

# 今日内容:

1. JQuery 高级
2. 动画
3. 遍历
4. 事件绑定
5. 案例
6. 插件

## JQuery 高级

### 动画

#### 三种方式显示和隐藏元素

##### 默认显示和隐藏方式

1. show([speed],[easing],[fn]))

1. 参数:

1. speed: 动画的速度。三个预定义的值("slow","normal","fast")或表示动画时长的毫秒数值(如: 1000)

2. easing: 用来指定切换效果, 默认是"swing", 可用参数"linear"

- swing: 动画执行时效果是 先慢, 中间快, 最后又慢
- linear: 动画执行时速度是匀速的

2. fn: 在动画完成时执行的函数, 每个元素执行一次。

2. hide([speed],[easing],[fn]))

3. toggle([speed],[easing],[fn]))

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
  <script type="text/javascript" src="../../js/jquery-3.3.1.min.js"></script>
</head>
<body>
<input id="hide" type="button" value="点击按钮隐藏div" onclick="hideFn()">
<input id="show" type="button" value="点击按钮显示div" onclick="showFn()">
<input id="toggle" type="button" value="点击按钮切换div显示和隐藏"
onclick="toggleFn()">

<div id="showDiv" style="width:300px;height:300px;background:pink">
```

```


显示和隐藏
</div>
<script>
    function hideFn() {
        $(function () {
            $("div").hide("slow","swing",function () {
                alert("完成");
            });
        });
    }

    function showFn() {
        $(function () {
            $("div").show("normal","linear",function () {
                alert("完成");
            });
        });
    }

    function toggleFn() {
        $(function () {
            $("div").toggle("fast","swing",function () {
                alert("完成");
            });
        });
    }

</script>

</body>
</html>


```

## 滑动显示和隐藏方式

1. slideDown([speed],[easing],[fn]): 滑动显示
2. slideUp([speed],[easing],[fn]): 滑动隐藏
3. slideToggle([speed],[easing],[fn])

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
    <script type="text/javascript" src="../../js/jquery-3.3.1.min.js"></script>
</head>
<body>
<input id="hide" type="button" value="点击按钮隐藏div" onclick="hideFn()">
<input id="show" type="button" value="点击按钮显示div" onclick="showFn()">
<input id="toggle" type="button" value="点击按钮切换div显示和隐藏"
onclick="toggleFn()">

```

```

<div id="showDiv" style="width:300px;height:300px;background:pink">
  div显示和隐藏
</div>

<script>
  function hideFn() {
    $(function () {
      $("div").slideUp("slow","swing",function () {
        alert("完成");
      });
    });
  }

  function showFn() {
    $(function () {
      $("div").slideDown("normal","linear",function () {
        alert("完成");
      });
    });
  }

  function toggleFn() {
    $(function () {
      $("div").slideToggle("fast","swing",function () {
        alert("完成");
      });
    });
  }

</script>

</body>
</html>

```

## 淡入淡出显示和隐藏方式

1. fadeIn([speed],[easing],[fn]): 显示
2. fadeOut([speed],[easing],[fn]): 隐藏
3. fadeToggle([speed],[easing],[fn])

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
  <script type="text/javascript" src="../js/jquery-3.3.1.min.js"></script>
</head>
<body>

```

```

<input id="hide" type="button" value="点击按钮隐藏div" onclick="hideFn()">
<input id="show" type="button" value="点击按钮显示div" onclick="showFn()">
<input id="toggle" type="button" value="点击按钮切换div显示和隐藏"
onclick="toggleFn()">

<div id="showDiv" style="width:300px;height:300px;background:pink">
  div显示和隐藏
</div>

<script>
  function hideFn() {
    $(function () {
      $("div").fadeOut("slow","swing",function () {
        alert("完成");
      });
    });
  }

  function showFn() {
    $(function () {
      $("div").fadeIn("normal","linear",function () {
        alert("完成");
      });
    });
  }

  function toggleFn() {
    $(function () {
      $("div").fadeToggle("fast","swing",function () {
        alert("完成");
      });
    });
  }

</script>

</body>
</html>

```

## 遍历

### js的遍历方式

- for(初始化值;循环结束条件;步长)

```

<!DOCTYPE html>
<html>

```

```

<head>
  <meta charset="UTF-8">
  <title></title>
  <script src="../../js/jquery-3.3.1.min.js" type="text/javascript" charset="utf-8"></script>
  <script type="text/javascript">

    </script>
</head>
<body>
<ul id="city">
  <li>北京</li>
  <li>上海</li>
  <li>天津</li>
  <li>重庆</li>
</ul>

  <script>

    $(function () {
      var citys = $("ul li");
      for(var i=0;i < citys.length;i++){
        //使用innerHTML获取属性值
        alert(i+": "+citys[i].innerHTML)
      }
    });

  </script>

</body>
</html>

```

## jq的遍历方式

### 1. jq对象.each(callback)

语法：

jquery对象.each(function(index,element){});

- index:就是元素在集合中的索引
  - element: 就是集合中的每一个元素对象
  - this: 集合中的每一个元素对象

回调函数返回值：

- true:如果当前function返回为false，则结束循环(break)。
- false:如果当前function返回为true，则结束本次循环，继续下次循环(continue)

```
<!DOCTYPE html>
```

```

<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <script src="../../js/jquery-3.3.1.min.js" type="text/javascript"
charset="utf-8"></script>
  <script type="text/javascript">

    </script>
</head>
<body>
<ul id="city">
  <li>北京</li>
  <li>上海</li>
  <li>天津</li>
  <li>重庆</li>
</ul>

  <script>

    $(function () {
      var citys = $("ul li");
      citys.each(function (index,element) {
        /*获取对象方式一: this,但是不能获取索引*/
        alert(this.innerHTML);
        /*获取方式二: 在回调函数中的参数*/
        /*使用js获取html文本*/
        alert(index+": "+element.innerHTML);
        /*使用jq获取html文本*/
        alert(index+": "+$(element).html());

        if ("上海"==$(element).html()){
          // return false;    结束循环, break
          // return true;     结束本次循环,continue
        }

      });
    });

  </script>

</body>
</html>

```

## 2. \$.each(object, [callback])

- Object: 与第一种还是有些区别的, 比如第一种必须传递jq对象, 而这里传递的可以是js对象
- callback: 回调函数

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <script src="../../js/jquery-3.3.1.min.js" type="text/javascript"
charset="utf-8"></script>
  <script type="text/javascript">

    </script>
</head>
<body>
<ul id="city">
  <li>北京</li>
  <li>上海</li>
  <li>天津</li>
  <li>重庆</li>
</ul>

  <script>

    $(function () {
      var citys = $("ul li");
      $.each(citys,function () {

        })
    });

  </script>

</body>
</html>

```

3. for..of: jquery 3.0 版本之后提供的方式，使用3.0之前的根本用不了  
for(元素对象 of 容器对象)，类似java中的增强for

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <script src="../../js/jquery-3.3.1.min.js" type="text/javascript"
charset="utf-8"></script>
  <script type="text/javascript">

    </script>
</head>
<body>

```

```
<ul id="city">
  <li>北京</li>
  <li>上海</li>
  <li>天津</li>
  <li>重庆</li>
</ul>

<script>

    $(function () {
        var citys = $("ul li");
        for(li of citys){
            alert($(li).html());
        }
    });

</script>

</body>
</html>
```

## 事件绑定

### jquery标准的绑定方式

- jq对象.事件方法(回调函数);
- 注：如果调用事件方法，不传递回调函数，则会触发浏览器默认行为。
- 表单对象.submit();//让表单提交

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <script src="../../js/jquery-3.3.1.min.js" type="text/javascript" charset="utf-8"></script>
  <script type="text/javascript">

    $(function () {
        /*
         * 最简单的点击事件
         * */
        $("#name").click(function () {
            alert("点击事件")
        });

        /*
```



```

    * 绑定事件，当然也可以使用链式编程
    * */
    $("#name").mouseover(function () {
        alert("鼠标移入");
    }).mouseout(function () {
        alert("鼠标移出");
    });

    /*
    * 获得焦点
    * */
    $("#name").focus();

});

</script>
</head>
<body>
<input id="name" type="text" value="绑定事件">
</body>
</html>

```

其他更多事件去看文档

## on绑定事件/off解除绑定

- jq对象.on("事件名称",回调函数)
- jq对象.off("事件名称")
- 如果off方法不传递任何参数，则将组件上的所有事件全部解绑

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title></title>
    <script src="../../js/jquery-3.3.1.min.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">

        $(function () {
            //使用on来绑定单击事件
            $("#btn").on("click",function () {
                alert("绑定单击事件");
            });

            $("#btn2").on("click",function () {
                $("#btn").off("click");//不传递参数那么就是解除所有的单击事件
                alert("解除btn绑定的单击事件");
            });
        });
    </script>

```

```

    </script>
</head>
<body>
<input id="btn" type="button" value="使用on绑定点击事件">
<input id="btn2" type="button" value="使用off解绑点击事件">
</body>
</html>

```

## 事件切换: toggle

- jq对象.toggle(fn1,fn2...): 点一下从事件fn1-->fn2, 再点fn2-->fn3, .....
- 当单击jq对象对应的组件后, 会执行fn1.第二次点击会执行fn2.....
- 注意: 1.9版本 .toggle() 方法删除,jQuery Migrate (迁移) 插件可以恢复此功能。

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title></title>
    <script src="../../js/jquery-3.3.1.min.js" type="text/javascript" charset="utf-8"></script>

    <!--使用插件来恢复toggle功能-->
    <script src="../../js/jquery-migrate-1.0.0.js"></script>
</head>
<body>

    <input id="btn" type="button" value="事件切换">
    <div id="myDiv" style="width:300px;height:300px;background:pink">
        点击按钮变成绿色, 再次点击红色
    </div>

    <script type="text/javascript">

        $(function () {

            $("#btn").toggle(function () {

                $("#myDiv").css("backgroundColor", "green");

            }, function () {

                $("#myDiv").css("backgroundColor", "pink");

            })

        });

    </script>
</body>

```

```
</html>
```

## 案例

### 广告显示和隐藏

```
<script>
    /*
    需求：
    1. 当页面加载完，3秒后。自动显示广告
    2. 广告显示5秒后，自动消失。

    分析：
    1. 使用定时器来完成。setTimeout（执行一次定时器）
    2. 分析发现JQuery的显示和隐藏动画效果其实就是控制display
    3. 使用 show/hide方法来完成广告的显示
    */

    //入口函数，在页面加载完成之后，定义定时器，调用这两个方法
    $(function () {
        //定义定时器，调用adShow方法 3秒后执行一次
        setTimeout(adShow,3000);
        //定义定时器，调用adHide方法，8秒后执行一次
        setTimeout(adHide,8000);
    });
    //显示广告
    function adShow() {
        //获取广告div，调用显示方法
        $("#ad").show("slow");
    }
    //隐藏广告
    function adHide() {
        //获取广告div，调用隐藏方法
        $("#ad").hide("slow");
    }
</script>
</head>
<body>
    <!-- 整体的DIV -->
    <div>

        <!-- 广告DIV -->
        <div id="ad" style="display: none;">
            
        </div>

        <!-- 下方正文部分 -->
        <div id="content">
            正文部分
        </div>
```

```
</div>
</body>
</html>
```

## 抽奖

```
<script language='javascript' type='text/javascript'>
    /*
        分析：
        1. 给开始按钮绑定单击事件
        1.1 定义循环定时器
        1.2 切换小相框的src属性
        * 定义数组，存放图片资源路径
        * 生成随机数。数组索引
        2. 给结束按钮绑定单击事件
        1.1 停止定时器
        1.2 给大相框设置src属性
    */
    var imgs = ["../img/man00.jpg",
        "../img/man01.jpg",
        "../img/man02.jpg",
        "../img/man03.jpg",
        "../img/man04.jpg",
        "../img/man05.jpg",
        "../img/man06.jpg",
    ];

    var startId;//开始定时器的id

    var index;//随机角标

    $(function () {

        //处理按钮是否可以使用的效果
        $("#startID").prop("disabled", false);
        $("#stopID").prop("disabled", true);

        //1. 给开始按钮绑定单击事件
        $("#startID").click(function () {

            // 1.1 定义循环定时器 20毫秒执行一次
            startId = setInterval(function () {

                //处理按钮是否可以使用的效果
                $("#startID").prop("disabled", true);
                $("#stopID").prop("disabled", false);

                //1.2生成随机角标 0-6
```

```

        index = Math.floor(Math.random() * 7);
        //0.000--0.999 --> * 7 --> 0.0-----6.9999

        //1.3设置小相框的src属性
        $("#img1ID").prop("src", imgs[index]);
    }, 20);
});

//2. 给结束按钮绑定单击事件
$("#stopID").click(function () {

    //处理按钮是否可以使用的效果
    $("#startID").prop("disabled", false);
    $("#stopID").prop("disabled", true);

    // 1.1 停止定时器
    clearInterval(startId);

    // 1.2 给大相框设置src属性
    $("#img2ID").prop("src", imgs[index]).hide();

    //显示1秒之后
    $("#img2ID").show(1000);
});
});
</script>

</head>
<body>

    <!-- 小像框 -->
    <div style="border-style:dotted;width:160px;height:100px">
        
    </div>

    <!-- 大像框 -->
    <div style="border-
style:double;width:800px;height:500px;position:absolute;left:500px;top:10px">
        
    </div>

    <!-- 开始按钮 -->
    <input
id="startID"
type="button"
value="点击开始"
style="width:150px;height:150px;font-size:22px">

    <!-- 停止按钮 -->
    <input

```

```
        id="stopID"
        type="button"
        value="点击停止"
        style="width:150px;height:150px;font-size:22px">
    </body>
</html>
```

## 插件：增强JQuery的功能

### 实现方式：

#### `$.fn.extend(object)`

- 增强通过Jquery获取的对象的功能，比如增强`$("#id")`，也叫增强对象级别的插件

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>01-jQuery对象进行方法扩展</title>
    <script src="../../js/jquery-3.3.1.min.js" type="text/javascript" charset="utf-8"></script>

</head>
<body>
<input id="btn-check" type="button" value="点击选中复选框" onclick="checkFn()">
<input id="btn-uncheck" type="button" value="点击取消复选框选中"
onclick="uncheckFn()">
<br/>
<input type="checkbox" value="football">足球
<input type="checkbox" value="basketball">篮球
<input type="checkbox" value="volleyball">排球

<script type="text/javascript">

    //使用插件，来给jq增加两个方法，check() 选中所有的复选框，uncheck() 取消选中所有的复选框

    //定义jq对象插件
    $.fn.extend({
        /*定义了一个check() 方法，所有的jq对象都可以来调用该方法*/
        check:function () {
            /*让复选框选中*/

            /*this: 调用该方法的jq对象*/
            this.prop("checked",true);
        },
        uncheck:function () {
            /*取消复选框的选中状态*/
```

```

        this.prop("checked", false);
    }
});

//使用check
$(function () {
    $("#btn-check").click(function () {
        $("input[type='checkbox']").check();
    });

    $("#btn-uncheck").click(function () {
        $("input[type='checkbox']").uncheck();
    });
});

</script>

</body>
</html>

```

## \$.extend(object)

- 增强JQuery对象自身的功能，比如增强\$或者jQuery，也是增强全局级别的插件

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>01-jQuery对象进行方法扩展</title>
    <script src="../../js/jquery-3.3.1.min.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">
        //对全局方法扩展2个方法，扩展min方法：求2个值的最小值；扩展max方法：求2个值最大值
        $.extend({
            max:function (a,b) {
                //返回两个数中的较大值
                return a>=b?a:b;
            }
        })

        $.max(1,2);
    </script>
</head>
<body>
</body>
</html>

```

