

« به نام خداوند بخشنده و مهربان »
هیچ نویسنده‌ای نباید از نوشتن خودداری کند
همان گونه که خدا او را آموزش داده است.
سوره‌ی بقره آیه‌ی 282



آموزش سریع و آسان jquery
نویسنده : سید محمدرضا حسینیان راوندی
وبلاگ: <http://barnamenevisi.blogsky.com>
وبسایت ناشر: <http://pachenar.ir>
وبسایت : <http://pachenar.ir>
رایانامه: ravand@chmail.ir
تاریخ انتشار: 1392/9/21

مقدمه:

jquery یکی از کتابخانه های جاوا اسکریپت است. که کار را برای ما خیلی ساده کرده به طوری که برای نوشتن یک برنامه با خود جاوا اسکریپت امکان دارد که ما مجبور بشیم زیاد کدنویسی کنیم ولی جی کوئری کار ما را راحت تر میکند. مثلاً ممکنه برای نوشتن یک برنامه با جاوا اسکریپت مجبور بشیم 30 خط کد بنویسیم ولی با جی کوئری این برنامه را مثلاً در 5 خط می نویسیم. و البته کدنویسی در جی کوئری نسبت به خود جاوا اسکریپت خیلی ساده تر و قابل فهم تر است.

پیش نیاز ها برای یادگیری jquery : html ، css و javascript
اگه شما قبل از یادگیری جی کوئری جاوا اسکریپت بلد باشید مطمئناً موفق تر خواهید بود.
ولی زیاد ضروری نیست.

ویژگی های jquery :

jQuery نسبت به خود جاوا اسکریپت با مرورگرها سازگار تره.
دارای افزونه های (plug in) زیادی می باشد.
سمت کلاینت بکار میره.
کدنویسی ساده ای داره.
ایجاد متحرک سازی . و پشتیبانی از ایجکس. و سورس باز بودن از دیگر ویژگی های جی کوئری است.

تعاریف:

سمت کلاینت (سمت کاربر) چیست؟

یعنی کاربر در کامپیوتر خودش به برنامه ی شما دسترسی داره. مثلاً کدهای html و jquery رو میشه بهش دسترسی داشت. ولی کدهای php که سمت سرور هست را نمیشه بهش دسترسی پیدا کرد. شما کدی را که با php نوشتی و روی سایت گذاشتی کسی نمیتونه بهش دسترسی پیدا کنه جز اینکه مجوز دسترسی به هاست شما رو داشته باشه.

پلاگین (plug in) چیست؟

خیلی از شرکت ها بعد از مدتی برای اینکه امکانات برنامه هاشون بیشتر بشه میان و یک سری امکانات رو اضافه بر ، برنامه ای که از قبل نوشتن ، را برای برنامه ی مورد نظرشون آماده میکنند که فرد با نصب کردن یا اضافه کردن به برنامه ی قبلی میتونه از این امکانات جدید استفاده کنه.

برای کار با جی کوئری به سایت <http://jquery.com/> بروید و فایل jquery را دانلود کنید.

یک مثال ساده با جی کوئری:

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      // کدهای جی کوئری
    </script>
  </head>
  <body>
    <!-- کدهایی که از کدهای جی کوئری فرمان میگیرند-->
  </body>
</html>
```

فایل بالا را با نام مثلا test.htm ذخیره کنید.

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("button").click(function () {
          alert("سلام راوندی");
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
  </body>
</html>
```

توضیحات:

برای خواننده شدن کدها همراه با بارگذاری صفحه از متد `ready()` استفاده کردیم. با کلیک روی دکمه ، متد `click()` دستور را دریافت کرده و دستورات داخل آن اجرا می شوند در اینجا دستور `alert` پیام مورد نظر را در یک پنجره نمایش می دهد. در این کتاب با متدهای بیشتری آشنا خواهیم شد.

می توانید از `id` و `class` هم برای ارتباط دکمه و متد استفاده کنید:

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("#btn").click(function () {
          alert("سلام راوندی");
        });
      });
    </script>
  </head>
  <body>
    <button id="btn">کلیک کنید</button>
  </body>
</html>

```

درست عین CSS . اگر بخواهید از class استفاده کنید به جای علامت # از نقطه استفاده کنید.

کم کم در طول آموزش با روش های دیگه هم آشنا می شوید.
به جای خط زیر :

```

$('document').ready(function(){
  // می توانید از خط زیر استفاده کنید که سریع تر عمل میکند:
  $(window).load(function() {
    // و یا حتی :

```

```

$(function () {
  // روش زیر وقتی صفحه به طور کامل بارگزاری شد باعث می شود تابع اجرا شود .
  $(document).ready()
  // یا استفاده از یک تابع به صورت زیر:

```

```

function tabe(){
  // کدهای داخل تابع
}
// و دکمه را به روش زیر ایجاد میکنیم که با کلیک روی دکمه تابع فعال شود و کدهای داخل
  // تابع اجرا شود:

```

```

<input type="button" value="ارسال" onclick="tabe()"/>
  // که این روش توی خود جاوا اسکریپت بکار می رود.

```

برنامه ی زیر را امتحان کنید ببینید چه اتفاقی می افتد؟

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      alert("اولین پیام");
      $('document').ready(function(){
        alert("بارگذاری کامل صفحه");
      });
      alert("دومین پیام");
    </script>
  </head>
  <body>
</body>
</html>

```

حتی میشه کدهای جی کوئری رو در قسمت <body></body> هم نوشت.

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
  </head>
  <body>
    <button id="btn">کلیک کنید</button>
    <script type="text/javascript">
      $("#btn").click(function () {
        alert("سلام راوندی");
      });
    </script>
  </body>
</html>

```

اینطوری دیگه از اون روش ها نیازی نیست استفاده کنیم.

افکت ها در jquery :

Hide(), Show(), Toggle(), slideUp(), slideToggle(),
slideDown(),fadeIn(),slideUp(),fadeOut(),fadeTo(),Animate(),
slideToggle()

hide() : مخفی کردن متن.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("#btn").click(function () {
          $("#matn").hide();
        });
      });
    </script>
  </head>
  <body>
    <button id="btn">کلیک کنید</button>
    <div id="matn">متن نوشته شده</div>
  </body>
</html>

```

کار مخفی کردن متن را متد `hide()` (به معنای مخفی) انجام میدهد.
ساختار:

`$(selector).hide(speed,easing,callback)`
 پارامترهایی که میگیرد: `speed,easing,callback`.
Speed: که سه مقدار `milliseconds` و `"slow"` و `"fast"` را می گیرد.
 مثال برای `milliseconds`:
 اگر متد را به این شکل در برنامه ی بالا بنویسیم:

```

$("#matn").hide(1000);

```

متن مدت 1000 میلی ثانیه طول می کشید که محو شود.
`"slow"` به معنای آهسته.
`"fast"` به معنای سریع، و سریع مخفی می شود.

پارامتر `easing`: مقادیر `"swing"` و `"linear"` را می گیرد..
`"swing"`: به معنای نوسان، تاب، نوعی رقص. شما وقتی ازش استفاده میکنی نوعی حالت انیمیشن ایجاد می شود. `"linear"` به همین شکل است ولی نوع متحرک سازیشون فرق داره.

Callback: کاری که بعد از کامل شدن عملیات متد `hide()` انجام میدهد.

```

$("p").hide(1000,function(){

```

```
alert("این پاراگراف مخفی شد");
});
```

کاری که بعد از کامل شدن عملیات یعنی کامل شدن کار مخفی سازی انجام شد دادن پیام بود.

دادن مقدار متغیر به جای سلکتور:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("button").click(function() {
          var mo = "php";
          $(". " + mo).hide();
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <div class="php">آموزش</div>
  </body>
</html>
```

حالا برای نمایش متن از متد show() استفاده میکنیم:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("#btn").click(function () {
          $("#matn").show();
        });
      });
    </script>
    <style type="text/css">
    #matn{
      display:none;
    }
  </style>
  </head>
  <body>
    <div id="matn">
      <div id="btn">
        دکمه
      </div>
      <div id="matn">
        متن
      </div>
    </div>
  </body>
</html>
```

```

    }
    </style>
</head>
<body>
<button id="btn">کلیک کنید</button>
<div id="matn">متن نوشته شده</div>
</body>
</html>

```

با کلیک روی دکمه متد کلیک فراخوانی میشه و برنامه ی داخل آن اجرا میشه . در قسمت css ما div را با استفاده از دستور `display:none;` به صورت مخفی درآوردیم . حالا با اجرای متد `show()` متن ما از حالت مخفی خارج میشه.

حالت های `display` :

متن به صورت افقی در میاد:

`display:inline;`

متن مخفی میشه:

`display:none;`

متن مخفی شده نمایش داده میشه:

`display:block;`

متد `show()` هم پارامترهای `hide()` رو میگیره.

`toggle()` : پارامترهای این متد هم مثل متد `hide()` است.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("button").click(function () {
          $("p").toggle("slow");
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <p>سید محمدرضا حسینیان راوندی</p>
    <p>http://pachenar.ir</p>
  </body>
</html>

```


با کلیک روی دکمه متن به صورت آهسته مخفی و با کلیک دوباره روی دکمه متن به صورت آهسته آشکار می شود. حالا میخوایم برنامه ای بنویسیم که با کلیک روی دکمه صفحه ی رویی محو بشه و با کلیک دوباره ، صفحه ی رویی نمایش داده بشه اونم به صورت افکت:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      #up
      {
        position: absolute;
        z-index: -1;
        background-color: #28DD39;
        height: 100px;
        width: 100px;
      }

      #down
      {
        position: absolute;
        z-index: -2;
        background-color: #dbbfff;
        height: 100px;
        width: 100px;
      }
    </style>
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("button").click(function ()
        {
          $("#up").toggle({"left" : "200px"});
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <div id="up">متن اول</div>
    <div id="down">متن دوم</div>
  </body>
</html>
```

ما با استفاده از دستورات زیر لایه ایجاد کردیم:

```
position:absolute;  
z-index:1;
```

برنامه ی زیر با کلیک روی دکمه، zaher فعال می شود. و مکعب نمایش داده می شود و با کلیک دوباره روی دکمه ، تابع makhfi فعال می شود.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style type="text/css">  
      .mokaab  
      {  
        display:none;  
        background-color:#d3b2f5;  
        height:200px;  
        width:200px;  
      }  
    </style>  
    <script src="jquery.js" type="text/javascript">  
    </script>  
    <script type="text/javascript">  
      $('document').ready(function () {  
        $("button").toggle(zaher,makhfi);  
      });  
      function zaher() {  
        $(".mokaab").show();  
      }  
      function makhfi() {  
        $(".mokaab").hide();  
      }  
    </script>  
  </head>  
  <body>  
    <button>کلیک کنید</button>  
    <div class="mokaab"></div>  
  </body>  
</html>
```

: slideUp()

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">
```

```

<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
$('#clickme').click(function() {
    $('#aks').slideUp('slow');
});
});
</script>
</head>
<body>
<div id="clickme">
کلیک کنید
</div>

</body>
</html>

```

با کلیک روی متن عکس به آرامی مخفی می شود. و تصویر مثل پرده کرکره به آرامی به بالا کشیده می‌شود و در نهایت مخفی می‌شود. پارامترهای این متد هم مثل متد hide() است.

: slideToggle()

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
$('#clickme').click(function() {
    $('#aks').slideToggle('slow');
});
});
</script>
</head>
<body>
<div id="clickme">
کلیک کنید
</div>

</body>
</html>

```

با کلیک روی متن عکس مثل پرده کرکره ظاهر می شود. و با کلیک دوباره روی متن عکس مثل پرده کرکره به سمت بالا می رود و در نهایت مخفی می شود. پارامترهای این متد هم مثل متد hide() است.

: slideDown()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      .mokaab {
        border: 2px solid #c5b5ec;
        background-color: #e2d9f9;
        width: 100px;
        height: 100px;
        text-align:center;
        display:none;
      }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $(".matn").click(function () {
          $(".mokaab").slideDown("slow");
        });
      });
    </script>
  </head>
  <body>
    <div class="matn">کلیک کنید</div>
    <div class="mokaab">راوندی</div>
  </body>
</html>
```

با کلیک روی متن مکعب از بالا به پایین نمایش داده می شود. و به خاطر slow() حرکت آرام تر است. پارامترهای این متد هم مثل متد hide() است.

: fadeIn()

```
<!DOCTYPE html>
<html>
  <head>
```

```

<meta charset="utf-8">
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
$('#clickme').click(function() {
    $("#aks").fadeIn("slow");
});
});
</script>
<style type="text/css">
#aks{
    display:none;
}
</style>
</head>
<body>
<div id="clickme">
کلیک کنید
</div>

</body>
</html>

```

عکس با دستور display:none مخفی است وقتی که روی متن کلیک می شود متد کلیک اجرا شده و متد fadeIn() که داخل متد کلیک هست اجرا می شود و به آهستگی با استفاده از ارتباط #aks با تگ عکس تصویر را نمایش می دهد.

پارامترهای این متد هم مثل متد hide() است.

slideUp و fadeIn هر دو با هم:

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
    $('#pic').slideUp().fadeIn();
});
</script>
</head>
<body>
<div id="pic"></div>
</body>
</html>

```

: fadeout()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $('#clickme').click(function() {
          $('#aks').fadeOut("slow");
        });
      });
    </script>
  </head>
  <body>
    <div id="clickme">
      کلیک کنید
    </div>
    
  </body>
</html>
```

با کلیک روی متن عکس محو می شود. پارامترهای این متد هم مثل متد hide() است.

:fadeTo()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("p:first").click(function () {
          $(this).fadeTo("slow", 0.33);
        });
      });
    </script>
  </head>
  <body>
    <p>متن کم کم کمرنگ می شود</p>
    <p>پاراگراف دوم</p>
  </body>
</html>
```

دستور `p:first` به پاراگراف اول اشاره دارد .
دستور `this` هم میگوید همین عنصری که در خط قبل انتخاب شده است. یعنی میتونستیم به جای `this` بگیم `"p:first"` . شماره ی `0.33` هم مقدار تاری هست.
از دستور زیر هم می توانید استفاده کنید:
`$(this).fadeTo(1500, 0.3);`
پارامترهای این متد هم مثل متد `hide()` است.

: Animate()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("#dokme").click(function(){
          $("#matn").animate({
            width: "80%",
            opacity: 0.5,
            marginLeft: "0.5px",
            fontSize: "4em",
          }, 1500 );
        });
      });
    </script>
  </head>
  <body>
    <button id="dokme">کلیک کنید</button>
    <div id="matn">سید محمدرضا حسینیان راوندی</div>
  </body>
</html>
```

`animate()` همانطوری که از اسمش پیداست برای متحرک سازی به کار میره.
`width: "80%"`, عرض به صورت درصد است
`opacity: 0.5`, مقدار شفافیت و تاری
`marginLeft: "0.5px"`, مقدار فاصله از سمت چپ را نشون میده
`fontSize: "4em"`, اندازه ی فونت
مقدار سرعت حرکت متن هست 1500

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      #mokaab{
        width:50px;
        height:50px;
        background-color:#e7e3f6;
      }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $("p").click(function(){
          $("#mokaab").animate({height:"120px"});
        });
      });
    </script>
  </head>
  <body>
    <p>متن</p>
    <div id="mokaab"></div>
  </body>
</html>

```

با کلیک روی متن ارتفاع مکعب 120 پیکسل میشه.
ساختار:

`(selector).animate({styles},speed,easing,callback)`

پارامتر:

Styles : این پارامتر ضروری است. تعریف یک یا بیشتر خصوصیت یا مقدار CSS برای متحرک سازی.

Speed : این پارامتر اختیاری است. مشخص کردن سرعت متحرک سازی.
مقادیری که ممکن است بگیرد:

Milliseconds

"slow"

"fast"

Easing : تعریف سرعت عنصر در نقاط مختلف متحرک سازی. این پارامتر اختیاری است.
مقدار پیشفرض "swing" است.

مقادیری که ممکن است بگیرد:
"swing" حرکت به آهستگی در ابتدا و انتها اما سریع تر در عمق.
"linear" در یک سرعت ثابت حرکت می کند.

Callback : این پارامتر اختیاری است. یک تابع که بعد از کامل شدن متحرک سازی اجرا می شود.

ساختار دیگر:

`(selector).animate({styles},{options})`

پارامترها:

Styles : این پارامتر ضروری است. تعریف یک یا بیشتر خصوصیت یا مقدار CSS برای متحرک سازی.

Options : این پارامتر اختیاری است. تعریف گزینه های اضافی برای متحرک سازی.
مقادیری که ممکن است بگیرد:

speed
easing
callback
step
queue
specialEasing

یک مثال دیگر از دادن استایل به متد `animate()` :

```
<!DOCTYPE html>
<html>
  <head>
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $('.box').animate({ "left" : "100px" }, 1000);
        $('.box').animate({ "left" : "20px" }, 500);
        $('.box').animate({ "left" : "50px" }, 500);
      });
    </script>
  </head>
  <body>
    <div class="box" style="height: 30px; width: 30px; background-color:
    #89BC38; position: absolute;"></div>
  </body>
</html>
```

با کلیک روی دکمه صفحه از سمت چپ باز شود و با کلیک دوباره بسته شود:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      .slide{
        display:none;
        border:1px navy solid;
        background-color:#dad7f8;
        height:250px;
        width:250px;
        text-align:center;
      }
    </style>
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $('document').ready(function () {
        $("button").click(function(){
          $(".slide").animate({ width: 'toggle' });
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <div class="slide">
      <h1>متن مورد نظر</h1>
    </div>
  </body>
</html>
```

روش دیگر: با کلیک روی دکمه ی اول از بالا به پایین ظاهر می شود و با کلیک روی دکمه ی دوم از پایین با بالا می رود و مخفی می شود.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      .slide{
        display:none;
        border:1px navy solid;
```

```

        background-color:#dad7f8;
        height:250px;
        width:250px;
        text-align:center;
    }
</style>
<script src="jquery.js" type="text/javascript">
</script>
<script type="text/javascript">
$('document').ready(function () {
    $('#btn1').click(function(){
        $(".slide").animate({height:'show'});
    });
    $('#btn2').click(function(){
        $(".slide").animate({height:'hide'});
    });
});
</script>
</head>
<body>
<button id="btn1">کلیک کنید</button><br/>
<button id="btn2">کلیک کنید</button>
<div class="slide">
    <h1>متن مورد نظر</h1>
</div>
</body>
</html>

```

Queue() : به معنای در صف ایستادن است. در کد زیر صبر میکنه تا صفحه کاملاً بارگذاری بشه . بعد پیام میده:
ساختار:

`$(selector).queue(queueName)`
queueName : یک رشته شامل نام صفت به صورت پیش فرض "fx" است.

```

<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function() {
                $('#pic').queue(function() {

```

```

        alert('صفحه کاملاً بارگذاری شد');
    });
});
</script>
</head>
<body>
    <div id="pic"></div>
</body>
</html>

```

رویداد:

Event (رویداد) چیست؟

رویداد به خاطر فعالیت کاربر یا قرار گرفتن صفحه ای در مرورگر به وقوع می پیوندد. مثلاً کاربر ماوس را کلیک میکند یا حرکت میدهد یا دکمه ای از صفحه کلید را فشار میدهد. در این کتاب با چند نمونه از رویدادها آشنا می شوید به عنوان مثال چند نمونه رویداد شامل موارد زیر است:

click(),keypress(),submit(),load(),dblclick(),keydown(),change(),resize(),mouseenter(),keyup(),focus(),scroll(),mouseleave(),blur(),unload()

: Mouseover()

ساختارها:

```

$(selector).mouseover()
$(selector).mouseover(function){
}

```

پارامتر:

Function: وقتی ماوس روی عنصر قرار میگیرد و رویداد اجرا می شود دستورات داخل آکولاد اجرا می شود.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <style type="text/css">
            #mokaab{
                width:50px;
                height:50px;
            }
        </style>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function () {
                $('p').mouseover(function(){

```

```

        $('#mokaab').css( "background", "#e7e3f6" );
    });
});
</script>
</head>
<body>
<div id="mokaab">
    <p>راوندی</p>
</div>
</body>
</html>

```

وقتی ماوس میره روی متن مکعب رنگی نمایش داده می شود.

: Mouseout()

ساختارها:

```
$(selector).mouseout()
```

```
$(selector).mouseout(function){
}
```

پارامتر:

Function: وقتی ماوس از روی عنصر برداشته می شود رویداد اجرا می شود و دستورات داخل آکولاد اجرا می شود.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <style type="text/css">
            #mokaab{
                width:50px;
                height:50px;
                background-color:#e7e3f6;
            }
        </style>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function () {
                $('p').mouseout(function(){
                    $('#mokaab').css( "background", "#FFFFFF" );
                });
            });
        </script>
    </head>
</body>

```

```
<div id="mokaab">
  <p>راوندی</p>
</div>
</body>
</html>
```

وقتی ماوس از روی متن برداشته میشه مکعب به رنگ سفید تغییر رنگ میده.

Mousedown() : با فشار ماوس، رویداد اجرا می شود و دستورات داخل آکولاد اجرا می شود.

```
$(selector).mousedown()
$(selector).mousedown(function){
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $('p').mousedown(function() {
          alert('با فشار کلیک ماوس عمل میکند');
        });
      });
    </script>
  </head>
  <body>
    <p>راوندی</p>
  </body>
</html>
```

Mousemove() : با حرکت ماوس از روی عنصر عمل میکند:

پارامتر:

:Function

```
$(selector).mousemove(function){
}
```

```

<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      #mokaab{
        width:50px;
        height:50px;
        background-color:#e7e3f6;
      }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $(document).mousemove(function(event){
          $("#mokaab").text(event.pageX + "," + event.pageY);
        });
      });
    </script>
  </head>
  <body>
    <div id="mokaab"></div>
  </body>
</html>

```

با حرکت ماوس مختصات ماوس در مکعب نمایش داده می شود.

: Hover()

ساختار:

`$(selector).hover(inFunction,outFunction)`

پارامترها:

inFunction : این پارامتر ضروری است. تعریف یک تابع ، این تابع زمانی اجرا می شود که رویداد مورد نظر با وارد شدن ماوس روی عنصر مشخص شده ، رخ دهد .

outFunction : این پارامتر اختیاری است. تعریف یک تابع ، این تابع زمانی اجرا می شود که رویداد مورد نظر با خارج شدن ماوس از روی عنصر مشخص شده ، رخ دهد .

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $(function () {
        $("p").hover(function () {
          $("p").css("color","blue");
        });
      });
    </script>
  </head>
  <body>
    <p><b>متن مورد نظر</b></p>
  </body>
</html>

```

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("p").hover(function(){
          $("p").css("background-color","yellow");
        },function(){
          $("p").css("background-color","Violet");
        });
      });
    </script>
  </head>
  <body>
    <p>راوندی</p>
  </body>
</html>

```

one() : رویداد فقط یک بار اتفاق می افتد:

ساختار:

`$(selector).one(event,data,function)`

پارامترها:

Event : این پارامتر ضروری است. که می توانید یک نوع رویداد را تعریف کنید مثلاً رویداد click ، dblclick و ...

Function : این پارامتر ضروری است. وقتی رویداد اتفاق می افتد تابع اجرا می شود.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("div").one('click', function () {
          $(this).css({
            border:"1px #CCCCCC solid",
            "background-color":"blue"
          });
        });
      });
    </script>
  </head>
  <body>
    <div>جدول اول</div>
    <div>جدول دوم</div>
    <div>جدول سوم</div>
  </body>
</html>
```

فقط با یک بار کلیک زمینه ی div به رنگ آبی می شود.

dblclick: وقتی رویداد دابل کلیک رخ می دهد که روی عنصر انتخاب شده دابل کلیک شود.

ساختار:

`$(selector).dblclick()`

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $("p").dblclick();
      });
    </script>
  </head>
  <body>
    <p ondblclick="alert('دابل کلیک')">dblclick</p>
  </body>
</html>

```

`$(selector).dblclick(function)`

در اینجا یک تابع را اضافه کردیم.
پارامتر:

function : این پارامتر اختیاری است. تابع وقتی اجرا می شود که رویداد دابل کلیک رخ دهد.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function () {
        $("p").dblclick( function () {
          alert("دابل کلیک شد");
        });
      });
    </script>
  </head>
  <body>
    <p>دابل کلیک کنید</p>
  </body>
</html>

```

باید حتما روی متن دابل کلیک شود تا پیغام ظاهر شود.

Submit() : این رویداد وقتی رخ می دهد که با ارسال فرم موافقت میشه.

ساختارهای متد :

`$(selector).submit()`
`$(selector).submit(function)`

پارامتر:

Function : این پارامتر اختیاری است. تابع وقتی اجرا می شود که رویداد submit اتفاق بیوفتد.

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("#form-name").submit(function(e){
          alert("Submitted");
        });
      });
    </script>
  </head>
  <body>
    <form id="form-name" method="get" action="get.php">
      <input type="text" name="name" />
      <input type="submit" value="ارسال" />
    </form>
  </body>
</html>
```

یک مثال دیگه:

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("form").submit(function(){
          alert("ارسال شد");
        });
        $("button").click(function(){
```

```

        $("form").submit();
    });
});
</script>
</head>
<body>
<form>
    name:<input type="text" name="FirstName">
</form>
    <button>send</button>
</body>
</html>

```

Keydown() : با فشار دکمه عمل می کند. یعنی وقتی دکمه پایین می آید و هنوز بالا نیامده متد عمل میکند.

```

<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).keydown(function(e){
                if (e.keyCode == 37) {
                    alert( " دکمه ی سمت چپ " );
                }
            });
        </script>
    </head>
<body>
</body>
</html>

```

برای اینکه دکمه ی مورد نظر شما وقتی کلیک شد برنامه پیغام بدهد باید شماره ی هر دکمه را به جای عدد 37 قرار بدهید.
مثلاً شماره ی 38 برای دکمه ی بالایی مکانما.
شماره ی 39 برای سمت راستی .
شماره ی 40 هم برای دکمه ی پایینی مکان نما و ...
ساختار:

\$(selector).keydown()

اضافه کردن یک تابع به رویداد **keydown** .

\$(selector).keydown(function)

پارامتر:


```

        </script>
    </head>
    <body>
        <input class="target" type="text">
        <button>کلیک کنید</button>
    </body>
</html>

```

Keyup() : بعد از فشار دکمه عمل نمیکند و باید حتماً دست خود را از روی دکمه بردارید یعنی دکمه به سمت بالا برود تا عمل کند. پارامترها و ساختار آن مثل پارامترها و ساختار متد `keydown()` است.

: Bind()

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function(){
                $("p").bind("click",function(){
                    alert("کلیک شد");
                });
            });
        </script>
    </head>
    <body>
        <p>کلیک کنید</p>
    </body>
</html>

```

رویداد کلیک را به عنصر `<p>` اضافه میکند.
با بارگذاری کامل صفحه پیغام ظاهر می شود:

```

<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function(){
                $(window).bind('load', function () {
                    alert("پنجره ی کاملاً بار گذاری شد");
                });
            });
        </script>
    </head>
    <body>
        <p>کلیک کنید</p>
    </body>
</html>

```

```

    });
    </script>
</head>
<body>
</body>
</html>

<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function () {
                $(document).bind("click", function(e){
                    var str = "( " + e.pageX + ", " + e.pageY + " )";
                    $("span").text(str);
                });
            });
        </script>
    </head>
    <body>
        <span></span>
    </body>
</html>

```

Document کل سند html را مشخص میکند. برای همین در کل صفحه اگر کلیک بکنید مختصات X و Y به شما نمایش داده خواهد شد.
ساختار:

`$(selector).bind(event,data,function,map)`

پارامترها :

Event : مشخص کردن یک یا چند رویداد. مثلا رویداد کلیک. این پارامتر ضروری است. بین هر رویداد یک فاصله قرار دهید.
مثالی برای چند رویداد:

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function () {
                $("p").bind("click mouseout", function() {
                    alert( "رویداد عمل کرد." );
                });
            });
        </script>
    </head>
    <body>
        <p>Hello World!</p>
    </body>
</html>

```

```

    });
    });
</script>
</head>
<body>
<p>متن</p>
</body>
</html>

```

وقتی روی پاراگراف کلیک شود پیام ظاهر می شود و وقتی ماوس از روی پاراگراف برداشته شود رویداد عمل میکند و پیام ظاهر می شود.

Data : این پارامتر اختیاری است. نمایش پیام همراه با اجرای یک تابع.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      function handlerName(e)
      {
        alert(e.data.msg);
      }
      $(document).ready(function(){
        $("p").bind("click", {msg: "پیام"},handlerName)
      });
    </script>
  </head>
  <body>
    <p>کلیک</p>
  </body>
</html>

```

Function : تعریف تابعی که بعد از رخ دادن رویداد اجرا می شود.

Map : شامل یک یا بیشتر رویداد اضافه شده به عنصر است. و تابع وقتی اجرا می شود که رویدادی اتفاق بیوفتد.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("img").bind({

```



```

        click:function(){
            $("p").hide();
        },
        mouseover:function(){
            $("span").css("color", "#374fff");
        }
    });
});
</script>
</head>
<body>
<p>این پاراگراف حذف می شود</p>
<span>این متن رنگی می شود</span>

</body>
</html>

```

Html در jquery :

معرفی متدهایی در رابطه با html و css

چند نمونه مثال :

text(),html(),attr(),val(),remove(),append(),prepend(),after(),before(),addClass(),removeClass(),toggleClass(),css(),width(),height(),...

remove() : به معنای برداشتن است. که ما در برنامه ی زیر هم همین کار را میکنیم با کلیک و با کمک همین دستور، پاراگراف از اینجا برداشته میشه.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function() {
                $('button').click(function() {
                    $('p').remove();
                });
            });
        </script>
    </head>
    <body>
        <button>کلیک کنید</button>

```

```
<p>متن حذف می شود</p>
</body>
</html>
```

ساختار:

```
$(selector).remove()
```

دستورات CSS :

رنگ زمینه:

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $(document.body).css("background", "green");
      });
    </script>
  </head>
  <body>
  </body>
</html>
```

document.body زمینه را در نظر می گیرد و تصویر زمینه سبز رنگ می شود. یعنی قسمت:

```
<body></body>
```

دادن استایل به div :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $('.matn').css("color", "red");
      });
    </script>
  </head>
  <body>
    <div class="matn">متن</div>
  </body>
</html>
```

متن به رنگ قرمز می شود.

دادن border :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $('.matn').css("border","3px solid red");
      });
    </script>
  </head>
  <body>
    <div class="matn">متن</div>
  </body>
</html>
```

روش های دیگه:

```
$('.matn').css("font-style", "italic");
$('.matn').css("background-color", "#bbbbff");
$('.matn').css("text-decoration", "line-through");
```

اگر CSS بلد باشید می توانید به روش بالا از کدهای CSS در jquery هم استفاده کنید.

دادن چند استایل:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $('.matn').css({backgroundColor: 'yellow',border:'3px solid red', fontWeight: 'bold'})
      });
    </script>
  </head>
  <body>
    <div class="matn">متن</div>
  </body>
</html>
```

هر خاصیتی از خاصیت دیگر با یک , از هم جدا شده و همه داخل آکولاد {} قرار گرفته است.

دادن یک استایل به چند عنصر:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("#matn,span,p,.matn").css("border","3px solid red");
      });
    </script>
  </head>
  <body>
    <p>راوندی</p>
    <div id="matn">متن کلاس</div>
    <div class="matn">متن آی دی</div>
    <span>متن آخر</span>
  </body>
</html>
```

بین هر کدام یک ویرگول میذاریم:

#matn,span,p,.matn

Attr() : خصوصیات و مقادیر عناصر انتخاب شده را بر می گرداند.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
      $(document).ready(function () {
        $("p").click(function(){
          $("img").attr({
            src: "/testi.gif"
          });
        });
      });
    </script>
  </head>
  <body>
    <p>متن</p>
    <img src="">
  </body>
</html>
```

با کلیک روی متن آدرس عکس به تگ عکس داده میشه. حالا اگه میخواهید که قبل از کلیک روی متن اون ضربدر قرمز رنگ در مرورگر اکسپلورر و دیگر مرورگرها نمایش داده نشه می توانید از دستور `display:none` استفاده کنید. و بعد از کلیک روی متن، `display:block` بشه. و تصویر نمایش داده بشه.

ساختارهای متد:

برگرداندن مقدار ویژگی:

`$(selector).attr(attribute)`

تنظیم ویژگی و مقدار:

`$(selector).attr(attribute,value)`

تنظیم ویژگی و مقدار استفاده شده ی یک تابع:

`$(selector).attr(attribute,function(index,currentvalue))`

تنظیم ویژگی ها و مقادیر مختلف:

`$(selector).attr({attribute:value, attribute:value,...})`

پارامترهای attr :

Attribute : خصوصیت ، ویژگی: در اینجا مشخصه ی نام یک صفت: مثلاً width

Value : مشخصه ی مقدار یک صفت: 100

Function(index,currentvalue) : تعیین یک تابع که مقدار ویژگی یک صفت را بر می گرداند.

Index : دریافت موقعیت شاخص عنصر در مجموعه.

Currentvalue : دریافت مقدار جاری عنصر انتخاب شده.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<script type="text/javascript" src="jquery.js">
```

```
</script>
```

```
<script type="text/javascript">
```

```
$(document).ready(function () {
```

```
$("#p").click(function(){
```

```
    $("#img").attr({width:"100"});
```

```
});
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p>متن</p>
```

```

```

```
</body>
```

```
</html>
```

یک مثال برای Function(index,currentvalue) :

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>attr</title>
    <meta charset="utf-8">
    <style type="text/css">
      .p1 {
        background-color: #FFEEEE;
      }

      .p2 {
        background-color: #EEFFEE;
      }

      .p3 {
        background-color: #EEEEFF;
      }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(function(){
        $("p").attr("class",function(index){
          return("p" + (index + 3));
        });
      });
    </script>
  </head>
  <body>
    <p>پاراگراف</p>
  </body>
</html>
```

با تغییر عدد 3 به اعداد یک تا سه . می توانید هر کلاسی که بخواهید به پاراگراف بدهید.
به جای class می توانید از id هم استفاده کنید و در قسمت استایل به جای نقطه از # استفاده کنید.

appendTo() : مقادیری رو اضافه میکنه.

ساختار:

`$(content).appendTo(selector)`

پارامتر:

Content : شامل اون چیزی هست که قرار هست در تگ مورد نظر در بخش selector وارد کنیم. که باید شامل تگ های html باشد.

Selector : عنصری را مشخص میکند که قرار است چیزی که توسط content تعریف شده به آن عنصر وارد شود.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          $('<p>راوندی</p>').appendTo('.matn');
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <div class="matn">
    </div>
  </body>
</html>
```

با کلیک روی متن با کمک appendTo کد <p>راوندی</p> به div که کلاس matn را دارد اضافه می شود.

: Text()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          var text = $('p').text();
          if (text == 'راوندی'){
            $('p').text('سلام راوندی');
          }
        });
      });
    </script>
  </head>
  <body>
    <p>راوندی</p>
  </body>
</html>
```

```

    });
    </script>
</head>
<body>
    <button>کلیک کنید</button>
    <p>راوندی</p>
</body>
</html>

```

متن داخل پاراگراف را داخل متغیر text قرار داده و در خط بعدی شرطی گذاشته شده که اگر مقدار داخل متغیر مساوی رشته ی راوندی باشد به جای پاراگراف قبلی نوشته شود سلام راوندی.

دو مساوی برای مقایسه به کار می رود.

ساختار و پارامتر:

برگرداندن محتوای متن:

`$(selector).text(content)`

Selector عنصر مورد نظر که قرار است مقدارش گرفته بشه یا مقداری بهش داده بشه را مشخص میکند.

Content : شامل مقداری است که قرار است به عنصر مورد نظر داده شود.

`$(selector).text(function(index,currentcontent))`

Index : موقعیت شاخص عنصر در مجموعه را بر می گرداند.

currentcontent : مقدار جاری عنصر انتخاب شده را بر می گرداند.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function(){
                $("button").click(function(){
                    $("p").text(function(n){
                        return n;
                    });
                });
            });
        </script>
    </head>
    <body>
        <button>کلیک کنید</button>
        <p>پاراگراف اول</p>

```



```
<p>پاراگراف دوم</p>
<p>پاراگراف سوم</p>
</body>
</html>
```

: Val()

```
<!DOCTYPE html>
<html>
  <head>
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $(function()
      {
        $('#btn').click(function()
        {
          var nami = $('#name').val();
          alert(nami);
        });
      });
    </script>
  </head>
  <body>
    <form dir="rtl" method="post">
      <input type="text" id="name"/><br>
      <input type="button" id="btn" value="ارسال"/>
    </form>
  </body>
</html>
```

مقداری رو در تکست وارد کنید و دکمه را کلیک کنید با کلیک روی دکمه متن نوشته شده نمایش داده میشود.

پارامترها:

\$(selector).val(value)

Value : مقداری که به عنصر انتخاب شده داده می شود.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $(function()
      {
        $('#btn').click(function()

```

```

        {
            $('#name').val("راوندی");
        });
    });
</script>
</head>
<body>
    <form dir="rtl" method="post">
        <input type="text" id="name"/><br>
        <input type="button" id="btn" value="ارسال"/>
    </form>
</body>
</html>

```

\$(selector).val(function(index,currentvalue))

function(index,currentvalue) : با استفاده از این تابع مقدار عنصر انتخاب شده برگردانده می شود.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script src="jquery.min.js" type="text/javascript">
        </script>
        <script type="text/javascript">
            $(function()
            {
                $('#btn').click(function()
                {
                    $('#name').val(function(n,c){
                        return c + "مقدار";
                    });
                });
            });
        </script>
    </head>
    <body>
        <form dir="rtl" method="post">
            <input type="text" id="name" value="مقدار قبلی"/><br/>
            <input type="button" id="btn" value="ارسال"/>
        </form>
    </body>
</html>

```

اعتبارسنجی فرم با جی کوئری :

می خواهیم برنامه ای بنویسیم در صورتی که متنی در تکست فرم نوشته نشده باشد اجازه ی ارسال فرم رو به ما ندهد.

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(function()
      {
        // با کلیک روی دکمه فعال میشه
        $('#btn').click(function()
        {
          // در حالت عادی مقداری در اسپن نیست
          $('#span').html('');
          // مقادیر داخل تکست رو داخل متغیر میذاره
          var nami = $('#name').val();
          // اگر مقدار ی داخل متغیر نیست
          if(nami == '')
          {
            // متن "تکست را پر کنید" را داخل اسپن قرار میده
            $('#span').html('تکست را پر کنید');
            // این دستور کاری میکنه که برنامه ادامه پیدا نکنه
            return false;
          }
          alert(nami);
        });
      });
    </script>
  </head>
  <body>
    <form dir="rtl" action="test.php" method="post">
      <input type="text" id="name">
      <span style="color:#ff0000"></span><br/>
      <input type="submit" id="btn" value="ارسال"/>
    </form>
  </body>
</html>
```

اگر متنی در تکست نباشه و بعد از اجرای برنامه ، شما قبلاً بعد از دستور `return false`; یک `alert("ravandi");` گذاشته باشی ، هیچی به شما نمایش نمیده. چون برنامه تا این کد متوقف میشه.

اگر در برنامه ی بالا شما قبل از `alert(nami);` یک دستور `return true`; قرار بدید بازم برنامه به کار خودش ادامه میده. و مقدار متغیر را نمایش میده. چون برنامه در اینجا متوقف نشده.

حالا اگر شما در تکست یک دکمه ی space را کلیک کنید و دکمه ارسال را بزنید پیغام فیلد را پر کنید نمایش داده نمیشه. پس باید با کمک `trim()` مقدار خالی را هم بررسی کنیم و با استفاده از `length` طول کاراکتر را بررسی کنیم. پس به جای خط `if(nami == '')` از خط زیر استفاده کنید:

```
if ( $("#name").val().trim().length == 0 )
```

دستور بالا دو چیز را بررسی میکند اینکه اگر تکست خالی باشد و طول کاراکترها مساوی صفر باشد .

نکته: اگر شما با برنامه های سمت کلاینت مثل جی کوئری و جاوا اسکریپت اعتبارسنجی کنید فرد میتونه این اعتبارسنجی رو دور بزنه . پس شما باید در سمت سرور اگر از asp یا php استفاده میکنید هم یک اعتبارسنجی انجام بدید.

یک مثال ساده برای radio button:

```
<!DOCTYPE html>
<html>
<html>
<head>
    <script src="jquery.js"></script>
    <script type="text/javascript">
        $(document).ready(function(){
            $('#btn').click(function() {
                var radioo=$("#input:radio[name='ray']:checked").val();
                alert(radioo);
            });
        })
    </script>
</head>
<body>
    <form dir="rtl" method="post">
        <br/>نظر شما در مورد این کتاب چیست؟
        <input type="radio" name="ray" value="1"/>عالی
        <input type="radio" name="ray" value="2"/>خوب
        <input type="radio" name="ray" value="3"/>متوسط
```

`<input type="radio" name="ray" value="4"/>` ضعیف

`<input type="button" value="انتخاب" id="btn"/>`

`</form>`

`</html>`

خط آبی رنگ کارش اینه که `input` ی را انتخاب کنه که از نوع `radio` باشه و مقدار `name` آن `ray` باشه که انتخاب شده باشه (`checked`). و مقدار اون `radio button` را داخل متغیر قرار میده و با `alert` نمایش میده.

یک مثال برای انتخاب چک باکس:

`<!DOCTYPE html>`

`<html>`

`<head>`

`<script type="text/javascript" src="jquery.js">`

`</script>`

`<script type="text/javascript">`

`function submit_form(){`

`var checkboxi =`

`$("input:checkbox[name='tedad']:checked").val();`

`alert(checkboxi);`

`}`

`</script>`

`</head>`

`<body>`

`<form dir="rtl">`

`<input type='checkbox' name='tedad' value='63'>`

`<input type="button" value="ارسال" onclick="submit_form();">`

`</form>`

`</body>`

`</html>`

برعکس مثال های قبلی به جای `ready()` و `click()` از `submit_form()` استفاده کردیم برای همین باید در قسمت دکمه از `onclick="submit_form();"` استفاده کنیم.

اعتبارسنجی radio button :

`<!DOCTYPE html>`

`<html>`

`<head>`

`<meta charset="utf-8">`

`<script type="text/javascript" src="jquery.js"></script>`

`<script type="text/javascript">`

`$(document).ready(function(){`

`$("#btn").click(function(){`

`// اعتبارسنجی رادیو`

```

if ($("#radioo:checked").length == 0 &&
    $("#radioi:checked").length == 0){
    $('.ekhtar').html('جنسیت را انتخاب کنید');
    return false;
}else{
    $('.ekhtar').html('');
}
});
});
</script>
</head>
<body>
<div style="text-align:right;direction:rtl;">
<form action="get.php" dir="rtl" method="post">
مرد<input type="radio" name="radio" id="radioo" value="مرد"/>
زن<input type="radio" name="radio" id="radioi" value="زن"/>
<input type="button" id="btn" value="ارسال"/>
</form>
<span class="ekhtar"></span>
</div>
</body>
</html>

```

برنامه ی بالا میگه اگر طول مقداری که به radio داده شده یعنی مقداری که داخل value هست مساوی صفر باشد.(دو مساوی برای مقایسه بکار میره) پیغام را نمایش بده. و در غیر این صورت هیچی نمایش نده.

با دستور return false کاری کردیم که برنامه در صورت دادن اخطار ادامه پیدا نکنه.

empty() : به معنای خالی. مقدار داخل عنصر مشخص شده را حذف می کند.

ساختار:

\$(selector).empty()

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script src="jquery.js" type="text/javascript">
</script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function () {
    $("p").empty();

```

```

    });
    });
</script>
</head>
<body>
    <button>کلیک کنید</button>
    <p>متن</p>
</body>
</html>

```

با کلیک روی دکمه ، <p>متن</p> از متن خالی می شود.

Get() : گرفتن نام عنصر.

ساختار:

`$(selector).get(index)`

دارای پارامتر `index` می باشد.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $('document').ready(function(){
                $("*").click(function (e) {
                    e.stopPropagation();
                    var onsor = $(this).get(0);
                    $("span:first").text("کلیک شد" + onsor.tagName + " روی");
                });
            });
        </script>
    </head>
    <body>
        <span>&nbsp;</span>
        <p>پاراگراف</p>
        <span>اسپن</span><br/>
        <input type="text"><br/>
        
        <div style="background:#D6E6FF;width:50px;height:50px;">
        <br/>
        </div>
    </body>
</html>

```

با کلیک روی هر یک از عناصر صفحه ، نام آن را نمایش میدهد.
Get() نام عنصر را میگیرد و به text() داده و آن هم نمایش میدهد.
span:first منظورش اولین span هست.

: toggleClass()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      .rang { background:#dbbfff; }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("p").click(function () {
          $(this).toggleClass("rang");
        });
      });
    </script>
  </head>
  <body>
    <p>متن اول</p>
    <p>متن دوم</p>
    <p>متن سوم</p>
  </body>
</html>
```

یک جورایی مثل toggle() عمل میکنه. ولی اینجا کارش با class هست. با کلیک روی هر پاراگراف رنگ زمینه ی هر پاراگراف را با classی که بهش میده تغییر میده . و با کلیک دوباره روی پاراگراف رنگ زمینه به حالت قبل بر میگرده.

\$(selector).toggleClass(classname,function(index,currentclass),switch)
Classname : این پارامتر ضروری است و برای تعریف یک یا چند کلاس بکار میرود . برای قرار دادن نام چند کلاس باید بین هر نام کلاس یک فاصله با دکمه ی space صفحه کلید ایجاد کنید.

function(index,currentclass) : این پارامتر اختیاری است. تعریف یک تابع که نام یک یا بیشتر کلاس را بر می گرداند.

Switch : اختیاری است. مقدار بولین (Boolean) یعنی صفر و یک درست و نادرست. داده می شود. اگر مقدار true باشد کلاس اضافه می شود و اگر false باشد کلاس حذف می شود.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("#add").click(function(){
          $("p").toggleClass("main",true);
        });
        $("#remove").click(function(){
          $("p").toggleClass("main",false);
        });
      });
    </script>
    <style type="text/css">
      .main
      {
        font-size:120%;
        color:red;
      }
    </style>
  </head>
  <body>
    <p class="main">متن مورد نظر</p>
    <button id="add">اضافه</button>
    <button id="remove">حذف</button>
  </body>
</html>
```

addClass() : خصوصیت های کلاس را به پاراگراف اضافه میکند.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      .rang { background:#dbbfff; }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
```

```

<script type="text/javascript">
$(document).ready(function(){
$("p").click(function () {
    $(this).addClass("rang");
});
});
</script>
</head>
<body>
<p>متن اول</p>
<p>متن دوم</p>
<p>متن سوم</p>
</body>
</html>

```

مثل برنامه ی قبل شد با این تفاوت که با کلیک دوباره روی پاراگراف رنگ زمینه اش تغییر نمیکند. الان با کلیک رنگ زمینه ی پاراگراف بنفش میشه چون کلاس در بخش استایل اینطوری مشخص کرده. اگه به جای `this` از خود `p` استفاده کنیم با کلیک روی هر کدام از پاراگراف ها (`<p></p>`) زمینه ی همه ی پاراگراف ها به رنگ بنفش میشه ولی اگر `this` باشه منظورش اینه که : این پاراگراف ...

```
$(selector).addClass(classname,function(index,oldclass))
```

پارامترها:

`Classname` یک یا بیشتر نام کلاس را می تواند بگیرد. و ضروری است.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style type="text/css">
.rang{
    color:blue;
}
.zakhim{
    font-weight:bold;
}
</style>
<script type="text/javascript" src="jquery.min.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
    $("#matn").addClass("rang zakhim");
});
</script>
</head>

```

```

<body>
  <span id="matn2">محمدرضا</span>
  <span id="matn">راوندی</span>
</body>
</html>

```

function(index,currentclass) : این پارامتر اختیاری است. تعریف یک تابع که یک یا چند نام کلاس را برای اضافه شدن، بر می گرداند.

Index : موقعیت شاخص عنصر را بر می گرداند.

Currentclass : نام کلاس جاری عنصر انتخاب شده را بر می گرداند.

append() : افزودن.

ساختار:

```
$(selector).append(content,function(index,html))
```

پارامترها:

Content : این پارامتر ضروری است. این پارامتر می تواند شامل تگ های html باشد.

مقادیری که می تواند بگیرد شامل: عناصر html ، اشیا jquery ، عناصر Dom .

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("p").append("<b>راوندی</b>");
      });
    </script>
  </head>
  <body>
    <p>سید محمدرضا حسینیان</p>
  </body>
</html>

```

کلمه ی راوندی را به صورت ضخیم به پاراگراف اضافه میکند.

مثالی دیگر:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $(".badi").appendTo(".ghabli");
      });
    </script>
  </head>
  <body>
    <span class="badi">راوندی</span>
    <span class="ghabli">سید محمدرضا حسینیان</span>
  </body>
</html>
```

پارامتر دیگر:

function(index,html) : این پارامتر اختیاری است.

یک تابع است شامل مقادیر زیر:

Index : موقعیت شاخص عنصر انتخاب شده را بر می گرداند.

Html : html جاری از عنصر انتخاب شده را بر می گرداند.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        var i = $('li').size() + 1;
        $("button").click(function(){
          $("<li>" + i + "<input type='text'/>
          </li><br/>").appendTo('ol');
          i++;
        });
      });
    </script>
  </head>
```

```

<body>
  <ol>
    <li>1<input type='text' /></li><br/>
  </ol>
</button>اضافه</button>
</body>
</html>

```

Prepend() : مقدار مشخص شده داخل متد را قبل از عنصر انتخاب شده قرار می دهد. درست برعکس متد append() است.

ساختار:

```
$(selector).prepend(content,function(index,html))
```

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("p").prepend("<b>محمدرضا </b>");
      });
    </script>
  </head>
  <body>
    <p>حسینیان</p>
  </body>
</html>

```

محمدرضا را به قبل حسینیان اضافه میکند.

ساختار:

```
$(selector).prepend(content,function(index,html))
```

پارامترها:

Content : این پارامتر ضروری است. تعریف مقداری برای وارد کردن که می تواند شامل مقادیر: عناصر html ، اشیاء جی کوئری ، عناصر DOM .

Function(index,html) : این پارامتر اختیاری است. شامل index و html است.

Index : موقعیت شاخص عنصر تنظیم شده را بر می گرداند .

Html : html جاری از عنصر انتخاب شده را بر می گرداند.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          $("p").prepend(function(n){
            return "شماره " + n;
          });
        });
      });
    </script>
  </head>
  <body>
    <p>متن اول</p>
    <p>متن دوم</p>
    <button>کلیک کن</button>
  </body>
</html>

```

prependTo() : این متد عناصر html را در شروع عناصر انتخاب شده وارد می کند. و بر عکس متد appendTo() می باشد.
ساختار:

`$(content).prependTo(selector)`

پارامترها:

Content : این پارامتر ضروری است. این پارامتر بیشتر برای تعریف مقداری شامل تگ های html است.
مثلاً:

`$("#<p>متن</p>").prependTo(".ghabli");`

Selector : این پارامتر ضروری است. تعریف مقدار قبلی.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $(".badi").prependTo(".ghabli");
      });
    </script>
  </head>
  <body>
    <span class="ghabli">راوندی</span>
    <span class="badi">سید محمدرضا حسینیان</span>
  </body>
</html>

```

Out: سید محمدرضا حسینیان راوندی

Wrap() : تعریف عناصر html دور عناصر انتخاب شده.

ساختار :

`$(selector).wrap(wrappingElement,function(index))`

پارامترها:

wrappingElement : این پارامتر ضروری است. تعریف عناصر html برای دور هر عنصر انتخاب شده.

می تواند شامل مقادیر زیر باشد:

عناصر html

اشیاء jquery

عناصر DOM

function(index) : این پارامتر اختیاری است. تعریف تابعی که عناصر بسته بندی شده(عناصری که دور آن ها تگ ها قرار گرفته است) را بر می گرداند. Index موقعیت شاخص عناصر انتخاب شده را بر می گرداند.

```

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

```

```

<script type="text/javascript">
$(document).ready(function() {
    $("span").wrap("<i><b></b></i>");
});
</script>
</head>
<body>
<span>متن مورد نظر اول</span>
<p>متن مورد نظر دوم</p>
</body>
</html>

```

متن داخل `<span>` را ضخیم (bold) و کج (italic) می کند.
 متد `unwrap()` هم بر عکس این متد عمل می کند به مثال زیر دقت کنید:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
    $("#btn1").click(function() {
        $("span").wrap("<i><b></b></i>");
    });

    $("#btn2").click(function() {
        $("span").unwrap();
    });
});
</script>
</head>
<body>
<button id="btn1">دکمه ی اول</button><br/><br/>
<button id="btn2">دکمه ی دوم</button><br/><br/>
<span>متن مورد نظر</span><br/>
</body>
</html>

```

replaceWith : عناصر انتخاب شده را با مقدار جدید جایگزین میکند.

ساختار:

`$(selector).replaceWith(content,function(index))`

پارامترها:

Content : این پارامتر ضروری است. تعریف مقداری برای جایگزین کردن، که می تواند شامل تگ های html هم باشد.

مقادیری که می تواند استفاده شود:

عناصر html

اشیاء jquery

عناصر DOM

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("p").replaceWith("<b>ضخیم</b>");
      });
    </script>
  </head>
  <body>
    <p>پاراگراف اول</p>
    <p>پاراگراف دوم</p>
  </body>
</html>
```

function(index) : این پارامتر اختیاری است. تعریف تابعی برای برگرداندن مقادیر برای جایگزین کردن.

Index : موقعیت شاخص عناصر انتخاب شده را بر می گرداند.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          $("p").replaceWith(function(n){
            return "پاراگراف شماره ی"<h3> + n + "</h3>";
          });
        });
      });
    </script>
  </head>
  <body>
    <button>بزرگ کردن</button>
  </body>
</html>
```

```

    });
    });
    });
    </script>
</head>
<body>
    <p>پاراگراف اول</p>
    <p>پاراگراف دوم</p>
    <p>پاراگراف سوم</p>
    <button>کلیک کنید</button>
</body>
</html>

```

replaceAll() : این متد عناصر انتخاب شده را با عناصر html جدید جایگزین می کند.

ساختار:

```
$(content).replaceAll(selector)
```

پارامترها:

Content : این پارامتر لازم است. تعریف مقداری برای جایگزین کردن، که می تواند شامل تگ های html هم باشد.

Selector : تعریف عناصری برای جایگزین شدن.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function() {
                $("<b>ضخیم</b>").replaceAll("p");
            });
        </script>
    </head>
    <body>
        <p>پاراگراف یک</p>
        <p>پاراگراف دو</p>
        <p>پاراگراف سه</p>
    </body>
</html>

```

insertAfter : به معنای قرار دادن بعد از ، متدی است برای وارد کردن عناصر html

بعد از عناصر انتخاب شده. متد insertBefore() هم برعکس این متد عمل می کند.

ساختار:

`$(content).insertAfter(selector)`

پارامتر:

Content : این پارامتر ضروری است. تعریف مقداری شامل تگ های html برای وارد کردن.

Selector : این پارامتر ضروری است. تعریف عنصری که مقادیر content بعد از آن قرار می گیرد.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          $("<h2>راوندی</h2>").insertAfter("hr");
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <p>محمدرضا</p>
    <hr>
  </body>
</html>
```

با هر بار کلیک روی دکمه ، متد click فعال شده و insertAfter متن داخل h2 را بعد از <hr> وارد میکند.

after() : به معنای پس از ، متدی است برای تعریف مقداری پس از عناصر انتخاب شده. بر عکس متد before() عمل می کند. ساختار:

`$(selector).after(content,function(index))`

پارامتر:

Content : این پارامتر ضروری است. تعریف مقداری برای وارد کردن که می تواند شامل مقادیر زیر باشد:

عناصر html .

اشیاء جی کوئری.

عناصر DOM .

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function () {
          $("p").after("ravandi");
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <p>متن</p>
  </body>
</html>

```

با کلیک روی دکمه متن ravandi پس از <p> اضافه می شود.

function(index) : تعریف تابعی که مقداری را برای وارد کردن بر می گرداند.
 Index موقعیت شاخص عناصر انتخاب شده را بر می گرداند.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.min.js">
    </script>
    <script>
      $(document).ready(function(){
        $("button").click(function(){
          $("p").after(function(n){
            return "<div>شماره ی " + n + "</div>";
          });
        });
      });
    </script>
  </head>
  <body>
    <p>پاراگراف اول</p>

```

```

</p>پاراگراف دوم</p>
</p>پاراگراف سوم</p>
</button>کلیک کنید</button>
</body>
</html>

```

prev() : این متد عنصر هم خانواده ی قبلی خود را از عنصر انتخاب شده بر می گرداند.
ساختار:

```
$(selector).prev(filter)
```

پارامتر:

Filter : این پارامتر اختیاری است. تعریف یک انتخاب کننده برای نمایش عناصر هم خانواده ی قبلی. که برنامه برای ما پیدا میکند.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("li").click(function () {
          $(this).prev().css('background-color', 'yellow');
        });
      });
    </script>
  </head>
  <body>
    <ul>
      <li>اولین متن</li>
      <li>دومین متن</li>
      <li>سومین متن</li>
      <li>چهارمین متن</li>
      <li>پنجمین متن</li>
    </ul>
  </body>
</html>

```

با کلیک روی هر متن در li متن قبلی آن به صورت زرد رنگ نمایش داده می شود.

next() : بعدی

اگر **next()** را به جای **prev()** در کد بالا قرار دهیم با کلیک روی هر متن ، متن بعدی زرد رنگ خواهد شد.

بدست آوردن عرض و ارتفاع تصویر در jquery :

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("img").load(function() {
          alert($(this).height());
          alert($(this).width());
        });
      });
    </script>
  </head>
  <body>
    
  </body>
</html>
```

عرض و ارتفاع تصویر را در پنجره نمایش می دهد.

متد height() :

ساختار:

`$(selector).height()`

مقدار ارتفاع را نمایش میدهد.

`$(selector).height(value)`

انتخاب ارتفاع ، مثلا به جای پارامتر value مقدار ارتفاع را وارد می کنید.

`$(selector).height(function(index,currentheight))`

پارامتری که می گیرد:

`function(index,current height)` : این پارامتر اختیاری است. تعریف تابعی که ارتفاع

جدید را از عناصر انتخاب شده بر می گرداند.

Index : موقعیت شاخص عنصر را بر می گرداند.

Current height : ارتفاع جاری عنصر انتخاب شده را بر می گرداند.

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
```

```

        $("div").height(function(n,c){
            return c+200;
        });
    });
</script>
</head>
<body>
<button>کلیک</button>
<br/><br/>
<div style="height:100px;border:4px solid;"></div>
</body>
</html>

```

حالا به جای این خط :

```
return c+200;
```

خط زیر را بنویسید:

```
alert(c);
```

با کلیک روی دکمه ارتفاع جاری یا همون current height را به شما نمایش میدهد.

متد های Traversing :

parent(),parents(),parentsUntil(),children(),find(),siblings(),
next(),prev(),prevAll(),prevUntil(),first(),last(),eq(),nextAll(),
nextUntil(),...

:Eq()

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function(){
                $("button").click(function(){
                    $("p").eq(0).css("color", "red");
                });
            });
        </script>
    </head>
    <body>
        <button>کلیک کنید</button>
        <p>متن اول</p>
        <p>متن دوم</p>
    </body>
</html>

```

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

Eq(0) اولین متن را نمایش میدهد. و اگر یک باشد متن دوم را نمایش میدهد و...

Find() : پیدا کردن.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $(".matn").click(function () {
          $(".matn").find("span").css('color','red');
        });
      });
    </script>
  </head>
  <body>
    <div class="matn">
      <span>متن اول</span>
      <p>متن دوم</p><br/>
      <span>متن دوم</span>
      <p>متن سوم</p>
    </div>
  </body>
</html>
```

میگرده توی کلاس matn. و تگ های span رو پیدا میکنه و با css رنگش رو قرمز میکنه.
ساختار:

`$(selector).find(filter)`

پارامتر:

Filter : این پارامتر ضروری است.

می تواند یک عبارت انتخاب کننده یا عنصر یا شی جی کوئری باشد برای جستجوی فرزندان.

برای توضیحات بیشتر : مثلاً ما یک div اصلی داریم که داخل این div چند تا div دیگه یا عنصر دیگه وجود دارند. برای اینکه اونها رو مشخص کنیم. مثلاً رنگی بهشون بدیم از متد

find برای پیدا کردن اون ها و از پارامتر filter برای مثلاً مشخص کردن اونها استفاده می کنیم.

Parent() : والد. (پدر) عنصر پدر عنصر انتخاب شده را بر می گرداند.

ساختار:

`$(selector).parent(filter)`

پارامتر:

Filter : این پارامتر اختیاری است. یک انتخاب کننده برای جستجوی پدر.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      .pedar{
        height:80px;
        width:80px;
        border:1px #e20337 solid;
        background-color:#83d0f6;
      }
      .farzand{
        height:40px;
        width:40px;
        border:1px #e20337 solid;
        background-color:#83d0f6;
      }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("button").click(function () {
          $('farzand').parent().css("background", "yellow");
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <div class="pedar">
    <div class="farzand">راوندی</div>
  </div>
</body>
</html>
```

با کلیک روی دکمه ، کلاس پدر farzand یعنی کلاس pedar به رنگ زرد در می آید.
یک مثال برای پارامتر Filter :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
$("p").parent(".matn").css("backgroundcolor","yellow");
});
</script>
</head>
<body>
<div class="matn">
    <p>پاراگراف اول</p>
    <p>پاراگراف دوم</p>
</div>
<div class="matn2">
    <p>پاراگراف اول</p>
    <p>پاراگراف دوم</p>
</div>
</body>
</html>
```

Out:

پاراگراف اول

پاراگراف دوم

پاراگراف اول

پاراگراف دوم

Children() : فرزندان. این متد همه ی فرزندان عنصر انتخاب شده را بر می گرداند.

ساختار:

`$(selector).children(filter)`

Filter : این پارامتر اختیاری است. یک انتخاب کننده برای جستجوی فرزندان.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $(".pedar").children("ul").css("color","red");
      });
    </script>
  </head>
  <body>
    <p>متن</p>
    <div class="pedar">
      <ul>
        <li>متن یک</li>
        <li>متن دو</li>
      </ul>
      <ol>
        <li>متن سه</li>
      </ol>
    </div>
    <ul>
      <li>متن بیرون</li>
    </ul>
  </body>
</html>
```

Out:

متن

- متن یک
- متن دو

1. متن سه

- متن بیرون

فرزندان کلاس pedar که از نوع ul هستند به رنگ قرمز مشخص شده است.
یک مثال برای پارامتر filter :

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" src="jquery.min.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
$(".matn").children("p").css("background-color","yellow");
});
</script>
</head>
<body>
<div class="matn">
    <p>پارامتر یک</p>
    <p>پارامتر دو</p>
</div>
    <p>پارامتر سه</p>
</body>
</html>
```

پارامتر یک

پارامتر دو

پارامتر سه

مشخص کردن فرزندان:

```
<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $('document').ready(function(){
                $(".button").click(function(){
                    $(".asli > li > ul").css("background-color","yellow");
                });
            });
        </script>
```

```

</head>
<body>
  <button>کلیک</button>
  <ul class="asli">
    <li>متن اول

    <ul>
      <li>متن اول داخلی</li>
      <li>متن دوم داخلی</li>
    </ul>

  </li>
  <li>متن دوم</li>

</ul>
</body>
</html>

```

دستور. `ul > li > asli` میاد کل `ul` زیر مجموعه ی `li` از کلاس `asli` را ، به رنگ زرد در میاره.

- متن اول
 - متن اول داخلی
 - متن دوم داخلی
- متن دوم

متدهای متفرقه:

`data(),each(),get(),index(),size(),toArray(),...`

: Each()

چند مثال برای دسترسی به مقادیر:
 1) دسترسی به مقادیر آرایه:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){

```

```

        $("button").click(function(){
        $.each( ['a','b','c'], function(i, l){
            alert( "Index #" + i + ": " + l );
        });
        });
        });
    </script>
</head>
<body>
    <button>کلیک کنید</button>
    <div class="matn">
    </div>
</body>
</html>

```

آرایه ['a','b','c'] مقادیر آرایه هست . آرایه از صفر شروع می شود. برای همین اولین شماره ی آرایه را صفر نمایش داد.

پارامتر:

function(index,element)

(2) دسترسی به مقادیر شی: این روش بر عکس روش بالا دسترسی به مقادیر شی هست.

```

<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
        $(document).ready(function(){
        $("button").click(function(){
        $.each({
            name: "mohammad",
            lastname: "ravandi"
        }, function(k, v){
            alert( "Key: " + k + ", Value: " + v );
        });
        });
        });
        </script>
    </head>
    <body>
        <button>کلیک کنید</button>
        <div class="matn">
        </div>
    </body>
</html>

```

ما با استفاده از key و value به شماره ها و مقادیر دسترسی پیدا کردیم.

(2) یک مثال که کاربرد های زیادی هم داره:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          $("li").each(function(){
            alert($(this).text());
          });
        });
      });
    </script>
  </head>
  <body>
    <button>کلیک کنید</button>
    <ul>
      <li>متن اول</li>
      <li>متن دوم</li>
    </ul>
  </body>
</html>
```

: index()

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $("div").click(function () {
          var mo = $("div").index(this);
          $("p").text(mo+": شماره");
        });
      });
    </script>
  </head>
  <body>
```

```

<p></p>
<div>متن اول</div>
<div>متن دوم</div>
<div>متن سوم</div>
<div>متن چهارم</div>
</body>
</html>

```

با کلیک روی هر div و فعال شدن متد کلیک ، مقادیر داخل متد هم فعال شده و شماره ی ایندکس(شاخص) هر عنصر div به متغیر mo داده می شود. و مقدار متغیر mo هم به متد text() داده میشه و با کمک text() مقدار در p (پاراگراف) یعنی در بین دو تگ <p></p> نمایش داده می شود.
ساختارها :

```

$(selector).index()
$(selector).index(element)

```

پارامتر:

Element : این پارامتر اختیاری است. تعریف یک عنصر که با متد index() موقعیت آن گرفته می شود.

jQuery.trim() : فضاهای خالی سمت چپ و راست رشته را حذف میکند.

ساختار:

```
jQuery.trim( str )
```

پارامتر:

Str : یک رشته.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="jquery.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function () {
          var str = "    متن با فاصله ی چپ رو راست    ";
          alert("'' + str + ''");
          str = jQuery.trim(str);
          alert("متن دوم" + str + "متن اول");
        });
      });
    </script>
  </head>
</html>

```



```

        </script>
    </head>
<body>
    <button>کلیک کنید</button>
</body>
</html>

```

اعتبارسنجی با \$.trim() :

```

<!DOCTYPE html>
<html>
    <head>
        <script src="jquery.js" type="text/javascript">
        </script>
        <script type="text/javascript">
            $(document).ready(function(){
                $("#btn").click(function(){
                    $('.ekhtar').html('');
                    if($.trim($('#msg').val()) == '') {
                        $('.ekhtar').html('متن را وارد نکردید. ');
                        return false;
                    }
                });
            });
        </script>
    </head>
<body>
    <div style="text-align:right;direction:rtl;">
    <form dir="rtl" action="test.php" method="POST">
    <input type="text" name="matn" id="msg">
    <input type="button" id="btn" value="ارسال" />
    </form>
    <div class="ekhtar"></div>
    </div>
</body>
</html>

```

متدهای setTimeout() و setInterval() از متدهای خود جاوا اسکریپت است ولی چون در این کتاب باهاش کار داریم من به آموزش این دو متد پرداختم:

setInterval() : به صورت ساده میشه گفت این متد یک فاصله ی زمانی ایجاد می کنه. با استفاده از زمانی که به این متد دادیم این متد متن راوندی را هر 5 هزار میلی ثانیه یک بار نمایش می دهد.

```
<script type="text/javascript" src="jquery.js"></script>
<script type='text/javascript'>
$(document).ready(function(){
setInterval(function() {
document.write("راوندی");
}, 5000);
});
</script>
```

ساختار:

setInterval(code,millisec,lang)

پارامترها:

Code : این پارامتر ضروری است. تابعی که اجرا خواهد شد.

Millisec : این پارامتر ضروری است. تعیین مدت فاصله ی زمانی به میلی ثانیه.

Lang : اختیاری است: JavaScript | VBScript | JScript

```
<script type="text/javascript" src="jquery.js"></script>
<script type='text/javascript'>
function autoupdate() {
    document.write("راوندی");
}
// هر یک ثانیه
$(document).ready(function() {
    setInterval("autoupdate()", 1000);
});
</script>
</script>
```

هر یک ثانیه تابع autoupdate() اجرا شده ، پس دستورات داخل تابع هم اجرا می شود.

setTimeout() : در برنامه ی زیر بعد از 5000 میلی ثانیه پیغامی نمایش می دهد. ولی برنامه ادامه پیدا نمیکند.

```
<script type="text/javascript" src="jquery.js"></script>
<script type='text/javascript'>
$(document).ready(function(){
setTimeout(function() {
    document.write("راوندی");
}, 5000);
});
</script>
```

ساختار برنامه :

setTimeout(code,millisec,lang)

پارامترها:

Code : این پارامتر ضروری است. تابعی که اجرا خواهد شد.

Millisec : این پارامتر ضروری است. تعیین مدت فاصله ی زمانی به میلی ثانیه.

Lang : اختیاری است: JavaScript | VBScript | JScript

چند تا برنامه ی کاربردی:

ساخت modal :

می خواهیم برنامه ای بنویسیم که با کلیک روی لینک ، یک پنجره ی کوچک را وسط صفحه باز کند به صورتی که این پنجره ی از بالا به پایین بیاید و با کلیک روی دکمه ی ضربدر به بالا رفته و محو شود.

```
<!DOCTYPE html>
<html>
  <head>
    <title>modal</title>
    <meta charset="utf-8">
    <script src="jquery.min.js" type="text/javascript">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        // کلیک روی لینک برای باز شدن پنجره
        $("#link").click(function ()
        {
          //pachenar.ir
          // سید محمدرضا حسینیان راوندی
          // به صورت آهسته ظاهر می شود
          $(".overlay").show().animate();
          $(".overlay").fadeIn(500);
          $("#divTestBox1").animate(
            {
              "top":"50px"
            },500);
        });
        // کلیک روی دکمه ی بستن
        $("#linki").click(function () {
          $("#divTestBox1").animate(
            {
```

```

        "top" : "-50px"
    },500);
    $(".overlay").fadeOut(500);
});
// کلیک روی صفحه ی سیاه برای بستن
$(".overlay").click(function ()
{
    $("#divTestBox1").animate(
    {
        "top":"-50px"
    },500);
    $(".overlay").fadeOut(500);
});
});
</script>
<style type="text/css">
.overlay {
    display:none;
    position: absolute;
    left: 0px;
    top: 0px;
    width:100%;
    height:100%;
    background-color:#aeaeae;
    margin-left:auto;
    margin-right:auto;
}
.close {
    float: right;
    font-size: 20px;
    font-weight: bold;
    line-height: 20px;
    color:#000000;
    text-shadow: 0 1px 0 #ffffff;
    opacity: 0.2;
    filter: alpha(opacity=20);
}
.close:hover,.close:focus {
    color:#000000;
    text-decoration:none;
    cursor: pointer;
    opacity: 0.4;
    filter:alpha(opacity=40);
}
button.close {
    padding:0;

```

```

        cursor: pointer;
        background: transparent;
        border: 0;
        -webkit-appearance: none;
    }
</style>
</head>
<body>
<div style="text-align: center">
<div class="overlay">
<div style="width: 305px; margin-left: auto; margin-right: auto; margin-top: -60px; border: 3px solid red;">
<div id="divTestBox1" style="width: 300px; background-color: #fff; border: 1px solid #000; border-radius: 5px; padding: 15px; text-align: center; position: absolute;">
<button type="button" id="linki" class="close" data-dismiss="modal" aria-hidden="true">&times;</button><br/>
<h3>متن مورد نظر</h3>
</div>
</div>
</div>
<a id="link" href="#">link</a><br/>
</div>
</body>
</html>

```

توضیحات: اول یک جدول با div در وسط صفحه ایجاد کردیم به صورتی که در قسمت استایل `margin-top: -60px;` باشد که در این صورت پنجره بالا قرار می گیرد و برای همین دیده نمی شود. وقتی روی `link` کلیک می کنید با استفاده از متد `animate()` مقدار `50px` به `divTestBox1` داده شده برای همین پنجره به پایین می آید.

`"top": "50px"`

به جدولی که `divTestBox1` داخل آن است ارتفاع ندادیم برای همین راحت بزرگ و کوچک می شود بدون اینکه مشکلی پیش بیاد.

وقتی روی علامت ضربدر یا زمینه کلیک می کنید مقدار `"-50px"` : `"top"` به `divTestBox1` داده میشه پس پنجره به سمت بالا میرود.

نکته : کلاس `overlay` همه ی صفحه را فرا گرفته است.

متد `fadeIn()` باعث ظاهر شدن کلاس `overlay` می شود و متد `fadeOut()` باعث مخفی شدن کلاس `overlay` به همراه افکت می شود.

ساخت يك اسلايدر ساده:

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      .slideshow {
        height:300px;
        width:568px;
        overflow:hidden;
      }
      .slideshow ul {
        list-style:none;
        margin:0;
        padding:0;
        position:relative;
      }
      .slideshow ul li {
        margin:0;
        padding:0;
        position:absolute;
      }
    </style>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript">
      var x = 2;
      function slideSwitch() {
        var m = 5;
        x += 1;
        if (x > m) {
          x = 1;
        }
        $(".slideshow ul li:nth-child(n)").animate(
        {
          opacity:0
        });
        $(".slideshow ul li:nth-child(" + (x) + ")").animate(
        {
          opacity:1
        });
      }
      $(document).ready(function() {
        setInterval( "slideSwitch()", 5000 );
      });
    </script>
  </head>
```

```

<body>
  <div class="slideshow">
    <ul>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </div>
</body>
</html>

```

توضیحات:

دستور زیر کل پاراگراف ها را رنگی میکند:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      p:nth-child(n)
      {
        background:#0000FF;
      }
    </style>
  </head>
  <body>
    <h1>عنوان</h1>
    <p>پاراگراف اول</p>
    <p>پاراگراف دوم</p>
    <p>پاراگراف سوم</p>
    <p>پاراگراف چهارم</p>
  </body>
</html>

```

دستور زیر اولین پاراگراف را رنگی میکند:

```

p:nth-child(2)
{
  background:#0000FF;
}

```

دستور زیر آمد و مقدار شفافیت همه ی تصاویر را صفر کرد. یعنی شفافیتی نداشته باشند:

```
$(".slideshow ul li:nth-child(n)").animate({opacity:0});
```

اگر یک بار مقدار 0.5 و یک بار عدد یک به بالا را به دستور opacity بدهید منظور ما را متوجه می شوید.

دستور بعدی یعنی دستور زیر میاد شماره ی هر عکس را به متغیر x میدهد و با اجرای تابع slideSwitch() در هر 5000 میلی ثانیه ، یعنی 5 ثانیه ای یکبار یک شماره ی جدید به متغیر x داده شده و یک عکس دیگر نمایش داده می شود.

```
$(".slideshow ul li:nth-child(" + (x) + ")").animate(  
{  
opacity:1  
});
```

دستور زیر هم هر پنج ثانیه ای یک عدد به متغیر x اضافه می کند:

```
x += 1;
```

تعداد تصاویر موجود 5 عدد هست و ما شرط گذاشتیم اگه مقدار متغیر x بیشتر از عدد 5 شد. مقدار متغیر x یک باشد. پس وقتی عکس پنجمی نمایش داده شد. عملیات متوقف نشده و از تصویر اولی دوباره شروع به نمایش دادن می کند. دستور زیر هم تماس li های زیر مجموعه ی ul که زیر مجموعه یا فرزند کلاس slideshow هستند را مشخص می کند.

```
.slideshow ul li
```

چند مثال با کمک پلاگین های جی کوئری:

گرد کردن div با جی کوئری:

اگه دیده باشید توی خیلی از سایت ها گوشه ی جدولی را که با div میسازند را گرد میکنند. حالا یکی با عکس اینکار را میکنه یکی با کدنویسی . بیشتر مشکل توی مرورگر اکسپلورر هست. اونم ورژن های پایین.

اگه بخوای با عکس این کار را بکنی می توانی با همه ی مرورگر ها سازگارش بکنی . ولی یک کم حجمش میره بالا و وقت گیره. یه راه دیگه هم هست که برای مرورگر اکسپلورر استفاده میشه که با استفاده از فایل PIE.htc اینکار انجام میشه که آموزشش را گذاشتم:

<http://barnamenevisi.blogsky.com/1391/10/16/post-109/>

ولی یک روش دیگه ام برای گرد کردن div در جی کوئری هست که با همه ی مرورگر ها و مرورگر اکسپلورر 9 به بالا سازگار و ولی نسخه های 8 اکسپلورر به پایین سازگار نیست.

در اینجا یک پلوگین بهتون معرفی میکنم که با کمک آن می توانید div را در jquery گرد کنید. به آدرس زیر بروید و فایل jquery.corner.js را دانلود کنید.

<http://jquery.malsup.com/corner/>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type="text/javascript" src="jquery.corner.js">
    </script>
    <script type="text/javascript">
      $('document').ready(function(){
        $('#gerd').corner();
      });
    </script>
    <style>
      #gerd{
        border:1px #e20337 solid;
        width:100px;
        height:100px;
      }
    </style>
  </head>
  <body>
    <div id="gerd">متن نوشته شده</div>
  </body>
</html>
```

<http://malsup.github.com/jquery.corner.js>

کشیدن و انداختن:

```
<!DOCTYPE html>
<html>
  <head>
    <script
      src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js">
    </script>
    <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-
      ui.min.js">
    </script>
    <style type="text/css">
      .draggable {
        width: 100px;
        height: 100px;
```

```

        background: #e7d5f3;
        text-align:center;
    }
</style>
<script type="text/javascript">
$(document).ready(function() {
    $(".draggable").draggable();
});
</script>
</head>
<body>
    <div class="draggable">بکشید</div>
</body>
</html>

```

تغییر اندازه ی div با Resizable():

```

<!DOCTYPE html>
<html>
<head>
<link
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-
ui.css" rel="stylesheet" type="text/css"/>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js">
</script>
<script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-
ui.min.js">
</script>
<style type="text/css">
#resizable {
    width: 100px;
    height: 100px;
    background: silver;
}
</style>
<script type="text/javascript">
$(document).ready(function() {
    $("#resizable").resizable();
});
</script>
</head>
<body>
    <div id="resizable"></div>
</body>
</html>

```

بررسی نوع مرورگر: یک بار پیش میاد که میخوای اگه نوع مرورگر اکسپلورر یا سفری یا کروم باشه فلان کد اجرا بشه برای این کار می توانید از روش زیر استفاده کنید:
اول افزونه ی jquery.mb.browser.js از آدرس زیر دانلود کنید:

<https://github.com/pupunzi/jquery.mb.browser/archive/master.zip>

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" src="jquery.min.js">
</script>
<script type="text/javascript" src="jquery.mb.browser.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
/* دریافت نوع مرورگر */
$.browser.chrome=/chrome/.test(navigator.userAgent.toLowerCase());
/* اگر مرورگر کروم بود */
if ($.browser.chrome){
    alert("این مرورگر کروم است");
return false;
}
$.browser.safari=/safari/.test(navigator.userAgent.toLowerCase());
/*اگر مرورگر سفری باشد*/
if ($.browser.safari){
    alert("این مرورگر سفری است");
}
if ( $.browser.msie){
    alert("این مرورگر اکسپلورر است");
}
else if ($.browser.mozilla){
    alert("این مرورگر فایرفاکس است");
}
else if ( $.browser.opera ){
    alert("این مرورگر اپرا است");
}
});
</script>
</head>
<body>
</body>
</html>
```

ایجکس در jquery : ایجکس یک زبان برنامه نویسی نیست بلکه یک تکنیک است. و راهی برای بروزرسانی بخش هایی از یک صفحه ی وب بدون نیاز به بارگزاری کل صفحه ی وب است. Ajax روشی است برای ارسال و دریافت قسمتی از اطلاعات به سرور بدون رفرش شدن صفحه. بطور مثال شما یک فرم میسازي که به محض کلیک بر روی submit بدون رفرش شدن ، اطلاعات را ارسال کنه و پیام موفقیت را بدهد به این روش ایجکس می گویند. متدهایی که در جی کوئری استفاده میشه:

```
$.load()  
$.post()  
$.get()  
$.getJSON()  
$.ajax()  
$.ajaxComplete()
```

...

http://www.w3schools.com/jquery/jquery_ref_ajax.asp

Load() : در برنامه ی زیر ما با استفاده از load() مقادیر یک صفحه را در صفحه ی جاری نمایش دادیم:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <script type="text/javascript" src="jquery.js">  
    </script>  
  </head>  
  <body>  
    <button id="b">ajax load</button>  
    <div id="result"> </div>  
    <script type="text/javascript">  
      $("#b").click(function(){  
        $('#result').load('test.php');  
      });  
    </script>  
  </body>  
</html>
```

ساختار:

```
$(selector).load(url,data,function(response,status,xhr))
```

پارامترها:

url : این پارامتر ضروری است. تعیین آدرسی که بارگذاری می شود.

Data : این پارامتر اختیاری است و پاسخ درخواست را به شما می دهد.

function(response,status,xhr) : تعریف تابعی که بعد از کامل شدن اجرای متد ، اجرا می شود.

پارامترهای اضافی:

Response : شامل داده های حاصل از درخواست.

Status : شامل وضعیت درخواست :

("success", "notmodified", "error", "timeout", یا "parsererror")

Xhr : شامل شی XMLHttpRequest

قبل از اینکه به بحث در مورد \$.post() بپردازم لازم است یک مثال ساده را به شما نشان دهم:

یک فرم ساده که با روش get مقادیر را ارسال میکند میسازیم:

```
<form action="test.php" method="get">
<input type="text" name="matn">
<input type="submit" value="ارسال">
</form>
```

Test.php

```
<?php
echo $_GET['matn'];
?>
```

وقتی در فرم کلمه ی ravandi را می نویسید و روی دکمه کلیک میکنید یک آدرس به شکل زیر در adres bar ایجاد می شود:

<http://localhost/test.php?matn=ravandi>

و سپس متن ravandi در صفحه ی test.php نمایش داده می شود.

شما اگر همین آدرس را هم بدون فرم وارد کنید همین متن ravandi نمایش داده می شود.

\$.post()

ساختار:

\$(selector).post(URL,data,function(data,status,xhr),dataType)

پارامترها:

URL : تعریف آدرسی که می خواهید از آن اطلاعات را درخواست کنید.

Data : این پارامتر اختیاری است و پاسخ درخواست را به شما می دهد.

Function(data,status,xhr) : این پارامتر اختیاری ، یک تابع است که اگر درخواست ، پاسخ موافقی بدهد اجرا می شود.

شامل پارامتر های اضافی زیر است: data,status,xhr

Data : شامل اطلاعات حاصل از درخواست.

Status : شامل وضعیت درخواست :

("success", "notmodified", "error", "timeout", یا "parsererror")

Xhr : شامل شی XMLHttpRequest

dataType : این پارامتر اختیاری است. بر اساس تنظیمات پیش فرض سرور.

انواع آنها عبارتند از:

"xml": یک سند xml

"html": به عنوان متن ساده

"text": یک رشته ی متنی ساده

"script": در پاسخ ، به عنوان جاوا اسکریپت اجرا می شود و آن را به عنوان متن ساده بر می گرداند.

"json": در پاسخ ، به عنوان JSON اجرا می شود. و یک شی جاوا اسکریپت را بر می گرداند.

از روش زیر می توانید مثلاً برای حذف یک رکورد از جدول بدون بارگذاری دوباره ی صفحه برای نمایش پیام انجام عملیات استفاده کنید.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <script type="text/javascript" src="jquery.js">
```

```
  </script>
```

```
    <script type="text/javascript">
```

```
      $(document).ready(function() {
```

```
        $("#dokme").click(function(){
```

```
          $.post("http://localhost/Delete.php?id=4", function(data) {
```

```
            alert("رکورد حذف شده" + data);
```

```
          });
```

```
        });
```

```
      });
```

```
    </script>
```

```
  </head>
```

```
<body>
```

```
  <button id="dokme">کلیک کنید</button>
```

```
</body>
```

```
</html>
```

Delete.php

```
<?php  
echo $_GET['id'];  
?>
```

فایل delete.php فقط برای نمایش ساختیم . شما می توانید در عمل هم تست کنید.
وقتی روی دکمه کلیک می شود مقدار عدد با کمک \$_GET['id'] گرفته شده و با echo نمایش داده می شود. نتیجه هم به data داده می شه و با alert() در همون صفحه ای که هستید نمایش داده میشه.

وقتی این برنامه را نوشتید حالا متوجه می شوید که ایجکس چه مزیتی داره!
می توانید این خط را بردارید:

```
alert("رکورد حذف شده" + data);
```

و به جاش این خط را بذارید:

```
$(".payam").html(data);
```

و پایین دکمه هم این خط را قرار بدید:

```
<span class="payam"></span>
```

اینطوری نتیجه در قسمت span نمایش داده میشه.

روش بالا با استفاده از post بود و این روش هم با استفاده از get :

```
$.get(URL,data,function(data,status,xhr),dataType)
```

پارامترها:

URL : تعیین URL شما ، مایل به درخواست.

Data : این پارامتر اختیاری است ، شی ساده یا رشته ای است که به سرور با درخواست ارسال می شود.

Function(data,status,xhr) : این پارامتر اختیاری ، یک تابع است که اگر درخواست ، پاسخ موافقی بدهد اجرا می شود.

شامل پارامترهای اضافی زیر است: data,status,xhr

Data : شامل اطلاعات حاصل از درخواست.

Statuse : شامل وضعیت درخواست :

("success", "notmodified", "error", "timeout", "parsererror")

Xhr : شامل شی XMLHttpRequest

dataType : این پارامتر اختیاری است. بر اساس تنظیمات پیش فرض سرور.

انواع آنها عبارتند از:

"xml": یک سند xml

"html" : به عنوان متن ساده

"text" : یک رشته ی متنی ساده

"script" : در پاسخ به عنوان جاوا اسکریپت اجرا می شود و آن را به عنوان متن ساده بر می گرداند.

"json" : در پاسخ به عنوان JSON اجرا می شود. و یک شی جاوا اسکریپت را بر می گرداند .

ارسال متن به دیتابیس:

قبل از بحث در این مورد میخوایم ببینیم اصلا این val() چه کاری انجام میدهد؟

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <script type="text/javascript" src="jquery.js">
```

```
  </script>
```

```
    <script type="text/javascript">
```

```
      $(document).ready(function(){
```

```
        $("#btn").click(function(){
```

```
          alert($("#iname").val());
```

```
        });
```

```
      });
```

```
    </script>
```

```
  </head>
```

```
<body>
```

```
  <form dir="rtl" method="post">
```

```
    نام:<input type="text" id="iname" name="name">
```

```
    <input type="button" id="btn" value="ارسال">
```

```
  </form>
```

```
</body>
```

```
</html>
```

وقتی متنی را داخل text می نویسیم و روی دکمه کلیک میکنیم ، همان متن در پنجره برای ما نمایش داده میشه.

اول دیتابیستی با نام: database بسازید و سپس دستورات sql را در phpmyadmin و در قسمت sql وارد کنید:

```
CREATE TABLE `test` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(30) collate utf8_persian_ci NOT NULL,  
  `lastname` varchar(30) collate utf8_persian_ci NOT NULL,  
  PRIMARY KEY (`id`)
```



```
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
COLLATE=utf8_persian_ci AUTO_INCREMENT=1 ;
```

جدولی هست با سه فیلد id,name,lastname

Send.php

```
<?php  
// این خط برای اینکه وقتی صفحه رفرش میشه مقادیر ارسال نشه  
if(isset($_GET['name']) and isset($_GET['lastname'])) {  
echo 'نام : ' . $_GET['name'] . ' - فامیلی : ' . $_GET['lastname'];  
exit;  
}  
?>  
<!DOCTYPE html>  
<html>  
<head>  
<script type="text/javascript" src="jquery.js"></script>  
<script type="text/javascript">  
function SendData(){  
$.get("get.php",{ name:$('#iname').val(),lastname:$('#ilastname').val()}),  
function(data){  
    alert(data);  
});  
}  
</script>  
</head>  
<body>  
<form dir="rtl" method="get" action="get.php">  
<input id="iname" name="name" type="text" size="15">  
<input id="ilastname" name="lastname" type="text" size="15">  
<input type="button" onclick="SendData()" value="ارسال">  
</form>  
</body>  
</html>
```

Get.php

```
<?php  
$Link= mysql_connect("localhost","root","");  
$Dbname="database";  
$name=$_GET['name'];  
$lastname=$_GET['lastname'];  
mysql_set_charset('utf8',$Link);  
$query="INSERT into `test` (`name`,`lastname`) values  
('$name','$lastname')";  
if(mysql_db_query($Dbname,$query,$Link)){  
    echo "اطلاعات ثبت شد";  
}
```

```
}else{  
    echo "خطایی در ثبت اطلاعات رخ داده است";  
}  
?>
```

`$.get()` مقادیر را به صفحه ی `test.php` ارسال کرده و نتیجه را به متغیر `data` می دهد و با کمک `alert()` مقادیر را چاپ میکند.
اگر خط زیر:

```
<form dir="rtl" method="get" action="get.php">
```

به این صورت هم نوشته می شود:

```
<form>
```

باز برنامه ی ما کار میکند. ولی چون روش بالا استاندارد بود من اون شکلی نوشتم.

اگر `php` بلد نیستید کتاب آموزش `php` به زبان ساده را که خودم نوشتم از آدرس زیر دانلود کنید:

<http://www.barnamenevisi.blogsky.com/1391/01/06/post-96/>

`$.getJSON()` : این متد برای بدست آوردن داده های `json` استفاده می شود برای تکنیک ایجکس و با روش درخواست `get` .
ساختار:

`$(selector).getJSON(url,data,success(data,status,xhr))`
پارامترها:

url : پارامتری ضروری، برای تعریف آدرس برای فرستادن درخواست.
Data : پارامتری اختیاری ، شی ساده یا رشته ای است که به سرور با درخواست ارسال می شود.

success(data,status,xhr) : پارامتری اختیاری ، اگر درخواست با موفقیت انجام شود تابع اجرا می شود.

Data : شامل اطلاعات بازگردانده شده از سرور.

Status : شامل یک رشته حاوی وضعیت درخواست ("موفق"، "تغییر نیافته"، "خطا"، "وقفه"، "خطای تجزیه کننده(parser error)"

Xhr : شامل شی `XMLHttpRequest`


```

        document.write("lastname: " + json.lastname);
    });
});
</script>
</head>
<body>
</body>
</html>

```

Get.php

```

<?php
$handle = mysql_connect("localhost","root","");
$db = mysql_select_db("database");
mysql_set_charset('utf8',$handle);
$res=mysql_query("SELECT * FROM test");
while ($row=mysql_fetch_assoc($res))
{
    $namei[]=$row['name'];
    $lastnami[]=$row['lastname'];
    /*
    مقادیر را به صورت آرایه در آورده ایم تا راحت بتوانیم
    هر مقداری را به راحتی بهش دسترسی پیدا کنیم.
    */
    $arr=array("name" => $namei,"lastname" =>$lastnami);
}
echo json_encode($arr);
?>

```

\$.ajax() : این متد برای اجرا کردن یک درخواست ایجکس به کار برده می شود.

ساختار:

\$.ajax({name:value, name:value, ... })

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script type="text/javascript" src="jquery.js">
        </script>
        <script type="text/javascript">
            $(document).ready(function(){
                $("#btn").click(function () {
                    var namei = $("#iname").val();
                    var lasti = $("#ilast").val();
                    $.ajax({
                        type: "POST",

```

```

        url: "test.php",
        data: {'name':namei, 'lastname':lasti},
        success: function(javab){
            $("#javabi").html(javab);
        }
    });
});
});
</script>
</head>
<body>
<form dir="rtl" action="test.php" method="post">
نام:<input type="text" name="name" id="iname" />
نام خانوادگی:<input type="text" name="lastname" id="ilast" />
<input type="button" value="ارسال" id="btn" />
</form>
<span id="javabi"></span>
</body>
</html>

```

```

Test.php
<?php
echo $_POST['name'];
echo "<br/>";
echo $_POST['lastname'];
?>

```

در قسمت type می توانید به جای POST از GET استفاده کنید در این صورت باید در کدهای php به این شکل عمل کنید:

```

echo $_GET['name'];

```

برنامه ی بالا نیازی به توضیح نداره چون با رنگ ها ارتباط کدهای برنامه درست نشان داده شده.

ارسال مقادیر چک باکس به صورت آرایه:

انتخاب مقادیر چک باکس را قبلاً یاد دادیم حالا میخوایم این مقادیر را با ایجکس ارسال کنیم.

شاید بخوایم چند گزینه رو به صورت آرایه ارسال کنیم. با روش زیر ارسال میکنیم:

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type='text/javascript'>
      $(document).ready(function(){
        $("#btn").click(function(){
          $.ajax({
            url: 'test.php',
            type: 'post',
            data:
              $("input[name='name[]']:checked").serialize(),
            success: function(data) {
              $('#javabi').html(data);
            }
          });
        });
      });
    </script>
  </head>
  <body>
    <p align="right">تخصص شما چیست؟</p>
    <form dir="rtl">
      <input type="checkbox" name="name[]" value="php" />php
      <input type="checkbox" name="name[]" value="jquery" />jquery
      <input type="button" value="ارسال" id="btn" />
    </form>
    <span id="javabi"></span>
  </body>
</html>

```

چون مقدار ما به صورت آرایه ارسال می شود ما موقتاً برای نمایش ، از دستور `print_r()` استفاده میکنیم به صورت زیر:

```

Test.php
<?php
print_r($_POST['name']);
?>

```

دستور زیر در قسمت `input` برای این است که مقادیر به صورت آرایه در نظر گرفته شود.

```
name="name[]"
```

حالا میخواهیم مقادیر داخل دیتابیس را انتخاب کنیم و بعد حذف کنیم:

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery.js">
    </script>
    <script type='text/javascript'>
      $(document).ready(function(){
        $('#btn').click(function () {
          $.ajax({
            url: 'test.php',
            type: 'post',
            data: $("input[name='id[]']:checked").serialize(),
            success: function(data) {
              $('#javabi').html(data);
            }
          });
        });
      });
    </script>
  </head>
  <body>
    <?php
      Header('Cache-Control: no-cache');
      $connection=mysql_connect("localhost","root","");
      $db=mysql_select_db("database",$connection);
      $sql="select * from contact";
      mysql_query("SET CHARACTER SET utf8",$connection);
      $result=mysql_query($sql,$connection);
      echo '<form dir="rtl">';
      echo '<p>عناوین مطالب</p><br/>';
      while ($row=mysql_fetch_array($result)){
        $id=$row['id'];
        $title=$row['tsubject'];
        //چک باکس به تعداد هر رکورد ایجاد می شود
        echo "<input type='checkbox' name='id[]' value=$id />".$title."<br/>";
      }
      echo '<input type="button" value="ارسال" id="btn" />';
      echo '</form>';
      mysql_close($connection);
    ?>
    <span id="javabi"><span>
  </body>
</html>

```

برای اینکه مقادیر رو به صورت آرایه ارسال بکنیم نام فیلد را به این شکل یعنی id[] در میاریم.

```
data: $('input[name='id[]']:checked').serialize(),
```

دستور بالا باعث می شود که تمام input هایی که مقدار name آن ها id[] هست و تیک خورده است با استفاده از serialize() به صورت آرایه در بیاد.

اگر رکورد ها را در داخل تگ های <table> قرار میدادید. بعد از کلیک روی دکمه رکورد ها ظاهراً حذف نمیشد و باید صفحه بارگذاری بشه تا شما از حذف رکورد ها مطلع شوید برای حل این مشکل می توانید از دستور زیر بعد از خط مشخص شده استفاده کنید:

```
$('#javabi').html(data);  
jQuery('input:checkbox:checked').parents("tr").remove();
```

Test.php

```
<?php
```

```
// برای جلوگیری از کش شدن برنامه
```

```
Header('Cache-Control: no-cache');
```

```
$host = "localhost";
```

```
$user = "root";
```

```
$password = "";
```

```
$database = "database";
```

```
// اتصال به هاست و نام کاربری و پسورد
```

```
$conn = mysql_connect($host, $user, $password);
```

```
// انتخاب بانک اطلاعاتی
```

```
mysql_select_db($database, $conn);
```

```
print_r($_POST['id']);
```

```
// اگر مقداری وجود ندارد
```

```
if(!isset($_POST['id'])) {
```

```
    die('مقداری را انتخاب نکردید.');
```

```
}
```

```
// دریافت مقدار
```

```
$id=$_POST['id'];
```

```
// اگر مقدار وجود دارد
```

```
if(isset($_POST['id'])) {
```

```
    // اگر مقادیر به صورت آرایه هست
```

```
    if(is_array($_POST['id'])) {
```

```
        // مقادیری که به صورت آرایه هست پشت سر هم به متغیر آی دی بده
```



```

foreach($_POST['id'] as $id){
    $delete = mysql_query("DELETE FROM `contact` WHERE `id`='$id'");
}
}
}
if($delete !== false && mysql_affected_rows() > 0) {
    echo 'حذف شد.';
}else{
    echo 'حذف نشد.';
}
mysql_close($conn);
?>

```

\$delete = mysql_query("DELETE FROM `contact` WHERE `id`='\$id'");
 با استفاده از دستور بالا رکوردی پاک می شود که مقدار id آن مساوی شماره ی رکوردی
 باشد که داخل متغیر \$id باشد. با دستور mysql_query پیامی را برای دریافت پاسخ
 ارسال میکنیم در سطرهای بعدی بررسی می شود که اگر پاسخ true بود (درست بود) پیام
 "حذف شد" داده شود.

ساخت فرم لوگین با ایجکس در jquery :

Login.php

```

<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
function SendData(){
$.get("admin.php", {
    user: $('#useri').val(),pass: $('#passi').val()
},
function(data){
if(data != 2){
    $(".asli").hide();
    $(".sahih").html(data);
}else{
    $(".kheta").html("نام کاربری یا پسورد اشتباه است.");
}
});
}
</script>
<style type="text/css">
.asli{
    direction:rtl;
    margin-left:auto;

```

```

        margin-right:auto;
    }
    A{
        text-decoration: none;
    }
    .kheta{
        color:red;
    }
</style>
</head>
<body>
<div style="text-align:center;">
<div class="asli">
<form dir="rtl" action="admin.php" method="get">
کاربر:<input type="text" size="20" id="useri" dir="ltr" name="user">
<br/>
پسورد:<input type="password" size="20" id="passi" dir="ltr"
name="pass">
<br/>
<input type="button" onclick="SendData()" value="ورود" id="btn">
</form>
<span class="kheta"></span>
</div>
<span class="sahih"></span>
</div>
</body>
</html>

```

قبل از اینکه بخواهم قسمت php رو توضیح بدم باید بگم ما میایم پسورد رو به صورت رمز گذاری شده در دیتابیس ذخیره میکنیم. من از روش زیر استفاده کردم:

```

<?php
echo hash('sha256',"ravandi");
?>

```

الان پسورد واقعی ما ravandi هست. شما فایل بالا را اجرا کنید. به همین خروجی به شما میدهد:

```
38c668c7073f7468b68d2c2152cf9b0f15eeb73a774789738943134c1453abbd
```

شما باید رمز بالا را در دیتابیس قرار بدید. این کار برای این است که هکر نتونه رمز واقعی شما رو پیدا کنه.

نکته: نباید هیچ گاه رمز عبور ساده برای خودتون بسازید.

مثلا این رمز عبور mohammad ساده است ولی این ساده نیست:

```
$&jgudgfsop
```

یک دیتابیس با نام database در phpmyadmin بسازید و کدهای زیر را در بخش sql وارد کنید:

```

CREATE TABLE `user` (
  `id` int(11) NOT NULL auto_increment,
  `user` varchar(50) collate utf8_persian_ci NOT NULL,
  `pass` varchar(150) collate utf8_persian_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_persian_ci AUTO_INCREMENT=2 ;

INSERT INTO `user` (`id`, `user`, `pass`) VALUES
(1, 'ravand',
'38c668c7073f7468b68d2c2152cf9b0f15eeb73a774789738943134c1453abbd
');
```

دستورات بالا یک جدول با نام user و با سه فیلد id,user,pass و سازگار با فارسی در جدول ایجاد میکند و مقادیر نام کاربری ravand و پسورد ravandi رمزگذاری شده را در جدول وارد میکند.

Conn.php

```

<?php
$mysql_hostnameu = "localhost";
$mysql_useru = "root";
$mysql_passwordu = "";
$mysql_databaseu = "database";
// اتصال با نام کاربری و پسورد به هاست محلی
$conn = mysql_connect($mysql_hostnameu, $mysql_useru,
$mysql_passwordu);
// انتخاب دیتابیس
mysql_select_db($mysql_databaseu, $conn);
?>
```

Admin.php

```

<?php
session_start();
// با دستور بالا سشن شروع میشه
// اگر سشنی از قبل وجود داشته باشه پاک میشه
if(isset($_SESSION['login'])) {
    unset($_SESSION['login']);
}
/* اتصال به هاست محلی با استفاده از نام کاربری و پسورد
و انتخاب دیتابیس به این فایل ضمیمه میشه
*/
```

```

include("conn.php");
// دریافت پسورد و ذخیره در متغیر
$passi=$_GET['pass'];
// دریافت نام کاربری و ذخیره در متغیر
$useri=$_GET['user'];
// تبدیل حروف بزرگ به کوچک
$useriu = strtolower($useri);
// یک روش رمزگذاری
$passiu =hash('sha256',$passi);
// سازگاری با فارسی
mysql_query('SET NAMES \'utf8\'');
$qry = "SELECT * FROM `user` WHERE (LOWER(`user`)='{ $useriu }'
AND `pass`='{ $passiu }')";
// برای بررسی عملیات یک پیام به جدول و دیتابیس ارسال می شود.
$result = mysql_query($qry);
// اگر نتیجه‌ی پاسخ بیشتر از صفر باشد.
if($result && mysql_num_rows($result) > 0) {
    echo "<a href='out.php'>خروج</a><br/>";
    // نام کاربری داخل سشن ذخیره می شود.
    // تا در صفحات بعد وجود کاربر بررسی شود.
    $_SESSION['login'] = $useriu;
    echo "خوش آمدید " . $useriu;
}
else {
    echo 2;
}
mysql_close($conn);
?>

```

می توانستیم یک فرم قشنگتر با امکانات بیشتر بنویسم ولی هدفم اینجا آموزشی بود . مثلاً باید برای فرم در قسمت کدهای php اعتبارسنجی میذاشتیم امنیت را رعایت میکردم ولی نکردم که کدها شلوغ نشه.

توضیحات: در فرم login.php وقتی فرد دکمه را کلیک میکند با استفاده از onclick تابع SendData() فراخوانی میشه . و دستورات داخل آن اجرا میشه. مقادیر داخل تکست را همانطوری که توضیح دادم برای فایل admin.php ارسال میشه و پاسخ آن در متغیر data قرار میگیره که ما با استفاده از شرط میگیریم اگر مقدار متغیر data نامساوی با عدد 2 هست در کلاس sahih. با استفاده از html() فایل admin.php را نمایش بده و کلاس asli. را که فرم داخل آن است را با hide() مخفی کن . وگرنه اگر با دو مساوی بود داخل کلاس kheta. با استفاده از html() متن نام کاربری یا پسورد اشتباه است را نمایش بده. می توانید

اگر اعتبارسنجی برای فرم ساختید براساس اعداد دیگه ای هم مثلا 3 بذارید و بگید که اگر با عدد دو و سه نامساوی بود این کارها را بکنه. یا حتی شرطی بذارید که اگه مقدار متغیر data با عدد 2 برابر بود پیغام بده. و با 3 برابر بود یک پیغام دیگه وگرنه لوگین انجام بشه.

Admin.php

خط آبی رنگ میگه انتخاب کن همه ی رکوردها از جدول user که فیلد user آن مساوی آن چیزی باشد که داخل متغیر \$useriu است. و فیلد pass آن مساوی مقدار داخل متغیر \$passiu است. نتیجه ی عملیات داخل متغیر \$qry قرار می گیرد. LOWER حروف بزرگ را به کوچک تبدیل میکند. چون اگه به اشتباه هم حروف بزرگ تایپ بشه، با این روش به حروف کوچک تبدیل بشه.

حالا نوبت به فایل out.php می رسد که با کلیک روی لینک خروج، این سشن که بوجود آمده پاک شود که در صفحات بعدی کسی نتونه وارد بشه. البته ما صفحات بعدی رو نساختیم.

Out.php

```
<?php  
include("admin.php");  
session_destroy();  
header("location:login.php");  
?>
```

برای اطلاعات بیشتر به کتاب آموزش php به زبان ساده مراجعه کنید:

<http://www.barnamenevisi.blogspot.com/1391/01/06/post-96/>

کتاب های من را فقط از سایت یا وبلاگم به آدرس های زیر دانلود کنید:

<http://pachenar.ir>

<http://www.barnamenevisi.blogspot.com>

آپلود عکس با کمک ایجکس:

Send.php

```
<html>  
<head>  
<script type="text/javascript" src="http://code.jquery.com/jquery-1.10.1.min.js">  
</script>  
<script type="text/javascript"  
src="http://malsup.github.io/jquery.form.js">  
</script>  
<script type="text/javascript">  
$(document).ready(function(){  
$("#btn").click(function(){  
$("#upload-form").ajaxForm({target:'#preview'}).submit();
```

```

});
});
</script>
</head>
<body>
<form id="upload-form" method="post" enctype="multipart/form-data"
action='upload.php'>
<input type="file" name="aks" /><br/>
<input type="text" name="matn"/><br/>
<input type="button" value="ارسال" id="btn"/>
</form>
<div id='preview'></div>
</body>
</html>

```

Upload.php

```

<?php
// شما می توانید این مقدار را به متغیر دهید و بعد وارد دیتایس کنید.
echo $_POST['matn']. "<br/>";
$target = "upload/" . basename( $_FILES['aks']['name']);
if(move_uploaded_file($_FILES['aks']['tmp_name'], $target))
{
    echo $target. "<br/>". "آپلود شد ";
}
else {
    echo "مشکلی در آپلود فایل بوجود آمده است ";
}
?>

```

یک پوشه هم با نام upload در پوشه ی www ایجاد کنید.

بقیه ی دستورات ایجکس را می توانید در این آدرس ببینید:

http://www.w3schools.com/jquery/jquery_ref_ajax.asp

و در آخر از شما می خواهیم که حق کپی رایت را رعایت کنید. اگر زمانی به یکی دو تا از برنامه های این کتاب نیاز داشتید که مثلاً به یکی از دوستان در انجمنی نشان دهید لطفاً منبع آن را یعنی کتاب آموزش سریع و آسان jquery را ذکر کنید. حتماً نظراتتان را در مورد این کتاب برای ما ارسال کنید. اگر مطلب آموزشی جدیدی نوشتید خوشحال میشم برای من هم ارسال کنید.

منابع کتاب :

<http://www.w3schools.com>

<http://api.jquery.com>

<http://barnamenevis.org>

<http://stackoverflow.com>

<http://www.hacktweaks.com/2010/09/worlds-smallest-jquery-slideshow.html>

<https://github.com>

<http://barnamenevisi.blogsky.com>