

در ادامه مطلب قبلی آموزش [\(jQuery\) جی کوئری #2](#) به ادامه بحث می‌پردازیم.

## انتخاب عناصر صفحه

در پستهای قبل ( [^](#) و [^](#) ) با بسیاری از توانایی‌ها و کارکردهای jQuery شامل توانایی‌های آن برای انتخاب عناصر موجود در صفحه تا تعریف توابع جدید و استفاده از آنها به محض آماده شدن صفحه آشنا شدیم.

در این پست و پست بعدی توضیحات تکمیلی در خصوص دو مورد از توانایی‌های jQuery و البته تابع (\$) خواهیم داشت که مورد اول، انتخاب عناصر صفحه با استفاده از انتخاب کننده‌ها و مورد دوم ایجاد عناصر جدید می‌باشد.

در بسیاری از مواقع برای تعامل با صفحه اینترنتی نیاز به تغییر دادن بخشی از یکی از اشیا موجود در صفحه داریم. اما پیش از آنکه قادر باشیم آنها را تغییر دهیم، ابتدا باید با استفاده از مکانیزمی شی مورد نظر را مشخص و سپس آن را انتخاب کنیم تا پس از آن قادر به اعمال تغییری در آن باشیم. بنابراین اجازه دهید تا به یک بررسی عمیق از راه‌های مختلف انتخاب عناصر صفحه و ایجاد تغییر در آنها بپردازیم.

### 1- انتخاب عناصر صفحه برای ایجاد تغییر

اولین قدم برای استفاده از هر گونه تابع jQuery، مشخص کردن و انتخاب عناصری است که می‌خواهیم تابع روی آن عناصر اعمال شود. گاهی اوقات انتخاب این مجموعه عناصر با یک توضیح ساده مشخص می‌شود، برای مثال "تمام عناصر پاراگراف موجود در صفحه". اما گاهی اوقات مشخص کردن این مجموعه نیاز به توضیح پیچیده‌تری دارد، برای مثال "تمام عناصر لیست در صفحه که دارای کلاس listElement هستند و لینکی دارند که اولین عضو آن لیست می‌باشد". خوشبختانه jQuery یک مکانیزم بسیار قوی و قدرتمند ارائه کرده است که انتخاب هر عنصری از صفحه را به سادگی امکان پذیر می‌سازد. انتخاب کننده‌های jQuery از ساختار مربوط به CSS استفاده می‌کنند، بنابراین ممکن است شما هم اکنون با تعداد زیادی از آنها آشنا باشید. در ادامه شمار بیشتر و قدرتمندتری خواهید آموخت. برای درک بهتر شما از مطالب مربوط به بخش انتخاب کننده‌ها، یک مثال آماده مختص به این مبحث، در قالب یک صفحه اینترنتی، را در [فایل صفحه کارگاهی](#) قرار داده ایم، این فایل در آدرس [chapter2/1ab.selector.htm](http://chapter2/1ab.selector.htm) قابل دسترسی می‌باشد. این مثال از پیش آماده و کامل (نوشته شده توسط نویسنده کتاب)، این امکان را به شما می‌دهد تا با وارد کردن یک رشته، به عنوان پارامتر انتخاب کننده، در همان زمان عنصر انتخاب کننده در صفحه را رویت کنید. زمانی که این صفحه را اجرا می‌کنید تصویری مانند زیر ظاهر خواهد شد.

**Selectors Lab**

**Selector**

Type a selector into the text field below and click the Apply button.

Selector:

Apply

**DOM Sample**

Some images:

This is a <div> with an id of aomebäv  
Hello  
Goodbye

- jQuery supports
  - CSS1
  - CSS2
  - CSS3
  - Basic XPath
- jQuery also supports
  - Custom selectors
  - Form selectors

Language	Type	Invented
Java	Static	1995
Ruby	Dynamic	1993
Smalltalk	Dynamic	1972
C++	Static	1983

Text:

Radio group: ☐ A ☐ B ☐ C

Checkboxes: ☐ 1 ☐ 2 ☐ 3 ☐ 4

Submit

**HTML for DOM Sample**

```
<div><label>Some images:</label></div>
<div>
  
```

برای درک بهتر مطالب این سلسله پست‌ها می‌توانید فایل‌های کتاب را از [آدرس اصلی](#) آن یا از [این آدرس در همین سایت](#) دانلود نمایید.

این صفحه سه پنجره مجزا دارد. در پنجره سمت چپ، یک `textBox` و یک دکمه دیده می‌شود، که با وارد کردن یک انتخاب کننده در `textBox` و فشردن دکمه، عنصر مورد نظر در پنجره سمت راست انتخاب می‌شود. برای شروع در `textBox` عبارت `li` را بنویسید و دکمه `Apply` را کلیک کنید. با انجام این عمل تصویر زیر باید خروجی شما باشد. می‌توانید حالت‌های دیگر را خودتان امتحان کنید.

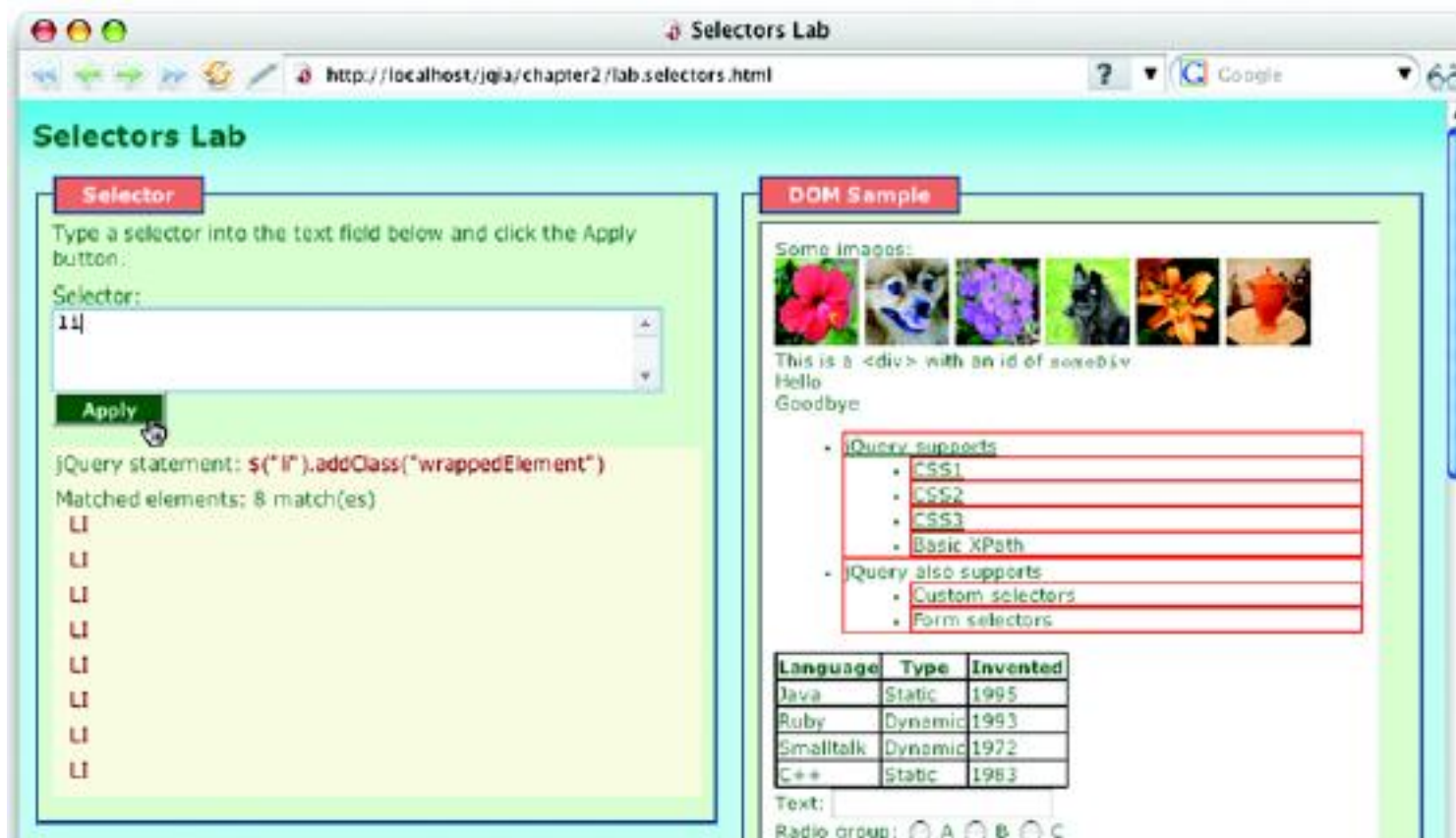


Figure 2.2 A selector value of li matches all <li> elements when applied as shown by the display results.

### 1-1- استفاده از انتخاب کننده‌های ابتدایی CSS

برنامه نویسان وب برای اعمال فرمت‌های ظاهری گوناگون به بخش‌ها و عناصر مختلف یک صفحه اینترنتی، از یک راه بسیار ساده، در عین حال قدرتمند و کارا استفاده می‌کنند که در تمام مرورگرهای مختلف نیز جوابگو باشد. این انتخاب کننده‌ها عناصر را بر اساس نام شناسه آنها، نام کلاس و یا ساختار سلسله مراتبی موجود در صفحه انتخاب می‌کنند.

در زیر به معرفی چند نمونه از این انتخاب کننده‌های ساده CSS می‌پردازیم:

a : تمام عناصر <a> را انتخاب می‌کند.

#specialID : عنصری را که دارای ID با عنوان specialID باشد انتخاب می‌کند.

.specialClass : عنصری را که دارای کلاس specialClass هستند انتخاب می‌کند.

a#specialID.specialClass : این عبارت عنصری را انتخاب می‌کند که شناسه آن specialID باشد، به شرط آنکه این عنصر <a> باشد و دارای کلاس specialClass نیز باشد را انتخاب می‌کند.

p a.specialClass : تمام عناصر لینک (<a>) را که دارای کلاس specialClass باشند و درون یک عنصر پاراگراف (<p>) قرار گرفته باشند را انتخاب می‌کند.

این انتخاب کننده‌ها شاید ساده به نظر برسند، اما در بسیاری از مواقع پاسخگوی ما می‌باشند؛ به علاوه آنه که با ادغام این انتخاب کننده‌های ساده، ما می‌توانیم انتخاب کننده‌های پیچیده‌تر و تخصصی‌تر ایجاد کنیم.

نکته مثبت در مورد انتخاب کننده‌های CSS این است که از همین انتخاب کننده‌ها می‌توانیم در jQuery نیز استفاده کنیم. برای این کار تنها کافیست انتخاب کننده مورد نظر را به تابع (\$) ارسال کنیم. در زیر یک نمونه را مشاهده می‌کنید:

```
$("#p a.specialClass")
```

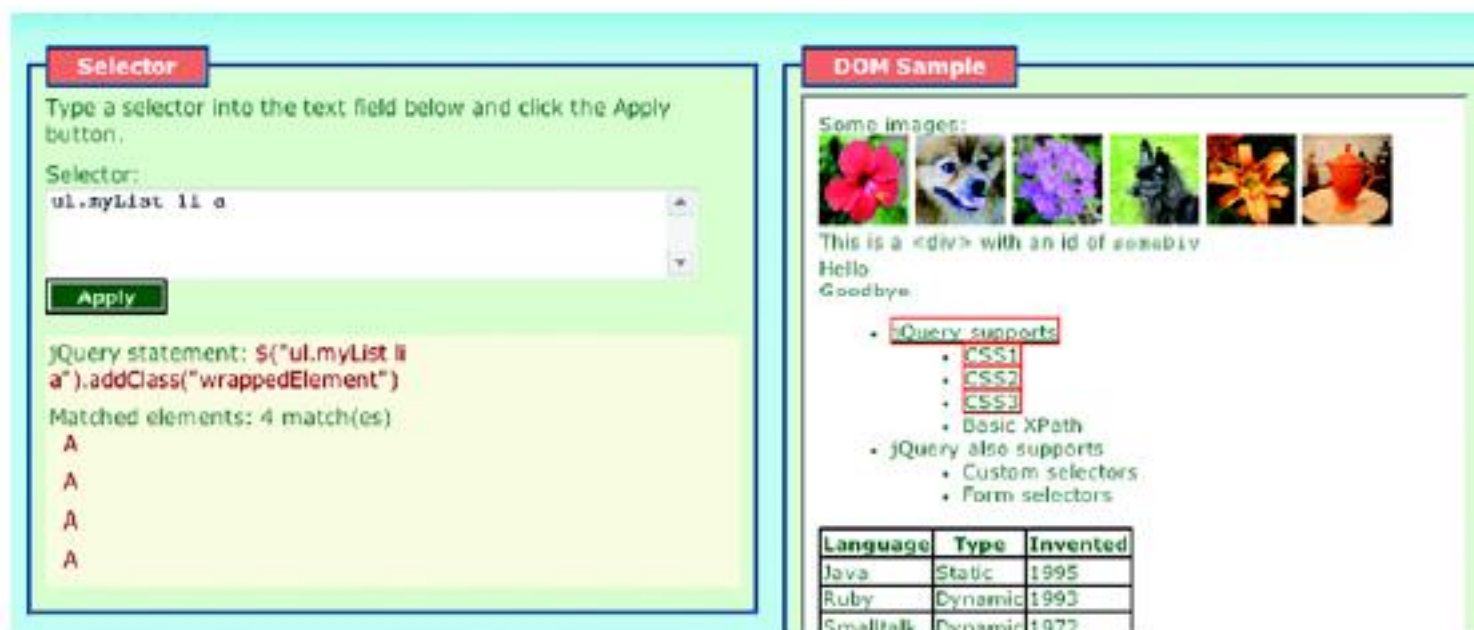
به جز چند مورد خاص که استثنا وجود دارد، [CSS3](#) و jQuery کاملاً با هم سازگاری دارند. بنابراین انتخاب عناصر به این شکل طبیعی خواهد بود. به عبارتی دیگر هر عنصر که از این طریق توسط CSS انتخاب شود، همان انتخاب حاصل انتخاب کننده jQuery نیز خواهد بود. اما باید به این نکته توجه داشت که jQuery وابسته به CSS نیست و اگر مرورگری پیاده سازی استاندارد برای CSS نداشته باشد، انتخاب کننده jQuery به مشکل بر نمی خورد، بلکه jQuery انتخاب خود را به درستی انجام می دهد، چرا که jQuery از قوانین استاندارد [W3C](#) تبعیت می کند.

## 2-1- استفاده از انتخاب کننده های فرزند (Child) ، نگهدارنده (Container) و صفت (Attribute)

برای انتخاب کننده های پیشرفته تر، jQuery از جدیدترین مرورگرهایی که CSS را پشتیبانی می کنند، استفاده می کند که می توان به Mozilla Firefox, Internet Explorer 7, Safari و سایر مرورگرهای پیشرفته (مدرن) اشاره کرد. این انتخاب کننده های پیشرفته شما را قادر می سازند تا مستقیماً فرزند یک عنصر را انتخاب کنید و یا از ساختار سلسله مراتبی عناصر صفحه، مستقیماً به عنصر مورد نظر دسترسی داشته باشید و یا حتی تمام عناصری که یک صفت خاص را شامل می شوند، انتخاب کنید. گاهی اوقات انتخاب فرزندی از یک شی برای ما مطلوب است. برای مثال ممکن است ما به چند مورد از یک لیست احتیاج داشته باشیم، نه یک زیر مجموعه ای از آن لیست. به قطعه کد زیر که از صفحه کارگاهی این پست گرفته شده است دقت نمایید:

```
<ul>
  <li><a href="http://jquery.com">jQuery supports</a>
    <ul>
      <li><a href="css1">CSS1</a></li>
      <li><a href="css2">CSS2</a></li>
      <li><a href="css3">CSS3</a></li>
      <li>Basic XPath</li>
    </ul>
  </li>
  <li>jQuery also supports
    <ul>
      <li>Custom selectors</li>
      <li>Form selectors</li>
    </ul>
  </li>
</ul>
```

حال فرض کنید از این ساختار، لینک وب سایت jQuery مد نظر ماست و این کار بدون انتخاب سایر لینک های مربوط به CSS مطلوب است. اگر بخواهیم از دستورهای انتخاب کننده CSS استفاده کنیم، دستوری به شکل `ul.myList li a` خواهیم داشت. اما متأسفانه این دستور تمام لینک های این ساختار را انتخاب میکند، زیرا همه آنها لینک هایی در عنصر `li` می باشند. با نوشتن این دستور در صفحه کارگاهی خروجی به شکل زیر خواهد بود:



**Figure 2.3** All anchor tags that are descendents, at any depth, of an <li> element are selected

ul.myList li a

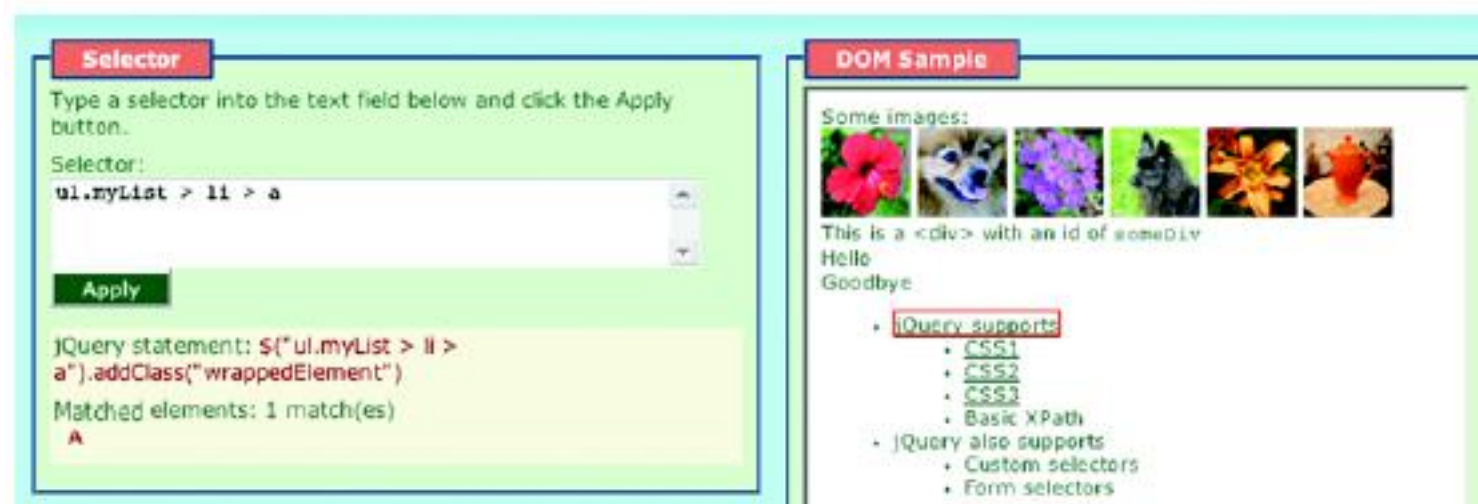
راه حل مناسب برای انتخاب چنین حالتی استفاده از **انتخاب فرزند** می باشد که به این منظور Parent (والد) و Child (فرزند)، به وسیله یک کاراکتر > از یکدیگر جدا می شوند:

p > a

این دستور تنها لینک (<a>) هایی را بر می گرداند که فرزند مستقیم یک عنصر <p> می باشند. بنابراین اگر در یک <p> لینکی در عنصر <span> معرفی شده باشد، این لینک انتخاب نمی شود، چرا که فرزند مستقیم <p> به حساب نمی آید. در مورد مثال لینک های موجود در لیست، می توانیم دستور زیر را به منظور انتخاب لینک مورد نظرمان استفاده کنیم:

ul.myList > li > a

دستور انتخاب فوق از میان عناصر <ul>، عنصری را که دارای کلاس myList می باشد، انتخاب می کند و پس از آن لینک هایی (<a>) که فرزند مستقیم گزینه های آن هستند، برگردانده می شوند. همانگونه که در شکل زیر مشاهده می کنید لینک های زیرمجموعه عنصر <ul> انتخاب نمی شوند، زیرا فرزند مستقیم این عنصر محسوب نمی شوند.



**Figure 2.4** With the selector `ul.myList > li > a`, only the direct children of parent nodes are matched.

انتخاب کننده‌های صفت نیز بسیار قدرتمند می‌باشند و ما را توانا تر می‌سازند، فرض کنید برای منظوری خاص قصد دارید به تمام لینک‌های موجود در صفحه که به مکانی خارج از این وب سایت اشاره دارند، رفتاری را اضافه کنید (مثلاً مانند همین سایت به کنار آنها یک آیکن اضافه نمایید). فرض کنید این کد (کد موجود در مثال کارگاهی) را در صفحه خود دارید:

```
<li><a href="http://jquery.com">jQuery supports</a>
  <ul>
    <li><a href="css1">CSS1</a></li>
    <li><a href="css2">CSS2</a></li>
    <li><a href="css3">CSS3</a></li>
    <li>Basic XPath</li>
  </ul>
</li>
```

موردی که یک لینک با اشاره به وب سایت خارجی را از سایر لینک‌ها متمایز می‌سازد، شروع شدن مقدار صفت href آن با `http://` می‌باشد. انتخاب لینک‌هایی که مقدار href آنها با `http://` آغاز می‌شود، به سهولت و از طریق دستور زیر صورت می‌پذیرد:

```
a[href^=http://]
```

این دستور باعث انتخاب تمام لینک‌هایی که مقدار صفت href آنها دقیقاً با `http://` آغاز می‌شود، می‌گردد. علامت `^` موجب می‌شود تا بررسی، لزوماً از ابتدای مقادیر صورت پذیرد و از آنجا که استفاده از این کاراکتر در سایر عبارات منظم به همین منظور صورت می‌پذیرد، به خاطر سپردن آن دشوار نخواهد بود. می‌توانید این کد را در صفحه کارگاهی تست کنید. راهای دیگری برای استفاده از انتخاب کننده‌های صفت وجود دارد.

```
form[method]
```

این دستور تمام عناصر `<form>` را که یک صفت `method` دارند را انتخاب می‌کند.

```
input[type=text]
```



این انتخاب کننده تمام عناصر input را که type آنها برابر text باشد انتخاب می‌کند.  
دستور زیر مثالی دیگر برای بررسی یک مقدار بر اساس کاراکترهای نخست آن می‌باشد:

```
div[title^=my]
```

همانطور که از دستور فوق بر می‌آید، عناصر div که مقدار title آنها با رشته my آغاز می‌شود، هدف این انتخاب کننده خواهد بود. اما اگر بخواهیم تنها بر اساس کاراکترهای انتهایی انتخابی انجام دهیم، دستور مناسب چه خواهد بود؟ برای چنین منظوری مانند زیر عمل می‌کنیم:

```
a[href$=.pdf]
```

این دستور کاربرد زیادی برای شناسایی لنک‌های اشاره کننده به فایل‌های pdf دارد. ساختار زیر نیز زمانی استفاده می‌شود که یک عبارت منظم در جایی از یک صفت قرار گرفته باشد، خواه این عبارت از کاراکتر دوم آغاز شده باشد و یا از هر جای دیگر.

```
a[href*=jquery.com]
```

همانگونه که انتظار می‌رود این انتخاب کننده، تمام لینک‌هایی که به وب سایت jQuery اشاره دارند را برمی‌گرداند. فراتر از خصوصیات، بعضی مواقع ما می‌خواهیم بررسی کنیم که آیا یک عنصر شامل عنصر دیگری هست یا خیر. در مثال‌های قبلی فرض کنید ما می‌خواهیم بدانیم که آیا یک li شامل a هست یا خیر، jQuery با استفاده از انتخاب کننده‌های Containerها این را پشتیبانی می‌کند:

```
li:has(a)
```

این انتخاب کننده همه li هایی را برمی‌گرداند که شامل لینک (<a>) هستند. دقت کنید که این انتخاب گر مانند li a نیست، انتخاب گر دوم تمامی لینک‌هایی را که در li هستند بر می‌گرداند اما دستور بالا li هایی را بر می‌گرداند که دارای لینک (<a>) هستند.

تصویر زیر انتخاب گرهایی را نشان می‌دهد که ما می‌توانیم در jQuery استفاده نماییم.

**Table 2.1 The basic CSS Selectors supported by jQuery**

Selector	Description
*	Matches any element.
E	Matches all element with tag name E.
E F	Matches all elements with tag name F that are descendents of E.
E>F	Matches all elements with tag name F that are direct children of E.
E+F	Matches all elements F immediately preceded by sibling E.
E-F	Matches all elements F preceded by any sibling E.
E:has(F)	Matches all elements with tag name E that have at least one descendent with tag name F.
E.C	Matches all elements E with class name C. Omitting E is the same as *.C.
E#I	Matches element E with id of I. Omitting E is the same as *#I.
E[A]	Matches all elements E with attribute A of any value.
E[A=V]	Matches all elements E with attribute A whose value is exactly v.
E[A^=V]	Matches all elements E with attribute A whose value begins with v.
E[A\$=V]	Matches all elements E with attribute A whose value ends with v.
E[A*=V]	Matches all elements E with attribute A whose value contains v.

انشالله در پست‌های بعدی ادامه مباحث را بررسی خواهد شد.