');`

## 五.属性和css样式的获取修改

- 1.css样式的获取和设置  
`$('#id').css('border');`  
`$('#id').css('background','red');`
- 2.属性的获取和设置  
`$('img').attr('src');`  
`$('img').attr('src','images/1.jpg');`  
`$('img').prop('src','images/2.jpg');`
- 3.prop和attr的区别  
对于html标签本身固有属性, 在处理时, 使用prop方法;  
对于html标签自定义的属性, 在处理时, 使用attr方法;  
如checked、selected、disabled;

## 六.\$(document).ready事件的说明

```
$(document).ready(function(){  
    //jquery定义的在页面dom加载完成之后开始执行此处代码  
    //类似于页面onload执行完毕后的代码  
});
```

## 七.\$获取到的dom和通过document.getElementById(id)的区别和相互转换

- 1.jquery通过`$('#id')`获取到的是一个jquery对象;

2.document.getElementById('id');获取到的是一个dom对象;

3.转化:

1.jquery dom转化为dom对象:

```
$('#id')[0] == document.getElementById('id');
```

2.dom对象转化为jQuery对象:

```
let domObj = document.getElementById('id');
```

```
let jqObj = $(domObj);//直接将dom对象用$包装起来就好
```

## 八.赋值

1.html赋值

```
$('#div').html();//字符串型赋值
```

2.text()赋值

```
$('#div').text();//纯文字赋值
```

3.val()

```
$('#input').val();//获取文本框种的值
```

```
$('#input').val('asdasd');//设置文本框种的值
```

## 九.动画效果

1.显示隐藏

```
$('#p').show();$('#p').hide();$('#p').toggle();
```

2.滑动

```
$('#p').slideDown('time');//time单位是ms 用900毫秒时间将段落滑下
```

```
$('#p').slideUp('time');
```

```
$('#p').slideToggle('time');
```

3.淡入淡出

```
$('#p').fadeIn('time');//单位是ms, 用time毫秒时间将段落淡入, 比如time=900
```

```
$('#p').fadeOut('time');
```

```
$('#p').fadeToggle('time');
```

```
$('#p').fadeTo('slow',0.6);//slow可以为normal, fast等值, 用time毫秒时间将段
```

落的透明度调整到0.6

## 十.插件拓展机制

```
jQuery.fn.extend({  
    check:function(){...},  
    uncheck:function(){...}  
});
```

```
$('#p').check();
```

//或者这样使用

```
jQuery.extend({
    min:function(a,b){return a < b ? a : b},
    max:function(a,b){return a < b ? b : a}
});
$('p').min(1,2);//1
$('p').max(2,3);//3
```

## 十一.链式操作

1.原理：每次调用都要返回一个当前的this对象

2.实现：

```
function Person(){
    Person.prototype = {
        set:function(value){
            this.value = value;
            return this;//这步是关键
        },
        get:function(){
            return this.value * 2;
        }
    };
    function makeObj(){
        return new Person();
    }
}
```

```
let person = makeObj();
person.set(5).get();
```

3.缺点：只能在最后一步返回值。

4.优点：使得异步编程更加清晰。