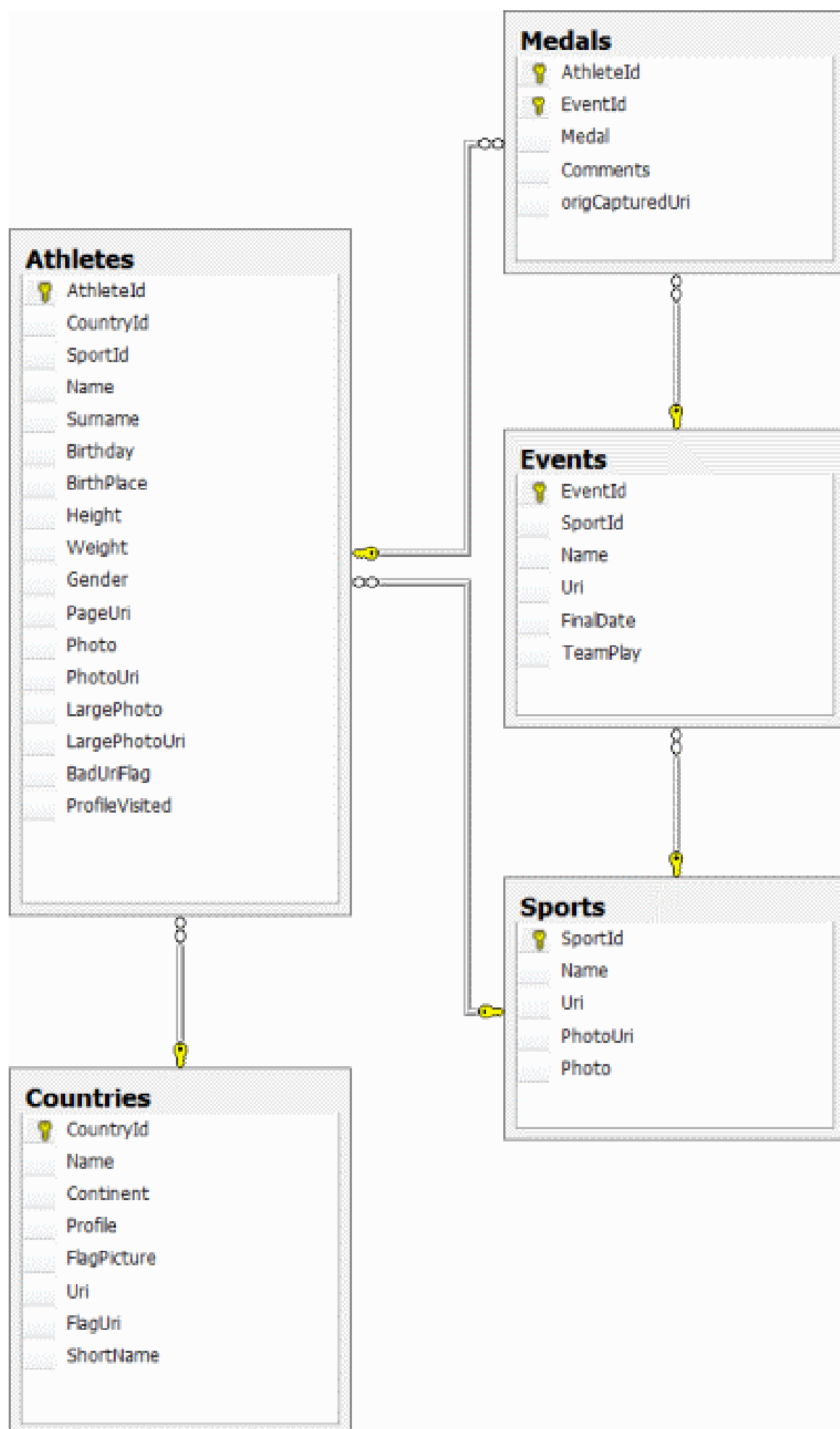


چند مدت پیش موقعی که تب المپیک بود و جدول <http://www.london2012.com/medals/medal-count> رو زیاد نگاه می‌کردم به نظرم رسید که کاشکی به اطلاعاتی مثل اینکه چند نفر از مدال آورها خانم و یا آقا هستند و یا اینکه در روزهای مختلف تعداد مدال‌ها چطور توزیع می‌شند و بشه با یک jQuery UI Slider روزهای مختلف رو انتخاب کرد و جدول رو دید. برای این کار اولین چیزی که لازم بود دریافت و ذخیره اطلاعات بود که من برای این کار از Entity framework 4.1 Database- first و کتابخانه HAP - htmlagilitypack استفاده کردم. طراحی دیتابیس نهایی به این صورت شد



خوب در تلاش اول و مبتدیانه و بدون استفاده از این کتابخانه مفید چون اکثر صفحات وب XHTML نیستند و بالاخره چند تگ درست بسته نشده دارند و شما اگر بخواهید در آبجکت XmlDocument این htmlهای به ظاهر سالم رو لود کنید فوراً با استثنای زیر مواجه می‌شوید

```
XmlException Was unhandled  
The 'img' start tag on line 1 position 1604 does not match the end tag of 'a'. Line 1, position 1766
```

راه حل ساده اینه که این کتابخونه رو با کمک NuGet نصب کنید

```
PM> Install-Package HtmlAgilityPack
```

و از اینجا به بعد با کدی مثل این میتونید از کلاس XmlDocument و مشابه XmlDocument ولی بدون ارور استفاده کنید. مثلاً با کد زیر میشه تاریخ تولد یک ورزشکار رو بدست آورد. توابع دیگه ای که خیلی جاها میتونه بدرد خورد GetAttributeValue و ChildNodes هست که یک نمونه نحوه استفادشو در ادامه میبینید

```
HtmlDocument xhtml = Crawler.GetXHtmlFromUri("http://www.london2012.com/athlete/hadadi-ehsan-1077408/");  
HtmlNode tempNode = xhtml.DocumentNode.SelectSingleNode("//table[@class='athleteBio']/tbody/tr[4]");
```

```
string temp = tempNode.FirstChild.FirstChild.InnerText.Replace("&nbsp;", "").Trim();  
athlete.Birthday = DateTime.Parse(temp.Substring(0, 10), new CultureInfo("en-GB"));
```

```
tempNode = xhtml.DocumentNode.SelectSingleNode("//div[@class='athletePhotoMedals']/div/div/img");  
athlete.LargePhotoUri = tempNode.GetAttributeValue("src", "");
```

البته تابع GetXHtmlFromUri رو جدا باید با کمک HttpRequest بنویسید و توی خود HAP متاسفانه چنین تابعی توکار نشده نکته اصلی هم پیدا کردن محل دقیق اطلاعاته که با ابزاری مثل Firebug خیلی راحت تر میشه این کارو انجام داد. کافیه روی تاریخ تولد راست کلیک و inspect element by Firebug رو بزنید و حالا اگر تویه dom روی هر المنت html نگه دارید بهتون XPath کامل رو میده که میتونید تویه تابع DocumentNode.SelectSingleNode ارزش استفاده کنید.

Ehsan Hadadi

[Overview](#) | [Events and results](#) |

[illegible]

برای درک بهتر XPath هم این 2 تا صفحه `xpath_syntax` و `xpath_examples` خیلی میتونه کمکتون بکنه.

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳:۵۳ ۱۳۹۱/۰۶/۰۴

این روش برای استخراج مطالب همین سایت خیلی مفیده!

نویسنده: محسن کریمی
تاریخ: ۱۳:۵۷ ۱۳۹۱/۰۶/۰۴

با تشکر از مطلبتون
نکته: این Htmlagilitypack متاسفانه با صفحات فارسی و UTF8 مشکلات زیادی داره و واقعا همیشه ارزش استفاده کرد.

نویسنده: جمال
تاریخ: ۱۴:۱۹ ۱۳۹۱/۰۶/۰۴

Crawler.GetXHtmlFromUri
شی Crawler از چه نوعیه؟ موقع اجرا خطا میگیره؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۸ ۱۳۹۱/۰۶/۰۴

شروع کار به این صورت هم می تواند باشد:

```
var doc = new HtmlDocument
{
    OptionCheckSyntax = true,
    OptionFixNestedTags = true,
    OptionAutoCloseOnEnd = true,
    OptionDefaultStreamEncoding = Encoding.UTF8
};
doc.LoadHtml(content);
```

OptionDefaultStreamEncoding رو به UTF8 تنظیم کنید.

نویسنده: محسن کریمی
تاریخ: ۱۵:۴۳ ۱۳۹۱/۰۶/۰۴

با تشکر

ولی طبق تجربه خود من کد بالا هم کمک نمی کنه و با تنظیم OptionDefaultStreamEncoding به UTF8 مشکل حل نمیشه ولی یه راه که قبلا من پیدا کرده بودم تو خوندن کد صفحات اینه که شما صفحات رو به صورت استریم دریافت کرده بعد توسط متد LoadHtml بخونید به این صورت مشکل برطرف میشه! (البته این تو سایت فارسی که من قصد خوندشو داشتم، بود با سایتهای دیگه تست نکردم)

```
var request = (HttpWebRequest)WebRequest.Create("آدرس سایت");
request.Method = "GET";
using (var response = (HttpWebResponse)request.GetResponse())
{
    using (var stream = response.GetResponseStream())
    {
        htmlDoc.Load(stream, Encoding.UTF8);
    }
}
```

نویسنده:

وحید نصیری

تاریخ:

۱۶:۴۹ ۱۳۹۱/۰۶/۰۴

بستگی داره content نظر قبلی رو به چه فرمتی (چه Encoding ایی) از وب دریافت کردید. مابقی آن توسط این کتابخانه بدون مشکل پردازش می‌شود.

```
using System.Net;
//...
var content = new WebClient { Encoding = Encoding.UTF8 }.DownloadString(url);
```

نویسنده:

ایمان عبیدی

تاریخ:

۲۰:۴۶ ۱۳۹۱/۰۶/۰۴

Crawler همونطور که در متن هم نوشته شده دست سازه و مهم نیست و تابع GetXHtmlFromUri میتونه مثل نمونه زیر باشه و دقت کنید خالی نبودن UseAgent خيله مهمه وگرنه ارور (409) Conflict The remote server returned an error: رو میده. من با همین تابع یک سایت فارسی رو چک کردم و اروری نداد و متن فارسی قابل کوئری گرفتن بود. کامل تر و با ارور هندلینگ بهترش رو میتونید در برنامه مفید [plrip](#) آقای وحید نصیری ببینید

```
private static HtmlDocument GetXHtmlFromUri(string uri) {
    HtmlDocument htmlDoc = new HtmlDocument()
    {
        OptionCheckSyntax = true,
        OptionFixNestedTags = true,
        OptionAutoCloseOnEnd = true,
        OptionDefaultStreamEncoding = Encoding.UTF8
    };

    var request = (HttpWebRequest)WebRequest.Create(uri);
    request.Method = "GET";

    //important
    request.UserAgent = "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)";
    request.Accept = "text/html";
    requestAutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;

    using (var response = request.GetResponse() as HttpWebResponse)
    {
        using (var stream = response.GetResponseStream())
        {
            htmlDoc.Load(stream, Encoding.UTF8);
        }
    }
    return htmlDoc;
}
```

اینم روش دوم که بازم UserAgent باید اضافه بشه

```
private static HtmlDocument GetXHtmlFromUri2(string uri) {
    WebClient client = new WebClient() { Encoding = Encoding.UTF8 };
    client.Headers.Add("user-agent", "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)");

    HtmlDocument htmlDoc = new HtmlDocument()
    {
        OptionCheckSyntax = true,
        OptionFixNestedTags = true,
        OptionAutoCloseOnEnd = true,
        OptionDefaultStreamEncoding = Encoding.UTF8
    };

    htmlDoc.LoadHtml(client.DownloadString(uri));

    return htmlDoc;
}
```

نویسنده: افشار محبی
تاریخ: ۹:۳۷ ۱۳۹۱/۰۶/۰۵

من هم از این ابزار در کارهایم استفاده کرده‌ام. خیلی خوب جواب می‌دهد.

نویسنده: ایمان عبیدی
تاریخ: ۰:۴ ۱۳۹۱/۰۶/۰۶

برای مشاهده نتایج بدست آمده رده بندی المپیک 2012 لندن به همراه اطلاعات جنسیت مدال گیرها و همچنین وضعیت جدول در روز هایه مختلف میتونید به لینک هایه زیر مراجعه کنید.

تویه این صفحات از پلاگین tableSorter و یکم جاوا اسکریپت هم در لینک اول برای کش کردن اطلاعات json استفاده کردم .
<http://olympics2012.iabidi.ir/Home/DailyMedals>
<http://olympics2012.iabidi.ir/Home/RankingsFull>

نویسنده: پریسا
تاریخ: ۲۰:۵۸ ۱۳۹۲/۰۲/۰۲

چرا وقتی XPath از Firebug استفاده می‌کنم با استفاده از SelectSingleNode جواب دقیقی نمیده یا هیچی نیاره

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۴ ۱۳۹۲/۰۲/۰۳

علت این است که html ایی که در فایرباگ بررسی میشه عموما به دلیل یک سری از نرمال سازی‌ها توسط موتور فایرفاکس و همچنین خودش، با html اصلی یک سایت متفاوت است. به همین جهت XPath استخراجی از آن روی سایت اصلی کار نخواهد کرد.
[یک برنامه کمکی](#) برای یافتن XPathها به همان نحوی که هستند.

نویسنده: mahsan
تاریخ: ۱۰:۳۸ ۱۳۹۲/۰۳/۲۳

این متد رو در چه صفحه ای باید بنویسم؟ آیا باید در همون صفحه ای که میخوام اطلاعات لود بشه بنویسم؟
uri مقدارش را باید داخل تابع بگیره؟ در page load فرمم چی بنویسم؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۲۹ ۱۳۹۲/۰۳/۲۳

- تفاوتی نمی‌کنه [کجا فراخوانی بشه](#) ؛ در page load یا در یک روال رخداد گردان کلیک و یا در یک سرویس مستقل.
- بهتره نتیجه رو بعد از فراخوانی برای مدتی کش کنی، تا هربار اطلاعات از وب درخواست نشود.

نویسنده: mahsan
تاریخ: ۱۲:۱۰ ۱۳۹۲/۰۳/۲۳

بخشین که دوباره مزاحمتون شدم: وقتی من کد بالا رو در page load کپی میکنم زیر بعضی کلمات مانند Crawler , Birthday, LargePhotoUri, GetAttributeValue , Parse, XmlNode, HtmlNode, Parse, Birthday, LargePhotoUri, GetAttributeValue یک سند html که من باید ایجادش میکردم؟ فقط همین کدها رو بنویسم؟ جواب میگیرم با باید چیز دیگه ای هم اضافه بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۳ ۱۲:۴۱

- شما باید از طریق نیوگت با دستور Install-Package HtmlAgilityPack این بسته رو نصب کنید. یا اینکه فایل DLL اون رو [از سایتش دریافت](#) و به ارجاعات پروژه اضافه کنید.
- کدهای کلاس Crawler چند کامنت بالاتر ارسال شدن. تابع GetXHtmlFromUri که [ملاحظه می کنید](#).
- مواردی مانند Birthday, LargePhotoUri یک سری متغیر هستند که از طرف نویسنده مقاله تعریف شدن. مهم نیستند. حذفشون کنید.
- [یک مثال دیگر در مورد استفاده از کتابخانه HtmlAgilityPack با کد قابل دریافت](#).

نویسنده: mahsan
تاریخ: ۱۳۹۲/۰۳/۲۵ ۹:۰۶

سلام ممنون که جوابم رو دادین
من کلاسی بنام Crawler ایجاد کردم ولی حالا زیر

OptionDefaultStreamEncoding

OptionAutoCloseOnEnd OptionFixNestedTags OptionCheckSyntax

و

LoadHtml

خط قرمز میکشه: (لطف میکنید بگید دلیلش چیه و باید چیکار کنم؟
و در فرم هم زیر این کلمه GetXHtmlFromUri
مینویسه
generate metod stub for 'GetXHtmlFromUri in "crawel

وقتی من متد GetXHtmlFromUri را در کلاس crawl تعریف کردم چرا باید این پیغام رو بده ؟ آیا باید این گزینه رو انتخاب کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۵ ۹:۴۳

پیشنهاد من این است که یک دوره سی شارپ مقدماتی رو بگذرونید. با مباحثی مانند نحوه تعریف فضای نام و روش فراخوانی یک متد استاتیک از کلاس متناظر با آن آشنا شوید.
[یک دوره خوب مقدماتی سی شارپ](#)

نویسنده: ایمان توکلی
تاریخ: ۱۳۹۲/۱۱/۲۶ ۱۷:۱۰

با سپاس
در صورتی که مقداری در querystring مربوط به صفحه اضافه شود و در هر درخواست این مقدار تغییر کند چطور می توان صفحه را خواند مثال: <http://website.com?log=1731004>
سایت برای هر بازدید یک لاگ جدید می نویسد یعنی هر درخواست جدید در صورتی که لاگ معتبر نباشد به صفحه دیگر ارسال می کند. حال اطلاعات این صفحه را چطور می توان خواند ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۲۶ ۱۷:۲۱

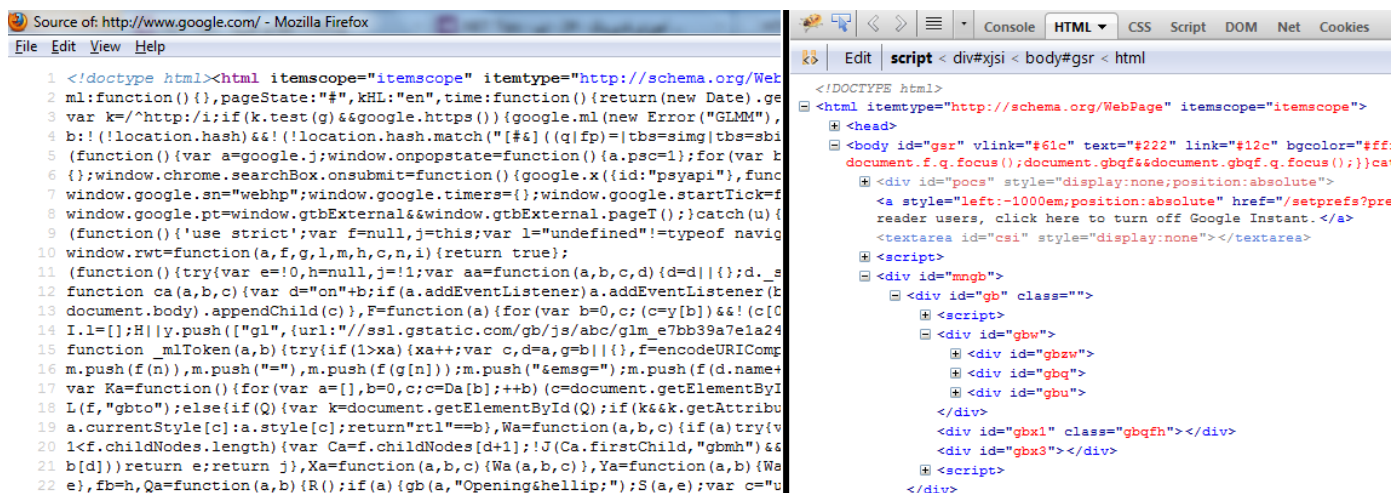
این نوع مسایل رو نمی‌تونید با HtmlAgilityPack حل کنید. فقط کارش آنالیز اطلاعات ثابتی هست که بهش می‌دید و نهایتاً خواندن نودهای HTML نهایی. موردی که عنوان کردید نیاز به آنالیز همزمان هدرهای دریافتی به همراه جاوا اسکریپت سایت مورد نظر داره.

در این سری از [مقالات آموزش FireBug](#) ، به صورت ترتیبی پیش رفتیم و ابتدا توضیحات تقریباً مفصلی در مورد [پنل Console](#) دادیم.

اکنون به پرکاربردترین بخش آن ، یعنی پنل HTML می‌رسیم.

محتویاتی که در این پنل نمایش داده می‌شود ، کدهای صفحه‌ی جاری ، بصورت زنده است و با چیزی که از سمت سرور به مرورگر ارسال می‌شود متفاوت است ، تگ‌های HTML بصورت درختی مرتب شده اند و رنگی نمایش داده می‌شوند و امکان ویرایش محتوا ، ویرایش استایل ، مشاهده‌ی آرایش تگ‌ها بصورت بصری و ... وجود دارد.

نگاهی به کدهای ارسال شده به مرورگر در آدرس [Google.com](http://www.google.com) و نحوه‌ی نمایش آن در فایرباگ را ملاحظه بفرمایید.
(کدهای ارسال شده سمت چپ - نمایش در فایرباگ سمت راست)



این پنل به سه بخش اصلی تقسیم می‌شود :

بخش اصلی یا NodeView که محتوای صفحه را بصورت درختی و مرتب و رنگی نمایش می‌دهد.

Panel Toolbar که در بالای پنل قرار دارد.

Side Panels که شامل پنل‌های DOM , Layout , Computed , Style می‌شود.

که به ترتیب برای نمایش و ویرایش استایل‌ها ، مشاهده استایل‌های محاسبه شده ، مشاهده Layout یا آرایش و نمایش اطلاعات DOM تگ انتخاب شده در NodeView است.

در این مقاله با دو بخش NodeView و Panel Toolbar ، و در مقاله‌ی بعد با پنل‌های سمت راست یعنی Side Panels آشنا می‌شویم.

ویرایش HTML

هر تگ HTML از یک سری Attribute تشکیل می‌شود که در فایرباگ نام ویژگی بصورت آبی تیره و مقدار آن با رنگ قرمز مشخص شده است. برای ویرایش هریک از آن‌ها کافیهست برویش کلیک کنید تا آن مقدار در یک باکس ویرایش به نمایش دربیاید. با ویرایش مقدار ، این تغییر در لحظه بروی صفحه اعمال می‌شود.

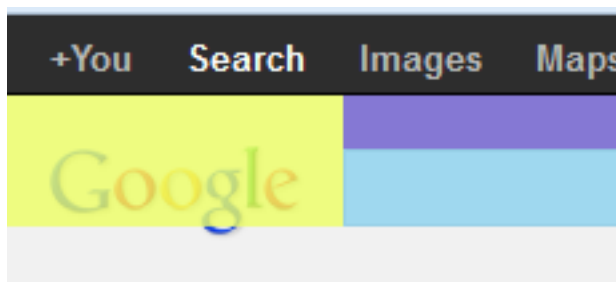


برای اضافه کردن یک Attribute جدید به تگ هم بروی تگ مورد نظر راست کلیک می‌کنید و سپس گزینه‌ی New Attribute را انتخاب می‌کنید. ابتدا نام ویژگی ، یک Enter ، سپس مقدار را وارد می‌کنید. با زدن کلیدهای Enter متوالی ، می‌توانید به وارد کردن ویژگی‌ها ادامه دهید.

برای ویرایش کردن یک تگ و تگ‌های فرزندش ، بروی تگ مورد نظر کلیک کنید تا به حالت انتخاب دربیاید ، سپس بروی دکمه‌ی Edit در بالای پنل کلیک کنید. در نهایت بعد از انجام ویرایش ، مجدداً بروی دکمه‌ی Edit کلیک کنید.

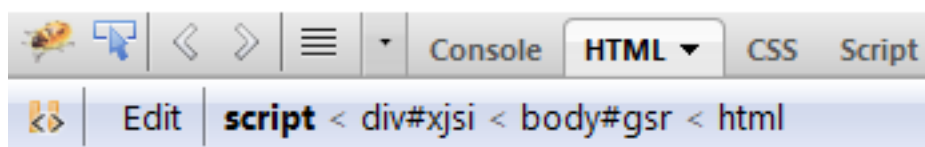
Node View

NodeView نام بخش اصلی پنل HTML است که محتویات صفحه را بصورت مرتب شده و درختی نمایش می‌دهد. تگ (Node) هایی که دارای فرزند می‌باشند ، یک علامت + یا - در کنارشان وجود دارد که با کلیک بروی آن می‌توانید آن تگ را باز/بسته کنید. همچنین این کار با کلیدهای + و - یا Right Arrow و Left Arrow از روی کیبورد هم قابل انجام است. برای باز کردن یک لینک در این قسمت هم از ترکیب Ctrl و کلیک موس کمک بگیرید. در نهایت زمانی که موس را بروی یک تگ قرار می‌دهید ، محیطی که توسط آن تگ در صفحه اشغال شده است بصورت رنگی مشخص می‌شود. که رنگ زرد به معنی محیط margin ، رنگ آبی تیره به معنی محیط padding و رنگ آبی روشن هم به معنی محیط محتوا است.



Panel Toolbar

این قسمت در بالای پنل HTML قرار دارد که گزینه‌های زیر را دارا می‌باشد:



Break On Mutate

این دکمه امکان توقف کد JavaScript ای که سعی در ویرایش محتوای صفحه را دارد ، می‌دهد. زمانی که FireBug تشخیص دهد که کدی سعی در ویرایش محتوا دارد ، شما را به خط مورد نظر از کد ، در پنل Script منتقل می‌کند.

Edit


این دکمه برای ویرایش مستقیم محتوای یک تگ بکار می‌رود نکته‌ی جالب در ویرایش محتوا در فایرباگ این است که تغییرات در لحظه اعمال می‌شوند و نیاز به عمل بروزرسانی جداگانه نیست. برای مثال در همین قسمت Edit ، با هر ویرایش محتوا ، تغییرات در لحظه اعمال می‌شوند.

Element Path

زمانی که یک تگ را در FireBug انتخاب می‌کنید ، لیستی از تگ‌ها که از تگ جاری شروع و به تگ ریشه ختم می‌شود ، نمایش داده می‌شود که با کلیک بروی هر کدام ، همان به عنوان تگ فعلی یا انتخاب شده تعیین می‌شود. نتیجه‌ی عملیاتی که بروی این تگ‌های انجام می‌دهید (حرکت موس و راست کلیک کردن) معادل همان عملیات بروی تگ‌های نمایش داده شده در قسمت اصلی (NodeView) است.

Options Menu

هر تب یا پنل در فایرباگ دارای یک سری تنظیمات است که Options Menu نام دارد. تب HTML هم دارای یک سری تنظیمات است که دانش‌تان آنها بسیار به شما کمک خواهد کرد. این منو با کلیک کردن بروی فلش تب (



(یا راست کلیک کردن بروی تب ظاهر می‌شود.

Show Full Text

در صورت فعال بودن ، متون بصورت کامل نمایش داده می‌شوند ، در غیراینصورت بصورت خلاصه شده نمایش داده می‌شوند.

Show White Space

در صورت فعال بودن ، فضاهاى خالى ، کرکترهاى خط جدید و ... را نمایش می‌دهد.

```
<p>
  This menu is reachable via the little
  arrow in the panel tab (
  <a class="image" href="/wiki/index.php
  /File:OptionsMenuArrow.png">
  ) or by right-
  clicking on on the panel tab (since
  <a title="Firebug Release
  Notes" href="/wiki/index.php
  /Firebug_Release_Notes#Firebug_1.9">Fire
  bug 1.9</a>
  ) .
</p>
```

Show Comments

در صورت فعال بودن ، کامنت‌ها را هم نمایش می‌دهد

```
<div id="jump-to-nav">
  <!-- start content -->
  <div class="thumb tright">
    <p>The HTML panel displays the generated
    HTML/XML of the currently opened page.
```

سه گزینه ی Show Entities As Unicode و Show Entities As Symbols ، Show Entities As Names ، نوع نمایش کرکترهای ویژه را تعیین می‌کنند.

Highlight Changes

در صورت فعال بودن ، تگ تغییر یافته توسط JavaScript (یا تگ والدی که قابل مشاهده باشد) Highlight می‌شود

Expand Changes

در صورت فعال بودن ، زمان تغییر دادن یک تگ توسط JavaScript ، والدهای آن تگ باز (Expand) می‌شوند

Scroll Changes Into View

در صورت فعال بودن این گزینه ، هنگام تغییر یک تگ در صفحه توسط JavaScript ، قسمت NodeView به قسمت تغییر بافته حرکت می‌کند.

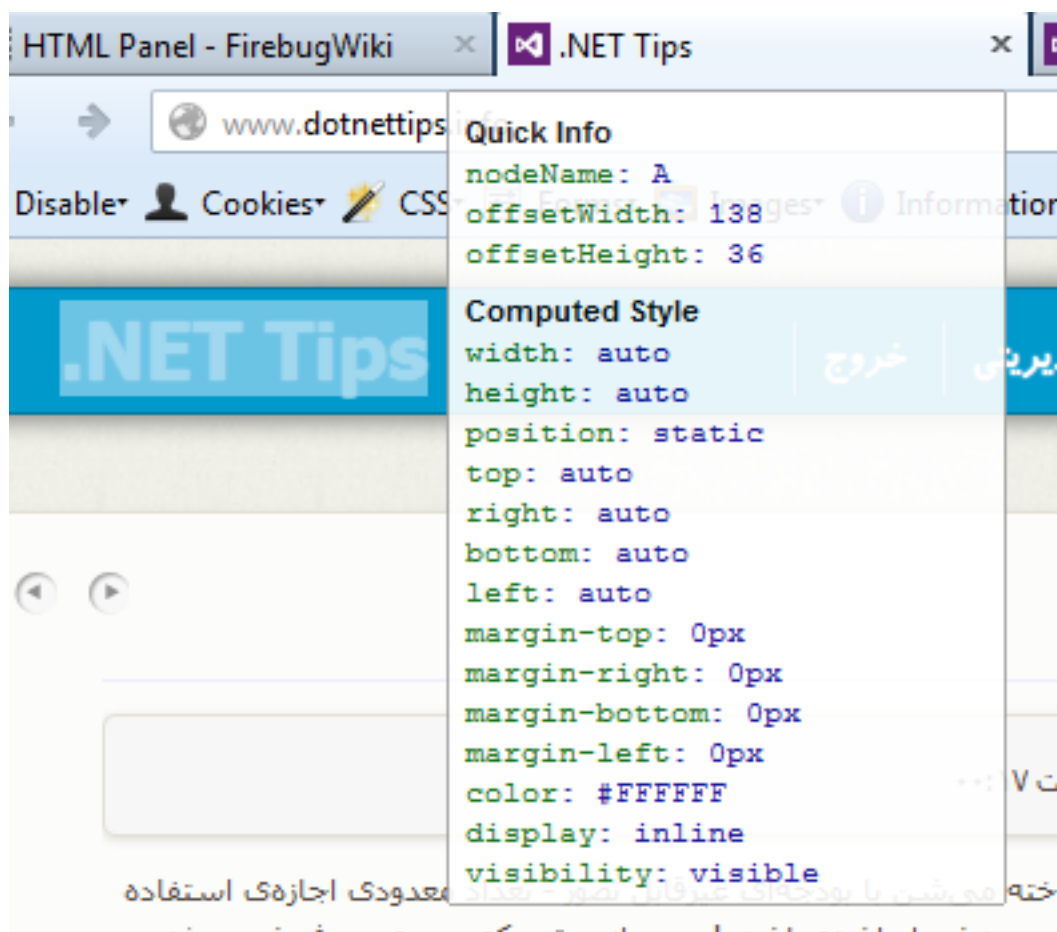
(فعال بودن این گزینه هنگام بررسی سایت هایی که اسلایدر یا سیستم هایی مشابه دارند ، باعث میشه که نتوانید بروی قسمت‌های سایت تمرکز کنید و مدام اسکرول به قسمت تغییرات منتقل می‌شود.)

Shade Box Model

در صورت فعال بودن ، فضای padding , margin و content را به شکلی که در بالا گفته شد نمایش می‌دهد ، در غیر اینصورت فقط یک خط دور تگ نشان می‌دهد.

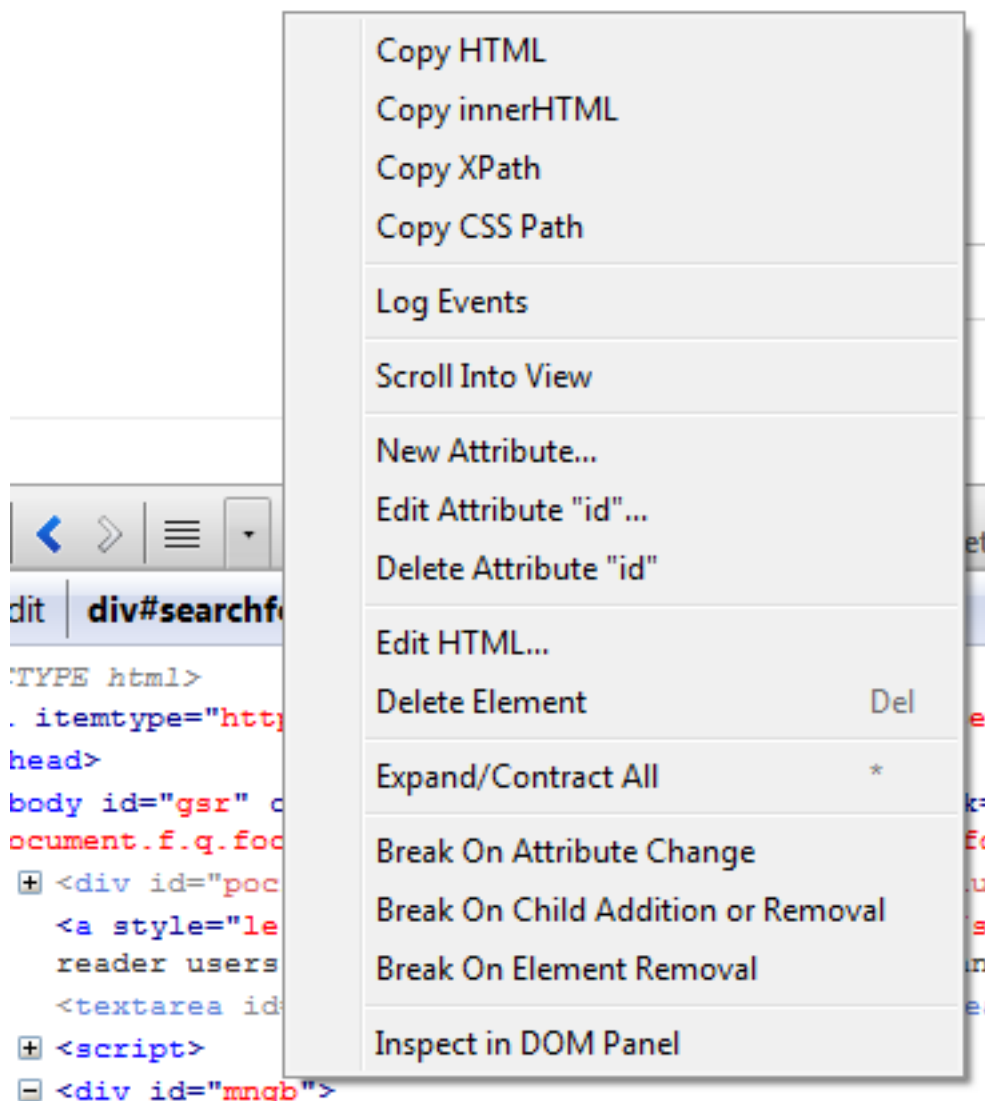
Show Quick Info Box

در صورت فعال بودن ، یک پاپ‌آپ به همراه اطلاعات مختصری از تگ در صفحه نمایش می‌دهد.



Context Menu

اگر بروی یک تگ راست کلیک کنید ، یک منو نمایش داده می‌شود ، در این قسمت با گزینه‌های این منو آشنا می‌شویم.

**Copy HTML**

خود تگ و محتوایش را بصورت HTML در حافظه کپی می‌کند.

Copy innerHTML

محتوای تگ را در حافظه کپی می‌کند.

Copy XPath

آدرس [XPath](#) تگ را در حافظه کپی می‌کند.

Copy CSS Path

[CSS Selector](#) تگ را در حافظه کپی می‌کند.

Log Events

رویدادهای تگ را در پنل Console ثبت می‌کند. (کلیک موس ، فشردن کلید ، ...) برای لغو لاگ کردن ، مجدداً بروی تگ راست کلیک کرده و این گزینه را از حالت انتخاب خارج کنید.

Scroll Into View

صفحه را به جایی که تگ قابل نمایش است منتقل می‌کند.

...New Attribute

یک attribute جدید به تگ اضافه می‌کند.
برای لغو عملیات ، دکمه‌ی Esc را بزنید.

..."<Edit Attribute "<attribute name

اگر بروی یک attribute راست کلیک کرده باشید ، این گزینه و گزینه‌ی بعدی را مشاهده خواهید کرد.
معادل کلیک بروی نام attribute است ، نام را به حالت ویرایش درمی آورد.

"<Delete Attribute "<attribute name

attribute را حذف می‌کند.

...Edit HTML

تگ را به حالت ویرایش می‌برد.
معادل انتخاب تگ ، زدن کلید Edit است.

Delete Element

تگ را حذف می‌کند.
راه دیگر حذف یک تگ ، انتخاب تگ و فشردن کلید Del از کیبورد است.

Expand/Contract All

تگ و Childهایش را باز/بسته می‌کند. (بجز تگ های <script> , <style> , <link>)
می‌توان با ترکیب کلید **+ Shift *** هم این کار را انجام داد که در این حالت تگ‌های فوق هم شامل باز/بسته شدن می‌شوند.

Break On Attribute Change

مانع اجرای کد JavaScript ای که attribute تگ را تغییر می‌دهد می‌شود و فایرباگ به خطی که باعث این عمل شده است در پنل Script منتقل می‌شود.
به عبارتی دیگر یک Break Point در خط JavaScript ای که باعث ویرایش attribute می‌شود قرار می‌دهد.

Break On Child Addition or Removal

مشابه توضیح قبل ، Break Point را در خطی که باعث اضافه/حذف شدن تگ Child شده است قرار می‌دهد.

Break On Element Removal

مشابه توضیح قبل ، Break Point را در خطی که باعث حذف شدن تگ شده است قرار می‌دهد.

Inspect in DOM Tab

تگ فعلی را در پنل DOM ، برای بررسی باز می‌کند.

در [قسمت بعدی](#) با پنل‌های سمت راست (Side Panels) آشنا می‌شویم.

نظرات خوانندگان

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۱۲/۲۵ ۱۵:۰

در [ورژن 1.11](#) از این افزونه ، امکانی برای paste کردن Html به صفحه ، به Context منوی پنل HTML اضافه شده که کار ویرایش محتویات Html را ساده تر کرده.
در حالی که قبلا برای این کار می بایست تگ را به حالت ویرایش درآورد و محتویات را اضافه کرد.

