# 今天课程目的

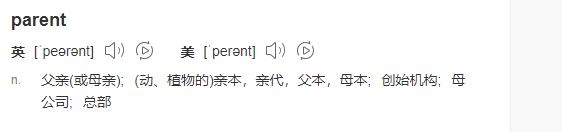
* 今天需要将jQuery函数库中常用方法学习完毕
* 今天需要通过jQuery函数库开发一些使用效果

## 1.1节点关系函数

概述：jQuery函数库中拥有一些节点（标签:element）关系一些方法提供给我们使用。

比如：可以获取某一个标签的父元素、兄弟元素、儿子元素等等.

### 1.1.1parent



概述：它是jQuery函数库提供一个节点关系方法，主要的作用是可以获取到某一个标签的父元素。

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!--利用JQ时候切记需要通过script标签进行引包  -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            width: 500px;

            height: 100px;

            border: 1px solid black;

            margin: 100px auto;

        }

        p {

            float: left;

            width: 100px;

            height: 100px;

            background-color: pink;

            margin-right: 10px;

        }

    </style>

</head>

<body>

    <div>

        <p></p>

        <p></p>

        <p></p>

        <p></p>

    </div>

</body>

</html>

<script>

    //给P标签绑定单击事件，要他们的父元素div背景颜色为青色

    $("p").click(function () {

        $("p").parent().css({

            "background": "cyan"

        })

    })

</script>

* parent方法可以获取到某一个标签父元素.

### 1.1.2this【函数的上下文】

概述：this不是jQuery函数库提供的，因为系统当中函数就有上下文this。

函数的上下文this也可以在JQ当中使用。

总结规律：函数作为事件处理函数执行，上下文是当前这个触发事件的元素。

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!--利用JQ时候切记需要通过script标签进行引包  -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            width: 500px;

            height: 100px;

            border: 1px solid black;

            margin: 100px auto;

        }

        p {

            float: left;

            width: 100px;

            height: 100px;

            background-color: pink;

            margin-right: 10px;

        }

    </style>

</head>

<body>

    <div>

        <p>1</p>

        <p>2</p>

        <p>3</p>

        <p>4</p>

    </div>

</body>

</html>

<script>

    //单击某一个p标签，当前被单击的这个p标签背景颜色为天蓝色

    //给全部P标签绑定单击事件

    $("p").click(function () {

        //单击某一个p，当前被单击的这个p标签背景颜色为天蓝色

        $(this).css({

            "background": "skyblue"

        })

    });

</script>

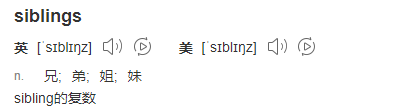
* 函数的上下文this，也可以在JQ当中使用
* 它代表的是当前触发这个事件的元素（标签）

函数上下文判断五大规律？

函数的上下文(this)，函数不同执行方法，上下文是不同的.

1. 函数名+小括号=>window
2. 函数作为事件的处理函数执行=>当前触发这个事件元素
3. 函数作为定时回调函数每隔一段事件执=>window
4. 函数作为数组元素枚举出来执行=>当前数组
5. 函数作为对象的方法执行，谁最后打点调用上下文即为谁

### 1.1.3siblings



概述：它是jQuery函数库提供一个关系方法，主要的作用是可以获取到某一个标签其他的姊妹元素们。

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!--利用JQ时候切记需要通过script标签进行引包  -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            width: 500px;

            height: 100px;

            border: 1px solid black;

            margin: 100px auto;

        }

        p {

            float: left;

            width: 100px;

            height: 100px;

            background-color: pink;

            margin-right: 10px;

        }

    </style>

</head>

<body>

    <div>

        <p>1</p>

        <p>2</p>

        <p>3</p>

        <p>4</p>

    </div>

</body>

</html>

<script>

    //siblings:获取姊妹元素

    //单击某一个P让他的背景颜色为黄色，他的其余兄弟元素们为绿色

    //给P标签绑定单击事件

    $("p").click(function () {

       $(this).css({"background": "yellow"}).siblings().css({"background": "green"})

    });

    //链式语法【来源于OC语言:搞iOS苹果开发语言】

    //为了方便(书写语句的时候)可以从左到右连续打点

    //看着比较整洁

    //链式语法执行顺序：从左到右一次执行

</script>

### 1.1.4children

概述：它也是jQuery函数库提供的一个节点关系函数，主要的作用是可以获取到某一个节点儿子元素。

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!--利用JQ时候切记需要通过script标签进行引包  -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            width: 500px;

            height: 100px;

            border: 1px solid black;

            margin: 100px auto;

        }

        p {

            float: left;

            width: 100px;

            height: 100px;

            background-color: pink;

            margin-right: 10px;

        }

    </style>

</head>

<body>

    <div>

        <p>1</p>

        <p>2</p>

        <p>3</p>

        <p>4</p>

        <span>5</span>

    </div>

</body>

</html>

<script>

    //将四个P标签背景颜色为黑色

    /\*第一种写法：

    $("p").css({

        "background": "black"

    });

    \*/

    $("div").children().css({

        "background": "black"

    });

</script>

### 1.1.5折叠卡片特效

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!-- 引包 -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        ul {

            width: 600px;

            border: 1px solid black;

            margin: 0px auto;

            list-style: none;

        }

        li {

            border-bottom: 1px dashed red;

            text-align: center;

            cursor: pointer;

        }

        p {

            background-color: black;

            color: white;

            display: none;

        }

        .open {

            display: block;

        }

    </style>

</head>

<body>

    <ul>

        <li>

            <h4>万众一心加油干 越是艰险越向前 共谱时代华章</h4>

            <p class="open">

                　历史的发展，总有一些关键的时间节点。习近平主席强调，

                “2020年是具有里程碑意义的一年。我们将全面建成小康社会

                ，实现第一个百年奋斗目标。”这既是决胜期，也是攻坚期。以必胜的信心、昂扬的斗志、扎实的工作，确保全面建成小康社会圆满收官，乘势而上开启全面建设社会主义现代化国家新征程

                ，向第二个百年奋斗目标进军，这是历史赋予我们的光荣使命和时代重任。

            </p>

        </li>

        <li>

            <h4>万众一心加油干 越是艰险越向前 共谱时代华章</h4>

            <p>

                　历史的发展，总有一些关键的时间节点。习近平主席强调，

                “2020年是具有里程碑意义的一年。我们将全面建成小康社会

                ，实现第一个百年奋斗目标。”这既是决胜期，也是攻坚期。以必胜的信心、昂扬的斗志、扎实的工作，确保全面建成小康社会圆满收官，乘势而上开启全面建设社会主义现代化国家新征程

                ，向第二个百年奋斗目标进军，这是历史赋予我们的光荣使命和时代重任。

            </p>

        </li>

        <li>

            <h4>万众一心加油干 越是艰险越向前 共谱时代华章</h4>

            <p>

                　历史的发展，总有一些关键的时间节点。习近平主席强调，

                “2020年是具有里程碑意义的一年。我们将全面建成小康社会

                ，实现第一个百年奋斗目标。”这既是决胜期，也是攻坚期。以必胜的信心、昂扬的斗志、扎实的工作，确保全面建成小康社会圆满收官，乘势而上开启全面建设社会主义现代化国家新征程

                ，向第二个百年奋斗目标进军，这是历史赋予我们的光荣使命和时代重任。

            </p>

        </li>

        <li>

            <h4>万众一心加油干 越是艰险越向前 共谱时代华章</h4>

            <p>

                　历史的发展，总有一些关键的时间节点。习近平主席强调，

                “2020年是具有里程碑意义的一年。我们将全面建成小康社会

                ，实现第一个百年奋斗目标。”这既是决胜期，也是攻坚期。以必胜的信心、昂扬的斗志、扎实的工作，确保全面建成小康社会圆满收官，乘势而上开启全面建设社会主义现代化国家新征程

                ，向第二个百年奋斗目标进军，这是历史赋予我们的光荣使命和时代重任。

            </p>

        </li>

    </ul>

</body>

</html>

<script>

    //jQuery函数库核心理念 写的少但是做的多

    $("h4").click(function () {

        //链式语法执行顺序:从左到右

        $(this).siblings().slideDown(2000).parent().siblings().children("p").slideUp(2000);

    })

</script>

## 1.2animate函数

概述：它也是jQuery函数库提供一个动画函数，主要的作用是可以让元素添加一些2D动画效果。

使用格式如下：

$(selector).animate(JSON,TIME,CALLBACK);

* JSON：动画属性设置【left、width、top等等】
* TIME:动画时间设置（单位是MS）
* CALLBACK：函数，这个函数会在动画结束的时候执行一次

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!-- 引包 -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            /\* 定位元素才有left|top|bottom|right \*/

            position: relative;

            width: 100px;

            height: 100px;

            background-color: pink;

            border: 1px solid black;

        }

    </style>

</head>

<body>

    <div>中国</div>

</body>

</html>

<script>

    $("div").animate({

        "left": 1000,

        "top": 100,

        "width": 150,

        "height": 50,

        "background": "cyan",

        "color": "white",

        "border": "20px solid red"

    }, 5000, function () {

        //让div遍椭圆

        $(this).css({

            "borderRadius": "50%",

            "textAlign": "center"

        })

    });

</script>

* animate是JQ封装一个函数动画，并不是全部的样式都可以参与动画（涉及颜色一般都不能参与动画）

### 1.2.1动画积累问题

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            position: relative;

            width: 100px;

            height: 100px;

            border-radius: 50%;

            background-color: red;

        }

    </style>

</head>

<body>

    <button class="btn1">去北京</button>

    <button class="btn2">去东京</button>

    <div></div>

</body>

</html>

<script>

    //第一个按钮

    $(".btn1").click(function () {

        $("div").stop(true).animate({

            "left": 1000

        }, 2000);

    });

    //第二个按钮

    $(".btn2").click(function () {

        $("div").stop(true).animate({

            "left": 0

        }, 2000)

    })

</script>

* stop(true):将匹配元素全部动画清除

## 1.3实战案需求

1：淘宝轮播图

* 自动轮播（每隔一段时间换一张图）
* 鼠标放上去（自动轮播停止），鼠标移出（自动轮播有开始）
* 有左右按钮，单击左右按钮可以切换图片
* 单击不同的分页器，可以显示不同图片

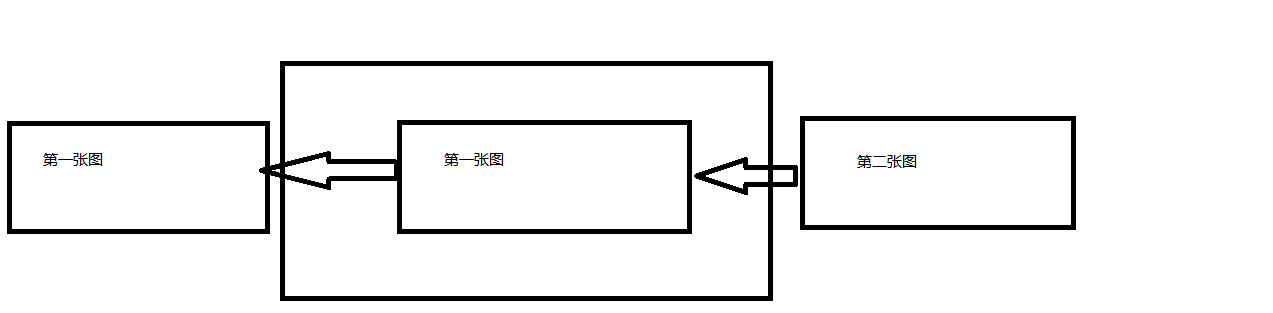
2：百度新闻导航效果



* 鼠标移上去、移下来事件

# 三位轮播图

概述：轮播图英文（carousel）,类似于淘宝、京东首页轮播图，前端工程师经常叫做‘三位轮播|传统轮播’



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!-- 引入外部样式 -->

    <link rel="stylesheet" href="./css/default.css">

    <!-- 引包，引入JQ -->

    <script src="./js/jQuery.min.js"></script>

</head>

<body>

    <div>

        <ul>

            <li class="current"><img src="./img/1.jpg" alt=""></li>

            <li><img src="./img/2.jpg" alt=""></li>

            <li><img src="./img/3.jpg" alt=""></li>

            <li><img src="./img/4.jpg" alt=""></li>

            <li><img src="./img/5.jpg" alt=""></li>

        </ul>

        <button class="lbtn">&lt;</button>

        <button class="rbtn">&gt;</button>

        <ol>

            <li class="cur">1</li>

            <li>2</li>

            <li>3</li>

            <li>4</li>

            <li>5</li>

        </ol>

    </div>

</body>

</html>

<script>

    //右侧按钮单击事件

    //信号量：全局变量（控制某一个li运动）

    var idx = 0;

    //右侧按钮的代码进行封装

    function rhandle() {

        //当前显示这张图片从left为0位置运动到-600

        $("ul li").eq(idx).css({

            "left": 0

        }).stop(true).animate({

            "left": -600

        }, 500);

        //全局变量累加1【下一张图要进入容器里面】

        idx++;

        idx = idx > 4 ? 0 : idx;

        //下一张图进入

        $("ul li").eq(idx).css({

            "left": 600

        }).stop(true).animate({

            "left": 0

        }, 500);

        //底下分页器也跟着变化

        $("ol li").eq(idx).addClass("cur").siblings().removeClass();

        //addClass代表给标签添加类名、removeClass代表给标签移出类名

    }

    $(".rbtn").click(rhandle);

    //开启定时器让轮播图自动轮播

    var timer = setInterval(rhandle, 1500);

    //鼠标移上事件

    $("div").mouseenter(function () {

        //清除定时器

        clearInterval(timer);

    });

    //鼠标离开事件

    $("div").mouseleave(function () {

        timer = setInterval(rhandle, 1500);

    });

    //左侧按钮的单击事件

    $(".lbtn").click(function () {

        //当前显示图片运动的动画

        $("ul li").eq(idx).css({

            "left": 0

        }).stop(true).animate({

            "left": 600

        }, 500);

        //全局变量需要累减

        idx--;

        idx = idx < 0 ? 4 : idx;

        //下一张图片动画

        $("ul li").eq(idx).css({

            "left": -600

        }).animate({

            "left": 0

        }, 500);

        //底下分页器也跟着变化

        $("ol li").eq(idx).addClass("cur").siblings().removeClass();

    });

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

    //底下的分页器业务

    $("ol li").click(function () {

        //获取单击的那个分页器li索引值

        var index = $(this).index();

        //分两种情况讨论

        if (index > idx) {

            //当前显示这张图片动画

            $("ul li").eq(idx).css({

                "left": 0

            }).stop(true).animate({

                "left": -600

            }, 500);

            //单击那个分页器对应的图片动画

            $("ul li").eq(index).css({

                "left": 600

            }).stop(true).animate({

                "left": 0

            }, 500);

            //全局变量idx从新赋值

            idx = index;

            //底下分页器也跟着变化

            $("ol li").eq(idx).addClass("cur").siblings().removeClass();

            //addClass代表给标签添加类名、removeClass代表给标签移出类名

        } else {

            //当前显示这张图片动画

            $("ul li").eq(idx).css({

                "left": 0

            }).stop(true).animate({

                "left": -600

            }, 500);

            //单击那个分页器对应的图片动画

            $("ul li").eq(index).css({

                "left": 600

            }).stop(true).animate({

                "left": 0

            }, 500);

            //全局变量idx从新赋值

            idx = index;

            //底下分页器也跟着变化

            $("ol li").eq(idx).addClass("cur").siblings().removeClass();

            //addClass代表给标签添加类名、removeClass代表给标签移出类名

        }

    })

</script>

* addClass||removeClass:分别是给标签添加类名+移动类名
* eq方法：可以获取到某一个索引值标签
* index:可以获取到某一个标签索引值

# 百度新闻导航

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <!-- 引入JQ -->

    <script src="./js/jQuery.min.js"></script>

    <style>

        \* {

            margin: 0;

            padding: 0;

        }

        div {

            position: relative;

            width: 100%;

            height: 40px;

            background-color: #01204f;

        }

        ul {

            position: absolute;

            width: 100%;

            height: 40px;

            list-style: none;

        }

        ul li {

            float: left;

            padding: 10px;

            color: white;

        }

        div.slider {

            position: absolute;

            width: 52px;

            height: 41px;

            background: red;

            opacity: .8;

        }

    </style>

</head>

<body>

    <div>

        <div class="slider"></div>

        <ul>

            <li>首页</li>

            <li>国内</li>

            <li>国际</li>

            <li>军事</li>

            <li>财经</li>

            <li>娱乐</li>

            <li>体遇</li>

            <li>互联网</li>

            <li>科技</li>

            <li>科技</li>

            <li>游戏</li>

            <li>女人</li>

            <li>汽车</li>

        </ul>

    </div>

</body>

</html>

<script>

    //全部li添加鼠标以上事件

    $("li").mouseenter(function () {

        //红色滑块算法(要获取到鼠标进入的这个li,前面全部姊妹元素之和)

        //prevAll:代表的是获取某一个元素前面全部姊妹元素

        //累加器

        var sum = 0;

        $(this).prevAll().each(function () {

            //获取每一个姊妹元素宽度

            sum += $(this).innerWidth();

        });

        //让红色滑块运动起来

        $(".slider").stop(true).animate({

            "left": sum,

            "width": $(this).innerWidth()

        }, 500);

    });

    //鼠标移出，红色滑块归还原始位置

    $("div").mouseleave(function () {

        $(".slider").stop(true).animate({

            "left": 0,

            "width": $("ul li").eq(0).innerWidth()

        }, 500);

    })

</script>

* prevAll：可以获取到某一个节点前面全部姊妹元素
* each：可以遍历到每一个匹配节点
* innerWidth:可以获取某一个标签宽度(也算上内边距数据)

# 工作部分组成

项目经理:管理项目的（PMP证书）

后台工程师：玩数据库【给全端提供数据来源】

前端工程师：web、iOS、安卓.开发页面的

测试工程师：看你的程序是否有问题

产品经理：设计原型图（网站写成什么样子，按照他的图进行开发）

需求工程师：和产品差不多，天天去甲方哪里开会（收集甲方需求）