|  |  |  |
| --- | --- | --- |
| Selector | Example | Description |
| id selector | $(“#menu”) |  |
| class selector | $(“.list-item”)  $(“li.list-item”)  $(“.product.fruit”) | İki class’ı da uygulayanlar. (Sadece birini uygulayanlar seçilmez.) |
| element selector | $(“p”) |  |
| child selector | $(“p > a”) |  |
| descendant selector | $(“p a”) |  |
| universal selector | $(“\*”) | Tüm öğeler seçilir. |
|  | $(“body, p, h1”) |  |
| adjacent sibling selector | $(“h1 + p”) | Aynı ebeveyn içerisinde bulunan belirlediğimiz elemandan hemen sonra gelen elemanı seçmek için kullanılır. |
| general sibling selector | $(“h1 ~ p”) | Aynı ebeveyn içerisinde bulunan belirlediğimiz elemandan sonra gelen elemanları seçmek için kullanılır. |
| attribute selector | $(“[href]”)  $(“a[href]”)  $(“a[href=’#’]”)  $(“a[href^=’http’]”)  $(“a[href$=’com’]”)  $(“li[class\*=’product’]”) |  |

:first

Example

//İlk elemanı seçer.

$("li.list-item:first")

:last

Example

//Son elemanı seçer.

$("li.list-item:last")

:even

Example

//index numarası çift olan elemanları seçer.

$("li.list-item:even")

:odd

Example

//index numarası tek olan elemanları seçer.

$("li.list-item:odd")

:not()

Example

//İlk çocuk element olmayan liste elemanlarını seçer.

$("li:not(:first-child)")

:eq(index{number})

Example

//index numarası 1 olan elemanı seçer.

$("li.list-item:eq(1)")

:gt(index{number})

Example

//index numarası 2'den büyük olan elemanları seçer.

$("li.list-item:gt(2)")

:lt(index{number})

Example

//index numarası 2'den küçük olan elemanları seçer.

$("li.list-item:lt(2)")

:root

Kök elementi seçer.(html elementi)

:lang

Example

<p lang="tr">Türkçe</p>

<p lang="en">English</p>

<p lang="en-US">English</p> <!-- stil uygulanır -->

<script>

$(document).ready(function(){

$("p:lang(en-US)").css("background-color", "yellow");

});

</script>

:animated

Animasyon halindeki öğeleri seçmek için kullanılır. (Animasyon halindeki öğe sayısı birden fazla ise hepsini seçer.)

:nth-child()

* index numaraları 1’den başlamaktadır.

Example

//index numarası 1 olan liste elemanları seçilir.

$("li:nth-child(1)")

:nth-last-child()

Example

$("li:nth-last-child(even)")

:nth-of-type()

:nth-last-of-type()

:first-child

:last-child

:first-of-type

:last-of-type

:only-child

Tek çocuk element olan elemanları seçmek için kullanılır.

Example

//Ebeveyn içerisinde p'den başka element node olmaması gerekir.

$("p:only-child")

:only-of-type

Example

//Ebeveyn içerisinde birden fazla element olabilir. p bir adet olmalıdır.

$("p:only-of-type")

:contains(text)

Parametre olarak verilen dizeyi içeren öğeleri seçer.

Example

<ul class="cities">

<li class="list-item">Kocaeli</li>

<li class="list-item">İstanbul</li>

<li class="list-item">Ankara</li>

<li class="list-item">Edirne</li> <!-- stil uygulanır -->

</ul>

<script>

$(document).ready(function(){

$(".list-item:contains('Edirne')").css("background-color", "yellow");

});

</script>

:empty

İçeriği boş(text ve element içermeyen) öğeleri seçmek için kullanılır.

:parent

İçeriği boş olmayan(text ya da element node içeren) öğeleri seçmek için kullanılır.

:has(selector)

İçerisindeki elemanlardan seçici ile eşleşen öğeleri seç anlamına gelmektedir.

:hidden

Gizli (display:none olan) öğeleri seçmek için kullanılır.

:visible

Görünür öğeleri seçmek için kullanılır.

:enabled

Aktif form elemanlarını seçmek için kullanılır.

:disabled

Pasif form elemanlarını seçmek için kullanılır.

:checked

Check edilmiş elemanları seçmek için kullanılır.

:selected

Seçili olan açılır liste öğesini seçmek için kullanılır.

.first()

.last()

.eq(index)

Index numarası ile eleman seçmek için kullanılır.

* index parametresi zorunlu olan pozitif veya negatif bir sayıdır. Negatif bir sayı kullanmak dizin sayımını seçilen öğelerin başlangıcı yerine sonundan başlatır.

.has(selector{string})

İçerisindeki elemanlardan seçici ile eşleşen öğeleri seç anlamına gelmektedir.

.is(selector{string})

Seçilen nesnenin parametre olarak verilen seçici ile eşleşip eşleşmediğini bool türünde döndürür.

.filter()

Example

<ul class="cities">

<li>Kocaeli</li> <!-- stil uygulanır -->

<li>İstanbul</li>

<li>Ankara</li> <!-- stil uygulanır -->

<li>Edirne</li>

</ul>

<script>

$(document).ready(function(){

$("li").filter(":even").css("background-color", "yellow");

});

</script>

Example

<ul class="cities">

<li class="list-item-1">Kocaeli</li>

<li class="list-item-2">İstanbul</li>

<li class="list-item-3">Ankara</li>

<li class="list-item-4">Edirne</li> <!-- stil uygulanır -->

</ul>

<script>

$(document).ready(function(){

$("li").filter(".list-item-4").css("background-color", "yellow");

});

</script>

Example

<ul class="cities">

<li>Kocaeli</li> <!-- stil uygulanır -->

<li>İstanbul</li> <!-- stil uygulanır -->

<li>Ankara</li>

<li>Edirne</li>

</ul>

<script>

$(document).ready(function(){

$("li").filter(function(index, element){

return index < 2;

}).css("background-color", "yellow");

});

</script>

.each(function(index, element){})

Seçilen elemanları bir döngüden geçirir.

* index parametresi geçerli elemanın index numarasını vermektedir.
* element parametresi geçerli elemanın kendisini vermektedir. this anahtar kelimesi de aynı işlevi görmektedir.
* return false, döngüyü erken durdurmak için kullanılabilir.

Example

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<p>Paragraph 4</p>

<p>Paragraph 5</p>

<script>

$(document).ready(function(){

$("p").each(function(index, element){

console.log(index, $(element).text());

});

});

</script>

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

.slice(startIndex{number}, endIndex{number})

İki adet index noktası arasındaki elemanları seçmek için kullanılır.

* İlk parametre başlangıç ikinci parametre bitiş indeksidir. Bitiş indeksi belirtilmezse son elemana kadar seçilir. Bitiş indeksi negatif değer aldığında sondan bitiş indeksi kadar eleman seçilmez.

.not(selector{string})

|  |
| --- |
| **NOT**  jQuery komutlarını yazarken **$** sembolü yerine **jQuery** ifadesini kullanabilirsiniz.  Example  jQuery(".list-item") |

children(filter:selector{string})

Seçilen elemanın çocuklarını döndürür. (Sadece element node’lar döner)

* filter parametresi opsiyoneldir. Çocuklar için aramayı daraltmayı sağlayan bir seçicidir.
* Bu metod sadece çocuk düğümleri döndürür. Yani DOM ağacında bir alt seviyeye geçiş yapar.

Example

<!-- tüm liste öğelerine stil uygulanır -->

<ul class="cities">

<li>Kocaeli</li>

<li>İstanbul</li>

<li>Ankara</li>

<li>Edirne</li>

</ul>

<script>

$(document).ready(function(){

$(".cities").children().css("background-color", "yellow");

});

</script>

Example

<ul class="cities">

<li>Kocaeli</li> <!-- stil uygulanır -->

<li>İstanbul</li>

<li>Ankara</li> <!-- stil uygulanır -->

<li>Edirne</li>

</ul>

<script>

$(document).ready(function(){

$(".cities").children(":even").css("background-color", "yellow");

});

</script>

.contents()

children() metodundan farklı olarak elementin içerisindeki text ve comment düğümleri de döndürür.

.closest(filter)

Parametre olarak verdiğiniz seçiciyle eşleşen en yakın üst ebeveyni döndürür.

* filter parametresi bir seçici ya da bir jquery objesi olabilir.
* Bu metod ilk atayı bulmak için geçerli öğeden belgenin kök öğesine kadar yukarı doğru hareket eder.

.parent(selector{string})

Seçilen öğenin bir üst öğesini döndürür.

* selector parametresi opsiyoneldir. Üst aramayı daraltmak için kullanılan bir seçicidir.

.parents(selector{string})

Seçilen öğenin tüm üst öğelerini döndürür.

* selector parametresi opsiyoneldir. Ebeveynler için aramayı daraltmayı sağlayan bir seçicidir. Birden fazla seçici virgülle ayrılır.

Example

<div id="div1">

div

<div id="div2">p

<h1 id="h1">h1</h1>

</div>

</div>

<script>

$(document).ready(function(){

console.log($("#h1").parents("#div1, #div2")); //#div1 ve #div2 seçilir

});

</script>

.offsetParent()

position özelliği ile konumlandırılmış ilk üst öğeyi döndürür.

.find(filter)

Seçilen öğenin tüm alt öğelerini döndürür. Bunun için \* seçicisi kullanılır.

* filter parametresi zorunludur. Bir seçici ya da bir jquery objesi olabilir. Birden fazla seçici virgülle ayrılır.
* Bu metod DOM öğelerinin alt öğeleri boyunca son alt öğeye kadar aşağı doğru hareket eder.

Example

<div id="div1">

div

<div id="div2">p

<h1 id="h1">h1</h1>

</div>

</div>

<script>

$(document).ready(function(){

console.log($("#div1").find("#h1")); //#h1 seçilir

});

</script>

.siblings(selector{string})

Seçilen öğenin tüm kardeş öğelerini döndürür.

* selector parametresi opsiyoneldir. Kardeş öğeler için aramayı daraltmayı sağlayan bir seçicidir.

.next(selector{string})

Seçilen öğenin sonraki kardeş öğesini döndürür.

* selector parametresi opsiyoneldir. Sonraki kardeş aramasını daraltmayı sağlayan bir seçicidir.

.nextAll(selector{string})

Seçilen öğeden sonraki tüm kardeş öğeleri döndürür.

* selector parametresi opsiyoneldir. Sonraki kardeşler aramasını daraltmayı sağlayan bir seçicidir.

.prev(selector{string})

Bir önceki kardeş öğeyi döndürür.

* selector parametresi opsiyoneldir. Önceki kardeş aramasını daraltmayı sağlayan bir seçicidir.

.prevAll(selector{string})

Seçilen öğeden önceki tüm kardeş öğeleri döndürür.

* selector parametresi opsiyoneldir. Önceki kardeşler aramasını daraltmayı sağlayan bir seçicidir.

.end()

Seçiciye yapılan manipüle(seçim) işlemini iptal eder.

Example

<ul class="cities">

<li class="list-item">Kocaeli</li>

<li>İstanbul</li>

<li class="list-item">Ankara</li>

<li>Edirne</li>

</ul>

<script>

$(document).ready(function(){

console.log($("li").filter(".list-item").length); //2

//end() sayesinde filter() metodu ile yapılan seçim iptal edildi

console.log($("li").filter(".list-item").end().length); //4

});

</script>

.length

Seçilen eleman sayısını verir.

.index()

Bir öğenin index numarasını öğrenmek için kullanılır.

Example

<ul class="cities">

<li class="list-item-1">Kocaeli</li>

<li class="list-item-2">İstanbul</li>

<li class="list-item-3">Ankara</li>

<li class="list-item-4">Edirne</li>

</ul>

<script>

$(document).ready(function(){

console.log($(".list-item-3").index()); //2

});

</script>

.toArray()

Seçilen öğeleri bir dizi içerisinde döndürür.

.data(key{string}, value{any})

Bir nesneye gizli bir değer atar ya da gizlenmiş değeri döndürür.

Example

<p id="p">Paragraph 1</p>

<script>

$(document).ready(function(){

$("#p").data("number", 10);

console.log($("#p").data("number")); //10

});

</script>

Example

<p id="p">Paragraph 1</p>

<script>

$(document).ready(function(){

$("#p").data("number1", 10);

$("#p").data("number2", 15);

console.log($("#p").data()); //{number1: 10, number2: 15}

});

</script>

* Değer olarak bir obje kullanılabilmektedir.

Example

<p id="p">Paragraph 1</p>

<script>

$(document).ready(function(){

$("#p").data("numbers", {number1: 10, number2: 15});

console.log($("#p").data("numbers")); //{number1: 10, number2: 15}

console.log($("#p").data("numbers").number1); //10

});

</script>

.removeData(name{string})

data() metodu ile atanılan değişkenleri silmek için kullanılır.

Example

$("#p").removeData(); //Tüm değişkenleri siler.

.attr()

Bir elementin niteliklerini ayarlar ya da döndürür.

Example

<img id="grapefruit" width="512" height="512" src="./grapefruit.png" alt="grapefruit">

<script>

$(document).ready(function(){

let element = $("#grapefruit");

console.log(element.attr("width")); //512

});

</script>

Example

let element = $("#grapefruit");

element.attr("width", "256");

element.attr("height", "256");

Example

let element = $("#grapefruit");

element.attr({width: "128", height: "128"});

Example

<img id="grapefruit" width="512" height="512" src="" alt="grapefruit">

<script>

$(document).ready(function(){

$("#grapefruit").attr("src", function(){

return this.alt + ".png";

});

});

</script>

Example

<img id="grapefruit" width="512" height="512" src="" alt="">

<script>

$(document).ready(function(){

$("#grapefruit").attr({

src: function(){

return this.id + ".png";

},

alt: function(){

return this.id;

}

});

});

</script>

Example

<input type="button" value="Düğme">

<input type="checkbox">

<script>

$(document).ready(function(){

$("input[type='button']").attr("disabled", "disabled");

$("input:checkbox").attr("checked", "");

});

</script>

.removeAttr()

Bir elementin niteliklerini silmek için kullanılır.

.css()

Bir elementin css özelliklerini ayarlar ya da döndürür.

Example

<img width="512" height="512" src="./grapefruit.png">

<script>

$(document).ready(function(){

console.log($("img").css("width")); //512px

});

</script>

Example

<p>Paragraph 1</p>

<script>

$(document).ready(function(){

$("p").css("background-color", "yellow");

});

</script>

Example

<p>Paragraph 1</p>

<script>

$(document).ready(function(){

$("p").css({

"background-color": "yellow",

"color": "white",

"font-size": "24px"

});

});

</script>

Example

<img width="512" height="512" src="./grapefruit.png">

<script>

$(document).ready(function(){

$("img").css({

width: function(index, value){

return parseFloat(value) \* 2;

},

height: function(index, value){

return parseFloat(value) \* 2;

}

});

});

</script>

.addClass(className{string})

Seçilen öğelere bir veya daha fazla sınıf adı ekler.

* Birden fazla sınıf adı eklemek için sınıf adları boşlukla ayrılır.

.removeClass(className{string})

Seçilen öğelerden bir veya daha fazla sınıf adını kaldırır.

* Parametresiz kullanıldığında tüm sınıf adları kaldırılır.

.toggleClass(className{string})

Seçilen öğelere bir veya daha fazla sınıf adı ekleme ve çıkarma arasında geçiş yapar.

.hasClass(className{string})

Seçilen öğelerden herhangi biri belirtilen sınıf adına sahipse true aksi halde false değer döner.

.html(content)

Seçilen öğelerin html içeriğini(innerHTML) döndürür ya da ayarlar.

* content parametresi HTML elementlerini içerebilmektedir.
* Bu metodun içerisinde fonksiyon çalıştırılabilir

Example

<div id="div1">

div

<div id="div2">p

<h1 id="h1">h1</h1>

</div>

</div>

<script>

$(document).ready(function(){

$("#div1").html(function(){

return "<h1>Heading</h1>";

});

});

</script>

.text(content)

Seçilen öğelerin metin içeriğini döndürür ya da ayarlar.

.val()

Seçilen öğenin value değerini döndürür ya da ayarlar.

* Bu metodun içerisinde fonksiyon çalıştırılabilir.

Example

<input type="text" name="age" value="30">

<script>

$(document).ready(function(){

$("input:text").val(function(index, value){ //value parametresi 30 olacaktır

return parseFloat(value) + 10; //value değeri 40 olacaktır.

});

});

</script>

.after(content)

Parametre olarak verilen içeriği seçilen öğelerin sonrasına ekler.

* content parametresi bir html içeriği ya da bir jquery objesi olabilir.

.before(content)

Parametre olarak verilen içeriği seçilen öğelerin öncesine ekler.

* content parametresi bir html içeriği ya da bir jquery objesi olabilir.

.append(content)

Parametre olarak verilen içeriği seçilen öğelerin sonuna ekler.

* content parametresi bir html içeriği ya da bir jquery objesi olabilir.

.prepend(content)

Parametre olarak verilen içeriği seçilen öğelerin başına ekler.

* content parametresi bir html içeriği ya da bir jquery objesi olabilir.

.replaceWith(content)

Seçilen öğeleri yeni içerikle değiştirir.

* content parametresi bir html içeriği ya da bir jquery objesi olabilir.

.wrap(wrappingElement{string})

Seçilen öğelerin etrafına parametre olarak verilen elementi ekler.

Example

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<p>Paragraph 4</p>

<script>

$(document).ready(function(){

$("p").wrap("<div></div>");

});

</script>

<!-- Result :

<div><p>Paragraph 1</p></div>

<div><p>Paragraph 2</p></div>

<div><p>Paragraph 3</p></div>

<div><p>Paragraph 4</p></div>

-->

.wrapAll(wrappingElement{string})

wrap() metodundan farklı olarak seçilen öğelerin hepsini birden içine alır.

Example

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<p>Paragraph 4</p>

<script>

$(document).ready(function(){

$("p").wrapAll("<div></div>");

});

</script>

<!-- Result :

<div>

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<p>Paragraph 4</p>

</div>

-->

.wrapInner()

Example

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<p>Paragraph 4</p>

<script>

$(document).ready(function(){

$("p").wrapInner("<span><em></em></span>");

});

</script>

<!-- Result :

<p><span><em>Paragraph 1</em></span></p>

<p><span><em>Paragraph 2</em></span></p>

<p><span><em>Paragraph 3</em></span></p>

<p><span><em>Paragraph 4</em></span></p>

-->

.unwrap()

wrap() metodunun tersini yapmaktadır.

Example

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p>Paragraph 3</p>

<p>Paragraph 4</p>

<script>

$(document).ready(function(){

$("p").wrap("<div></div>")

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

$(this).unwrap()

});

});

</script>

.clone()

Alt düğümler, metin içeriği ve özellikler dahil olmak üzere seçili öğelerin bir kopyasını oluşturur.

* Parametre olarak true değeri verilirse datalar ve olaylar da kopyalanır.
* İkinci parametre olarak true değeri verilirse alt öğelerin data ve olayları da kopyalanır.

.empty()

Seçili öğelerin tüm alt düğümlerini ve içeriğini kaldırır. (Öğenin kendisini kaldırmaz)

.remove()

Seçili öğeleri kaldırır. (Tüm alt düğümler ve içerik de kaldırılır.) (Seçilen öğelerin olayları da kaldırılır.)

.detach()

remove() metodundan farklı olarak seçilen öğelerin olayları kaldırılmaz. Ayrıca kaldırılan öğelerin bir kopyasını tutar ve bu daha sonra yeniden yerleştirilmelerine olanak tanır.

Example

<body>

<p>Paragraph 1</p>

<input type="button" value="Sil">

<input type="button" value="Ekle">

<script>

$(document).ready(function(){

var x;

$("input:button").first().click(function(){

x = $("p").detach();

});

$("input:button").last().click(function(){

$("body").prepend(x);

});

});

</script>

</body>

.width(value)

Seçilen öğelerin genişliğini ayarlar ya da döndürür.

* Genişlik değeri için CSS ölçü birimlerini kullanabilirsiniz. (px varsayılan)

.height(value)

Seçilen öğelerin yüksekliğini ayarlar ya da döndürür.

* Genişlik değeri için CSS ölçü birimlerini kullanabilirsiniz. (px varsayılan)

.innerWidth()

Seçilen öğenin genişliğini ayarlar ya da döndürür. (Genişliğe padding değeri dahildir.)

* Genişlik değeri için CSS ölçü birimlerini kullanabilirsiniz. (px varsayılan)

.innerHeight()

Seçilen öğenin yüksekliğini ayarlar ya da döndürür. (Yüksekliğe padding değeri dahildir.)

* Genişlik değeri için CSS ölçü birimlerini kullanabilirsiniz. (px varsayılan)

.outerWidth()

Seçilen öğenin genişliğini ayarlar ya da döndürür. (Genişliğe padding ve border değerleri dahildir.)

* Parametre olarak true değeri verilirse genişliğe margin değeri de eklenir.
* Genişlik değeri için CSS ölçü birimlerini kullanabilirsiniz. (px varsayılan)

.outerHeight()

Seçilen öğenin yüksekliğini ayarlar ya da döndürür. (Yüksekliğe padding ve border değerleri dahildir.)

* Parametre olarak true değeri verilirse yüksekliğe margin değeri de eklenir.
* Genişlik değeri için CSS ölçü birimlerini kullanabilirsiniz. (px varsayılan)

.offset()

Seçilen öğenin belgenin sol üst köşesine göre koordinatlarını döndürür ya da ayarlar.

* Geriye x ve y koordinat değeri içeren bir obje döner.
* Parametre opsiyoneldir.

Example

<div class="container">div element</div>

<script>

$(document).ready(function(){

let element = $(".container");

element.offset({top: 50, left: 50});

console.log(element.offset()); //{top: 50, left: 50}

});

</script>

.position()

Konumlandırılmış öğenin top ve left değerlerini döndürür.

* Geriye x ve y koordinat değeri içeren bir obje döner.

.scrollTop(value{number})

Dikey kaydırma çubuğunun konumunu döndürür ya da ayarlar.

* value parametresi opsiyoneldir. Piksel cinsinden uzaklığı temsil etmektedir.
* Elementler ve window objesi üzerinde kullanılabilir.

.scrollLeft(value{number})

Yatay kaydırma çubuğunun konumunu döndürür ya da ayarlar.

* value parametresi opsiyoneldir. Piksel cinsinden uzaklığı temsil etmektedir.
* Elementler ve window objesi üzerinde kullanılabilir.

Events(Olaylar)

on(event{string}, function)

Seçilen öğelere bir veya daha fazla event handler tanımlamak için kullanılır.

* event parametresi olay adını temsil etmektedir. Birden fazla olay tanımı için olay adları boşlukla ayrılır.
* function parametresi olay gerçekleştiğinde çalışacak olan fonksiyondur.

Example

$("p").on("click", function(){

console.log("Paragrafa tıklandı.");

})

Example

$("p").on("click dblclick", function(event){

console.log(event.type);

})

off(event{string}, function)

on() metodu ile tanımlanmış olayları kaldırmak için kullanılır.

* function parametresi opsiyoneldir.

Example

$("p").off("click");

Example

function function1(){

console.log("function1 çalıştı.");

}

function function2(){

console.log("function2 çalıştı.");

}

$("p").on("click", function1);

$("p").on("click", function2);

//function2'yi kaldırdık. click olayı gerçekleştiğinde çalışmayacak.

$("p").off("click", function2);

one(event{string}, function)

on() metodundan farklı olarak tanımlı olaylar yalnızca bir kez çalıştırılır.

click()

click olayını tetiklemek için kullanılır.

Example

$("p").click()

dblclick()

dblclick olayını tetiklemek için kullanılır.

Example

$("p").dblclick()

blur()

blur olayını tetiklemek için kullanılır.

focus()

focus olayını tetiklemek için kullanılır.

Example

<input type="text">

<script>

$(document).ready(function(){

$("input:text").focus();

});

</script>

change()

Bir öğenin içeriği değiştiğinde change olayı meydana gelmektedir. Bu metod change olayını tetiklemek için kullanılır.

* Sadece input, textarea ve select elementleri üzerinde çalışmaktadır.

|  |
| --- |
| **NOT**  Seçim menüleri için bir seçenek seçildiğinde change olayı meydana gelmektedir. Metin alanları için içerik değiştirildikten sonra alan odağı kaybettiğinde change olayı meydana gelir. |

keydown()

Klavyeden bir tuşa basıldığında keydown olayı gerçekleşmektedir. Bu metod keydown olayını tetiklemek için kullanılır.

keypress()

keypress olayı keydown’dan sonra gerçekleşir. Bu metod keypress olayını tetiklemek için kullanılır.

* keypress CTRL, ALT, SHIFT, TAB gibi bazı tuşlar için çalışmaz. keydown ve keyup’da böyle bir sınırlama yoktur.

keyup()

Klavyeden basılan tuş bırakıldığında keyup olayı gerçekleşmektedir. Bu metod keyup olayını tetiklemek için kullanılır.

|  |
| --- |
| **NOT**  Tuş basılı olarak tutulursa keydown ve keypress olayları devamlı gerçekleşir bu sırada keyup çalışmaz. keyup basılan tuş bırakıldığında çalışır. |

mousedown()

Fare ile bir öğeye tıklandığında mousedown olayı gerçekleşmektedir. Bu metod mousedown olayını tetiklemek için kullanılır.

mouseup()

Tıklama olayından sonra farenin düğmesi bırakıldığında mouseup olayı gerçekleşmektedir. Bu metod mouseup olayını tetiklemek için kullanılır.

mouseenter()

Fare ile bir öğenin üzerine gelindiğinde mouseenter olayı gerçekleşmektedir. Bu metod mouseenter olayını tetiklemek için kullanılır.

* mouseenter olayı mouseover’dan farklı olarak elementin içerisindeki öğeler için tekrar tetiklenmez.

mouseleave()

Fare öğenin üzerinden ayrıldığında mouseleave olayı gerçekleşmektedir. Bu metod mouseleave olayını tetiklemek için kullanılır.

* mouseleave olayı mouseout’dan farklı olarak elementin içerisindeki öğeler için tetiklenmez.

mouseover()

Fare ile bir öğenin üzerine gelindiğinde mouseover olayı gerçekleşmektedir. Bu metod mouseover olayını tetiklemek için kullanılır.

* mouseenter olayından farklı olarak fare işaretçisi öğenin yanı sıra herhangi bir alt öğeye girerse mouseover olayı yeniden tetiklenir.

mouseout()

Fare öğenin üzerinden ayrıldığında mouseout olayı gerçekleşmektedir. Bu metod mouseout olayını tetiklemek için kullanılır.

* mouseleave olayından farklı olarak fare işaretçisi öğenin yanı sıra herhangi bir alt öğeden ayrılırsa mouseout olayı tetiklenir.

mousemove()

Fare bir öğenin üzerinde hareket ettiğinde mousemove olayı gerçekleşmektedir. Bu metod mousemove olayını tetiklemek için kullanılır.

hover(inFunction, outFunction)

Fare işaretçisi seçili öğelerin üzerine geldiğinde çalıştırılacak olan iki fonksiyon belirtir.

* Bu metod hem mouseenter hem de mouseleave olaylarını tetikler.

Example

$("p").hover(function(){

$(this).css("background-color", "yellow");

}, function(){

$(this).css("background-color", "white");

})

select()

Bir metin alanı seçildiğinde select olayı gerçekleşmektedir. Bu metod select olayını tetiklemek için kullanılır.

Example

<input type="text">

<script>

$(document).ready(function(){

$("input:text").select(function(){

console.log("selected");

});

});

</script>

submit()

Bir form gönderildiğinde submit olayı gerçekleşmektedir. Bu metod submit olayını tetiklemek için kullanılır.

* Sadece form elementi üzerinde çalışmaktadır.

Example

$("form:first").submit()

load

Example

$(window).on("load", function(){

});

scroll()

Bir öğenin kaydırma çubuğu hareket ettirildiğinde scroll olayı gerçekleşmektedir. Bu metod scroll olayını tetiklemek için kullanılır.

* scroll olayı tarayıcı penceresi de dahil tüm kaydırılabilir öğeler için çalışmaktadır.

Example

$(window).scroll(function(){

console.log($(this).scrollTop());

})

resize()

Tarayıcı penceresinin boyutu değiştiğinde resize olayı gerçekleşmektedir. Bu metod resize olayını tetiklemek için kullanılır.

Example

$(window).resize(function(){

console.log($(this).width());

})

ready(function)

DOM yüklendiğinde ready olayı gerçekleşmektedir.

* ready olayı sadece document objesi üzerinde kullanılabilmektedir.

Example

$(document).ready(function(){

});

Example

$().ready(function(){

});

Example

$(function(){

});

Event Object

event.type

Gerçekleşen olayın adını döndürür.

event.which

Hangi klavye tuşuna tıklandığını ya da hangi fare düğmesine basıldığını döndürür.

event.target

Olayı tetikleyen DOM öğesini döndürür.

event.currentTarget

event.CurrentTarget **this** anahtar kelimesine eşittir.

event.pageX

Fare işaretçisinin belgenin sol kenarına göre konumunu döndürür.

event.pageY

Fare işaretçisinin belgenin üst kenarına göre konumunu döndürür.

event.screenX

Fare işaretçisinin ekranın sol kenarına göre konumunu döndürür.

event.screenY

Fare işaretçisinin ekranın üst kenarına göre konumunu döndürür.

event.offsetX

Fare işaretçisinin gezdiği katmanın sol kenarına göre konumunu döndürür.

event.offsetY

Fare işaretçisinin gezdiği katmanın üst kenarına göre konumunu döndürür.

event.stopPropagation()

Olayın yayılmasını önler. (event bubbling, event capturing)

event.isPropagationStopped()

Bu metod olay için stopPropagation() metodunun çağrılıp çağrılmadığını kontrol eder. true ya da false değer döndürür.

event.stopImmediatePropagation()

Tanımlı olayın geri kalanının yürütülmesini durdurmak için kullanılır. Ayrıca olayın DOM ağacında yayılmasını da önler.

Example

function function2(){

console.log("function2 çalıştı.");

}

function function3(){

console.log("function3 çalıştı.");

}

$("p").on("click", function(event){

console.log("function1 çalıştı.");

//function2 ve function3 artık çalışmayacak.

event.stopImmediatePropagation();

});

$("p").on("click", function2);

$("p").on("click", function3);

event.isImmediatePropagationStopped()

Bu metod olay için stopImmediatePropagation() metodunun çağrılıp çağrılmadığını kontrol eder. true ya da false değer döndürür.

event.preventDefault()

Bu metodu bir formun gönderilmesini ya da bir bağlantının açılmasını önlemek için kullanabilirsiniz.

Example

$("a").click(function(event){

event.preventDefault();

});

event.isDefaultPrevented()

Bu metod olay için preventDefault() metodunun çağrılıp çağrılmadığını kontrol eder. true ya da false değer döndürür.

Fonksiyon Tanımlama

jQuery’de üç farklı yöntemle fonksiyon tanımlanabilmektedir.

Example

function addNumbers(x, y){

console.log(x + y);

}

Example

$.addNumbers = function(x, y){

console.log(x + y);

}

$(document).ready(function(){

$.addNumbers(1, 2); //3

});

Example

$.process = {

Process1: function(){

console.log("Process1");

},

Process2: function(){

console.log("Process2");

},

Process3: function(){

console.log("Process3");

}

}

$(document).ready(function(){

$.process.Process1(); //Process1

});

Plugins(Eklenti Yazma)

$.fn.extend

Example

<p>Paragraph 1</p>

<input type="button" value="Change Color">

<script>

$(document).ready(function(){

$.fn.extend({

changeBgColor: function(value){

$(this).css("background-color", value);

}

});

$("input[type='button']").click(function(){

$("p").changeBgColor("yellow");

});

});

</script>