

ژنرال وب دیزاین جلسه نهم



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

کلیک

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
<script>
  $('#btn1').click(
function() {
  $('#box').hide();
}
)
</script>
</body>
</html>
```



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

دابل کلیک

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
<script>
$('#btn1').dblclick(

function() {
  $('#box').hide();
}
)
</script>
</body>
</html>
```



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
<script>
  $('#btn1').mouseenter(
function() {
  $('#box').hide();
}
)
</script>
</body>
</html>
```

mouseenter

آمدن ماوس روی شی
یا وارد شدن ماوس



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

mousedown

پایین نگه داشتن ماوس

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
<script>
  $('#btn1').mousedown(
function() {
  $('#box').hide();
}
)
</script>
</body>
</html>
```



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

mouseup

برداشتن دست از کلیک

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
<script>
$('#btn1').mouseup(
function() {
  $('#box').hide();
}
)
</script>
</body>
</html>
```



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200px;height:200px background:blue">
  </div>
  <button class="btn1"> اجرا کن </button>

  <script>
    $(#btn1).hover(
      function() {
        $('#box').hide();
      },function() {
        $('#box').show();
      }
    );
  </script>
</body>
</html>
```

hover

قرار گرفتن ماوس روی شی و یا خارج شدن ماوس از روی آن

یعنی دو تا رویداد داره

یکی **mouseenter**

دیگری **mouseleave**



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200px;height:200px background:blue">
  </div>
  <input id="textbox1" type="text">
  <script>
    $('#textbox1').focus(
function() {
    $('#box').hide();
  }
)
</script>
</body>
</html>
```

focus

فعال کردن شی توسط کاربر
(رویداد غیر ماوسی هست)



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی رویدادها

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200px;height:200px background:blue">
  </div>
  <input id="textbox1" type="text">
  <script>
    $('#textbox1').blur(
function() {
    $('#box').hide();
  }
)
</script>
</body>
</html>
```

blur

معکوس عمل **focus** یعنی
وقتی شی توسط کاربر از
فوکوس خارج می شود
یعنی داخل شی هستیم
بیرون آن کلیک کنیم
(رویداد غیر ماوسی هست)



مرور توابع ، رویدادها و افکت های مهم جی کوئری

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
<div id="box" style="width:200px;height:200px background:blue; display:none">
</div>
<button class="btn1"> مخفی کن </button>
<br>
<button class="btn2"> نمایش بده </button>
<script>
$('#btn1').click(
function() {
    $('#box').hide();
}
)
$('#btn2').click(
function() {
    $('#box').show();
}
)
</script>
</body>
</html>
```

تست عملی افکت ها

افکت ها:

Hide() : مخفی کردن

Show() : نمایش

Fadein()

Fadeout()

Slideup()

Slieodown()

Slidetoggle()

Fadetoggle()

برای تست نمایش میتونیم **display:none** را هم برای باکس قرار بدیم و اینجوری تست کنیم



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی افکت ها

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
<div id="box" style="width:200px;height:200px background:blue">
</div>
<button class="btn1"> اجراکن </button>
<br>
<button class="btn2"> اجراکن </button>
<script>
$('#btn1').click(
function() {
    $('#box').fadein();
}
)
$('#btn2').click(
function() {
    $('#box').fadeout();
}
)
</script>
</body>
</html>
```

افکت ها:

Hide()

Show()

Fadein(): معادل show باز شدن به نرمی

Fadeout(): معادل hide مخفی شدن به نرمی

Slideup()

Slieodown()

Slidetoggle()

Fadetoggle()

برای تست نمایش میتونیم **display:none** را هم برای باکس قرار بدیم و اینجوری تست کنیم



مرور توابع ، رویدادها و افکت های مهم جی کوئری

تست عملی افکت ها

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
  <br>
  <button class="btn2"> اجراکن </button>
</script>
$('#btn1').click(
function() {
  $('#box').slideup();
}
)
$('#btn2').click(
function() {
  $('#box').slideDown();
}
)
</script>
</body>
</html>
```

افکت ها:

Hide()

Show()

Fadein()

Fadeout()

hide مخفی شدن اسلایدی معادل Slideup()
show باز شدن اسلایدی معادل Slieodown()

Slidetoggle()

Fadetoggle()

برای همه میتونیم یک پارامتر برای زمان
انجام کار بدهیم که به میلی ثانیه هست



مرور توابع ، رویدادها و افکت های مهم جی کوئری

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
<div id="box" style="width:200;height:200px background:blue">
</div>
<button class="btn1"> اجراکن </button>
<br>
<button class="btn2"> اجراکن </button>
<script>
$('#btn1').click(
function() {
    $('#box').slidetoggle();
}
)
$('#btn2').click(
function() {
    $('#box').fadetoggle();
}
)
</script>
</body>
</html>
```

تست عملی افکت ها

افکت ها:

Hide()

Show()

Fadein()

Fadeout()

Slideup()

Slieodown()

Slidetoggle()

Fadetoggle()

Toggle کار مخفی و نمایش را باهم انجام میدهد : اگر مخفی باشد نمایش میدهد و اگر آشکار باشد مخفی می کند برای **fade** و **slide** هر کدام را با روش و نحوه نمایش خودشان این کار را انجام میدهد



مرور توابع ، رویدادها و افکت های مهم جی کوئری

نکته بسیار مهم

آیا لازم هست از ابتدا همه اینها را حفظ کنیم؟
قطعاً نه

برنامه نویسی حفظ کردن نیست یادگیری هست
هرجا یادت رفت میتونی سرچ کنی یا حتی از سایر برنامه نویس ها بپرسی
اشکال و یا **error** جزئی از برنامه نویسی هست

اصلاً دلیل نمیشه کسی که حین برنامه نویسی به **error** میخوره بلد نیست ، برنامه نویسی هر لحظه اش ممکنه یک موضوع جدید برات باشه ، بنابراین باید این موضوع را خیلی دقت کنید

بعضی مواقع تمرکز نیست، اشتباه میشه، بعضی مواقع یک چیز پیش پا افتاده شما را به چالش می کشه

اصلاً نباید نگران باشید باید محکم باشید ، خسته نشید، تلاش کنید
هر روز تمرین، هر روز یادگیری بطوریکه جزئی از زندگیتان بشه



متغیر - نحوه تعریف آن - انواع متغیر هادر JS

متغیر: متغیرها ظرف های نامداری هستند که میتوانند در یک زبان برنامه نویسی تعریف شده و مقداردهی شوند.

اگر یک متغیر را همچون یک جعبه تصور کنیم مفهوم آن برای ما قابل درک می شود. برای مثال متغیری به نام message را می توان همچون یک جعبه تصور کرد که روی آن برچسب "message خورده و مقدار "Hello!" در آن قرار دارد:



متغیر- نحوه تعریف آن- انواع متغیرها در JS

ثابت:

```
const myBirthday = '18.04.1982';  
Const pi=3.141516;
```



متغیر - نحوه تعریف آن - انواع متغیرها در JS

متغیر: متغیرها ظرف های نامداری هستند که میتوانند در یک زبان برنامه نویسی تعریف شده و مقداردهی شوند.

JavaScript به ما اجازه ی کار با سه داده نوع اولیه و اصلی را میدهد :

اعداد - برای مثال ۱۲۳ ، ۱۲۰,۰۵ و ...

رشته ها - برای مثال " This text string " و یا "این رشته از متن" و ...

بولین - برای مثال true یا false

جاوا اسکریپت همچنین دارای دو داده نوع اصلی جزئی نیز

هست، `null` و `undefined` که هر کدام از آنها تنها یک مقدار را تعریف میکنند. به

علاوه ی این داده نوع های اصلی ، جاوا اسکریپت از داده نوع های مرکب و ترکیبی

که با نام " `object`" شناخته میشوند نیز پشتیبانی میکند.



متغیر - نحوه تعریف آن - انواع متغیر هادر JS

```
<script type="text/javascript">  
  var money;  
  var name;  
</script>
```

حوزه و یا قلمرو یک متغیر ناحیه ایست که متغیر در آن تعریف شده است. متغیر های جاوا اسکریپت تنها میتوانند دارای دو قلمرو باشند.

متغیر های سراسری - یک متغیر سراسری دارای قلمرو سراسری است. این به این معناست که آن میتواند در هر جایی از برنامه ی جاوا اسکریپت تعریف و استفاده شود.

متغیر های محلی - متغیر های محلی تنها در یک تابعی که در آن تعریف شده اند قابل استفاده اند. پارامتر های تابع همیشه برای آن تابع محلی هستند.



متغیر - نحوه تعریف آن - انواع متغیر هادر JS

```
<html>
```

```
<body>
```

```
<script>
```

```
    var myVar = "global"; // Declare a global variable
```

```
    function checkscope( )
```

```
{
```

```
    var myVar = "local"; // Declare a local variable
```

```
    document.write(myVar);
```

```
}
```

```
    checkscope();
```

```
    document.write(myVar);
```

```
</script>
```

```
</body>
```

```
</html>
```



متغیر - نحوه تعریف آن - انواع متغیرها در JS

تعریف متغیر با استفاده از کلمه `let`

```
let message;
```

```
message = 'Hello'; // store the string
```

```
alert(message); // shows the variable content
```

```
let message = 'Hello!'; // define the variable and assign the value
```

```
let user = 'John', age = 25, message = 'Hello';
```

در اسکریپت های قدیمی تر از کلمه `var` برای تعریف متغیرها استفاده می شد



توابع

من میخوام چندین متغیر رو باهمدیگه جمع ریاضی کنم، خب در حالت عادی و بدون استفاده از تابع میایم مثلا چنتا متغیر با مقدار مشخص تعریف میکنیم و بعدش متغیرهارو با همدیگه جمع میکنیم. (چیزی شبیه به دستورات زیر)

```
script>  
var a = 5;  
var b = 5;  
var c = 7;  
var d = 7;  
document.write(a + b);  
document.write("<br>");  
document.write(c + d);  
</script>
```

خب حالا فکرشو کنید شاید بخوایم ۱۰۰۰ تا عدد رو باهم جمع کنیم خب اونوقت باید چکار کرد، میتونیم همانند دستورات بالا عمل کنیم ولی خب میدونید چقد حجم برنامهون میره بالا اینجاست که تابع بدردمون میخوره و میتونه از استفاده تکراری (همینکه بخوایم تعداد زیادی متغیر تعریف و مقدار بدیم بهشون) دستورات جلوگیری کنه در نتیجه سرعت اجرای برنامهون رو بالا ببره، به همین سادگی



نحوه تعریف تابع در زبان جاوااسکریپت

برای ایجاد یا تعریف یک تابع در زبان جاوااسکریپت ما باید از کلمه کلیدی `function` استفاده کنیم بعدش نام تابع مون رو مشخص کنیم و در نهایت به پرانتز باز و بسته ایجاد کنیم:

```
function Name ( Parameter1 , Parameter2 , Parameter3 , ... ) {  
    // کدهایی که میتونن اجرا شن  
}
```

پارامتر و آرگومان:

```
<script>  
function test( Parameter1 , Parameter2 ){  
    .....  
}  
test( Argument1 , Argument2 );  
</script>
```



نحوه تعریف تابع در زبان جاوااسکریپت

تعریف یک تابع بدون پارامتر (چون هیچ پارامتری نداریم پس پرانتز خالی میمونه)

```
<script>  
function ShowMessage(){  
alert(“سلام خوبی”);  
{  
</script>
```

تعریف یک تابع با پارامتر:

```
<script>  
function ShowMessage( name ){  
Alert(“سلام”+name+”خوبی”)  
}  
</script>
```



کلمه کلیدی Return در توابع

```
function Jame( Number_1 , Number_2 ){  
var Result = Number_1 + Number_2;  
document.write(Result);  
}
```

// برای اجرا کردن

```
Jame(5,6);
```

// خروجی همیشه 11

```
document.write( "<br>" );
```

```
document.write( Jame(10,16) );
```

// خروجی همیشه 26



ایجاد یک تابع در درون یه تابع دیگه

```
<script>  
function ShowMessage( Name , Family ){  
    function Show(){  
        document.write(Name + Family);  
    }  
    return Show();  
}  
ShowMessage( "Sadegh" , " Asadi" );  
</script>
```



ادامه ساخت منو

سایت اصلی را ببینیم وقتی منو ماوس نرفته روش فقط سطح اول دیده میشه پس ما باید سطح دوم و سوم را اول از همه مخفی کنیم به طوریکه دیده نشن:
خیلی ساده هست: **display** سطح دو و سه را **none** می کنیم

```
Menu_top>ul>li>ul{
```

```
...
```

```
Display:none;
```

```
}
```

```
#menu_top3{ (دقت کنیم نام سطح ۳ درست باشه)
```

```
...
```

```
Display:none;
```

```
}
```



ادامه ساخت منو

حالا باید از جی کوئری استفاده کنیم
اول میریم پروژه مونو به جی کوئری وصل می کنیم
یک پوشه JS میسازیم و فایل داندلود شده جی کوئری را در داخلش کپی می کنیم
سپس با دستور script که در قسمت head می نویسیم ارتباط را برقرار می کنیم:

```
<head>
```

```
<script src="js/jquery.js"></script>
```

```
</head>
```

و کد بزیم وقتی رفتیم روی منوی سطح بالاتر یعنی روی گزینه ها یا li ها ، باید display یعنی ul داخل اون li از حالت none خارج بشه به صورت زیر:

```
<script>
```

```
$( 'menu_top li' ).hover( function{
```

```
    $( ' >ul', this ).fadein();
```

```
}, function{
```

```
    $( ' >ul', this ).fadeout();
```

```
}}
```

```
</script>
```

نکته ۲: کدهای جی کوئری یک تگ را
باید بعد از تگ بنویسید

نکته:

This یعنی همین سلکتوری که الان
توش هستیم یعنی menu_top li

یعنی \$(' >ul', this).fadein();
برو تو سلکتوری که توش هستیم ul
اون رو نمایش بده

ادامه ساخت منو

حالا یک اجرا میگیریم خروجی را میبینیم:

(بهتر هست منوها هر کدامشان گزینه های زیرمنو داشته باشند)

اگر زمان ظاهر شدن یا مخفی شدن کند باشه میتونیم پارامتر توابع را صفر بزاریم : `fadein(0)` و `fadeout(0)`
چرا منوی سطح سه باز نمیشه ؟ چون اسمش را عوض کردیم بریم برای همان اسم هم دو تابع بالا را بنویسیم
البته در فانکشن خودش:

```
$('# >menu_top3',this).fadein();
```

```
$('# >menu_top3',this).fadeout();
```

حالا تست می کنیم درسته

آیا تمام شد؟



ادامه ساخت منو

در همه آموزشها بله تا همین جا تموم شده ولی من میگم نه ، یه کارای دیگه باید بکنیم از نظر من منو خیلی کارای دیگه هم داره وقتی ماوس از روی هر گزینه ای رد میشه سریع گزینه هاش بسته میشن درحالی که در سایتهای بزرگ مثل دیجی کالا اینجوری نیست ، بریم روی سایت دیجی کالا روی منوها با ماوس حرکت کنیم و تفاوت را ببینیم، حین رد شدن یک چند ثانیه و یا کمتر گزینه هاش باز میمونه چطور اینکار را بکنیم؟

یک دستور تاخیر داریم در جی کوئری به نام **delay** و یک دستور به نام **setTimeout**

```
setTimeout(function, millisecond, param1, param2, ...)
```

مثال:

```
<script>
alert('salam'); // نمایش یک پنجره پیام
setTimeout(function(){ alert('hello');},4000) // نمایش یک پنجره پیام با تاخیر ۴ ثانیه
</script>
```

مثال دیگر:

```
<script>
var t; // تعریف یک متغیر - شرح متغیرها و نحوه تعریف آن در برنامه نویسی
t= setTimeout(function(){ alert('hello');},3000) // 3 ثانیه نمایش یک پنجره پیام با تاخیر
clearTimeout(t); // خنثی شدن زمان دستور بالا
</script>
```



ادامه ساخت منو

حالا ببینیم از دستوراتی که گفتیم در منوها استفاده کنیم

اگر یکبار دیگر به دیجی کالا رجوع کنیم میبینیم وقتی ماوس با تاخیر روی گزینه ای قرار میگیره ، گزینه هاش باز میشه ولی موقعی که سریع از روش رد میشه گزینه هاش باز نمیشه
کد قبلی را به صورت زیر تغییر میدیم:

```
<script>
```

```
var timer; //تعریف یک متغیر برای ذخیره زمان
```

```
$('#menu_top li').hover(function{
```

```
var tag=$(this); //تعریف یک متغیر که سلکتور جاری در آن ذخیره شود
```

```
timer= setTimeout(function(){
```

```
$('# >ul',tag).fadeIn();
```

```
$('# >menu_top3', tag).fadeIn();
```

```
},2000);
```

```
},function{
```

```
var tag=$(this); //تعریف یک متغیر که سلکتور جاری در آن ذخیره شود
```

```
timer= setTimeout(function(){
```

```
$('# >ul', tag).fadeIn();
```

```
$('# >menu_top3', tag).fadeIn();
```

```
},2000);
```

```
})
```

```
</script>
```

نکته:

بعد از نوشتن کدهای روبرو اگر خروجی را ببینیم میبینیم که منوها بعد از ۲ ثانیه ماندن روش و یا ۲ ثانیه بیرون رفتن از روش باز و بسته میشه

شرح تعریف متغیر در برنامه نویسی



ادامه ساخت منو

حالا باید در تابع قبلی در هر دو حالت تایمر را از اول شروع کنیم و مقدار قبلی زمان را حذف کنیم :
کد قبلی را به صورت زیر تغییر میدیم:

```
<script>
  var timer;
  $('menu_top li').hover(function{
  var tag=$(this);
  clearTimeout(timer);
  timer= setTimeout(function(){
    $(' >ul',tag).fadein();
    $(' >menu_top3', tag).fadein();
  },2000);
},function{
  var tag=$(this);
  clearTimeout(timer);
  timer= setTimeout(function(){
    $(' >ul', tag).fadein();
    $(' >menu_top3', tag ).fadein();
  },2000);
})
</script>
```

نکته:

با دو دستور فوق ، شمارش ثانیه برای باز و بسته شدن هر منو وقتی که یک لحظه جابجا میشه از اول شروع میشه



ادامه ساخت منو

حالا باید در تابع قبلی در هر دو حالت تایمر را از اول شروع کنیم و مقدار قبلی زمان را حذف کنیم :
که قبلی را به صورت زیر تغییر میدیم:

```
<script>
  var timer;
  $('menu_top li').hover(function{
  var tag=$(this);
  clearTimeout(timer);
  timer= setTimeout(function(){
    $(' >ul',tag).fadein();
    $(' >menu_top3', tag).fadein();
  },2000);
},function{
  var tag=$(this);
  clearTimeout(timer);
  timer= setTimeout(function(){
    $(' >ul', tag).fadein();
    $(' >menu_top3', tag ).fadein();
  },2000);
})
</script>
```

نکته: ul که والد آن tag هست را نمایش بده

نکته:

با انتخاب یک متغیر برای تایمر، این تایمر برای همه منوها همیشه درحالی که برای هر منو باید تایمر جداگانه استفاده نمائیم به همین دلیل از آبجکت یا آرایه باید استفاده کنیم نحوه انجام آن را در اسلایدهای بعدی میبینید
شرح آرایه ها در برنامه نویسی



object یا شی در جاوا اسکریپت

آبجکت ها یکی از دیتا تایپ (انواع داده) های بسیار مهم و پر کاربرد جاوا اسکریپت هستند که برای ذخیره سازی جزئیات هر شی استفاده می شود.

آرایه:

تعریف آرایه:

```
Arrayname=[ ]
```

```
let users = ["reza", "mona", "ali"]
```

در آرایه میتونیم n تا مقدار را نگهداری کنیم اما نمیتوانیم مقادیر مختلفی را در خانه هاش نگهداری کنیم که دسته بندی شده باشد به عنوان مثال نمی توانیم اطلاعات نام، کدملی و حقوق چهار نفر را به طور جداگانه و دسته بندی شده داخل یک آرایه ذخیره سازی کنیم.

به همین دلیل از آبجکت ها استفاده می کنیم.



object یا شی در جاوا اسکریپت

برای تعریف یک آبجکت دقیقا مثل آرایه عمل می کنیم، با این تفاوت که به جای براکت ([]) از آکُلاد ({}) استفاده می کنیم.
تعریف آبجکت:

```
let objectname = { }
```

هر آبجکت از چندین Property یا ویژگی تشکیل شده است. مثلا برای یک شخص می توان پروپرتی هایی مثل اسم، کدملی ، حقوق، موبایل و دیگر ویژگی های آن شخص را در نظر گرفت که تمامی آن ها را داخل آبجکت مربوطه ذخیره سازی می کنیم.

هر پروپرتی از دو قسمت Key (کلید، عنوان) و Value (ولیو، مقدار) تشکیل شده است. اسم هر ویژگی همان Key و مقدار هر ویژگی برابر است با مقدار آن.

به عنوان مثال یک شخص را در نظر بگیرید که ۲ پروپرتی دارد. اولی با کلید name و با مقدار ali و دومی با کلید ssn و مقدار 1234567890.



object یا شی در جاوا اسکریپت

مثال:

یک شخص را در نظر بگیرید که ۲ پروپرتی دارد. اولی با کلید `name` و مقدار `ali` و دومی با کلید `ssn` و مقدار `1234567890`.

به صورت زیر تعریف می شود:

```
Let person= {  
  name:'ali',  
  ssn: 1234567890  
}
```



آرایه ای از آبجکت ها

می توانیم آرایه ها را با آبجکت ها ترکیب کنیم

مثال

```
let persons = [  
  {id: 1, name: 'Ali', age: 21},  
  {id: 2, name: 'Amin', age: 12},  
  {id: 3, name: 'Amir', age: 32},  
  {id: 4, name: 'Hasan', age: 15},  
  {id: 5, name: 'Babak', age: 25}  
]
```

در آرایه بالا ما اطلاعات ۵ نفر که شامل آیدی، نام و سن می باشد را در یک آرایه ذخیره کردیم. اطلاعات هر نفر یک آبجکت است



دسترسی به مقادیر آبجکت ها

به دو روش Dot Notation (کاراکتر .) و Bracket Notation (براکت []) می توانیم به مقادیر داخل هر آبجکت دسترسی داشته باشیم.
مثال:

```
Let person = {  
  name: 'ali',  
  age: 33  
}
```

خروجی با کنسول:

```
Console.log(person.name)
```

```
Console.log(person.age)
```



ادامه ساخت منو

حالا برای هر منو یک تایمر جداگانه با آبجکت یا آرایه تعریف می کنیم ، ابتدا برای دوگزینه سطح اول یعنی کالای دیجیتال و لوازم خانگی یک خصوصیت با نام دلخواه برای آن تعریف می کنیم

```
<div id=menu-top .....>
```

```
<ul>
```

```
<li data-menu="1" .....>
```

```
<a>
```

کالای دیجیتال

```
</a>
```

...

```
<li data-menu="2" .....>
```

```
<a>
```

لوازم خانگی

```
</a>
```

```
<script>
```

```
var timer={}; // تعریف یک آبجکت برای ذخیره زمان های هر منو
```

```
$('#menu_top li').hover(function{
```

```
var tag=$(this);
```

```
Var timerattr=tag.attr('data-menu');
```

```
Alert('timerattr');
```

...

نکته: برای مشاهده مقدار یک متغیر
برای تست میتوان از دستور **alert**
برای نمایش خروجی آن استفاده کرد



ادامه ساخت منو

ادامه کدها به صورت زیر تغییر می یابد:

```
<script>
var timer={ }; // تعریف یک آرایه یا آبجکت برای ذخیره زمان های هر منو
$('menu_top li').hover(function{
var tag=$(this); // تعریف یک متغیر که سلکتور جاری در آن ذخیره شود
Var timerattr=tag.attr('data-menu');
cleartimeout(timer[timerattr]);
timer[timerattr]= setTimeout(function(){
    $(' >ul',tag).fadein();
    $(' >menu_top3', tag).fadein();
},2000);
},function{
var tag=$(this); // تعریف یک متغیر که سلکتور جاری در آن ذخیره شود
Var timerattr=tag.attr('data-menu');
cleartimeout(timer[timerattr]);
timer[timerattr]= setTimeout(function(){
    $(' >ul', tag).fadein();
    $(' >menu_top3', tag ).fadein();
},2000);
})
</script>
```

نکته: در این جا عدد هر گزینه منو در آرایه آورده میشه و تایمر برای آن به صورت جداگانه ساخته میشه با دستورات روبرو برای هر گزینه منو مقدار آن آورده میشه و یک تایمر جدا برای آن ساخته شد

نکته: برای همه li هایی که زیرمنو دارند یک صفت **data-menu** جداگانه با مقدار دلخواه غیر تکراری ایجاد می کنیم



ادامه ساخت منو

حالا در منو یک تست می کنیم با قرار گرفتن روی هر منو تایم برای آن محاسبه میشه بهتر هست برای تست این حالت زمان را از چند ثانیه به زیر یک ثانیه تغییر بدیم با رد شدن سریع از روی یک گزینه زیرمنوهای آن باز نمیشه مگر اینکه بیشتر روی آن مکث کنیم.



شرح دستورات منو

عبارت **this** :

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
  <div id="box" style="width:200;height:200px
background:blue">
  </div>
  <button class="btn1"> اجراکن </button>
<script>
  $(#box).click(
function() {
  $(this).hide();
  //var tag=$(this);
  // $(tag).hide();
}
)
</script>
</body>
</html>
```

یعنی همان سلکتوری که الان
داخلش هستیم و داریم کد
میزنیم

دربحث شی گرایی **this** یعنی
همین شی

یکبار **\$(this)** را دریک
متغیر ذخیره می کنیم و بارها
میتونیم به جای آن از متغیر
استفاده کنیم

نکته: برای راحتی کار میتونیم **\$(this)** را در یک متغیر بریزیم
این دوخط معادل خط بالایی هستند



شرح دستورات منو

```
<html>
<head>
<script src="js/hquery3.6.4"> </script>
</head>
<body>
<div class="box1" data-color="red">

<button class="btn1"> اجراکن </button>
<script>
$('#btn1).click(

function() {
var attr=$('#.box1').attr('data-color');
alert(attr);
}
)
</script>
</body>
</html>
```

تعریف صفت یا اتریبیوت در html

میتونیم برای هر تگ html هرچقدر که نیاز داریم صفت تعریف کنیم و در کدنویسی از مقادیر آن استفاده کنیم فرم کلی آن به صورت زیر هست:

< tag data-* ="value">

اتریبیوت با data شروع میشه

میتونیم برای یک تگ بینهایت اتریبیوت تعریف کنیم

نکته: یک متغیر به نام **attr** تعریف شد و مقدار **data-color** را دریافت می کنیم
متد attr برای گرفتن مقدار اتریبیوت از سلکتور بکار میره ابتدا نام سلکتور سپس یک نقطه و سپس نام متد را مینویسیم و پارامترها یا مقادیر ورودی را در داخل پرانتز متد می نویسیم اصلا فرم کلی استفاده از متد به همین صورت هست میتونیم داخل پرانتز هر مقدار دیگری را هم دریافت کنیم

