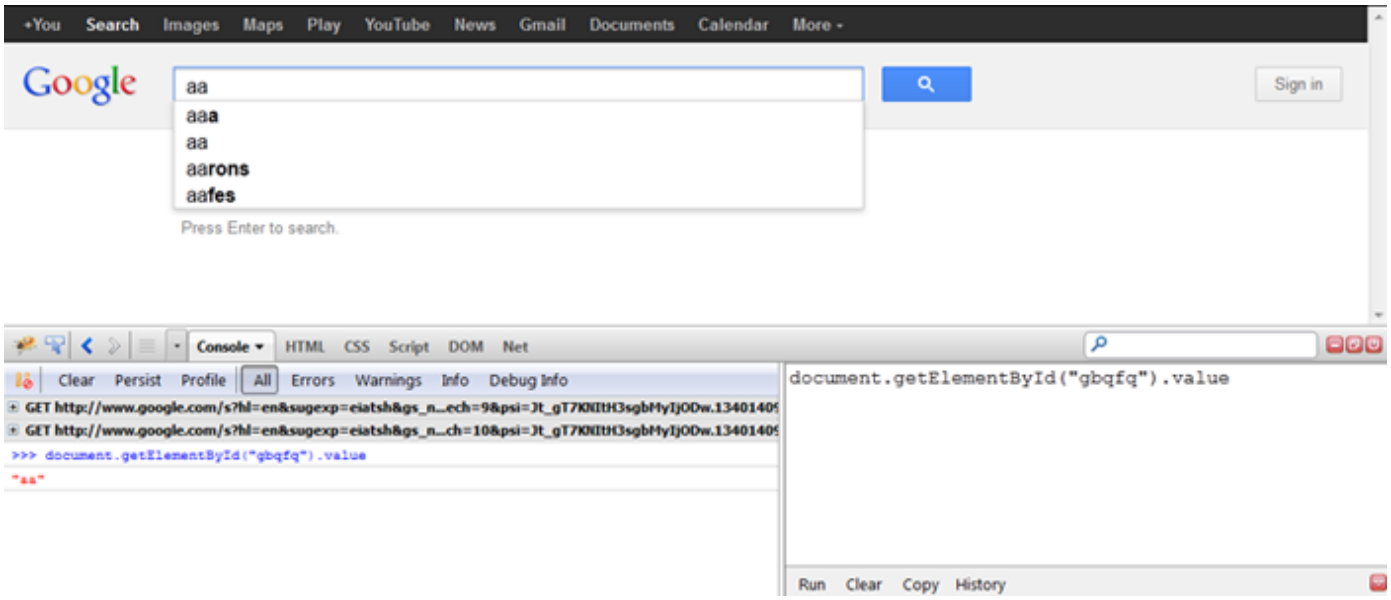
عنوان: **آموزش فایرباگ - #1**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۳/۳۱ ۲:۱۳
آدرس: www.dotnettips.info
برچسب‌ها: FireBug

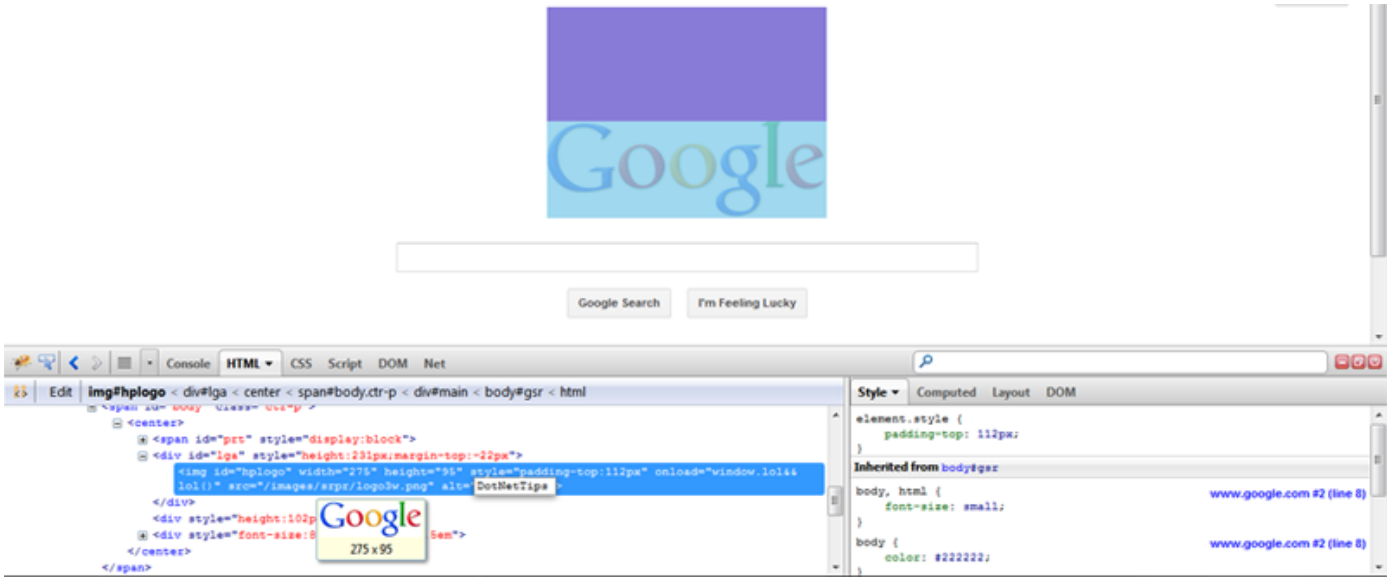
فایرباگ نام ابزاری است که همه ما نام آن را شنیده ایم یا با آن کار کرده ایم .
این ابزار امکانات و قابلیت‌های فراوانی دارد که فراگیری آنها برای یک توسعه گر یا طراح وب سایت ضروری به نظر می‌رسد .
در سری مقالات آموزش فایرباگ ، با این ابزار محبوب و پرکاربرد بیشتر آشنا می‌شویم .

نصب و اجرای فایرباگ :
ابتدا وارد این صفحه شوید و ورژن مناسب مرورگرتان را نصب نمایید .
پس از نصب و دوباره باز کردن مرورگر ، میتوانید با کلیک کردن بروی آیکون فایرباگ یا فشردن کلید F12 ، فایرباگ را اجرا کنید .

تب‌ها :
فایرباگ در حالت پیشفرض دارای 6 تب می‌باشد .
Console : در این تب دو بخش وجود دارد :
در بخش Log هشدارها ، پیغام‌ها ، درخواست‌های XHR و ... نمایش داده می‌شوند .
بخش دیگر هم که در سمت راست قرار دارد ، مخصوص اجرای کدهای جاوا اسکریپت می‌باشد .

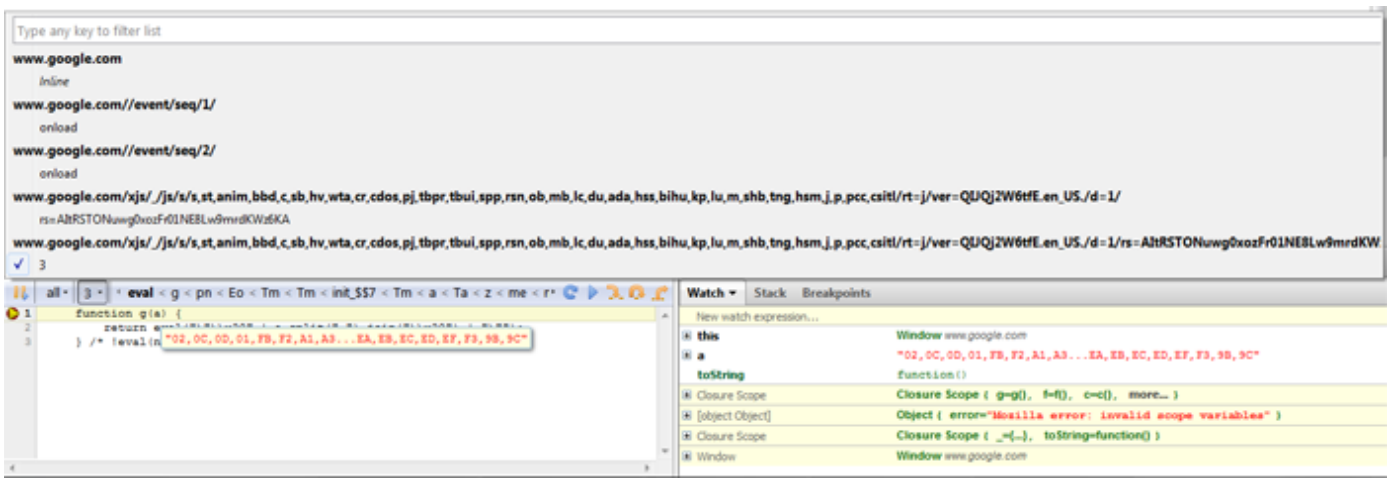


HTML : در این تب HTML صفحه را بصورت درختی و بصورت Live مشاهده می‌کنید .
Live بودن به این معنی است دقیقا چیزی را مشاهده می‌کنید که مرورگر در حال تفسیر هست و نه چیزی که به مرورگر ارسال شده .
همچنین در این تب می‌توانید محتوای HTML قسمت Body صفحه را ویرایش کنید .

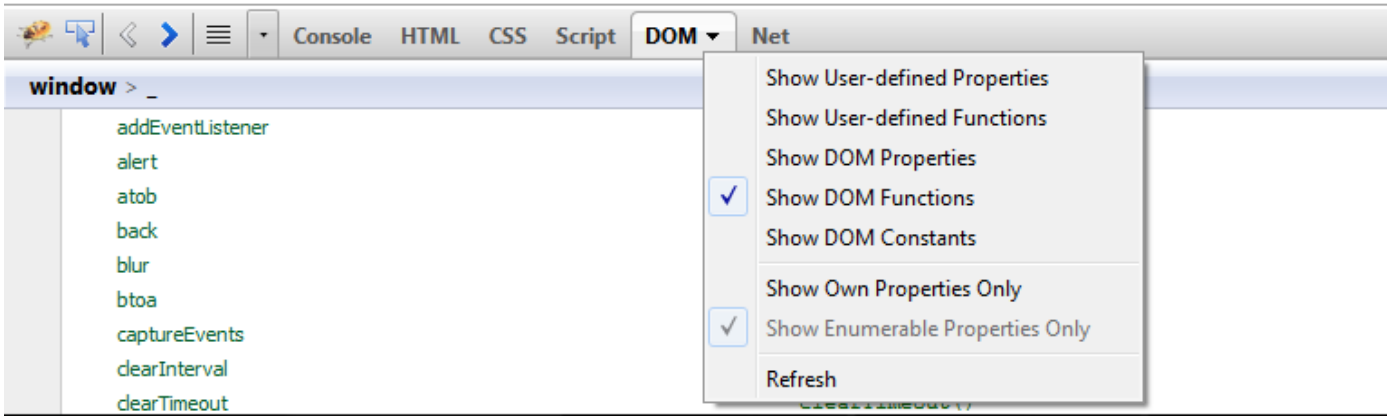


CSS : در این تب می‌توانید تعاریف CSS موجود در صفحه را مشاهده/ویرایش/حذف/فعال-غیرفعال نمایید .
همچنین می‌توانید تمام فایل‌های استایلی که به صفحه اضافه شده اند را مشاهده و تعیین کنید که تعاریف موجود در کدام فایل‌ها در این محیط نمایش داده بشوند .

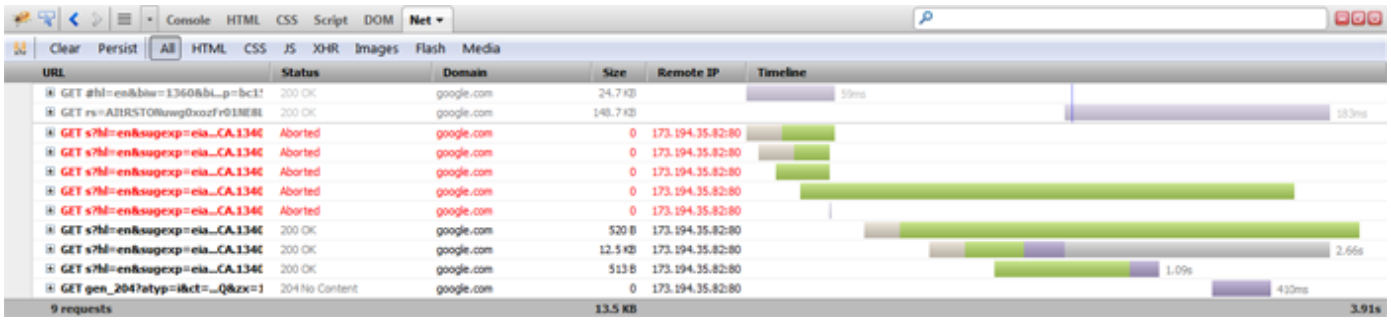
Script : در این تب می‌توانید به خطایابی کدهای جاوا اسکریپت بپردازید . همچنین مانند تب CSS می‌توانید یکی از فایل‌های اضافه شده به صفحه برای نمایش در این بخش ، چند مورد را انتخاب کنید .



DOM : در این تب می‌توانید ساختار DOM صفحه جاری را مشاهده نمایید .



Net : در این تب می‌توانید درخواست هایی که از صفحه ارسال می‌شوند را زیر نظر بگیرید .
این تب اطلاعات کاملی درباره‌ی هر درخواست ارائه می‌کند .
از جمله : نوع درخواست (Post/Get) ، حجم درخواست و پاسخ ، مدت زمان درخواست ، تعداد درخواست‌های ارسال شده ، وضعیت درخواست‌ها و ...



در قسمت‌های بعدی هر تب را به صورت جداگانه بررسی می‌کنیم .

نظرات خوانندگان

نویسنده: صالح
تاریخ: ۱۳۹۱/۰۳/۳۱ ۲۱:۵۲

ابزار بسیار کارامد و عالی هست. از این که آموزشش دادید. سپاس گذارم

نویسنده: سارا
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۰:۴۳

بسیار مقاله مفید و روانی بود با تشکر

نویسنده: سمیه
تاریخ: ۱۳۹۱/۰۴/۰۹ ۲۳:۴۲

بسیار عالی .مرسی

نویسنده: حسن
تاریخ: ۱۳۹۱/۱۲/۲۹ ۲۳:۲۲

اخوی ،میشه یه خرده بیشتر درمورد Dom بگید ؟
من نمیدونم Dom چیه ؟ ممنون از مطلب مفیدتون.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱:۱۰

ویکیپدیا

عنوان: **آموزش فایرباگ - #2 - تب Console**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۴/۰۵ ۱:۴
آدرس: www.dotnettips.info
برچسب‌ها: FireBug

در قسمت قبلی با تب‌های فایرباگ آشنا شدیم . در این قسمت و قسمت‌های بعدی ، با هر تب به صورت کامل آشنا خواهیم شد .

در این قسمت با تب Console آشنا خواهیم شد .
در قسمت قبل در مورد این تب گفتیم :

در این تب دو بخش وجود دارد :
در بخش Log هشدارها ، پیغام‌ها ، درخواست‌های XHR و ... نمایش داده می‌شوند .
بخش دیگر هم که در سمت راست قرار دارد ، مخصوص اجرای کدهای جاوا اسکریپت می‌باشد .

پس یک قسمت داریم برای نوشتن کدهای جاوا اسکریپت و یک قسمت هم برای مشاهده‌ی رویدادها .

اکنون 2 سوال مطرح می‌شود :

برای اجرای یک کد در حالت معمول چگونه عمل می‌کنیم ؟
پاسخ : کدهای مورد نظر را بین تگ باز و بسته‌ی script قرار می‌دهیم و صفحه مورد نظر را در مرورگر باز می‌کنیم و مرورگر کد را اجرا می‌کند .
در صورتی که کدهای ما خطا داشته باشند ، چگونه خطایابی می‌کنیم ؟
پاسخ : در بین کدهای نوشته شده چند alert قرار می‌دهیم و سعی می‌کنیم مشکل را پیدا کنیم .

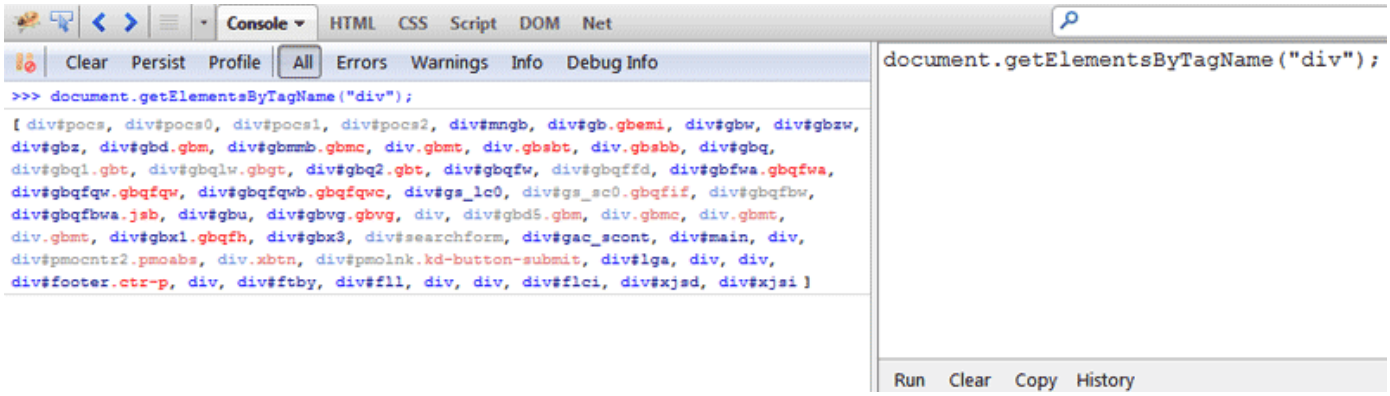
همین 2 سوال اهمیت و قدرت فایرباگ و قسمت Console آن را برای ما آشکار می‌کند ، زیرا ما می‌توانیم بدون Reload کردن صفحه ، کدهایمان را اجرا و نتیجه را مشاهده کنیم یا بوسیله توابع موجود خیلی ساده کدهایمان را خطایابی کنیم .

چگونه کدهای نوشته شده را سریع اجرا کنم ؟ کلید میانبر (HotKey) اجرای کدها چیست ؟
پاسخ : فشردن کلید CTRL + Enter

امتحان کنید :
کد زیر را در بخش کدنویسی تایپ کنید و بوسیله کلید میانبر گفته شده ، آن را اجرا کنید .

```
document.getElementsByTagName("div");
```

تصویر ذیل نتیجه اجرای کدی هست که اجرا کردیم .



چند نکته :
قسمت هایی که با رنگ قرمز و یک دات (.) بعد از نام تگ قرار گرفته اند ، کلاس‌های CSS ای هستند که المنت دارد .
قسمت هایی که با رنگ آبی تیره و یک شارپ (#) بعد از نام تگ قرار گرفته اند ، ID تگ‌ها هستند .
قسمت هایی که کمرنگ هستند ، المنت هایی هستند که در صفحه قابل نمایش نیستند .
اگر یک تگ چند کلاس داشته باشد ، فقط اولین کلاس نمایش داده می‌شود .

دکمه‌ها و حالت‌های نمایش بخش کد نویسی

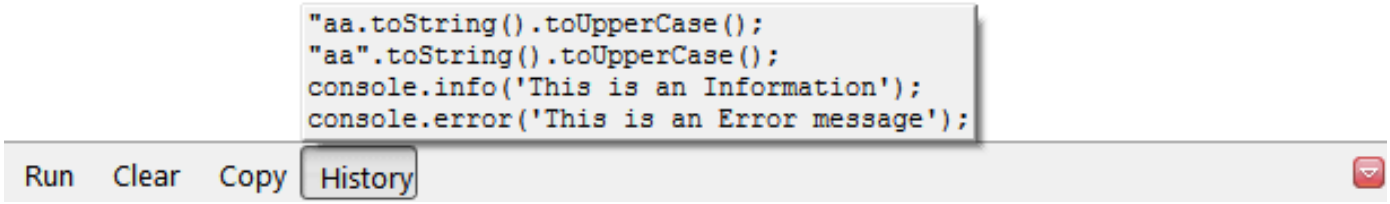
4 دکمه در قسمت کدنویسی وجود دارد :

Run : اجرای کد
Clear : خالی کردن بخش کد نویسی
Copy : کپی کردن کد موجود در بخش کد نویسی در حافظه
History : کدهای نوشته شده در نشست (Session) فعلی مرورگر

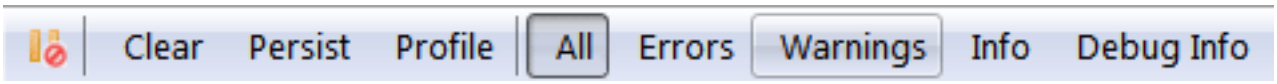
بخش کد نویسی می‌تواند به 2 شکل نمایش داده شود :

بصورت جعبه چند خطی (Command Editor)
بصورت تک خطی (Command Line)
با دکمه‌ی قرمز رنگ موجود در پایین - چپ می‌توانید بین این دو حالت سوئیچ کنید .

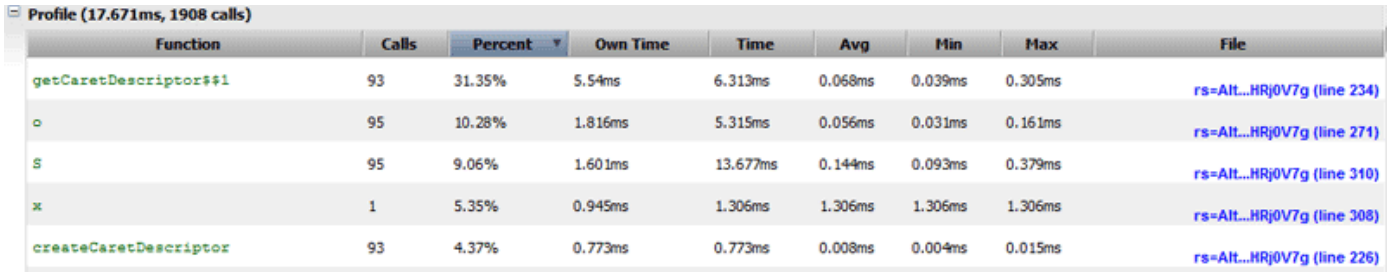
```
console.error('This is an Error message');
```



9 دکمه هم در بالای بخش log وجود دارد (به ترتیب از چپ به راست) :



دکمه ای با عنوان "Break On All Errors" . زمانی فعال شود ، در اولین اجرای یک کد از داخل صفحه ، به تب Script منتقل می‌شود و در خطی که کد در حال اجرا است توقف می‌کند .
Clear : بخش log را خالی می‌کند .
Persist : فعال بودن این دکمه باعث می‌شود که محتویات بخش Console در بارگزاری مجدد صفحه حفظ شود .
Profile : بوسیله این گزینه می‌توانید کدهای اجرایی خود در مدت زمان فعال بودن این دکمه ، تحت نظر بگیرید .
به این صورت که پس از غیر فعال کردن این دکمه (کلیک مجدد بروی آن) می‌توانید تابع‌های اجرا شده ، تعداد فراخوانی آنها ، مدت زمان اجرای هر یک ، میانگین زمان اجرای هر بار یک تابع و ... را مشاهده کنید .
5 دکمه‌ی بعدی هم برای فیلتر کردن Log هستند .

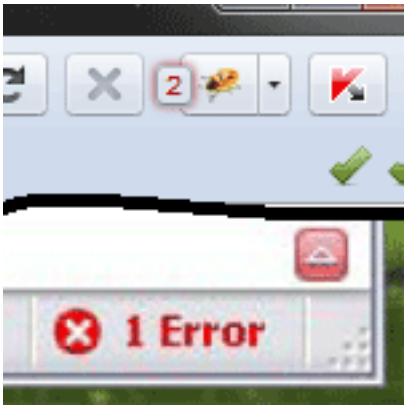

Profile (17.671ms, 1908 calls)

Function	Calls	Percent	Own Time	Time	Avg	Min	Max	File
getCaretDescriptor$$1	93	31.35%	5.54ms	6.313ms	0.068ms	0.039ms	0.305ms	rs=Alt...HRj0V7g (line 234)
o	95	10.28%	1.816ms	5.315ms	0.056ms	0.031ms	0.161ms	rs=Alt...HRj0V7g (line 271)
S	95	9.06%	1.601ms	13.677ms	0.144ms	0.093ms	0.379ms	rs=Alt...HRj0V7g (line 310)
x	1	5.35%	0.945ms	1.306ms	1.306ms	1.306ms	1.306ms	rs=Alt...HRj0V7g (line 308)
createCaretDescriptor	93	4.37%	0.773ms	0.773ms	0.008ms	0.004ms	0.015ms	rs=Alt...HRj0V7g (line 226)

نمایش تعداد خطاهای اتفاق افتاده ، در نوار وضعیت (Status Bar) :

اگر در زمان فعال بودن فایرباگ ، در صفحه خطایی رخ دهد ، در نوار وضعیت عدد 1 را نمایش می‌دهد و به ازای هر خطای جدید ،

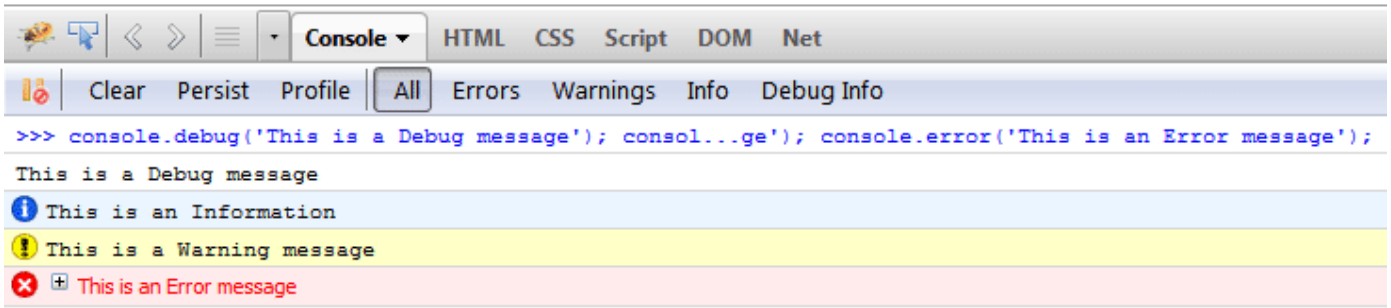
یکی به تعداد خطاها اضافه می‌کند .
البته اگر از ورژن‌های جدید فایرفاکس استفاده می‌کنید ، نوار وضعیت را نخواهید داشت و تعداد خطاها را در کنار آیکون فایرباگ خواهید داشت .



در کنار همه این امکانات ، فایرباگ یک مجموعه کامل از توابع کاربردی برای توسعه جاوا اسکریپت و خطایابی جاوا اسکریپت در اختیار ما می‌گذارد .

چند متد کمکی برای نوشتن Logهای مختلف در Console :

```
console.debug('This is a Debug message');
console.info('This is an Information');
console.warn('This is a Warning message');
console.error('This is an Error message');
```

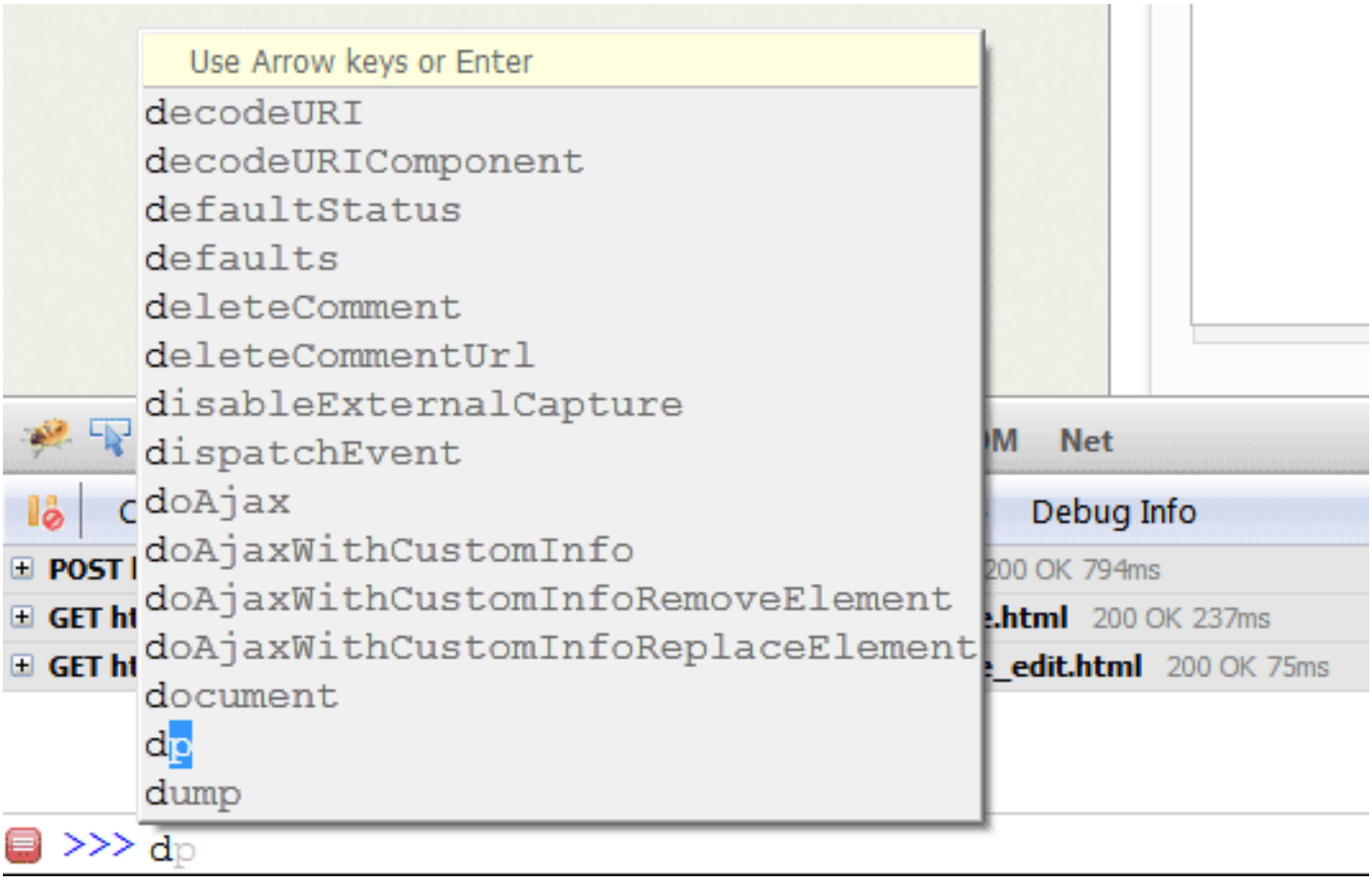


در قسمت بعدی توابع مربوط به توسعه جاوا اسکریپت (Command Line API & Console API) در فایرباگ را بررسی خواهیم کرد .

نظرات خوانندگان

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۴/۱۶ ۱:۵۲

در صورتی که محیط کدنویسی را بصورت Command Line قرار بدهید ، می‌توانید از امکان AutoComplete استفاده کنید که کار کدنویسی را بسیار ساده و شیرین می‌کند !



عنوان: آموزش فایرباگ - #3 JavaScript Development
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۴/۳۱ ۲:۲۱
آدرس: www.dotnettips.info
برچسب‌ها: JavaScript, xml, FireBug

توابع توسعه جاوا اسکریپت در فایرباگ به دو بخش تقسیم می‌شوند :
توابع خط فرمان - Command Line API
توابع کنسول - Console API

توابع خط فرمان توابعی هستند که فقط در خط فرمان قابل استفاده هستند و توابع کنسول هم توابعی هستند که خارج از محیط خط فرمان (، در بین کدهای جاوا اسکریپت برنامه) هم قابل استفاده هستند .
در این قسمت توابع خط فرمان را بررسی خواهیم کرد و در قسمت بعدی با توابع کنسول آشنا خواهیم شد .

توابع خط فرمان - Command Line API :

این توابع حدود 14 تا هستند که بوسیله آنها می‌توانیم در حین اجرای برنامه تست‌های مختلفی انجام داده یا اطلاعاتی از قسمت‌های مختلف بدست آوریم .
توجه : برای همراه شدن با تست‌های انجام شده در این مقاله می‌توانید کد صفحه‌ی زیر را ذخیره کنید و برای اجرای کدها ، آن‌ها را در قسمت خط فرمان (در تب کنسول) قرار بدهید و دکمه‌ی Run (یا Ctrl + Enter) را بزنید .

```
<div id="first" class="content">Content1 with css class and id</div>
<div class="content">
    Content2 with css class
    <a class="links" href="#">Link1</a>
    <a href="#">Link2</a>
</div>
<div>
    Content3 without css class and id
</div>
<input type="button" onclick="myFunc()" value="Run myFunc" />
<input type="text" id="myInput" />

<script type="text/javascript">
    function myFunc() {
        loop(1000);
        loop(50000);
    }
    function loop(number) {
        for (var i = 0; i < number; i++) { }
    }
</script>
```

قبل از توضیح این توابع ، یک مثال ساده می‌زنیم :
فرض کنید می‌خواهید همه‌ی المنت هایی که با یک selector مطابقت دارند را مشاهده کنید . (آرایه ای از المنت‌ها دریافت کنید)

```
$$("div.content");
```

نتیجه :

```
>>> $$("div.content");
[ div#first.content, div.content ]
```

می خواهید یکی از توابع جاوا اسکریپت برنامه را اجرا و آن را از لحاظ سرعت ، تعداد فراخوانی شدن و ... بررسی کنید .

```
profile("myFunc Testing");
myFunc();
profileEnd();
```

نتیجه :

```
>>> profile("myFunc Testing"); myFunc(); profileEnd();
```

myFunc Testing (0.291ms, 3 calls)

Function	Calls	Percent	Own Time	Time	Avg	Min	Max	File
loop	2	95.19%	0.277ms	0.277ms	0.139ms	0.123ms	0.154ms	1.htm (line 26)
myFunc	1	4.81%	0.014ms	0.291ms	0.291ms	0.291ms	0.291ms	1.htm (line 22)

اکنون با همه‌ی توابع خط فرمان آشنا می‌شویم :

$(id)

معادل دستور document.getElementById است که یک المنت با id داده شده بر می‌گرداند .

```
$("first");
```

نتیجه :

```
>>> $("first")
<div id="first" class="content">
```

$$(selector)

آرایه ای از المنت‌های مطابق با selector داده شده بر می‌گرداند .

```
$$("div.content")
```

نتیجه :

```
>>> $$("div.content")
[ div#first.content, div.content ]
```

به تفاوت دو دستور توجه کنید . خروجی دستور اول ، یک المنت است و خروجی دستور دوم یک آرایه از المنت که بین [و] قرار گرفته اند .

برای آشنایی بیشتر با CSS Seletorها به این لینک مراجعه کنید : http://www.w3.org/TR/css3-selectors

$x(xPathExpression)

آرایه ای از المنت هایی را بر می‌گرداند که با xPath داده شده مطابقت داشته باشند .

```
var objects = $x("html/body/div[2]/a")

for(var i = 0; i < objects.length; i++) {
   console.log(objects[i]);
}
```

نتیجه :

```
>>> var objects = $x("html/body/div[2]/a")  for(var ...cts.length; i++) {
console.log(objects[i]); }
<a class="links" href="#">
<a href="#">
```

برای آشنایی بیشتر با عبارات xPath به این لینک مراجعه کنید : http://www.w3schools.com/xpath

dir(object)

تمام خصوصیات شیء ارسال شده را لیست می‌کند .

```
var objects = $x("html/body/div[2]/a")

dir(objects);
```

نتیجه :

```
>>> var objects = $x("html/body/div[2]/a")  dir(objects);
⊞ 0                                  a.links #
⊟ 1                                  a #
    addEventListener                 addEventListener()
    appendChild                      appendChild()
    blur                             blur()
    cloneNode                        cloneNode()
    compareDocumentPosition          compareDocumentPosition()
    dispatchEvent                    dispatchEvent()
    focus                            focus()
```

(dirxml(node

سورس یک المنت را بصورت درختواره (tree) پرینت می‌کند . همچنین با کلیک بروی هر node ، فایرباگ آن node را در تب html نمایش می‌دهد .

```
var node = $("first");
dirxml(node);
```

نتیجه :

```
>>> var node = $("first"); dirxml(node);
<div id="first" class="content">
   Content1 with css class and id
</div>
```

توجه کنید که این دستور فقط یک node دریافت می‌کند . برای همین اگر از دستور $$("first#") استفاده می‌کنید ، چون این دستور یک آرایه بر می‌گرداند ، باید اولین عضو آرایه را دریافت و ارسال کنید .
یعنی :

```
var node = $$("#first")[0];
dirxml(node);
```

()clear

این دستور محیط console را خالی می‌کند . عملکرد این دستور معادل کلیک دکمه‌ی Clear (در بالا - چپ تب کنسول) است .

([inspect(object[,tabName

توسط این دستور می‌توانید یک شیء را در مناسب‌ترین تب فایرباگ یا یکی از تب‌های مورد نظر خود ، Inspect کنید .

```
var node = $("first");

inspect(node); // inspect in html tab
inspect(node,'dom'); // inspect in dom tab
```

(keys(object

آرایه ای از "نام" تمام خصوصیات شیء ارسال شده بر می‌گرداند .

```
var obj = $("first");
keys(obj)
```

(values(object

آرایه ای از "مقدار" تمام خصوصیات شیء ارسال شده بر می‌گرداند .

```
var obj = $("first");
values(obj)
```

(debug(fn) and undebug(fn

این متدها یک BreakPoint در ابتدای تابع مشخص شده اضافه/حذف می‌کنند . (در تب Script) . به همین ترتیب هنگامی که تابع مورد نظر فراخوانی شود ، در نقطه ای که BreakPoint قرار داده شده توقف خواهد کرد .
البته می‌شود BreakPoint را دستی هم قرار داد . در اصل این تابع ، این عملیات را ساده‌تر می‌کند .

```
debug(myFunc);
myFunc();
undebug(myFunc);
```

(monitor(fn) and unmonitor(fn

این متدها برای فعال/غیرفعال کردن Logging فراخوانی‌های یک تابع استفاده می‌شوند .
در حالت عادی برای پی بردن به اینکه یک تابع اجرا می‌شود یا نه ، در تابع مورد نظر یک alert قرار می‌دهیم و تست می‌کنیم . که این روش در برنامه برنامه‌های بزرگ صحیح نیست . زیرا در این حالت باید بین حجم زیادی کد به دنبال تابع مورد نظر بگردیم و سپس alert را قرار بدهیم و بعد اطمینان از صحت عملکرد تابع مجدد آن را حذف کرد ، که با اتلاف زمان و به خطر انداختن کدها همراه است .
اما با استفاده از این متدها ، تنها نیاز به داشتن اسم تابع داریم (و نه مکان تابع در کدهای برنامه) .
تست monitor :

```
monitor(myFunc);
// now click on "Run myFunc" button
```

تست unmonitor :

```
unmonitor(myFunc);
// now click on "Run myFunc" button
```

([monitorEvents(object[, types]) and unmonitorEvents(object[, types

این متدها عملیات Event Logging برای یک شیء را فعال/غیرفعال می‌کنند . در کنار شیء مورد نظر ، می‌توان نوع رویداد را هم به

متد ارسال کرد . در این صورت عملیات Logging فقط برای همان گروه رویداد/رویداد ، فعال/غیرفعال می‌شود .
منظور از گروه رویداد ، مجموعه رویدادهای یک شیء است . مثلا mousedown , moseover , mousemove , ... در گروه mouse قرار می‌گیرند . یعنی می‌توانید با ارسال کلمه‌ی mouse فقط رویدادهای mouse را تحت نظر بگیرید یا اینکه فقط یک رویداد را مشخص کنید ، مثل mousedown .
راه ساده‌تر فعال کردن Event Logging ، رفتن به تب Html ، راست کلیک کردن بروی المنت مورد نظر و فعال کردن گزینه‌ی Log Events می‌باشد .

```
var obj = $("myInput");
monitorEvents(obj,'keypress');
```

نتیجه پس از فشردن چند دکمه‌ی کیبورد در myInput :

keypress charCode=0, keyCode=38
keypress charCode=0, keyCode=38
keypress charCode=0, keyCode=38
keypress charCode=0, keyCode=39
keypress charCode=0, keyCode=13
keypress charCode=0, keyCode=13
keypress charCode=115, keyCode=0
keypress charCode=100, keyCode=0
keypress charCode=97, keyCode=0
keypress charCode=97, keyCode=0

توضیحات بیشتر : http://getfirebug.com/wiki/index.php/MonitorEvents

()profile([title]) and profileEnd

این متدها ، JavaScript Profiler را فعال/غیرفعال می‌کنند . هنگام فعال کردن می‌توانید یک عنوان هم برای پروفایل مشخص کنید . در قسمت قبلی مقاله در مورد این قابلیت توضیحاتی ارائه شد .
سه را برای اجرای Profiler وجود دارد :
1 - کلیک بروی دکمه‌ی Profiler در بالای تب کنسول .
2 - استفاده از کد console.profile("ProfileTitle") در کدهای جاوا اسکریپت .
3 - استفاده از متد profile("Profile Title") در خط فرمان .

```
profile("myFunc Testing");
myFunc();
profileEnd();
```

نتیجه :

```
>>> profile("myFunc Testing"); myFunc(); profileEnd();
```

myFunc Testing (0.335ms, 3 calls)

Function	Calls	Percent	Own Time	Time	Avg	Min	Max	File
loop	2	95.22%	0.319ms	5.545ms	2.773ms	2.662ms	2.883ms	1.htm (line 26)
myFunc	1	4.78%	0.016ms	5.561ms	5.561ms	5.561ms	5.561ms	1.htm (line 22)

ستون‌های Profiler :

Function : نام تابع اجرا شده .
Calls : تعداد دفعات فراخوانی تابع .
Percent : زمان اجرای تابع در زمان کل ، به درصد .
Own Time : زمان اجرای تابع به تنهایی . برای مثال در کد ما ، زمان اجرای تابع myFunc به تنهایی تقریبا صفر است زیرا عملیاتی در خود انجام نمی‌دهد و زمان صرف شده در این تابع ، برای اجرای 2 با تابع loop است . این زمان (Own Time) زمان اجرای تابع ، منهای زمان صرف شده برای فراخوانی توابع دیگر است .
Time : زمان اجرای تابع از نقطه‌ی آغاز تا پایان . مجموع زمان اجرای خود تابع به همراه زمان اجرای توابع فراخوانی شده . در کد ما ، این زمان ، مجموع زمان اجرای خود تابع به همراه دو بار فراخوانی تابع loop است .
Avg : میانگین زمان اجرای هربار تابع . فرمول : Avg = Time / Calls
Min & Max : حداقل و حداکثر زمان اجرای تابع .
File : نام فایل و شماره خطی که تابع در آن قرار دارد .

در قسمت بعد با توابع کنسول آشنا خواهیم شد .

منابع :
http://michaelsync.net/2007/09/15/firebug-tutorial-commandline-api
http://michaelsync.net/2007/09/09/firebug-tutorial-logging-profiling-and-commandline-part-i
http://getfirebug.com/wiki/index.php/Console_Panel
http://getfirebug.com/wiki/index.php/MonitorEvents

Packtpub.Firebug.1.5.Editing.Debugging.and.Monitoring.Web.Pages.Apr.2010

عنوان: **آموزش فایرباگ - #4 - JavaScript Development**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱۹:۵۹
آدرس: www.dotnettips.info
برچسب‌ها: FireBug

در قسمت قبل با توابع خط فرمان آشنا شدیم . در این قسمت با توابع کنسول آشنا خواهیم شد .

فایرباگ یک متغییر عمومی به نام console دارد که به همه‌ی صفحات باز شده در فایرفاکس اضافه می‌کند . این شیء متدهایی دارد که بوسیله آن‌ها می‌توانیم عملیاتی در برنامه مان انجام داده و اطلاعاتی را در کنسول چاپ کنیم .

بعضی از این متدها عملکردی مشابه متدهای خط فرمان (که در قسمت قبل شرح داده شدند ،) دارند که از توضیح مجدد آن‌ها اجتناب می‌کنیم .

توابع کنسول - Console API :

توجه : همانند قسمت قبل ، در این قسمت هم برای همراه شدن با تست‌ها ، کد صفحه‌ی زیر را ذخیره کنید و برای اجرای کدها ، آن‌ها را در قسمت خط فرمان (در تب کنسول) قرار بدهید و دکمه‌ی Run (یا Ctrl + Enter) را بزنید .

```
<input type="button" onclick="startTrace('Some Text')" value="startTrace" />
<input type="button" onclick="startError()" value="test Error" />

<script type="text/javascript">
    function startTrace(str) {
        return method1(100, 200);
    }
    function method1(arg1, arg2) {
        return method2(arg1 + arg2 + 100);
    }
    function method2(arg1) {
        var var1 = arg1 / 100;
        return method3(var1);
    }
    function method3(arg1) {
        console.trace();
        var total = arg1 * 100;
        return total;
    }

    function testCount() {
        // do something
        console.count("testCount() Calls Count .");
    }

    function startError() {
        testError();
    }

    function testError() {
        var errorObj = new Error();
        errorObj.message = "this is a test error";
        console.exception(errorObj);
    }

    function testFunc() {
        var t = 0;
        for (var i = 0; i < 100; i++) {
            t += i;
        }
    }
</script>
```

([...,console.log(object[,object

این دستور یک پیغام در کنسول چاپ می‌کند .

```
console.log("This is a log message!");
```

نتیجه :

```
>>> console.log("This is a log message!");
This is a log message!
```

این دستور را می‌توانیم به شکل‌های مختلفی فراخوانی کنیم .
مثلا :

```
console.log(1 , "+" , 2 , "=", (1+2));
```

نتیجه :

```
>>> console.log(1 , "+" , 2 , "=", (1+2));
1 + 2 = 3
```

در این دستور می‌توانیم از چند حرف جایگزین هم استفاده کنیم .

Pattern	Type
%s	String
%d, %i	Integer (numeric formatting is not yet supported)
%f	Floating point number (numeric formatting is not yet supported)
%o	Object hyperlink
%c	Style formatting

مثال :

```
console.log("Firebug 1.0 beta was %s in December %i.","released",2006);
```

نتیجه :

```
>>> console.log("Firebug 1.0 beta was %s in December %i.","released",2006);
Firebug 1.0 beta was released in December 2006.
```

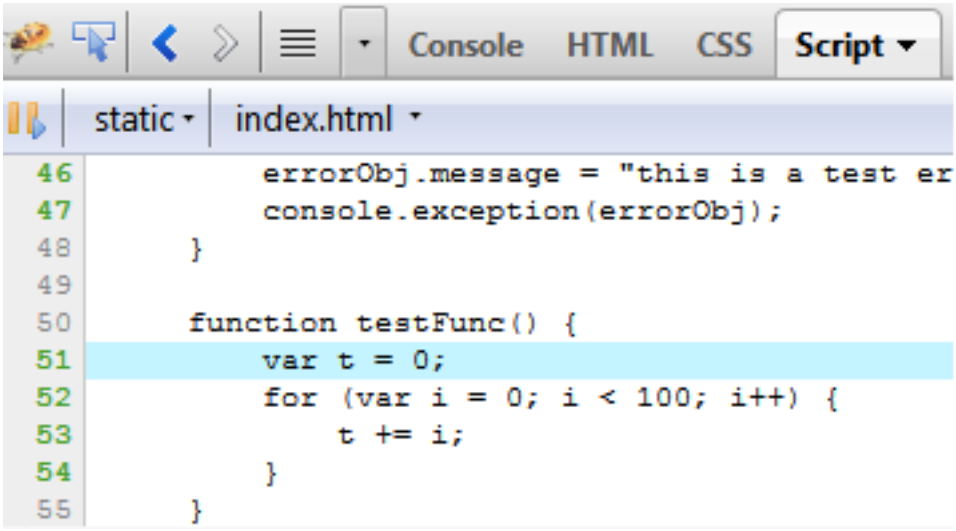
عملکرد 3 جایگزین نخست با توجه با مثال قبل مشخص شد . پس به سراغ جایگزین o% و c% می‌رویم .
اگر در رشته‌ی مورد نظر ، یک شیء (تابع ، آرایه ، ...) برای جایگزین o% ارسال کنیم ، در خروجی آن شیء بصورت لینک نمایش داده می‌شود که با کلیک بروی آن ، فایرباگ آن شیء را در تب مناسبش Inspect می‌کند .
مثال :

```
console.log("this is a test functin : %o",testFunc);
```

نتیجه :

```
>>> console.log("this is a test functin : %o",testFunc);
this is a test functin : testFunc()
```

و زمانی که بروی لینک testFunc کلیک کنیم :



یک ترفند : بوسیله جایگزین o% توانستیم به تابع مورد نظر لینک بدهیم . اگر بجای جایگزین o% از s% استفاده کنیم ، می‌توانیم بدنه‌ی تابع را ببینیم :

```
console.log("this is a test functin : %s",testFunc);
```

نتیجه :

```
>>> console.log("this is a test functin : %s",testFunc);
this is a test functin : function testFunc() {
    var t = 0;
    for (var i = 0; i < 100; i++) {
        t += i;
    }
}
```

توسط جایگزین c% هم می‌توانید خروجی را فرمت کنید .

```
console.log("%cThis is a Style Formatted Log","color:green;text-decoration:underline;");
```

نتیجه :

```
>>> console.log("%cThis is a Style Formatted Log","color:green;text-decoration:underline;");
This is a Style Formatted Log
```

console.log(object[, object, ...])

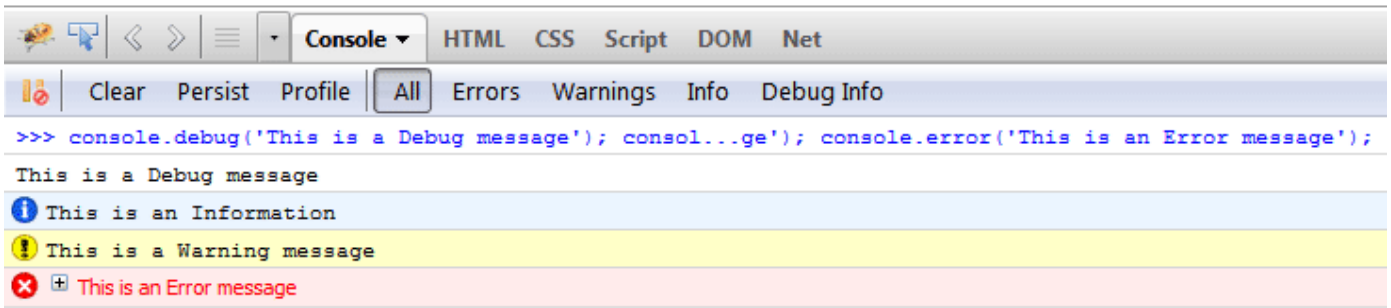
console.debug(object[, object, ...])

console.info(object[, object, ...])

console.warn(object[, object, ...])

console.error(object[, object, ...])

مشابه با دستور log عمل می‌کنند با این تفاوت که خروجی را با استایل متفاوتی نمایش می‌دهند .
همچنین هر یک از این دستورات ، توسط دکمه‌های همنام در کنسول قابل فیلتر شدن هستند .

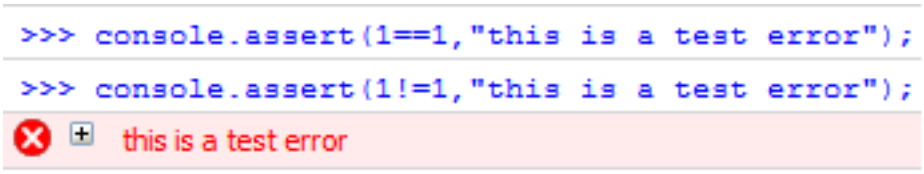


([... ,console.assert(expression[, object

چک می‌کند که عبارت ارسال شده true هست یا نه . اگر true نبود ، پیغام وارد شده را چاپ و یک استثناء ایجاد می‌کند .

```
console.assert(1==1,"this is a test error");
console.assert(1!=1,"this is a test error");
```

نتیجه :



()console.clear

(console.dir(object

(console.dirxml(node

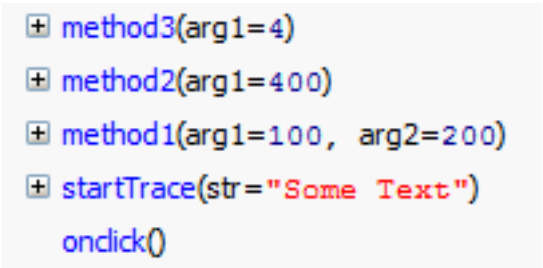
([console.profile([title

()console.profileEnd

این توابع معادل توابع همنامشان در خط فرمان هستند که در قسمت قبل با عملکردشان آشنا شدیم .

()console.trace

با این متد می‌توانید پی ببرید که از کجا و توسط چه متدهایی برنامه به قسمت trace رسیده . برای درک بهتر مجددا اسکریپت صفحه‌ی تست این مقاله را بررسی کنید (جایی که متد trace قرار داده شده است) .
اکنون صفحه‌ی تست را باز کنید و بروی دکمه‌ی startTrace کلیک کنید . خروجی ظاهر شده در کنسول را از پایین به بالا بررسی کنید .



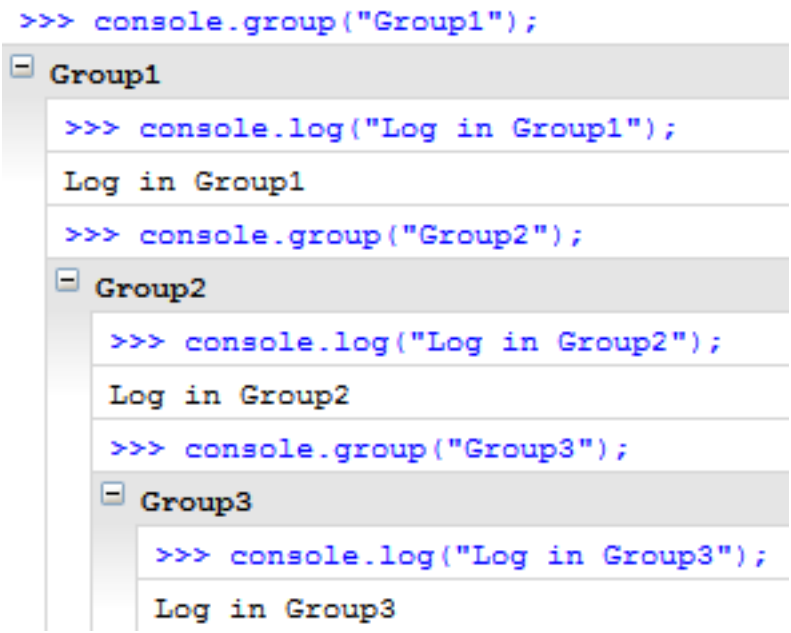
حتما متوجه شدید که متد method3 چگونه در کدهایمان فراخوانی شده است !؟
ابتدا با کلیک بروی دکمه‌ی startTrace ، متد startTrace اجرا شده و به همین ترتیب متد startTrace متد method1 ، متد method1 هم متد method2 و در نهایت method2 متد method3 را فراخوانی کرده است .
دستور trace زمانی که در حال بررسی کدهای برنامه نویسان دیگر هستید ، بسیار می‌تواند به شما کمک کند .

([... ,console.group(object[, object

با این دستور می‌توانید لاگ‌های کنسول را بصورت تو در تو گروه بندی کنید .

```
console.group("Group1");
console.log("Log in Group1");
console.group("Group2");
console.log("Log in Group2");
console.group("Group3");
console.log("Log in Group3");
```

نتیجه :



([... ,console.groupCollapsed(object[, object

این دستور معادل دستور قبلی است با این تفاوت که هنگام ایجاد ، گروه را جمع می‌کند .

()console.groupEnd

به آخرین گروه بندی ایجاد شده خاتمه می‌دهد .

(console.time(name

یک تایمر با نام داده شده ایجاد می‌کند . زمانی که نیاز دارید زمان طی شده بین 2 نقطه را اندازه گیری کنید ، این تابع مفید خواهد بود .

(console.timeEnd(name

تایمر همنام را متوقف و زمان طی شده را چاپ می‌کند .

```
console.time("TestTime");
var t = 1;
for (var i = 0; i < 100000; i++) { t *= (i + t) }
console.timeEnd("TestTime");
```

نتیجه :

```
>>> console.time("TestTime");  var t = 1;  for (var ...) { t *= (i + t) }
console.timeEnd("TestTime");
TestTime: 527ms
527
```

console.timeStamp()

توضیحات کامل را از اینجا دریافت کنید .

console.count([title])

تعداد دفعات فراخوانی شدن کدی که این متد در آنجا قرار دارد را چاپ می‌کند .
البته ظاهرا در ورژن 10.0.1 که بنده با آن کار می‌کنم ، این دستور بی عیب کار نمی‌کند . زیرا بجای آنکه در هربار فراخوانی ، در همان خط تعداد فراخوانی را نمایش بدهد ، فقط اولین لاگ را آپدیت می‌کند .

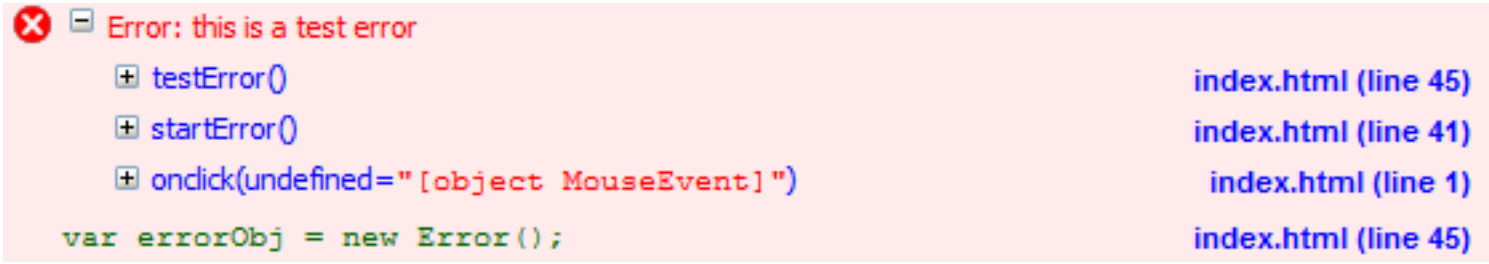
```
>>> testCount()
testCount() Calls Count . 5
undefined
>>> testCount()
undefined
>>> testCount()
undefined
>>> testCount()
undefined
>>> testCount()
undefined
```

console.exception(error-object[, object, ...])

یک پیغام خطا را به همراه ردیابی کامل اجرای کدها تا زمان رویداد خطا (مانند متد trace) چاپ می‌کند .
در صفحه‌ی تست این متد را اجرا کنید :

```
startError();
```

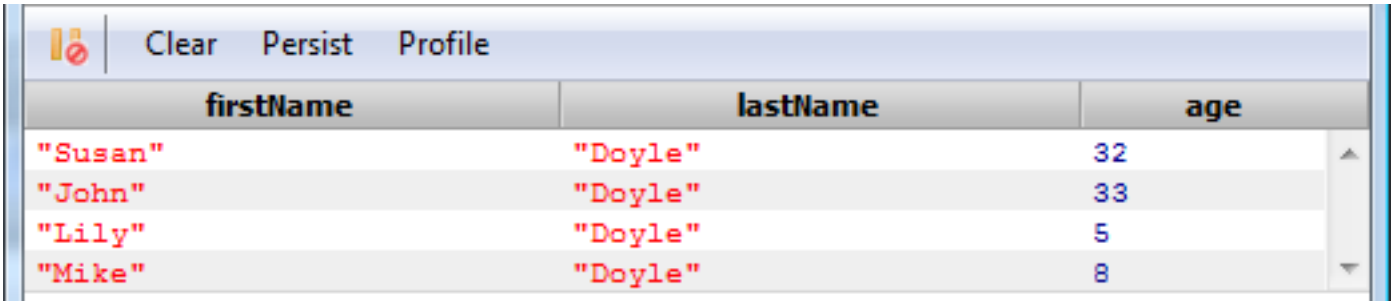
نتیجه :



توجه کنید که ما برای مشاهده‌ی عملکرد صحیح این دستور ، آن را در تابع testError قرار دادیم و بوسیله تابع startError آن فراخوانی کردیم .

console.table(data[, columns])

بوسیله این دستور می‌توانید مجموعه ای از اطلاعات را بصورت جدول بندی نمایش بدهید .
این متد از متدهای جدیدی است که در فایرباگ قرار داده شده است .



برای اطلاعات بیشتر به اینجا مراجعه کنید .

منابع : https://getfirebug.com/wiki/index.php/Console_API

نظرات خوانندگان

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۱ ۱۳:۷

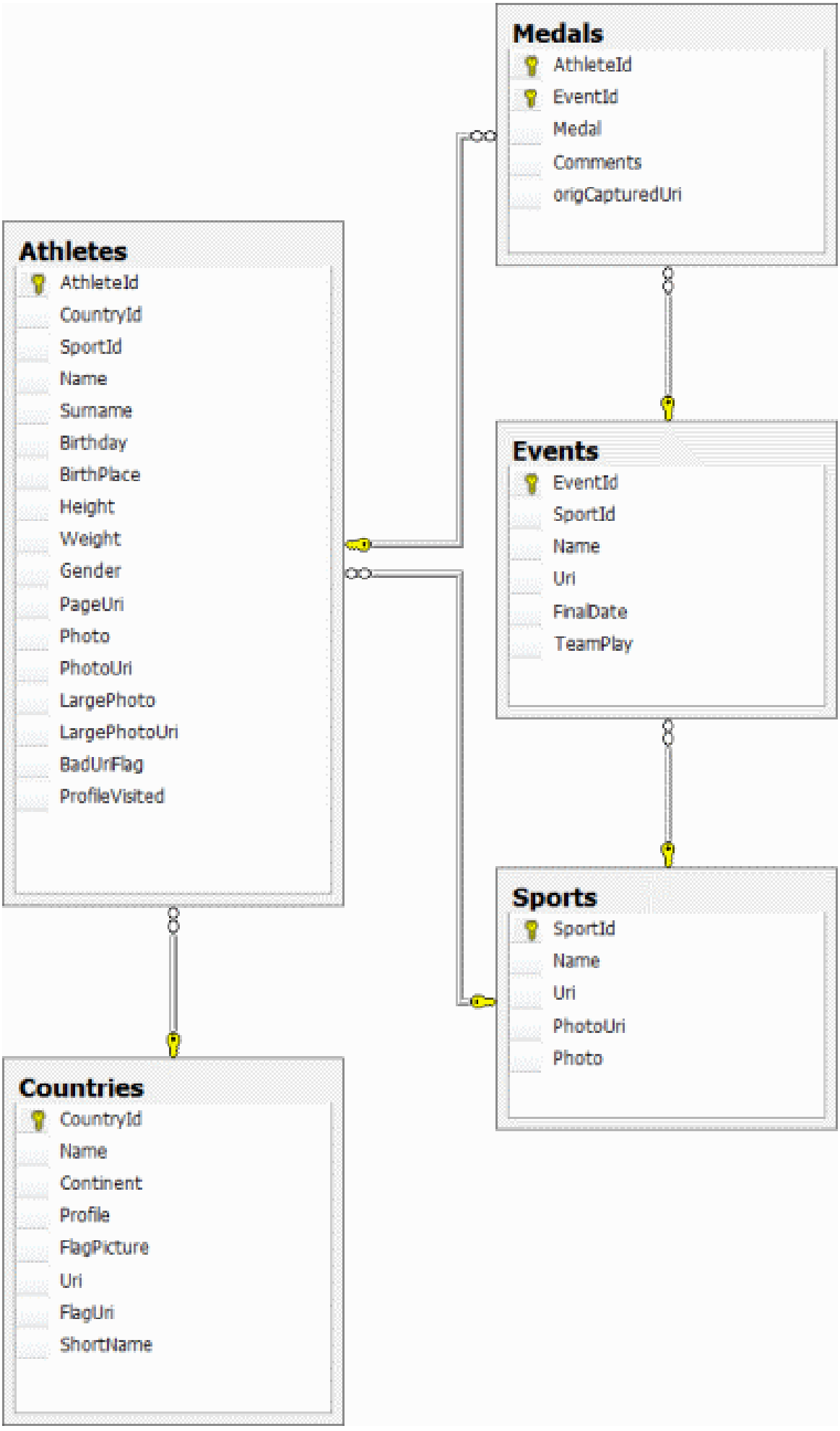
با سلام
می خواستم بپرسم فایرباگ چه محاسنی نسبت به developer tool مرورگر کروم داره؟ به نظر خیلی شبیه به هم می‌آیند.
با تشکر از زحمات شما

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۵/۱۲ ۷:۱۲

سلام
بنده خیلی کم با Developer Tool کار کردم . اما به نظر میرسه قوی و کامل‌تر از فایرباگ باشه .
نتایج جستجوی عبارت " google chrome developer tools vs firebug " در گوگل هم جالب و مفیده .

عنوان: **استخراج اطلاعات از صفحات وب با کمک HtmlAgilityPack**
نویسنده: ایمان عبیدی
تاریخ: ۱۳۹۱/۰۶/۰۳ ۲۳:۵۸
آدرس: www.dotnettips.info
گروه‌ها: XPATH, xml, Html Agility Pack, FireBug

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


Medals
AthleteId
EventId
Medal
Comments
origCapturedUri
Athletes
AthleteId
CountryId
SportId
Name
Surname
Birthday
BirthPlace
Height
Weight
Gender
PageUri
Photo
PhotoUri
LargePhoto
LargePhotoUri
BadUrlFlag
ProfileVisited
Events
EventId
SportId
Name
Uri
FinalDate
TeamPlay
Sports
SportId
Name
Uri
PhotoUri
Photo
Countries
CountryId
Name
Continent
Profile
FlagPicture
Uri
FlagUri
ShortName

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 فورا با استثنای زیر مواجه می‌شوید

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

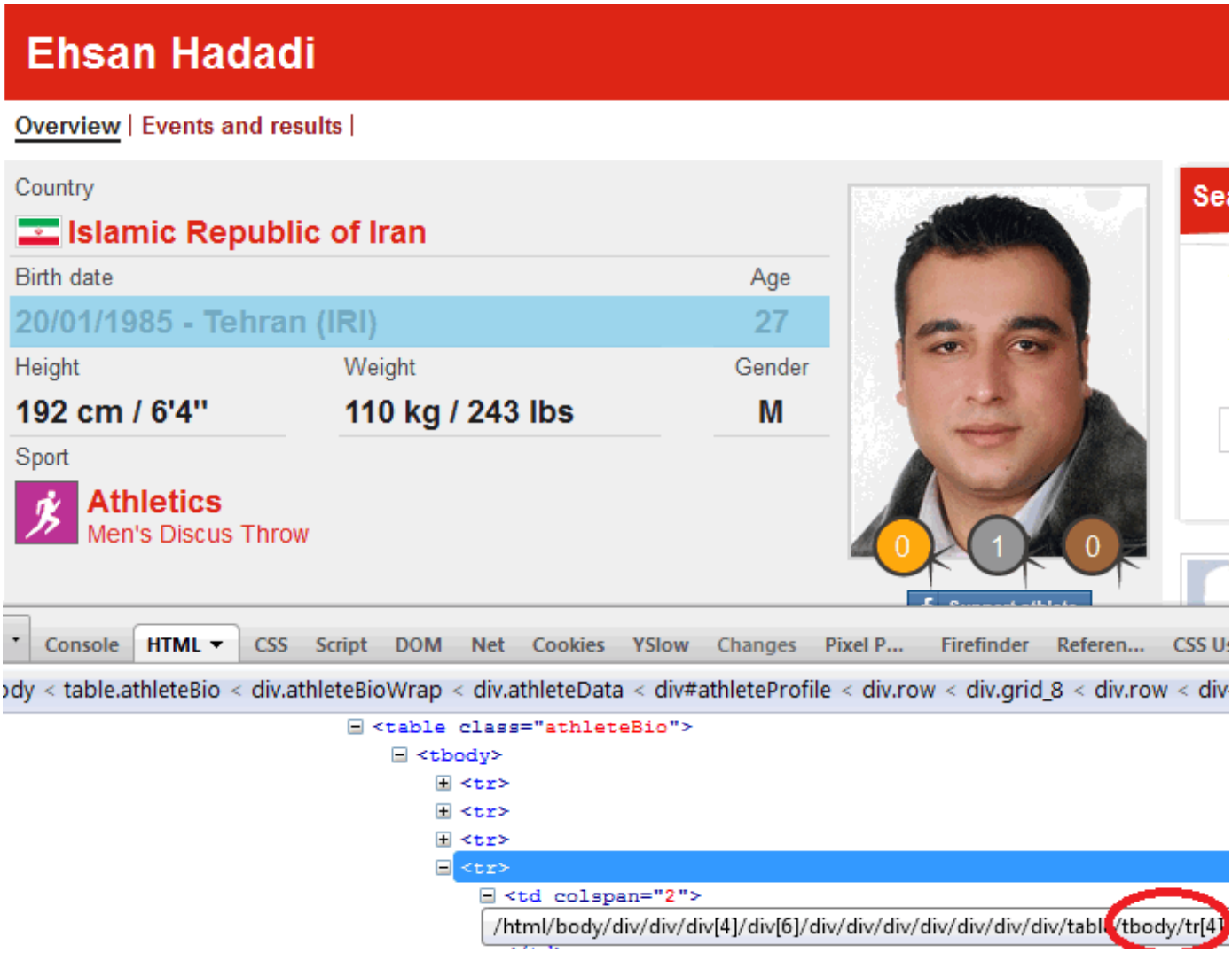
راه حل ساده اینه که این کتابخونه رو با کمک NuGet نصب کنید

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

و از اینجا به بعد با کدی مثل این میتونید از کلاس HtmlDocument و مشابه XmlDocument ولی بدون ارور استفاده کنید.
مثلا 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(" ", "").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 HttpWebRequest بنویسید و توی خوده HAP متاسفانه چنین تابعی توکار نشده نکته اصلی هم پیدا کردن محل دقیق اطلاعاته که با ابزاری مثل Firebug خیلی راحت‌تر 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 ازش استفاده کنید.



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

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۶/۰۴ ۱۲:۵۳

این روش برای استخراج مطالب همین سایت خیلی مفیده!

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

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

نویسنده: جمال
تاریخ: ۱۳۹۱/۰۶/۰۴ ۱۴:۱۹

Crawler.GetXHtmlFromUri
شی Crawler ازچه نوعیه؟موقع اجرا خطا میگیره؟

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

شروع کار به این صورت هم می‌تواند 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: ۱۳۹۱/۰۶/۰۴ ۱۵:۴۳

با تشکر

ولی طبق تجربه خود من کد بالا هم کمک نمی‌کنه 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: ۱۳۹۱/۰۶/۰۴ ۱۶:۴۹

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: ۱۳۹۱/۰۶/۰۴ ۲۰:۴۶

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 خیله مهمه وگرنه ارور The remote server returned an error: (409) Conflict رو میده.
من با همین تابع یک سایت فارسی رو چک کردم و اروری نداد و متن فارسی قابل کوئری گرفتن بود.
کامل‌تر 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";
            request.AutomaticDecompression = 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: ۱۳۹۱/۰۶/۰۵ ۹:۳۷

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

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

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.
توبه 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: ۱۳۹۲/۰۲/۰۲ ۲۰:۵۸

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

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

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: ۱۳۹۲/۰۳/۲۳ ۱۰:۳۸

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: ۱۳۹۲/۰۳/۲۳ ۱۱:۲۹

- 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: ۱۳۹۲/۰۳/۲۳ ۱۲:۱۰

ببخشین که دوباره مزاحمتون شدم: وقتی من کد بالا رو در page load کپی میکنم زیر بعضی کلمات مانند , Crawler
Birthday, LargePhotoUri, GetAttributeValue,HtmlNode,Parse,خط قرمز میکشه ! چرا؟ آیا HtmlDocument 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 می‌کنید .
- 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 ۹:۶

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 فرمم هم زیر این کلمه GetXHtmlFromUri
مینویسه
generate metod stub for 'GetXHtmlFromUri in "crawel

وقتی من متد GetXHtmlFromUri را در کلاس crawel تعریف کردم جرا 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 سی‌شارپ

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 می‌توان صفحه را خواند مثال: **http://website.com?log=1731004**
سایت برای هر بازدید یک لاگ جدید می‌نویسد یعنی هر درخواست جدید در صورتی که لاگ معتبر نباشد به صفحه دیگر ارسال می‌کند.حال اطلاعات این صفحه را چطور می‌توان خواند ؟

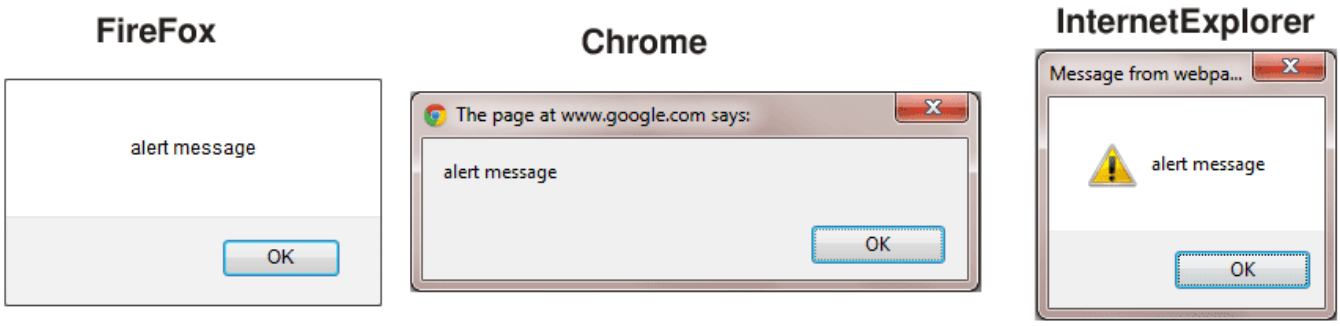
نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۲۶ ۱۷:۲۱

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

عنوان: **ایجاد alert,confirm,prompt هایی متفاوت با jQuery Impromptu**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۱۸ ۱۸:۴۰
آدرس: www.dotnettips.info
گروه‌ها: JavaScript, FireBug, jQuery

alert,confirm,prompt سه متد توکار JavaScript هستند که برای نمایش پیغام ، دریافت تایید و دریافت مقدار از کاربر هستند .

گرافیک این پیغام‌ها هم وابسته به مرورگر هستند و قابل تغییر نیستند . متن عنوان و دکمه‌ها هم با توجه به زبان سیستم عامل تعیین می‌شوند و قابل تغییر توسط برنامه نویس نیستند.



حال مواقعی پیش می‌آید که نیاز داریم پیغام هایی با گرافیک و عبارات متفاوت نمایش دهیم . برای رفع این نیاز می‌توانیم از پلاگین jQuery Impromptu استفاده کنیم . البته این پلاگین قابلیت‌های دیگری هم دارد که در این مقاله با آنها آشنا می‌شویم .

از ویژگی‌های این پلاگین می‌توان حجم کم (حدود 11 کیلوبایت) و قدرت شخصی سازی بالا اشاره کرد .

طرز استفاده به این شکل است :

```
$.prompt( msg , options )
```

msg می‌تواند یک string یا یک شئ از states باشد . string ارسال شده می‌تواند شامل کدهای html باشد .

یک شئ states هم شامل مجموعه ای از وضعیت‌های prompt است . برای مثال می‌توان یک prompt ایجاد کرد که مثل یک Wizard شامل چند مرحله باشد .

options هم تنظیماتی است که می‌توان مشخص کرد . تنظیماتی مثل : prefix,classes,persistent,timeout,...

msg :

گفتیم شئ states شامل وضعیت‌های مختلف prompt است . هر وضعیت (state) می‌تواند شامل بخش‌های زیر باشد :

html : مقدار Html وضعیت

buttons : یک شئ شامل متن و مقدار دکمه هایی که کاربر می‌تواند کلیک کند

focus : ایندکس دکمه‌ی focus شده در وضعیت

submit : تابعی که زمانی که یکی از دکمه‌های وضعیت انتخاب شود فراخوانی می‌شود .

اگر در این تابع false بازگشت داده شود یا متد preventDefault از event فراخوانی شود ، prompt باز می‌ماند . (روشی برای جلوگیری از بسته شدن prompt هنگام تغییر state یا اعتبار سنجی فرم)
همچنین شئ event شامل state (المنت state) و stateName (نام state) می‌باشد .

پیشفرض :

```
function(event, value, message, formVals){}
```

value مقدار دکمه ای است که بروی آن کلیک شده ، message مقدار html تعریف شده برای state است ، formVals هم در صورتی که در html تعریف شده برای state ، المنت‌های فرم وجود داشته باشد ، شامل نام/مقادیر آنها می‌باشد . (برای دریافت مقادیر فرم ، باید از نام المنت استفاده نمایید .)

position : مشخص کننده‌ی موقعیت state که شامل موارد زیر است :

```
position: { container: '#container', x: 0, y: 0, width: 0, arrow: 'lm' }
```

container : selector المنتی است که state باید در آن مکان قرار بگیرد .
x/y : موقعیت نسبی prompt نسبت به container
arrow : جهت نمایش فلش prompt است که می‌تواند یکی از این مقادیر باشد : tl, tc, tr, rt, tm, tb, br, bc, bl, lb, lm, lt.

نحوه تعریف یک wizard ساده با شئ states :

```
var tourSubmitFunc = function (e, v, m, f) {
    if (v === -1) {
        $.prompt.prevState();
```

```
        return false;
    } else if (v === 1) {
        $.prompt.nextState();
        return false;
    }
};

var states =
    {
        state0:
            {
                html: "State1",
                buttons: { Next: 1 },
                //position: { container: '#container', x: 10, y: 0, width: 350, arrow: 'lm' },
                submit: tourSubmitFunc
            },
        state1:
            {
                html: "State2",
                buttons: { Prev: -1, Next: 1 },
                submit: tourSubmitFunc
            },
        state2:
            {
                html: "State3",
                buttons: { Prev: -1, Done: 0 },
                submit: tourSubmitFunc
            }
    };

$.prompt(states);
```

تا به اینجا با پارامتر اول prompt آشنا شدیم و فهمیدیم که می‌توانیم یک رشته یا یک شئ states به عنوان message به prompt ارسال کنیم .

options :

اکنون با optionهای prompt (پارامتر دوم) آشنا خواهیم شد .

توجه کنید که زمانی که یک رشته به prompt ارسال کنید ، مقادیر buttons,focus,submit از این تنظیمات دریافت می‌شود .
به عبارت دیگر ، زمانی که یک شئ states به prompt ارسال کنید ، از مقادیر فوق که در تنظیمات است ، استفاده نمی‌شود .

loaded

یک تابع که زمانی که prompt کامل بارگزاری شده فراخوانی می‌شود .

```
$.prompt("Message",
    {
        loaded: function() {
            alert("Prompt Loaded !");
        }
    });
```

submit

یک تابع که زمانی که یکی از دکمه‌های state کلیک شود ، فراخوانی می‌شود .
(زمانی اتفاق میوفتد که یک رشته به عنوان متن به prompt ارسال کرده باشید و زمانی که یک شئ از states ارسال می‌کنید ، هنگام کلیک دکمه‌های آنها ، این تابع فراخوانی نمی‌شود .)
پیشفرض :

```
function(event){}
```

statechanging

یک تابع که زمانی که یک state در حال تعویض شدن هست فراخوانی می‌شود .

پیشفرض :

```
function(event, fromStateName, toStateName){}
```

برای لغو تغییر state ، مقدار return false کنید یا متد preventDefault از event را فراخوانی کنید .

statechanged

یک تابع که زمانی که یک state در حال تعویض شدن هست فراخوانی می‌شود .

پیشفرض :

```
function(event, toStateName){}
```

callback

یک تابع که زمانی که (یکی از دکمه‌های prompt کلیک شود و) prompt بسته شود ، فراخوانی می‌شود .

پیشفرض :

```
function(event[, value, message, formVals]){}
```

سه پارامتر آخر تنها زمانی که یک دکمه‌ی prompt کلیک شده باشد موجود هستند .

buttons

یک شئ شامل مجموعه ای از دکمه‌ها .

پیشفرض :

```
{ Ok : true }
```

شکل دیگر تعریف دکمه به این شکل است :

```
[
    {title: 'Hello World',value:true},
    {title: 'Good Bye',value:false}
]
```

prefix

یک پیشوند برای همه cssها class و idهای المنت‌های html که توسط prompt ایجاد می‌شود .

پیشفرض : jqi

classes

یک css class که به بالاترین سطح prompt داده می‌شود .

در حالت پیشفرض مقداری ندارد .

focus

ایندکس دکمه‌ی focus شده

پیشفرض : 0

zIndex

zIndex اعمال شده بروی prompt .
پیشفرض : 999

useiframe
استفاده از یک iframe برای overlay در IE6
پیشفرض : false

top
فاصله‌ی prompt از بالای صفحه
پیشفرض : 15%

opacity
میزان شفافیت لایه‌ی ای که صفحه را پوشانده است .
پیشفرض : 0.6

overlayspeed
سرعت نمایش افکت fadeOut , fadeIn لایه‌ی پوشاننده .
مقادیر قابل قبول : "slow", "fast", number(milliseconds)
پیشفرض : "slow"

promptspeed
سرعت نمایش prompt .
مقادیر قابل قبول : "slow", "fast", number(milliseconds)
پیشفرض : "fast"

show
نام یک متد jQuery برای animate کردن نمایش prompt .
مقادیر قابل قبول : "slideDown","fadeIn","show", ...
پیشفرض : "promptDropIn"

persistent
بسته شدن prompt زمانی که بروی لایه‌ی fade کلیک شود .
false : بسته شدن
پیشفرض : true

timeout
مدت زمانی که پس از آن ، prompt بصورت خودکار بسته می‌شود . (milliseconds)
پیشفرض : 0

returns :

مقدار بازگشتی متد prompt ، یک شئ jQuery ، شامل همه‌ی محتویات تولید شده توسط prompt است .

Events using Bind :

استفاده از Eventها با بایند کردن

promptloaded

معادل loaded در optionها .

promptsubmit

هنگام submit شدن (کلیک شدن یکی از دکمه‌های) state فعال می‌شود .

promptclose

معادل callback در optionها .

promptstatechanging

معادل statechanging در optionها .

promptstatechanged

معادل statechanged در optionها .

ظرز بایند کردن یک event به شئ prompt :

```
var myPrompt = $.prompt( msg , options );
myPrompt.bind('promptloaded', function(e){});
```

: **Helper Functions**

(jQuery.prompt.setDefaults(options

تعیین مقادیر پیشفرض برای همه‌ی promptها .

```
jQuery.prompt.setDefaults({
prefix: 'myPrompt',
show: 'slideDown'
});
```

(jQuery.prompt.setStateDefaults(options

تعیین مقادیر پیشفرض برای stateها .

```
jQuery.prompt.setStateDefaults({
buttons: { Ok:true, Cancel:false },
focus: 1
});
```

()jQuery.prompt.getCurrentState

یک شئ jQuery از state جاری برمی‌گرداند .

()jQuery.prompt.getCurrentStateName

نام state جاری را برمی‌گرداند .

(jQuery.prompt.getStateContent(stateName

یک شئ jQuery از state مشخص شده برمی‌گرداند .

(jQuery.prompt.goToState(stateName, callback
state مشخص شده را در prompt نمایش می‌دهد .
callback تابعی است که بعد از تغییر state فراخوانی می‌شود .

(jQuery.prompt.nextState(callback
prompt را به state بعدی منتقل می‌کند .

(jQuery.prompt.prevState(callback
prompt را به state قبلی منتقل می‌کند .

()jQuery.prompt.close
prompt را می‌بندد .

حال که با این پلاگین آشنا شدیم ، سه دستور جاوا اسکریپتی که در ابتدای مقاله ذکر کردیم را با این پلاگین پیاده سازی می‌کنیم .

alert

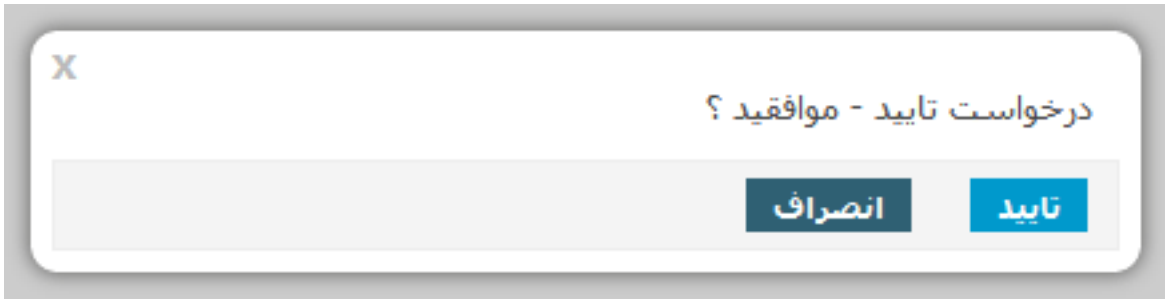
```
$.prompt("یک پیام تستی",
    {
        prefix: 'dnt',
        buttons: { 'تایید': true }
    });
```

نتیجه :

confirm

```
$.prompt("درخواست تایید - موافقید ؟",
    {
        prefix: 'dnt',
        buttons: { 'تایید': true, 'انصراف': false }
    });
```

نتیجه :



prompt

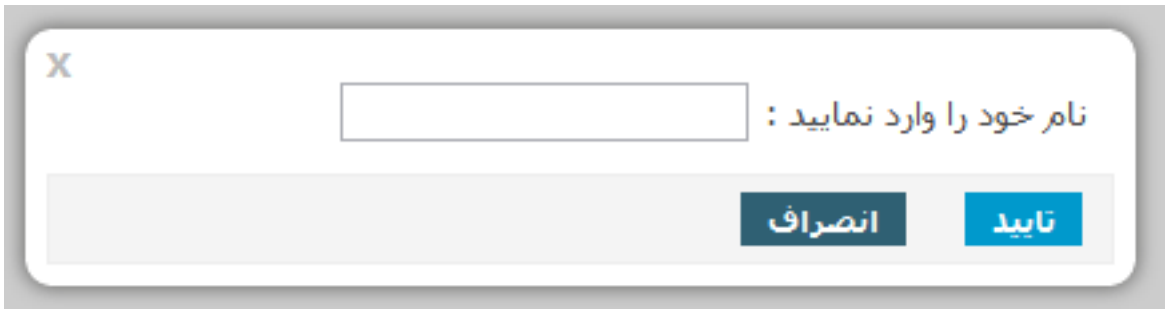
یک فرم html مخفی برای نمایش در prompt :

```
<div class="prompt-content" style="display: none;">
    <span>نام خود را وارد نمایید : </span>
    <span>
        <input type="text" name="name" />
    </span>
</div>
```

نمایش prompt :

```
$.prompt(
    $(".prompt-content").html(),
    {
        prefix: 'dnt',
        buttons: { 'تایید': true, 'انصراف': false }
    });
```

نتیجه :



در این سه مثال آخر ، از یک css متفاوت استفاده کردیم . این پلاگین یک سری کلاس دارد که نام این کلاس‌ها از ترکیب مقدار prefix که در option مشخص کردیم حاصل می‌شود .
برای مثال اگر مقدار prefix را برابر با dnt قرار بدهیم ، برای استایل دهی متن پیام باید از کلاس div.dnt .dntmessage استفاده کنید .
همانطور که در سه مثال آخر مشاهده کردید ، تغییری در استایل prompt داشتیم که با تغییر دادن استایل‌های مورد نظر انجام شد .

برای تهیه‌ی یک قالب سفارشی باید selector المنت هایی که نیاز به تغییر دارند را از قالب پیشفرض پیدا کنید ، سپس قسمت prefix از selector را به نام قالب خودتان تغییر بدید و استایل را ویرایش کنید . سپس هنگام ایجاد prompt ، مقدار prefix را برابر نام قالب قرار دهید .

مثلا می‌خواهید یک قالب به نام dnt داشته باشید . می‌خواهید متن قالب راست به چپ باشد .

```
div.dnt .dntmessage {
     color: #444444;
     line-height: 20px;
     padding: 10px;
/*       Edited      */
     direction:rtl;
     text-align:right;
}
```

البته راه بهتری هم هست که نیاز به آشنایی با فایرباگ دارد . در این روش ابتدا کل قالب jqi (موجود در قالب پیشفرض) را در برنامه خود کپی می‌کنیم ، مقادیر jqi را با نام قالب جایگزین می‌کنیم ، مقدار prefix را در prompt برابر با نام قالب قرار می‌هیم . اکنون در FireFox یک prompt ایجاد می‌کنیم و توسط فایرباگ استایل هایی که با نام قالب بروی prompt اعمال شده‌اند را مطابق سلیقه تغییر می‌دهیم . در مرحله آخر به تب CSS در فایرباگ می‌رویم و کل استایل‌های مربوط به قالب را کپی و جایگزین استایل قبلی در برنامه می‌کنیم .

قالبی که بنده برای سه دستور فوق استفاده کردم (dnt) ، به این شکل است :

```
/*    Start : DotNetTips Theme     */

.dntfade {
     background-color: #AAAAAA;
     position: absolute;
}

div.dnt {
     background-color: #FFFFFF;
     border-radius: 10px 10px 10px 10px;
     border: 1px solid #FFFFFF;
     box-shadow: 0px 0px 10px 1px #6D6D6C;
     font-family: tahoma;
     font-size: 11px;
     padding: 7px;
     position: absolute;
     text-align: left;
     width: 400px;
}

div.dnt .dntcontainer {
     font-size: small;
}

div.dnt .dntclose {
     color: #BBBBBB;
     cursor: pointer;
     font-weight: bold;
     position: absolute;
     top: 4px;
     width: 18px;
}

div.dnt .dntmessage {
     color: #444444;
     line-height: 20px;
     padding: 10px;
}

div.dnt .dntbuttons {
```

```
     background-color: #F4F4F4;
     border: 1px solid #EEEEEE;
     padding: 5px 0px;
     text-align: right;
}

div.dnt button {
     background-color: #2F6073;
     border: 1px solid #F4F4F4;
     color: #FFFFFF;
     font-size: 12px;
     font-weight: bold;
     margin: 0px 10px;
     padding: 3px 10px;
}

div.dnt button:hover {
     background-color: #728A8C;
}

div.dnt button.dntdefaultbutton {
     background-color: #0099CC;
}

.dnt_state {
     direction: rtl;
     text-align: right;
}

.dnt_state button {
     font-family: tahoma;
}

.dntwarning .dnt .dntbuttons {
     background-color: #CCDDFF;
}

.dnt .dntarrow {
     border: 10px solid transparent;
     font-size: 0px;
     height: 0px;
     line-height: 0;
     position: absolute;
     width: 0px;
}

.dnt .dntarrowtl {
     border-bottom-color: #FFFFFF;
     left: 10px;
     top: -20px;
}

.dnt .dntarrowtc {
     border-bottom-color: #FFFFFF;
     left: 50%;
     margin-left: -10px;
     top: -20px;
}

.dnt .dntarrowtr {
     border-bottom-color: #FFFFFF;
     right: 10px;
     top: -20px;
}

.dnt .dntarrowbl {
     border-top-color: #FFFFFF;
     bottom: -20px;
     left: 10px;
}

.dnt .dntarrowbc {
     border-top-color: #FFFFFF;
     bottom: -20px;
     left: 50%;
     margin-left: -10px;
}

.dnt .dntarrowbr {
     border-top-color: #FFFFFF;
     bottom: -20px;
```

```
     right: 10px;
}

.dnt .dntarrowlt {
     border-right-color: #FFFFFF;
     left: -20px;
     top: 10px;
}

.dnt .dntarrowlm {
     border-right-color: #FFFFFF;
     left: -20px;
     margin-top: -10px;
     top: 50%;
}

.dnt .dntarrowlb {
     border-right-color: #FFFFFF;
     bottom: 10px;
     left: -20px;
}

.dnt .dntarrowrt {
     border-left-color: #FFFFFF;
     right: -20px;
     top: 10px;
}

.dnt .dntarrowrm {
     border-left-color: #FFFFFF;
     margin-top: -10px;
     right: -20px;
     top: 50%;
}

.dnt .dntarrowrb {
     border-left-color: #FFFFFF;
     bottom: 10px;
     right: -20px;
}

/*    End : DotNetTips Theme     */
```

برای مشاهده‌ی مثال‌های بیشتر به صفحه‌ی اصلی jQuery Impromptu مراجعه نمایید .

نظرات خوانندگان

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۱/۰۶/۱۹ ۱۰:۳۶

با سلام و تشکر از شما

نمونه پست ارسالی شما در این آدرس نیز موجود است لطفا تفاوت کارایی‌های این دو را بفرمائید و از طرفی اگر امکان دارد نحوه استفاده آن در دات نت را برای تائید و دریافت ورودی توضیح دهید باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۹ ۲۳:۵۷

- تفاوت مهم آن در حجم کمتر است. jQuery-UI نسبتا حجم بالایی دارد و اگر صرفا قرار است پیام کوتاهی به کاربری نمایش داده شود، روش فوق حداقل از لحاظ حجم (11K) مقرون به صرفه‌تر است.
- استفاده از نتایج آن هم مطابق توضیحات مفصل مطلب جاری، در قسمت callback آن باید صورت گیرد. برای مثال

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۴:۵۹

ممنون،
یه سوال بنده یه فرم ثبت نام دارم که با کلیک بر روی Botton اطلاعات در دیتابیس درج می‌شود حالا می‌خوام بعد از ثبت اطلاعات یه alert به کاربر نشون بده که اطلاعات با موفقیت ثبت شد ولی مشکل اینجاست که وقتی کاربر روی Button کلیک میکنه چون صفحه PostBack میشه دیگه پیغام به کاربر نمایش داده نمیشه،اگه امکان داره بنده رو راهنمایی بفرمائید.
دقیقا مثل همین قسمت ارسال نظر سایت.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۵:۵۹

این مطلب می‌تونه مفید و مرتبط باشه.

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۷/۱۲ ۲:۳۲

سلام. یک linkbutton دارم تو گرید ویو که کار حذف رو انجام میده. در حالت پیش فرض با دستور

```
return confirm('')
```

و انتساب اون به خاصیت onClientClick میشد یک confirm ساده قبل حذف داشت. حالا وقتی میخام از این کنترل برای این کار استفاده کنم کار نمیکنه؟ میشه راهنمایی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۲ ۹:۴۶

مثالی که در این کامنت معرفی شد مرتبط به GridView هم هست (انتهای مقاله).

نویسنده: omid

تاریخ: ۱۳۹۱/۱۰/۱۲ ۸:۱۹

سلام
با تشکر از راهنمایی شما برای کنترل‌ها من از این کنترل می‌خواستم استفاده کنم ولی اجرا نشد اگه راهنماییم کنید ممنون میشم .

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
<title></title>

<link href="jquery-impromptu.css" media="all" rel="stylesheet" type="text/css" />
<script src="jquery-1.8.3.min.js" type="text/javascript"></script>
<script src="jquery-impromptu.js" type="text/javascript"></script>
<script type="text/javascript">

$(function(){
$show.click(function(e){
$.prompt("Hello World!");
});
});
});

</script>
</head>

<body>
<button class="show">ShowPrompt</button>
</body>
</html>
```

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۰:۲۴

- کد شما تعداد {} متوازنی ندارد. دو } بازه شده، سه تا بسته.
- به نظر با آخرین نگارش jQuery سازگار نیست. به صفحه خطای مرورگر مراجعه کنید تا خطای شناخته نشدن نوع مرورگر قابل مشاهده باشد. با نگارش‌های پایین‌تر مشکلی ندارد .

نویسنده: omid
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۰:۵۶

سلام
خیلی ممنون بابت راهنماییتون و آشنایی با سایت http://jsfiddle.net

نویسنده: neda
تاریخ: ۱۳۹۲/۰۱/۰۹ ۱۲:۲۶

ممنون از سایت خیلی خوب و مفیدتون
ممنون میشم اگه امکانش هست یه سمپل ساده و کوچیک از این کار هم در اختیار بگذارید
من از کدها استفاده کردم ولی موفق نشدم خروجی صحیح بگیرم
ممنونم

نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۱/۰۹ ۱۷:۱۷

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

```
<head>
<link rel="stylesheet" media="all" type="text/css" href="http://trentrichardson.com/Impromptu/jquery-
impromptu.css" />
<script type="text/javascript" src="http://code.jquery.com/jquery-1.9.0.min.js"></script>
<script type="text/javascript" src="http://trentrichardson.com/Impromptu/jquery-impromptu.js"></script>
</head>
<body>
<button class="show">ShowPrompt</button>
<script type="text/javascript">
$(function(){
   $(".show").click(function(e){
      $.prompt("Hello World!");
   });
});
</script>
</body>
</html>
```

نویسنده: سمیرا
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۳:۴۴

سلام؛ وقتی با استفاده از jQuery Impromptu یک inputbox ایجاد می‌کنیم و کاربر اطلاعاتی رو توی inputbox وارد می‌کنه، چجوری می‌تونیم مقداری که کاربر وارد کرده رو برگردونیم و ازش استفاده کنیم ؟

نویسنده: مژده
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۴:۲۳

سلام
من هم همین مشکلو داشتم
از این روش استفاده کردم جواب داد :

```
var myval = m.find('#name').val();
alert(myval);
```

نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۸:۴۴

سلام؛

طبق مطلب ارائه شده در مقاله:

تابع submit :

```
function(event, value, message, formVals){}
```

value مقدار دکمه ای است که بروی آن کلیک شده ، message مقدار html تعریف شده برای state است ، formVals هم در صورتی که در html تعریف شده برای state ، المنت‌های فرم وجود داشته باشد ، شامل نام/مقادیر آنها می‌باشد . (برای دریافت مقادیر فرم ، باید از نام المنت استفاده نمایید .)

خب حالا مشابه مثال زیر عمل کنید:

```
<div class="prompt-content" style="display: none;">
    <span>ایمیل خود را وارد نمایید : </span>
    <span>
        <input type="text" name="user_email" />
    </span>
```

```
</div>
```

```
$.prompt(
    $(".prompt-content").html(),
    {
        submit: function (e, v, m, f) {
            var userEmail = f["user_email"];
            console.log(userEmail);
        }
    });
```

نویسنده: مژده
تاریخ: ۱۳۹۲/۰۳/۰۹ ۲۱:۱۳

سلام
خسته نباشید
من توی سایتم از ajax استفاده کردم و برای حذف یه رکورد اول از کاربر تاییدیه گرفتم . اگه کاربر روی دکمه‌ی بله کلیک کنه ، عملیات شروع میشه و رکورد حذف میشه .
آخر سر توی تابعی که نتیجه برمیگرده توش (handleServerResponse)، بر اساس نتیجه‌ی برگشتی پیغام مناسبی بایدنمایش داده بشه. اما هیچ پیغامی نشون داده نمیشه
خیلی چیزا رو آزمایش کردم و آخر سر به این نتیجه رسیدم که چون دو تا پیغام میخواد پشت سر هم نمایش داده بشه ، این مشکل پیش میاد :(
حتی وقتی که پیغام تایید حذف رو برداشتم ، دیدم که پیغام دوم میاد !
واااقعا نمیدونم علتش چیه ! ممنون میشم کمکم کنید.
اینم قسمتی از کدمه :

```
function deleteFile(location, filename)//حذف فیزیکی و منطقی فایل
{
    $.prompt("آیا برای حذف فایل موجود اطمینان دارید ؟", {
        title: '<img src="Images/warning.png" />',
        buttons: { "بله": true, "خیر": false },
        focus: 2,
        submit: function (e, v, m, f) {
            if (v == true) {
                var getdate = new Date();  //Used to prevent caching during ajax call
                if (xmlhttp) {
                    $("p#vtip").fadeOut("slow").remove(); //محو شدن tooltip
                    var id = document.getElementById("id").value;
                    var i = '3';
                    params = "id=" + id + "&i=" + i + "&filename=" + filename + "&location=" +
location;
                    xmlhttp.open("POST", "FetchData/dbSearchDocument1.php", true);
                    xmlhttp.onreadystatechange = handleServerResponse4;
                    xmlhttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
                    xmlhttp.send(params);
                }
            }
            else {
            }
        }
    });
}
//-----------------------------------------------------------------------
function handleServerResponse4() {
    if ((xmlhttp.readyState == 1) || (xmlhttp.readyState == 2) || (xmlhttp.readyState == 3))//loading
    {
        document.getElementById("content").innerHTML = "<img src='Images/bigloading.gif'/>";
    }
    else if (xmlhttp.readyState == 4) //ready
    {
        if (xmlhttp.status == 200) {
            if (xmlhttp.responseText == 1) {
                $.prompt(". حذف فایل با موفقیت انجام شد ", {
```

```
                    title: '<img src="Images/warning3.png" />',
                    buttons: { "بستن": true }
                });
            }
            else {
                $.prompt("! حذف فایل با خطا مواجه شده است", {
                    title: '<img src="Images/warning2.png" />',
                    buttons: { "بستن": true }
                });
            }
        }
        else {
            alert("Error during AJAX call. Please try again");
        }
    }
}
//---------------------------------------------------------------------
```

نویسنده: مژده
تاریخ: ۱۳۹۲/۰۳/۰۹ ۲۱:۵۴

من الان یک بار دیگه این مطلبو خوندم و به نظرم رسید که باید از callback استفاده کنم.
اما حتی این توی سایتم جواب نمیده :

```
function mycallbackfunc(v,m){
      $.prompt('i clicked ' + v);
}
$.prompt('Example 8',{ callback: mycallbackfunc });

});
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۰ ۰:۱۶

زمانیکه از jQuery استفاده می‌کنید، دیگر نباید از Ajax خام استفاده کنید. خود jQuery تابع سازگار با انواع و اقسام مرورگرها رو برای کار با Ajax داره. این مثال تست شده و مشکلی نداره:

```
using System.Web.Mvc;

namespace jQueryImpromptu.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string postId, string filename)
        {
            return Content("ok");
        }
    }
}
```

```
@{
    ViewBag.Title = "Index";
    var postUrl = Url.Action(actionName: "Index", controllerName: "Home");
}
<h2>
    Index</h2>
<input type="button" id="btnDelete" value="delete" />
```

```
<div id="infoDiv">
    Some Info ....
</div>
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $('#btnDelete').click(function () {
                $.prompt("آیا برای حذف اطلاعات موجود اطمینان دارید ؟",
             {
                 title: 'Title',
                 buttons: { "بله": true, "خیر": false },
                 focus: 2,
                 submit: function (e, v, m, f) {
                     //debugger;
                     if (!v) {
                         return;
                     }
                     var postId = 10;
                     var fileName = "test.jpg"
                     $.ajax({
                         type: "POST",
                         url: '@postUrl',
                         data: JSON.stringify({ postId: postId, fileName: fileName }),
                         contentType: "application/json; charset=utf-8",
                         dataType: "json",
                         complete: function (xhr, status) {
                             var data = xhr.responseText;
                             if (status === 'error' || !data || data == "nok") {
                                 $.prompt("! حذف فایل با خطا مواجه شده است",
                                    {
                                        title: 'Title',
                                        buttons: { "بستن": true }
                                    });
                             }
                             else {
                                 $("#infoDiv").fadeOut("slow").remove();
                             }
                         }
                     });
                 }
             });
            });
        });
    </script>
}
```

نویسنده: مژده
تاریخ: ۱۳۹۲/۰۳/۱۱ ۱۱:۳۵

سلام
ممنون که جواب دادید. خیلی لطف کردید
من تونستم با استفاده از یه js دیگه ، با callback مشکلمو حل کنم . اما حالا این js ، عنوان (title) نداره ! D:

از این سایت استفاده کردم :
/http://www.shiguenori.com/material/jquery.impromptu

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۱ ۱۱:۴۴

کدی که نوشته شد بر اساس نسخه GitHub بود.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۰۶ ۲۱:۱۱

سلام ، اگه بخوایم مقدار ورودی‌های اکشنمون به صورت داینامیک باشه چیکار باید کنیم؟
منظورم به عنوان مثال مقادیر این خط در مثال شماست

```
var postId = 10;
var fileName = "test.jpg"
```

چنین مدلی دارم که میخوام دکمه‌ی حذفش به صورت Jquery Ajax باشه

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Title)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.IsRead)
        </td>
        <td>
            @Html.ActionLink("حذف پیام", MVC.User.ActionNames.DeleteMessageSend, MVC.User.Name, new{
id=item.Id }, new { @class="btn btn-danger" })
        </td>
    </tr>
}
```

ولی مقدار id رو باید به اکشن بفرستم ! اونم تو یه حلقه Foreach . باید کدهای جی کوئری تو حلقه باشه؟ به نظرتون راه حل چیه؟ ممنون

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۰۶ ۲۱:۳۶

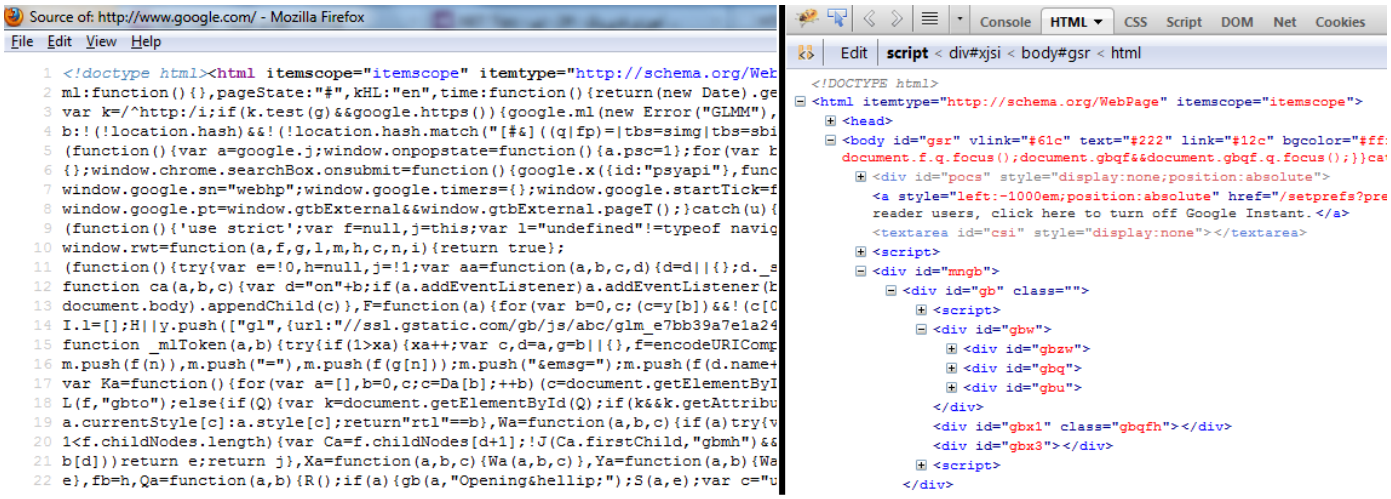
ببخشید تو این مطلب توضیح دادید.

عنوان: **آموزش فایرباگ - #5 - HTML Panel**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۸/۲۳ ۱۸:۱۵
آدرس: www.dotnettips.info
برچسب‌ها: JavaScript, FireBug, CSS, XPATH, HTML

در این سری از مقالات آموزش FireBug ، به صورت ترتیبی پیش رفتیم و ابتدا توضیحات تقریبا 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 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.

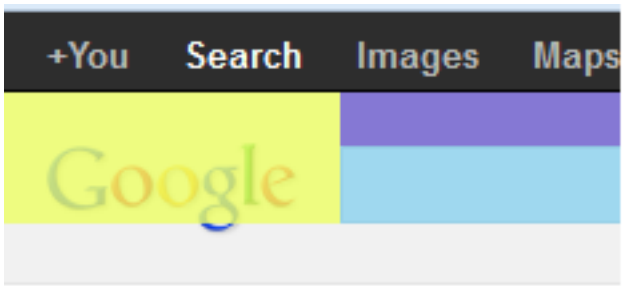
ویرایش HTML

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 کافیست 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 موردنظر کلیک کنید تا به حالت انتخاب دربیاید ، سپس بروی دکمه‌ی Edit در بالای پنل کلیک کنید. در نهایت بعد از انجام ویرایش ، مجددا بروی دکمه‌ی Edit کلیک کنید.

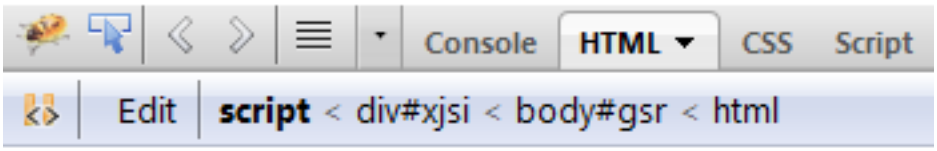
Node View

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 **Left Arrow و Right 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

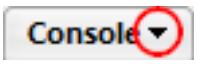
این دکمه برای ویرایش مستقیم محتوای یک تگ بکار می‌رود
نکته‌ی جالب در ویرایش محتوا در فایرباگ این است که تغییرات در لحظه اعمال می‌شوند و نیاز به عمل بروزرسانی جداگانه نیست. برای مثال در همین قسمت Edit ، با هر ویرایش محتوا ، تغییرات در لحظه اعمال می‌شوند.

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 هرکدام ، 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 دانشتن آنها بسیار به شما کمک خواهد کرد.
این منو با کلیک کردن بروی فلش تب (



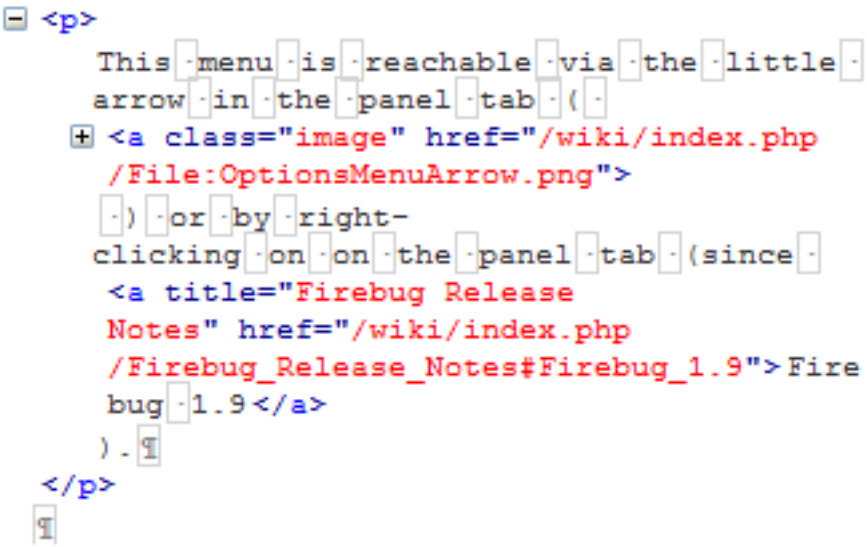
) یا راست کلیک کردن بروی تب ظاهر می‌شود.

Show Full Text

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

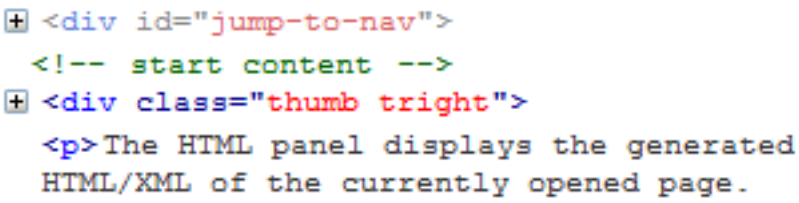
Show White Space

در صورت فعال بودن ، فضاهای خالی ، کرکترهای خط جدید و ... را نمایش می‌دهد.



Show Comments

در صورت فعال بودن ، کامنت‌ها را هم نمایش می‌دهد



سه گزینه ی Show Entities As Names ، Show Entities As Symbols و Show Entities As Unicode ، 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.

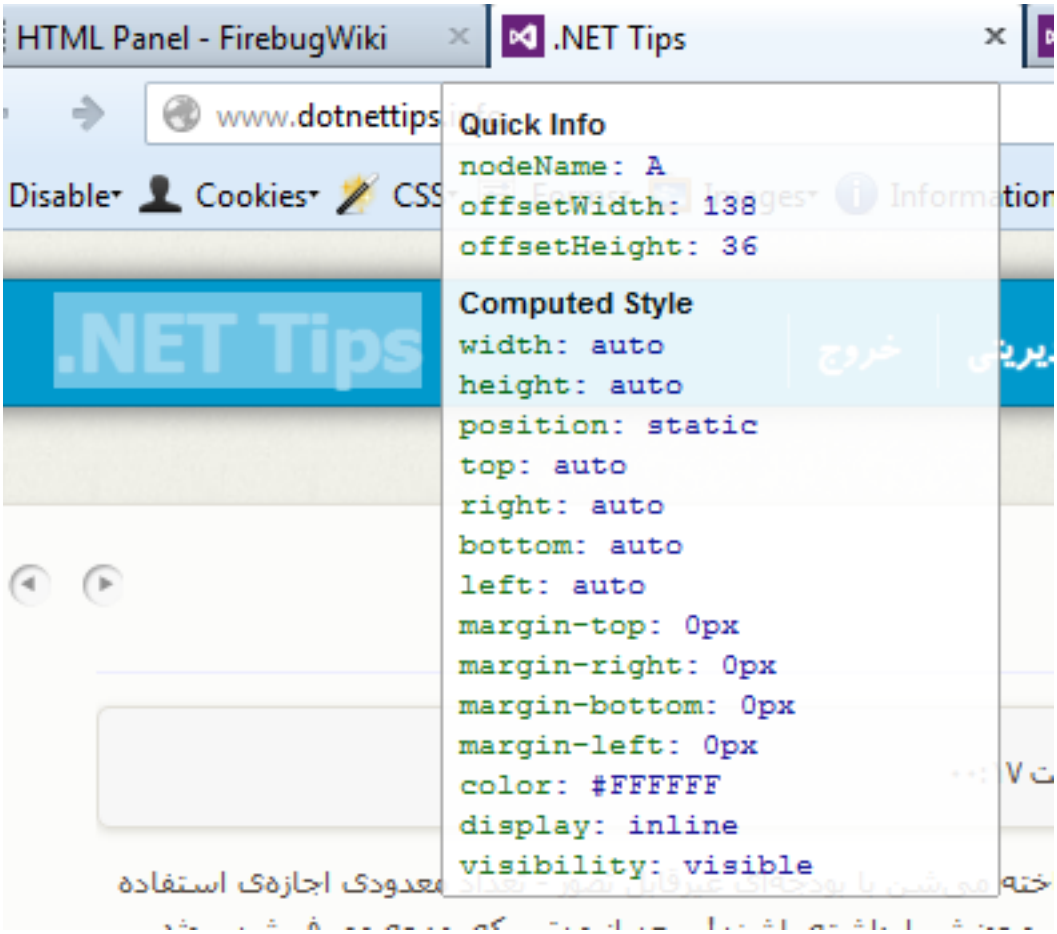
(فعال بودن این گزینه هنگام بررسی سایت هایی که اسلایدر یا سیستم هایی مشابه دارند ، باعث میشه که نتوانید بروی قسمت‌های سایت تمرکز کنید و مدام اسکرول به قسمت تغییرات منتقل می‌شود.)

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.

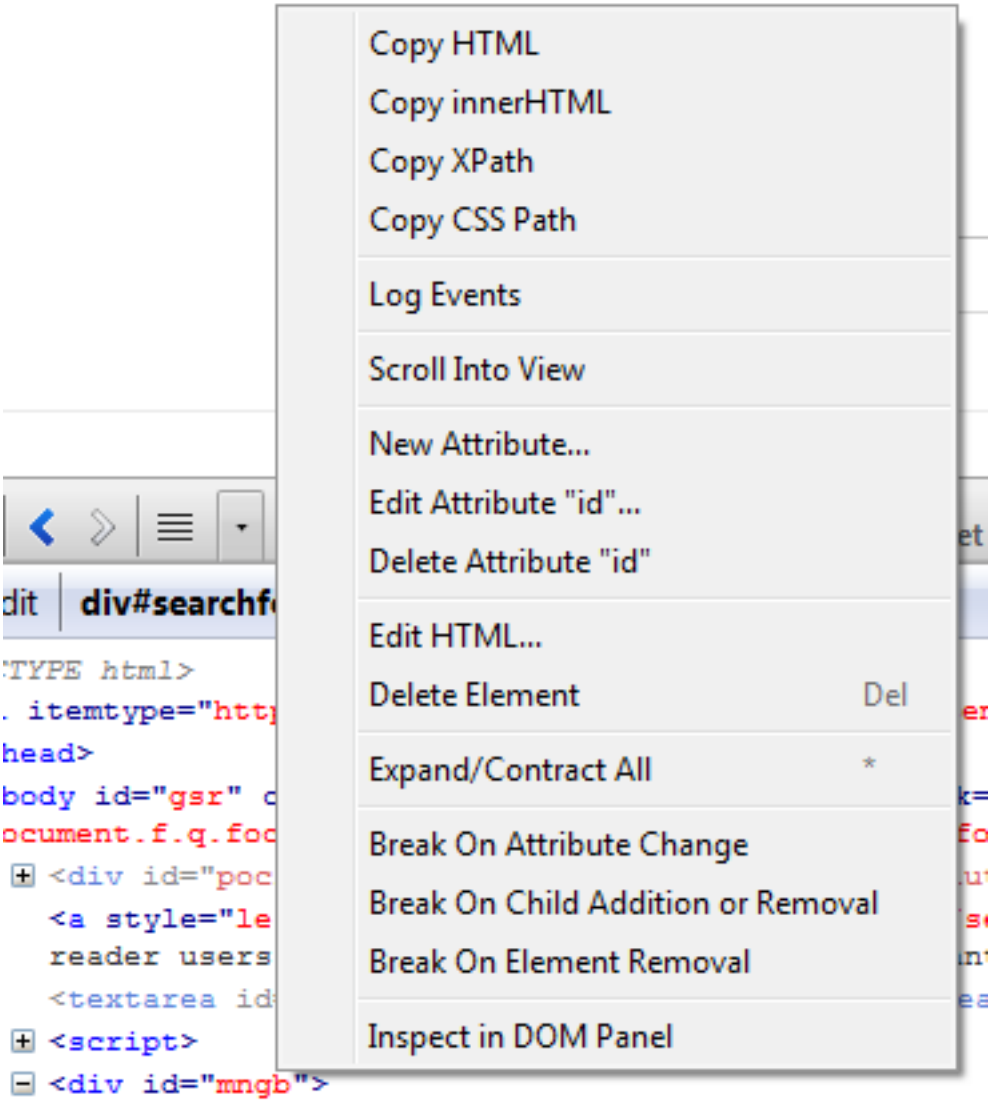
Show Quick Info Box

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 ، مجددا 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 <link> , <style> , <script>)
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 تگِ 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.

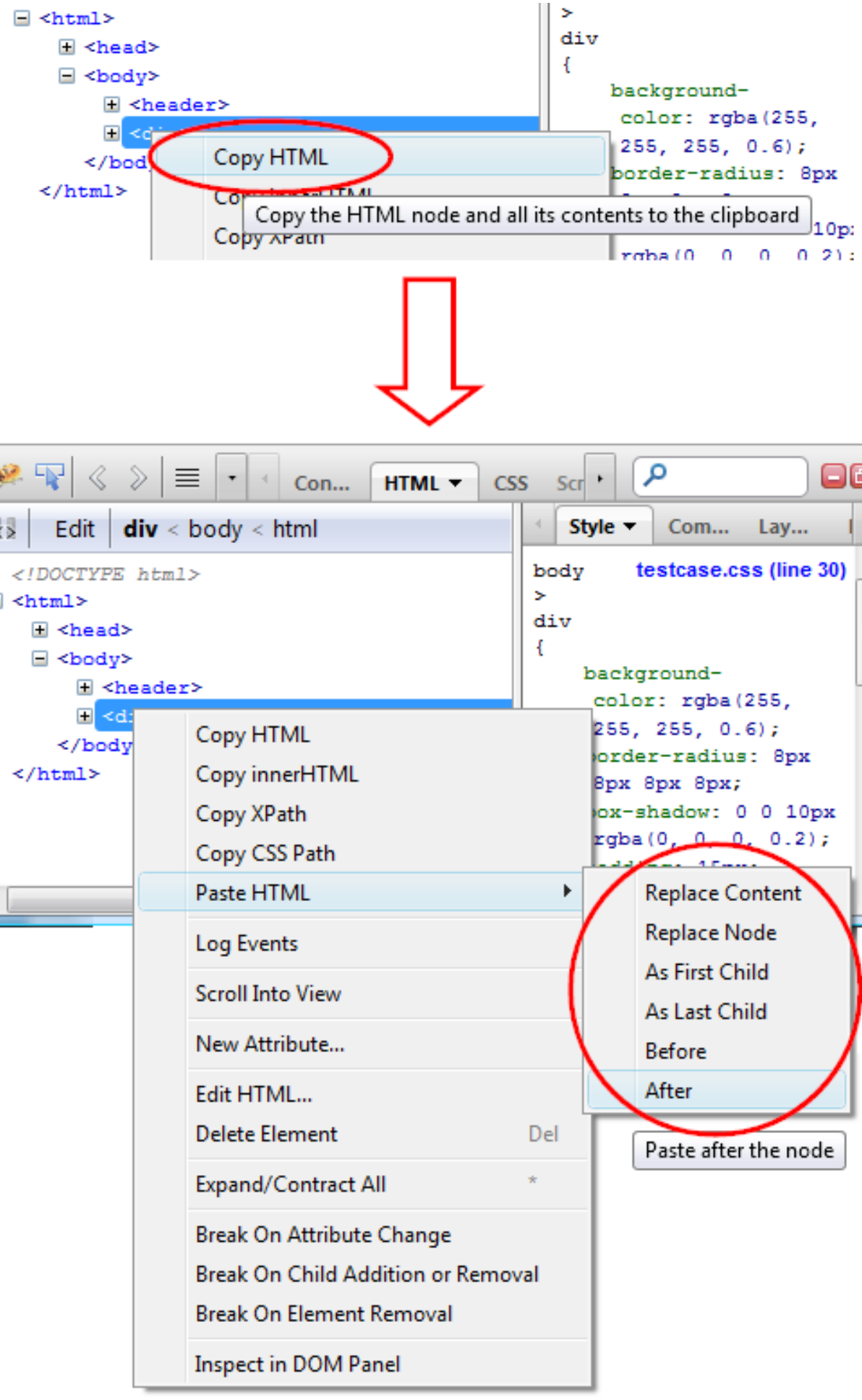
Inspect in DOM Tab
تگ فعلی را در پنل DOM ، برای بررسی باز می‌کند.

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 ساده‌تر کرده.
در حالی که قبلا برای این کار می‌بایست تگ را به حالت ویرایش درآورد و محتویات را اضافه کرد.


<html>
<head>
<body>
<header>
</body>
</html>
Copy HTML
Copy the HTML node and all its contents to the clipboard
Copy XPath
Con...
HTML
CSS
Edit
div < body < html
<!DOCTYPE html>
Style
Com...
Lay...
body
testcase.css (line 30)
Copy HTML
Copy innerHTML
Copy XPath
Copy CSS Path
Paste HTML
Log Events
Scroll Into View
New Attribute...
Edit HTML...
Delete Element
Del
Expand/Contract All
*
Break On Attribute Change
Break On Child Addition or Removal
Break On Element Removal
Inspect in DOM Panel
Replace Content
Replace Node
As First Child
As Last Child
Before
After
Paste after the node

عنوان: آموزش فایرباگ - #6 - HTML Panel - Side Panels
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۸/۲۵ ۰:۴۵
آدرس: www.dotnettips.info
برچسب‌ها: FireBug, CSS, HTML

در قسمت قبل توضیحاتی در مورد تب HTML ارائه کردیم.
در این قسمت توضیحات کاملی در مورد پنل‌های جانبی داخل پنل HTML می‌دهیم.

Side Panels

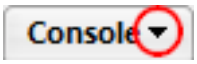
در پنل HTML درکنار ارائه امکاناتی برای مشاهده و کار با تگ‌های صفحه ، اطلاعات و امکانات دیگری هم برای تگ انتخاب شده در قسمت NodeView وجود دارد.
این امکانات در پنل هایی که سمت راست پنل اصلی قرار دارند گنجانده شده است که به ترتیب برای نمایش و ویرایش استایل‌ها ، مشاهده استایل‌های محاسبه شده ، مشاهده Layout یا آرایش و نمایش اطلاعات DOM تگ انتخاب شده در NodeView هستند.

Style - 1

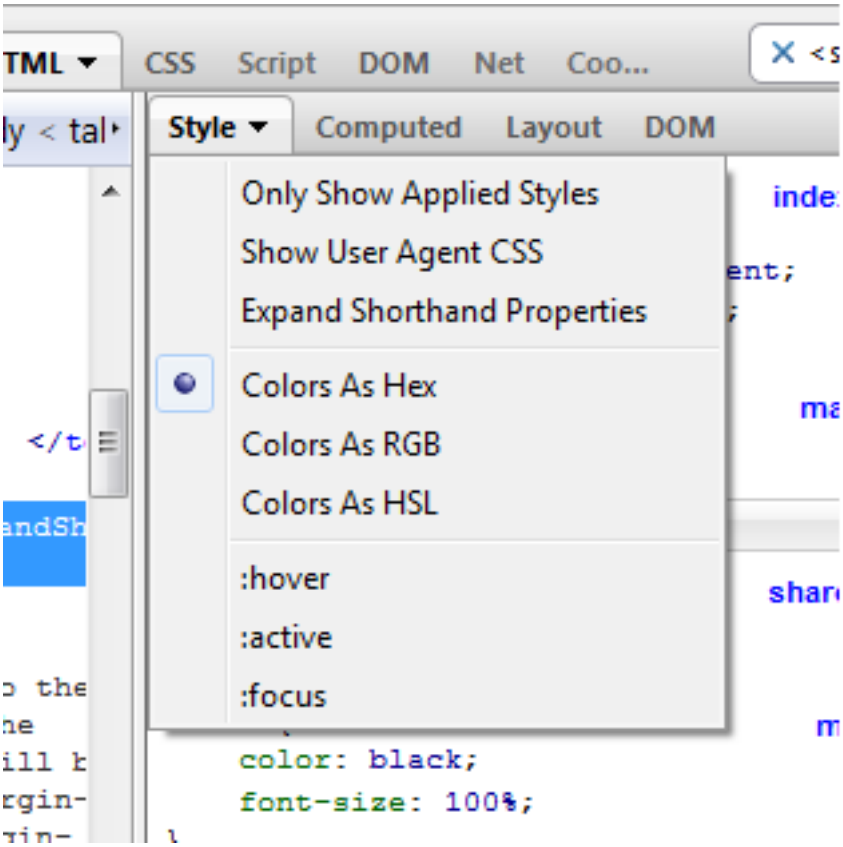
در این تب استایل هایی که در حال حاظر بروی تگ انتخاب شده اعمال شده اند ، نمایش داده می‌شود.
در صورتی که موس را بروی مقادیر استایل هایی که جلوه‌ی بصری دارند بگیرید ، یک پاپ‌آپ کوچک نمایان می‌شود که مقدار را نمایش می‌دهد.

Options Menu

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



) یا راست کلیک کردن بروی تب ظاهر می‌شود.



Only Show Applied Styles

در صورت انتخاب ، فقط استایل هایی که اعمال شده اند نمایش داده می‌شوند. (استایل‌های Overwrite شده نمایش داده نمی‌شوند.)

(این گزینه قابلیت خوبی است ، اما چندبار برای بنده پیش آمده که این مورد به اشتباه استایلی که اعمال شده بود را هم Overwrite شده در نظر گرفته بود. پس در هین طراحی استایل و کار با CSS اگر احیانا یکی از استایل هایتان وجود نداشت و از وجود آن اطمینان داشتید ، غیرفعال کردن این گزینه را امتحان کنید.)

Show User Agent CSS

با فعال کردن این گزینه ، استایل هایی که توسط مرورگر اعمال شده اند هم نمایش داده می‌شوند.

Expand Shorthand Properties

با فعال کردن این گزینه ، استایل هایی که بصورت کوتاه شده تعریف شده اند را بصورت گسترده و باز شده نمایش می‌دهد.

برای مثال ، دستور margin را بصورت margin-left , margin-bottom , margin-right , margin-top نمایش می‌دهد.

سه گزینه ی Colors As RGB ، Colors As Hex و Colors As HSL تعیین کننده‌ی فرمت نمایش رنگ‌ها هستند.

سه گزینه ی :hover ، active و :focus هم برای تعیین کلاس کاذب برای تگ جاری کاربرد دارند.

برای مثال شما می‌خواهید استایلی که یک لینک زمان موس برویش قرار دارد را بررسی کنید ، لینک را در NodeView انتخاب می‌کنید و سپس از گزینه‌ی :hover را فعال می‌کنید.

Panel

Element styles

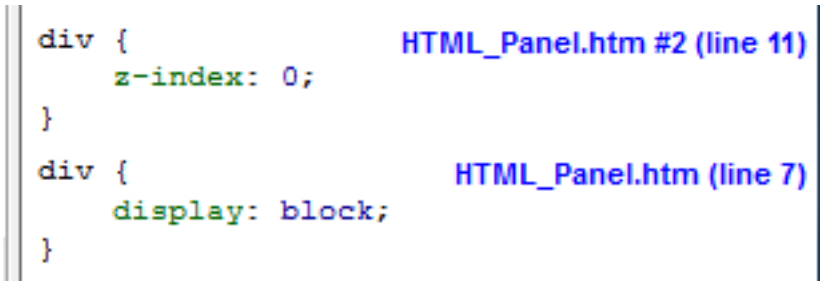
استایل هایی که بصورت inline (در خود تگ) تعریف شده اند هم در این قسمت نمایش داده می‌شود و نام rule آن element.style است.

Source Links

در بالا-راست هر بخش ، یک لینک قرار دارد که لینک فایل استایلی است که در همان قسمت وجود دارد و عددی که در پرانتز قرار دارد ، شماره خط استایل در همان فایل است.

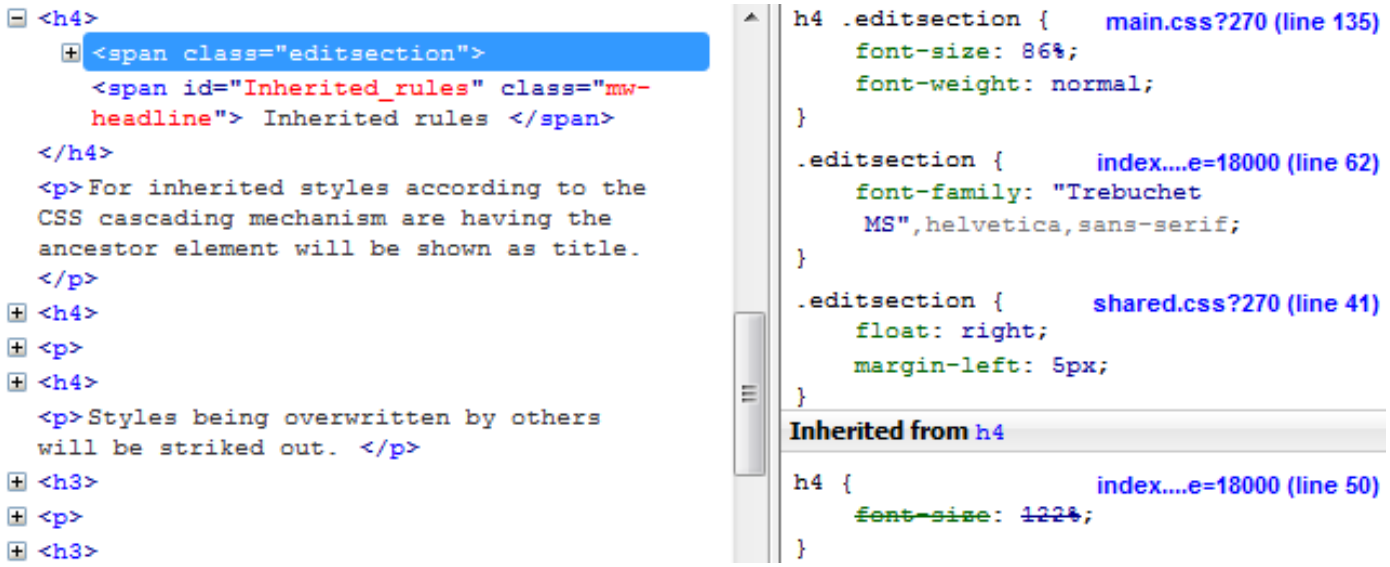
اگر نام فایل با نام صفحه‌ی جاری برابر باشد ، به معنی وجود استایل در تگ <style> در صفحه‌ی جاری است و شماره‌ی بعد از # هم ایندکس تگ <style> است.

(با کلیک بروی لینک فایل ، فایل در خط مورد نظر در پنل CSS نمایش داده می‌شود.)



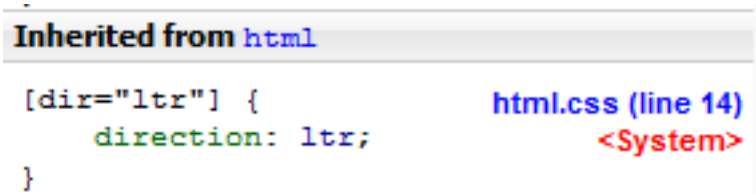
Inherited rules

ruleهای به ارث رسیده هم در قسمت‌های جداگانه به همراه استایل‌های به ارث رسیده نمایش داده می‌شود و تگی والد که استایل‌ها از آن به ارث رسیده اند هم در قسمت عنوان همان استایل‌ها نمایش قرار داده شده است. (با کلیک بروی آن ، در قسمت Nodeview انتخاب می‌شود.)



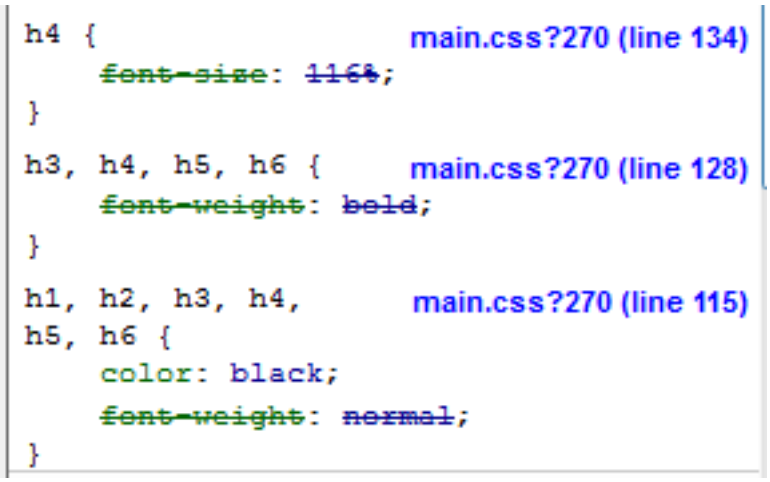
User agent rules

استایل هایی که توسط مرورگر اعمال شده اند (User agent rules) ، با عبارت <System> در زیر لینک منبع استایل ، مشخص شده اند.



Overwritten styles

استایل‌های overwrite شده ، با یک خط برویشان مشخص شده اند.

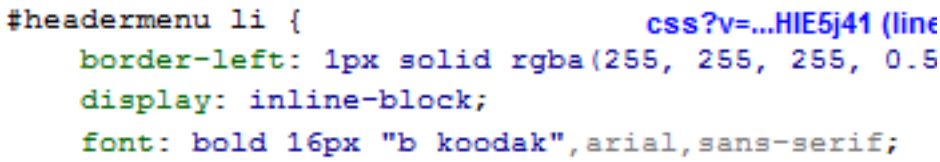


Inline editing

استایل‌های نمایش داده شده در این پنل را براحتی و با کلیک کردن بروی نام یا مقدار هر یک از دستورات می‌توانید تغییر دهید.
برای نوشتن دستورات و مقادیر آن‌ها می‌توانید از پیشنهادهای فایرباگ هم کمک بگیرید و با دکمه‌های Arrow Up و Arrow Down هم بین مقادیر مجاز حرکت کنید.
دستورات یا مقادیر نا صحیح در هین تایپ ، با رنگ قرمز و مقادیر صحیح با رنگ سبز مشخص می‌شوند.
(این امکان خیلی مفید است ، برای مثال می‌خواهید فونت‌های مختلف را برای یک استایل امتحان کنید ، دستور font-family را می‌نویسید و بعد از زدن Enter ، با دکمه های Arrow Up و Arrow Down در لحظه نتیجه‌ی اعمال فونت‌های مختلف و دردسترس را مشاهده می‌کنید و بهترین را بر می‌گزینید.
یا برای یافتن بهترین مقدار margin ، بعد از دستور margin ، زدن کلید Enter ، وارد کردن یک عدد برای شروع ، می‌توان باز هم با دکمه های Arrow Up و Arrow Down به سرعت تغییر را در صفحه مشاهده کرد.)

Rendered font highlighted

برای دستور font ، فایرباگ هوشمندانه عمل کرده و فونتی که در حال استفاده است را پررنگ می‌کند.
این امکان برای یافتن خطاهای متداول هنگام تعریف فونت‌های غیر سیستمی ، بسیار مفید است.



Context Menu

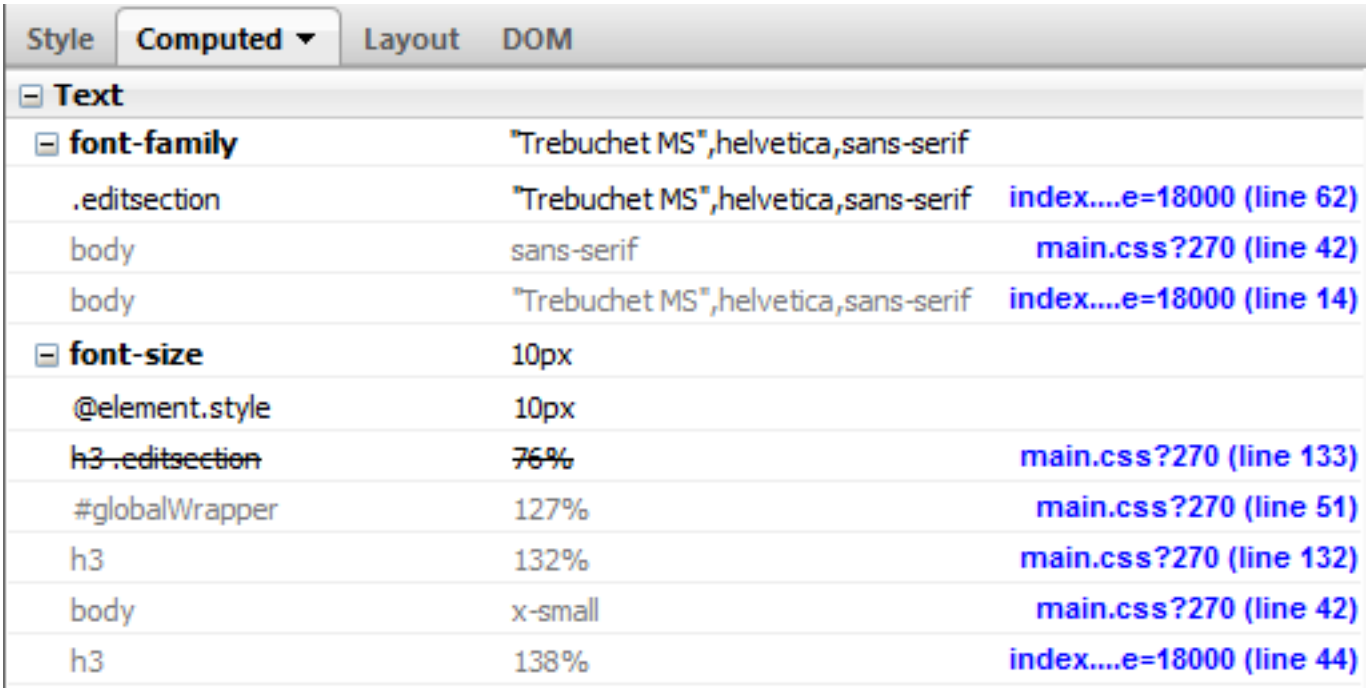
این منو زمانی که در پنل راست کلیک کنید ظاهر می‌شود و نسبت به منطقه (Context)ای که در آن راست کلیک کرده اید ، گزینه‌های متفاوتی را مشاهده خواهید کرد. در جدول زیر ، گزینه‌ها ، Contextشان و توضیح هر گزینه آمده است.

گزینه	Context	توضیحات
Copy Rule Declaration	CSS selector	CSS Rule فعلی را به همراه استایل هایش در clipboard کپی می‌کند
Copy Style Declaration	CSS selector	استایل‌های CSS Rule فعلی را در clipboard کپی می‌کند
Copy Location	source link	آدرس فایل تعریف CSS Rule را در clipboard کپی می‌کند
Open in New Tab	source link	آدرس فایل تعریف CSS Rule را در تب جدید باز می‌کند
Edit Element Style...	everywhere	امکان تعریف استایل‌های درون تگ (inline) را محیا می‌کند
Add Rule...	everywhere	یک Rule جدید ایجاد می‌کند CSS Rule هایی که در حال حاظر وجود دارد را هم پیشنهاد می‌دهد
Delete "<rule name>"	CSS selector	CSS Rule فعلی را حذف می‌کند

گزینه	Context	توضیحات
New Property...	CSS rule	یک استایل جدید به CSS Rule فعلی اضافه می‌کند
Edit "<property name>"...	CSS property	Property فعلی را به حالت ویرایش می‌برد راه دیگر ویرایش Property ، کلیک بروی آن است
Delete "<property name>"	CSS property	Property فعلی را حذف می‌کند
Disable "<property name>"	CSS property	Property فعلی را غیر فعال می‌کند را سریع‌تر ، کلیک کردن در ناحیه‌ی پشت Property ، بروی علامت قرمز رنگ است
Refresh	everywhere	محتویات پنل را بروز می‌کند
Inspect in DOM Panel	CSS rule	CSS Rule فعلی را در پنل DOM برای بررسی باز می‌کند
Inspect in CSS Panel	CSS rule	CSS Rule فعلی را در پنل CSS برای بررسی باز می‌کند
<Default Editor Name>	CSS rule	فایل تعریف استایل‌ها را در ادیتور تعریف شده باز می‌کند (این گزینه در صورت تعریف ادیتور در تنظیمات FireBug نمایش داده خواهد شد)

Computed - 2

دراین تب نتیجه‌ی پردازش استایل‌های ارائه شده توسط کاربر ، برای تگ مشخص شده در قسمت NodeView نمایش داده می‌شود. (مقادیر استایل هایی که در نهایت بروی تگ اعمال شده اند.)



Style Tracing

برای ردیابی استایل‌ها ، استایل‌ها به ترتیب اعمال شدنشان مرتب شده اند و اولین مقدار ، مقداری است که اعمال شده است.
مقادیر Overwrite بصورت خط کشیده شده و استایل‌های Overwrite شده بصورت خاکستری-کمرنگ نمایش داده می‌شوند.
هر استایل هم مانند تب Style ، یک لینک به منبع خود دارد.

Options Menu

Show User Agent CSS
در صورت انتخاب ، فقط استایل هایی که اعمال شده اند نمایش داده می‌شوند.

Sort alphabetically
در صورت انتخاب ، استایل‌ها به ترتیب الفبا ، و درصورت عدم انتخاب بصورت گروه بندی نمایش داده می‌شوند.

Show Mozilla Specific Styles
در صورت انتخاب ، استایل‌های مخصوص Mozilla را نمایش می‌دهد. (استایل هایی با پیشوند -moz-)

سه گزینه‌ی Colors As RGB ، Colors As Hex و Colors As HSL تعیین کننده‌ی فرمت نمایش رنگ‌ها هستند.

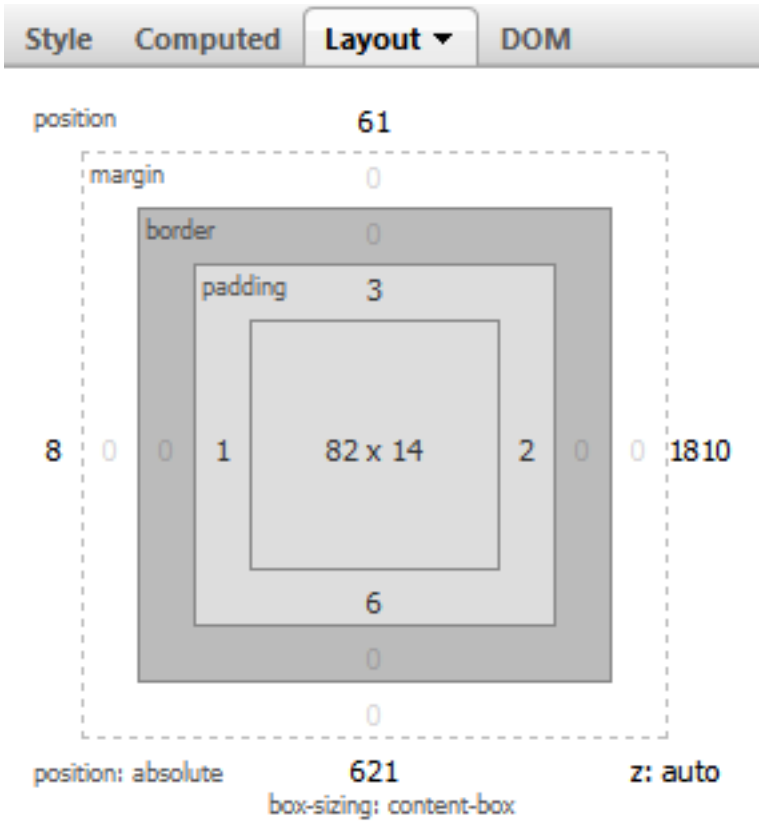
Context Menu

این منو زمانی که در پنل راست کلیک کنید ظاهر می‌شود و نسبت به منطقه (Context)ای که در آن راست کلیک کرده اید ، گزینه‌های متفاوتی را مشاهده خواهید کرد. در جدول زیر ، گزینه‌ها ، Contextشان و توضیح هر گزینه آمده است.

گزینه	Context	توضیحات
Expand All Styles	everywhere	CSS Rule فعلی را به همراه استایل هایش در clipboard کپی می‌کند
Collapse All Styles	everywhere	استایل‌های CSS Rule فعلی را در clipboard کپی می‌کند
Inspect in DOM panel	styles	آدرس فایل تعریف CSS Rule را در تب جدید باز می‌کند
Copy Location	style source link	امکان تعریف استایل‌های درون تگ (inline) را محیا می‌کند
Open in New Tab	style source link	یک Rule جدید ایجاد می‌کند CSS Rule هایی که در حال حاظر وجود دارد را هم پیشنهاد می‌دهد
Inspect in CSS panel	style source link	CSS Rule فعلی را حذف می‌کند
<Default Editor Name>	style source link	فایل تعریف استایل‌ها را در ادیتور تعریف شده باز می‌کند (این گزینه در صورت تعریف ادیتور در تنظیمات FireBug نمایش داده خواهد شد)

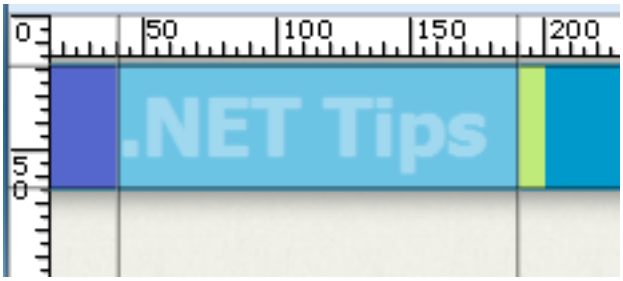
Layout - 3

در این تب ، مقادیر Box Model بصورت بصری نمایش می‌دهد. می‌توان با کلیک کردن بروی هریک از مقادیر ، آن را ویرایش کرد.
(این تغییر بصورت inline در تگ اعمال می‌شود.)
با حرکت موس بروی قسمت‌های مختلف ، می‌توان همان قسمت‌ها را در صفحه بصورت خط کشی شده مشاهده کرد.
(البته ظاهرا در ورژن 1.10.4 که بنده استفاده می‌کنم ، عملیات ویرایش مقادیر به درستی انجام نمی‌شود.)



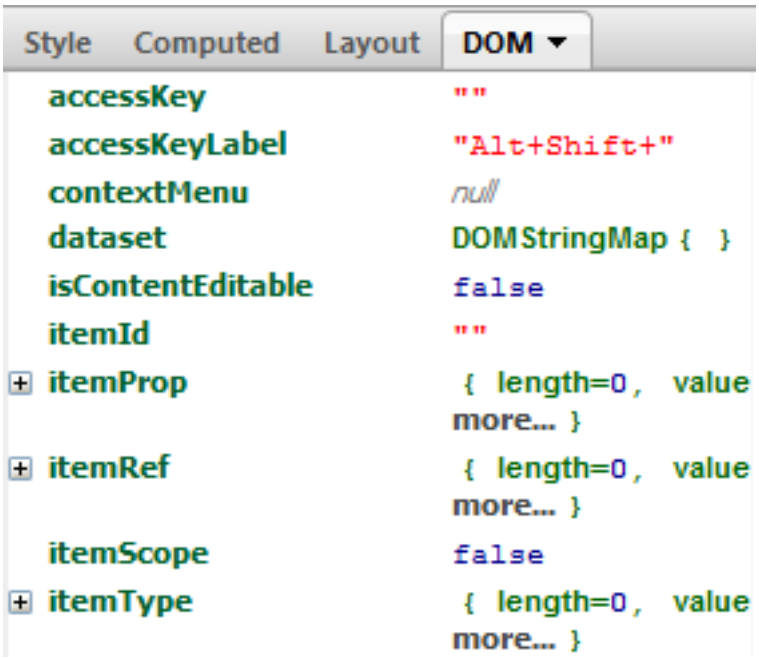
Options Menu

Show Rulers and Guides
در صورت انتخاب ، خطهای راهنما را هنگام حرکت موس بروی اجزای Box Model در صفحه نمایش می‌دهد.



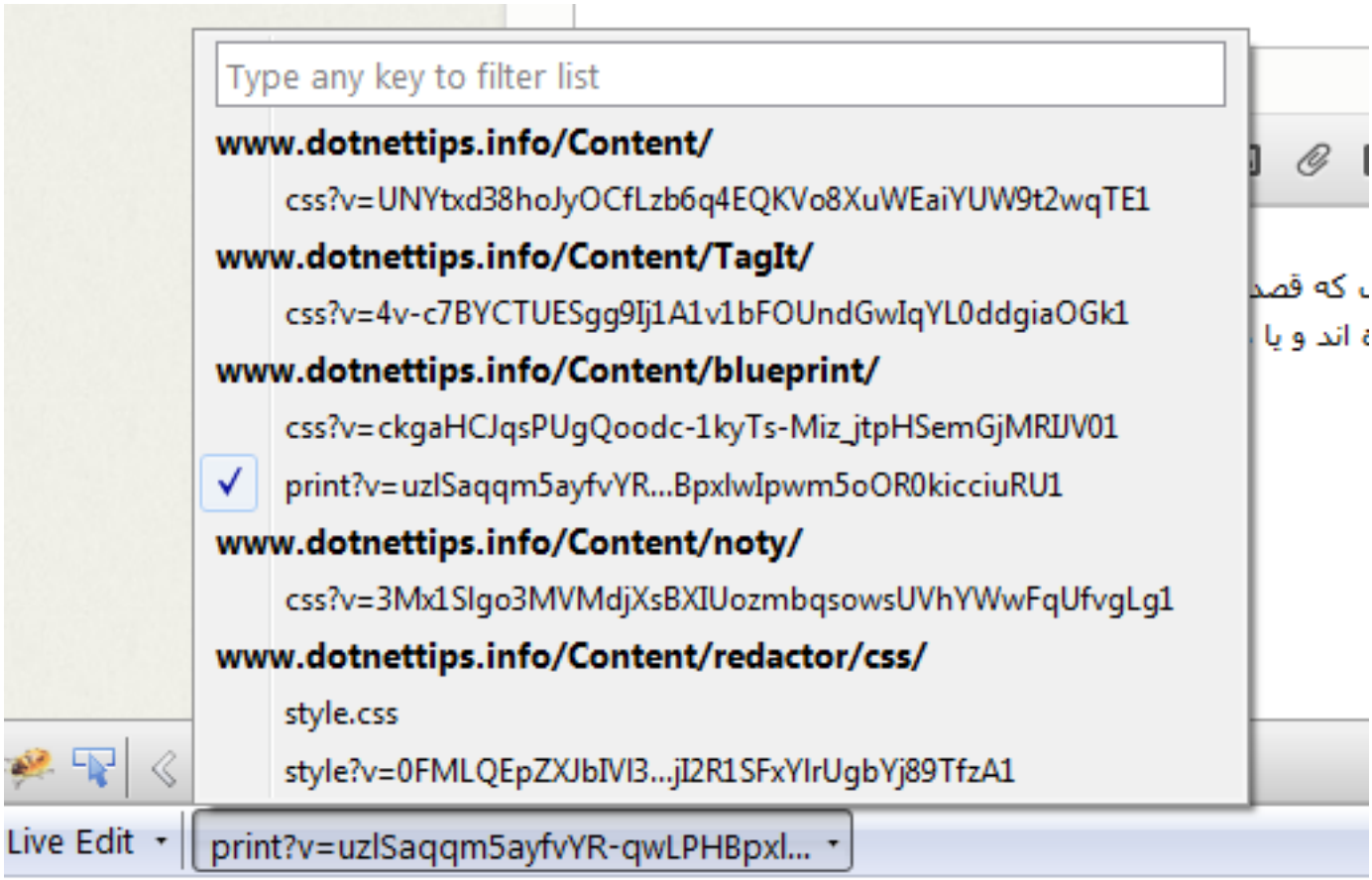
DOM - 4

این پنل اطلاعات DOM تگ جاری را نمایش می‌دهد.
این پنل تمام قابلیت‌های پنل DOM اصلی را دارا می‌باشد.
(در مقالات آینده با تب DOM آشنا خواهیم شد.)



عنوان: **آموزش فایرباگ - #7 - CSS Panel**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۱۲/۲۵ ۱۸:۵
آدرس: www.dotnettips.info
برچسب‌ها: FireBug, CSS, HTML

پنل CSS مانند پنل جانبیِ Style که در مقاله‌ی قبل بررسی کردیم است با این تفاوت که امکانات بیشتری برای افرادی که قصد تصریف استایل دارند محیا کرده است.
در این پنل می‌توان به اضافه ، ویرایش و حذف استایل هایی که به صفحه‌ی جاری توسط فایل‌های مختلف اضافه شده اند و یا داخل خود صفحه تعریف شده اند پرداخت.




```
body {
    background: none repeat scroll 0 center transparent;
    color: #000000;
    font-family: Tahoma;
    font-size: 10pt;
    line-height: 1.5;
}

.container {
    background: none repeat scroll 0 center transparent;
}
```

همچنین در این پنل امکان ویرایش یک فایل css بصورت کامل وجود دارد ، به این صورت که می‌توانید تمام محتویات فایل مورد نظر را در یک Text area ویرایش کنید.

مطابق با روالی که در قسمت قبل پیش گرفتیم عمل می‌کنیم و به ترتیب به تشریح ابزار پنل ، خود پنل ، منوی راست کلیک و پنل جانبیِ Elements می‌پردازیم.

Options Menu

با راست کلیک کردن بروی تب CSS و یا کلیک کردن بروی فلش کوچک روی تب CSS قابل دسترس است و امکانات زیر را محیا می‌کند:

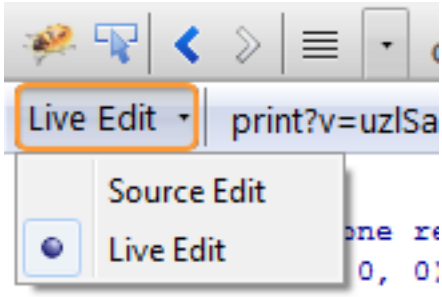
Expand Shorthand Properties : نمایش دستورات css بصورت کامل ، به این معنی که دستوراتی مانند margin که هم بصورت خلاصه و هم بصورت کامل تعریف می‌شوند را بصورت کامل نمایش می‌دهد. مثلا margin را چهار مقدار margin- margin-top , margin-right , margin-bottom , margin-left نمایش می‌دهد.

Color As Hex , Color As RGB , Color As HSL : با انتخاب یکی از سه مقدار ذکر شده ، فرمت نمایش رنگ‌ها به حالت انتخاب شده تغییر می‌کند.
مثلا دستور color : #000000 به این صورت نمایش داده می‌شود : color : rgb(0, 0, 0)

Refresh : این گزینه محتویات پنل را بروز می‌کند.

Panel Toolbar

ابزاری موجود در این پنل مشابه پنل‌های دیگر در بالای پنل و در زیر تب پنل‌ها قرار دارد و شامل ابزارهای زیر می‌شود:
Edit Button : این ابزار به دو صورت Live Edit و Source Edit در دسترس هست که با کلیک بروی فلش کوچکی که در سمت راست دکمه قرار دارد می‌توان نوع آن را تغییر داد.
انتخاب حالت Source Edit ، باعث می‌شود سورس فایل به همان صورتی که به مرورگر ارسال شده نمایش داده شود. (کامنت‌ها ، فرمت فایل و ... حفظ می‌شود.)
حالت Live Edit هم باعث نمایش محتویات فایل بصورت مرتب شده می‌شود. (کامنت‌ها و دستوراتی که توسط مرورگر تفسیر نشده اند نمایش داده نمی‌شوند.)

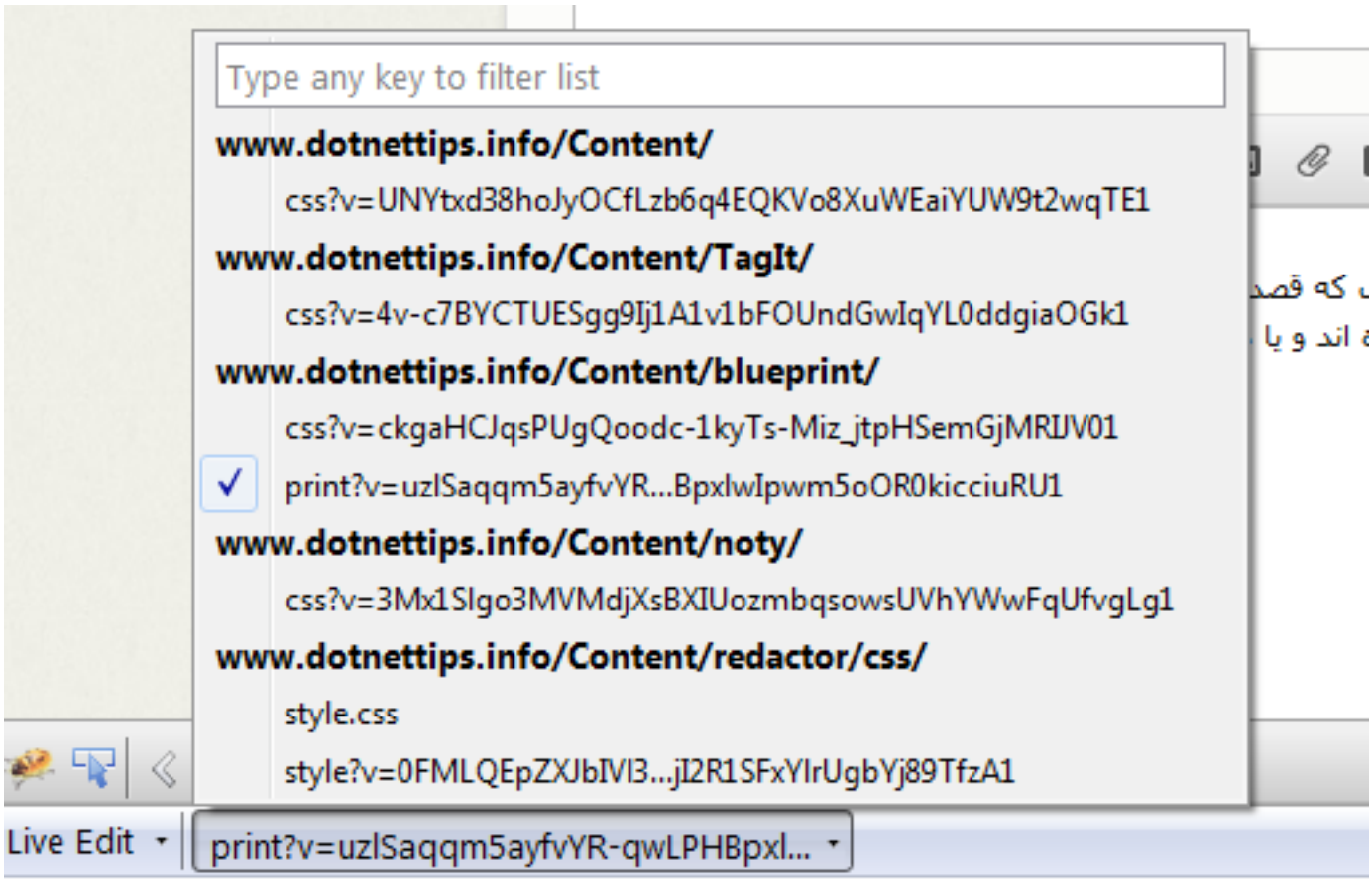


CSS Location Menu : نام فایل جاری که در پنل در حال نمایش است را نمایش می‌دهد و همچنین با کلیک بروی آن ، فایل‌های استایل دیگری که در صفحه بارگزاری شده اند را نمایش می‌دهد. با کلیک بروی هرکدام از فایل‌های نمایش داده شده ، همان فایل در پنل باز می‌شود.

استایل هایی که در خود صفحه تعریف شده اند با نام خود صفحه در این قسمت نمایش داده می‌شوند و اگر استایل‌های در چندین تگ style تعریف شده باشند ، دومین تگ با #2 ، سومین با #3 و ... مشخص می‌شوند.

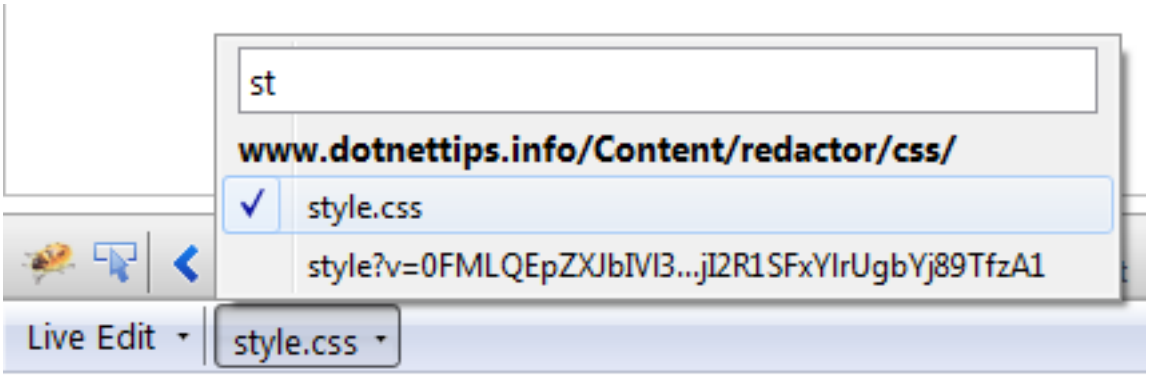
هنگام باز بودن :

-1 فایل هایی که از پوشه‌های مختلف در صفحه بارگذاری شده اند ، با نام پوشه از هم تفکیک شده اند:




```
body {
    background: none repeat scroll 0 center transparent;
    color: #000000;
    font-family: Tahoma;
    font-size: 10pt;
    line-height: 1.5;
}

.container {
    background: none repeat scroll 0 center transparent;
}
```

-2 با تایپ کردن عبارت دلخواه می‌توان نتایج را محدود کرد:



Panel

منظور از پنل ، قسمتی هست که استایل بصورت فرمت شده و نمایش داده شده است.

Infotips

توسط این قابلیت ، زمانی که موس را بروی آدرس تصاویر ، نوع فونت و ... ببرید ، پاپ آپ کوچکی باز می‌شود و اطلاعاتی در مورد مقادیر می‌دهد. مثلا اینکه آیا تصویر مورد نظر به درستی بارگذاری شده یا نه ، نمای کوچکی از تصویر ، شکل فونت و ...

Editing rules

برای ویرایش تعاریف CSS شامل Selectorها ، دستورات و مقادیر ، این پنل ابزارهای مفیدی ارائه می‌کند.
برای ویرایش یک selector ، دستور یا مقدار آن بروی آن عبارت کلیک کنید ، در همین حال یک text box ظاهر می‌شود و می‌توانید مقدار جدید را وارد کنید. پس از انجام ویرایش مورد نظر بروی قسمتی از صفحه کلیک کنید یا کلید Tab سپس Esc را بزنید. برای ویرایش دستورات بعدی ، کلید Tab را بزنید و برای لغو تغییر جاری ، کلید Esc را بزنید.

برای ایجاد یک rule جدید ، بروی قسمتی از پنل راست کلیک کرده و سپس گزینه‌ی " New Rule... " را برگزینید.
برای ایجاد یک property هم چند راه وجود دارد:
بروی قسمت از فضای خالی تعریف یک استایل دوبار کلیک کنید.
در قسمتی از تعریف یک استایل راست کلیک کرده و گزینه‌ی " New Property... " را انتخاب کنید.

بروی مقدار آخرین property کلیک کرده و کلید Tab را بزنید.

هنگام ویرایش یا ایجاد یک Rule , Propery یا مقدار بصورت inline ، حاشیه‌ی text box متناسب با مقدار وارد شده رنگی را که نمایانگر حالت ذخیره‌ی مقدار وارد شده است نمایش می‌دهد. برای مثال اگر مقداری که برای یک selector وارد شده است نامعتبر باشد ، رنگ حاشیه‌ی text box قرمز می‌شود.
جدول زیر حالت‌های مختلف را شرح می‌دهد:

رنگ حاشیه	Selectorها	نام Properyها	مقادیر Propertyها و Ruleها
خاکستری	تغییری ایجاد نشده است	تغییری ایجاد نشده است	تغییری ایجاد نشده است
قرمز	selector نامعتبر است	نام نامعتبر است	مقدار نامعتبر است
زرد	selector صحیح است اما بروی Element فعلی تاثیر ندارد	نام صحیح است اما مقدار Property غیر مجاز است یا وارد نشده است	
سبز	selector صحیح است	نام و مقدار صحیح هستند	مقدار صحیح است

Auto-completion

زمانی که در حال ایجاد/ویراش کردن یک Rule, Propery یا مقدار آنها هستید ، میتوانید از این قابلیت استفاده کنید. مثلا با وارد کردن # تمام Selector هایی که می‌توانند با # شروع شوند در دسترس شما هستند و با دکمه‌های Up/Down می‌توانید مقادیر ممکن را مرور کنید.

اگر در حال ویرایش یک مقدار عددی هستید ، می‌توانید با دکمه‌های Up/Down مقادیر را یک واحد یک واحد افزایش دهید. با نگه داشتن کلید Shift و فشردن Up/Down می‌توانید مقادیر را 10تا 10تا تغییر دهید و با نگه داشتن Ctrl بجای Sihft ، یک دهم یک دهم.

Toggling styles

با این امکان می‌توانید یک Property را بطور موقت فعال/غیرفعال کنید. با حرکت موس از یک Property یک آیکون قرمز رنگ در کنار آن نمایش داده می‌شود که با کلیک بروی آن ، Property و مقدار آن کمرنگ شده و آیکون قرمز رنگ کنار آن ثابت می‌شود. با کلیک مجدد بروی آن ، Property فعال می‌شود.

```
table.wikitable {
    border-collapse: collapse;
}
table {
    color: black;
    font-size: 100%;
```

Context Menu

این منو زمانی که در پنل راست کلیک کنید ظاهر می‌شود و نسبت به منطقه (Context)ای که در آن راست کلیک کرده اید ، گزینه‌های متفاوتی را مشاهده خواهید کرد. در جدول زیر ، گزینه‌ها ، Contextشان و توضیح هر گزینه آمده است.

گزینه	Context	توضیحات
Copy Location	CSS Location Menu	آدرس فایل استایل را در حافظه کپی می‌کند.
Open in New Tab	CSS Location Menu	فایل استایل را در یک تب جدید باز می‌کند.
Copy Image Location	image values	آدرس تصویر را در حافظه کپی می‌کند.
Open Image in New Tab	image values	تصویر را در یک تب جدید باز می‌کند.
Copy Color	color values	مقدار رنگ را در حافظه کپی می‌کند.
Copy Rule Declaration	CSS selector	Selector و Propertyها را در حافظه کپی می‌کند.
Copy Style Declaration	CSS selector	فقط Propertyها را در حافظه کپی می‌کند.
New Rule...	همه جای پنل	یک Rule جدید بالای قسمتی که راست کلیک شده ایجاد می‌کند.
Delete "<selector>"	CSS selector	Rule را حذف می‌کند.

گزینه	Context	توضیحات
New Property...	CSS rule	یک Property جدید در Ruleی که در آن راست کلیک شده ایجاد می‌کند.
Edit "<property name>"...	CSS property	Property فعلی به حالت ویرایش درمی آید. (راه ساده‌تر ، کلیک بروی Property است.)
Delete "<property name>"...	CSS property	Property فعلی را حذف می‌کند.
Disable "<property name>"...	CSS property	Property فعلی را غیرفعال می‌کند.
Refresh	همه جای پنل	محتویات پنل را بروز رسانی می‌کند.
Inspect in DOM panel	CSS Location Menu, CSS rule	فایل استایل یا Rule را در پنل DOM باز می‌کند.

Elements Side Panel

در این پنل که سمت راست پنل CSS قرار دارد ، با وارد کردن یک CSS Selector می‌توانید Elementهایی که در صفحه با آن مطابقت دارند را مشاهده کنید.

برای وارد کردن CSS Selector هم می‌توان مقدار مورد نظر را در قسمت Try a selector... وارد کرد هم می‌توان بروی یکی از Selectorهای پنل راست کلیک کرد و گزینه‌ی Get Matching Elements را برگزید.

Context Menu این قسمت هم مشابه Context Menu پنل HTML هست و فقط در ورژن 1.11 گزینه‌ی Paste HTML اضافه شده که در این کامنت از مقاله‌ی آموزش فایرباگ - #5 - HTML Panel توضیح داده شده است.

نظرات خوانندگان

نویسنده: راضیه
تاریخ: ۱۳۹۲/۰۱/۰۸ ۲۳:۴۶

عالی بود

عنوان: **نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۳۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, Debugging, MVC, FireBug, jQuery, jQuery Ajax

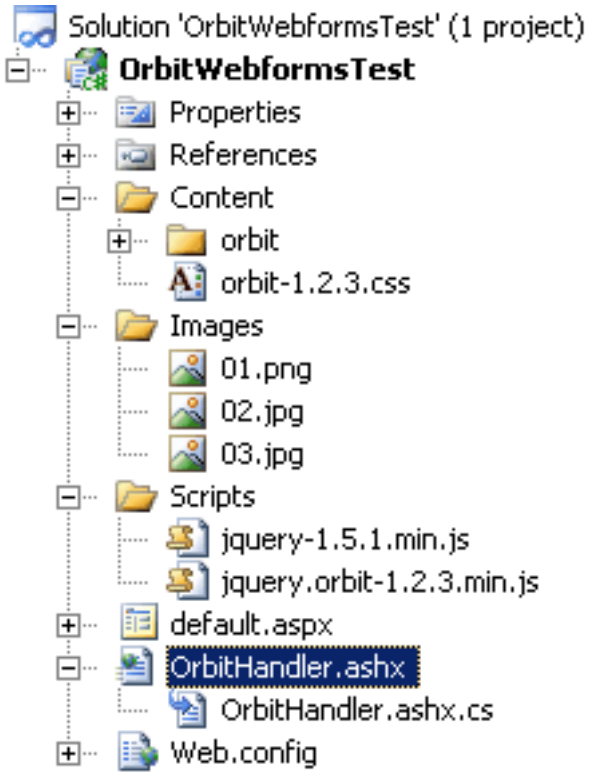
هر از چندگاهی سؤال «این مثال jQuery رو نمی‌تونم اجرا یا باز سازی کنم» در این سایت یا سایت‌های مشابه تکرار می‌شوند. بنابراین بهتر است نحوه عیب یابی برنامه‌های ASP.NET مبتنی بر jQuery را یکبار با هم مرور کنیم. در اینجا، مثال تهیه یک Image Slider را که پیشتر در سایت مطرح شده است، به نحوی دیگر بررسی خواهیم کرد:

1) فراموش می‌کنیم تا اسکریپت اصلی jQuery را به درستی پیوست و مسیردهی کنیم.
2) مسیر Generic handler دیگری را ذکر می‌کنیم.
3) مسیرهای تصاویری را که Image slider باید نمایش دهد، کاملا بی‌ربط ذکر می‌کنیم.
4) خروجی JSON نامربوطی را بازگشت می‌دهیم.
5) یکبار هم یک استثنای عمدی دستی را در بین کدها قرار خواهیم داد.

و ... بعد سعی می‌کنیم با استفاده از Firebug عیوب فوق را یافته و اصلاح کنیم؛ تا به یک برنامه قابل اجرا برسیم.

معرفی برنامه‌ای که کار نمی‌کند!

یک برنامه ASP.NET Empty web application را آغاز کنید. سپس سه پوشه Scripts، Content و Images را به آن اضافه نمائید. در این پوشه‌ها، اسکریپت‌های نمایش دهنده تصاویر، Css آن و تصاویری که قرار است نمایش داده شوند، قرار می‌گیرند:



سپس یک فایل default.aspx و یک فایل OrbitHandler.ashx را نیز به پروژه با محتویات ذیل اضافه کنید: (در این دو فایل، 5 مورد مشکل ساز یاد شده لحاظ شده‌اند)
محتویات فایل OrbitHandler.ashx.cs مطابق کدهای ذیل است:

```
using System.Collections.Generic;
using System.IO;
using System.Web;
using System.Web.Script.Serialization;

namespace OrbitWebformsTest
{
    public class Picture
    {
        public string Title { set; get; }
        public string Path { set; get; }
    }

    public class OrbitHandler : IHttpHandler
    {
        IList<Picture> PicturesDataSource()
        {
            var results = new List<Picture>();
            var path = HttpContext.Current.Server.MapPath("~/Images");

            foreach (var item in Directory.GetFiles(path, "*.*"))
            {
                var name = Path.GetFileName(item);
                results.Add(new Picture
                {
                    Path = /*"Images/" + name*/ name,
                    Title = name
                });
            }

            return results;
        }

        public void ProcessRequest(HttpContext context)
        {
            var items = PicturesDataSource();
            var json = /*new JavaScriptSerializer().Serialize(items)*/ string.Empty;
            throw new InvalidDataException("همینطوری");
            context.Response.ContentType = "text/plain";
            context.Response.Write(json);
        }

        public bool IsReusable
        {
            get { return false; }
        }
    }
}
```

در اینجا جهت سهولت دموی برنامه (و همچنین امکان باز تولید آن توسط خوانندگان)، از بانک اطلاعاتی استفاده نشده و عمدا از یک لیست جنریک تشکیل شده در حافظه کمک گرفته شده است. تصاویر برنامه در پوشه Images واقع در ریشه سایت، قرار دارند. بنابراین توسط متد PicturesDataSource، فایل‌های این پوشه را یافته و مطابق ساختار کلاس Picture بازگشت می‌دهیم. نهایتا این اطلاعات به ظاهر قرار است با فرمت JSON بازگشت داده شوند تا بتوان نتیجه را توسط افزونه Orbit استفاده کرد.

همچنین کدهای صفحه ASPX ایی که قرار است (به ظاهر البته) از این Generic handler استفاده کند به نحو ذیل است:

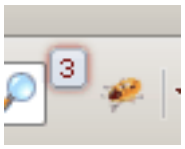
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="default.aspx.cs"
Inherits="OrbitWebformsTest._default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="Content/orbit-1.2.3.css" rel="stylesheet" type="text/css" />
    <script src="Script/jquery-1.5.1.min.js" type="text/javascript"></script>
    <script src="Scripts/jquery.orbit-1.2.3.min.js" type="text/javascript"></script>
</head>
<body>
    <form id="form1" runat="server">
    <div id="featured">
    </div>
    </form>
    <script type="text/javascript">
```

```
        $(function () {
            $.ajax({
                url: "Handler.ashx",
                contentType: "application/json; charset=utf-8",
                success: function (data) {
                    $.each(data, function (i, b) {
                        var str = '<img src="' + b.Path + '" alt="' + b.Title + '"/>';
                        $("#featured").append(str);
                    });
                    $('#featured').orbit();
                },
                dataType: "json"
            });
        });
    </script>
</body>
</html>
```

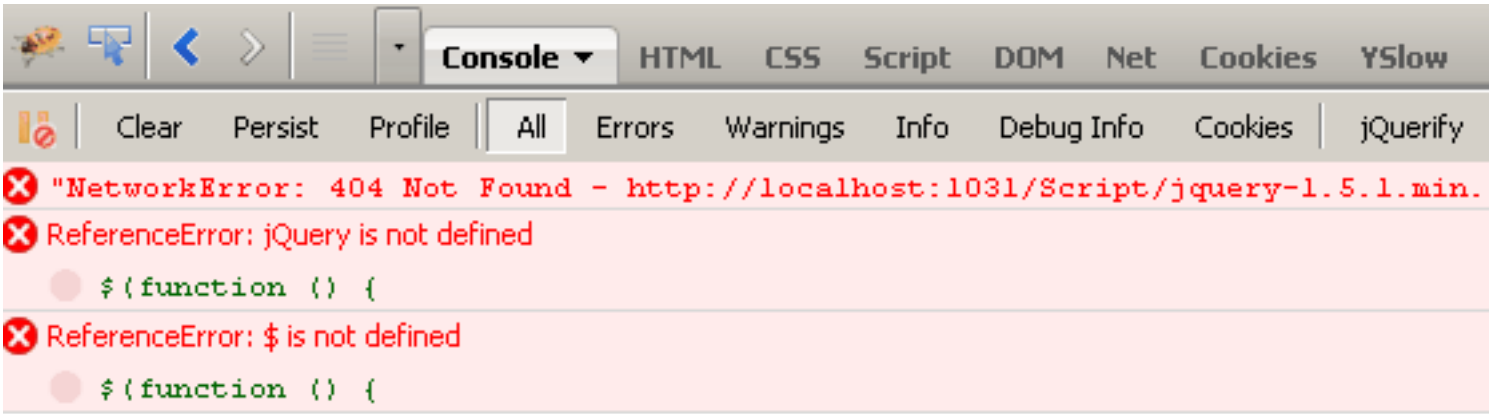
خوب! اگر پروژه را اجرا کنیم، کار نمی‌کند. یک مستطیل مشکی رنگ در کنار صفحه ظاهر شده و همین! حالا چکار باید کرد؟

مراحل عیب یابی برنامه‌ای که کار نمی‌کند!

ابتدا برنامه را در فایرفاکس باز کرده و سپس افزونه Firebug را با کلیک بر روی آیکن آن، بر روی سایت فعال می‌کنیم. سپس یکبار بر روی دکمه F5 کلیک کنید تا مجددا مراحل بارگذاری سایت تحت نظر افزونه Firebug فعال شده، طی شود.



اولین موردی که مشهود است، نمایش عدد 3، کنار آیکن فایرباگ می‌باشد. این عدد به معنای وجود خطاهای اسکریپتی در کدهای ما است.
برای مشاهده این خطاها، بر روی برگه Console آن کلیک کنید:

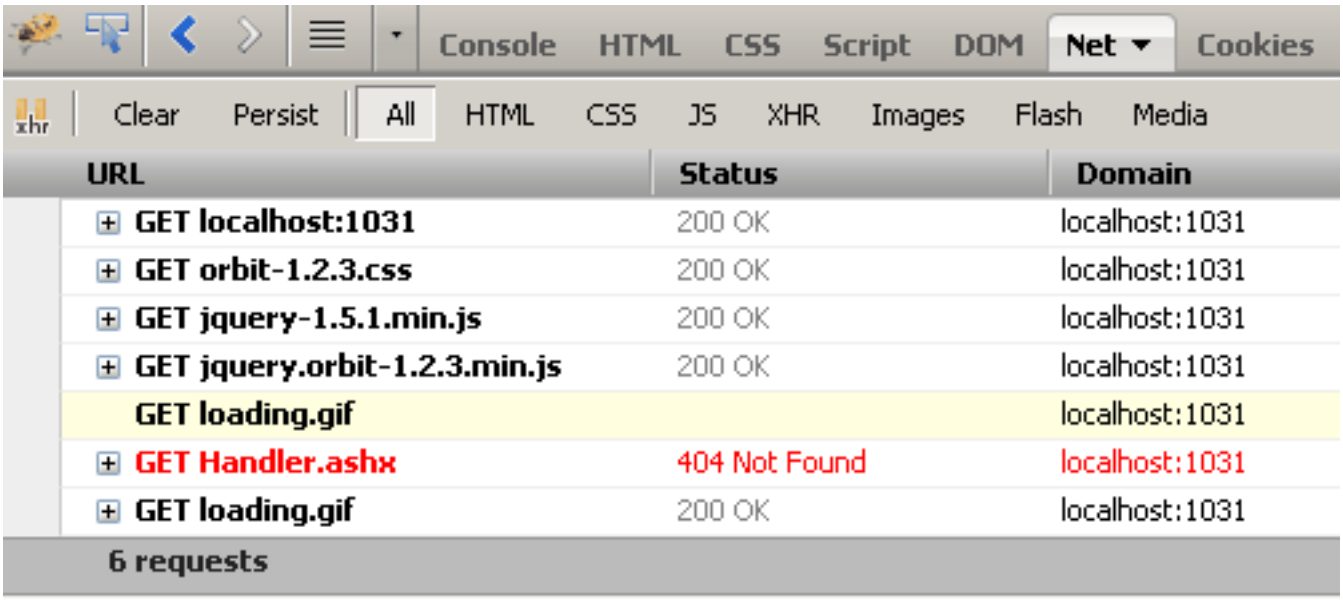


بله. مشخص است که مسیر دهی فایل jquery-1.5.1.min.js صحیح نبوده و همین مساله سبب بروز خطاهای اسکریپتی گردیده است. برای اصلاح آن سطر زیر را در برنامه تغییر دهید:

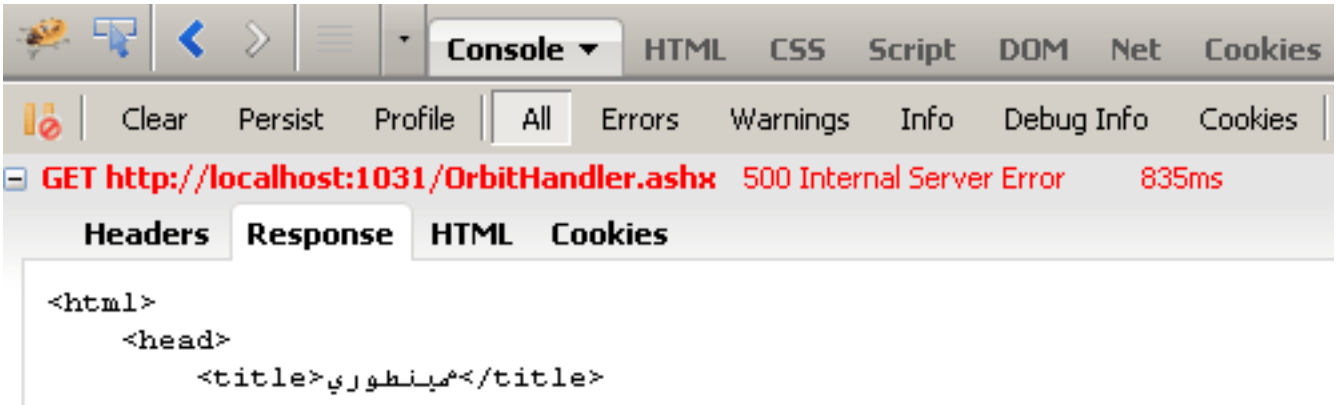
```
<script src="Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>
```

پیشتر پوشه Script ذکر شده بود که باید تبدیل به Scripts شود.

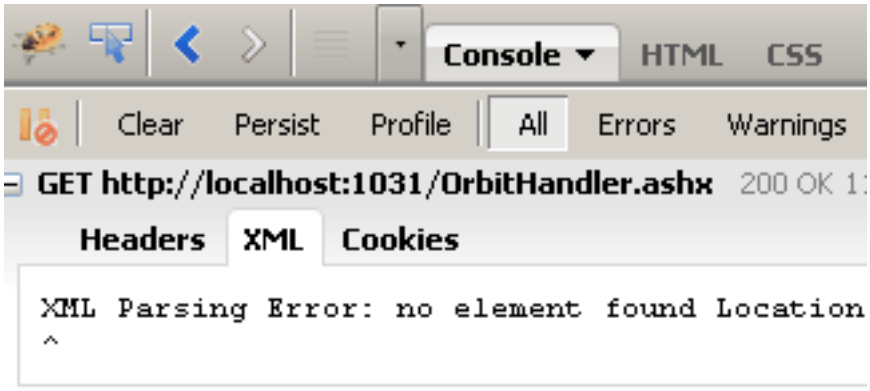
مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید. اینبار در برگه Console و یا در برگه شبکه فایرباگ، خطای یافت نشدن Generic handler نمایان می‌شوند:



برای رفع آن به فایل default.aspx مراجعه و بجای معرفی Handler.ashx، نام OrbitHandler.ashx را وارد کنید.
مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید.



اگر به برگه کنسول دقت کنیم، بروز استثناء در کدها تشخیص داده شده و همچنین در برگه Response پاسخ دریافتی از سرور، جزئیات صفحه خطای بازگشتی از آن نیز قابل بررسی و مشاهده است.
اینبار به فایل OrbitHandler.ashx.cs مراجعه کرده و سطر throw new InvalidDataException را حذف می‌کنیم. در ادامه برنامه را کامپایل و مجددا اجرا خواهیم کرد.

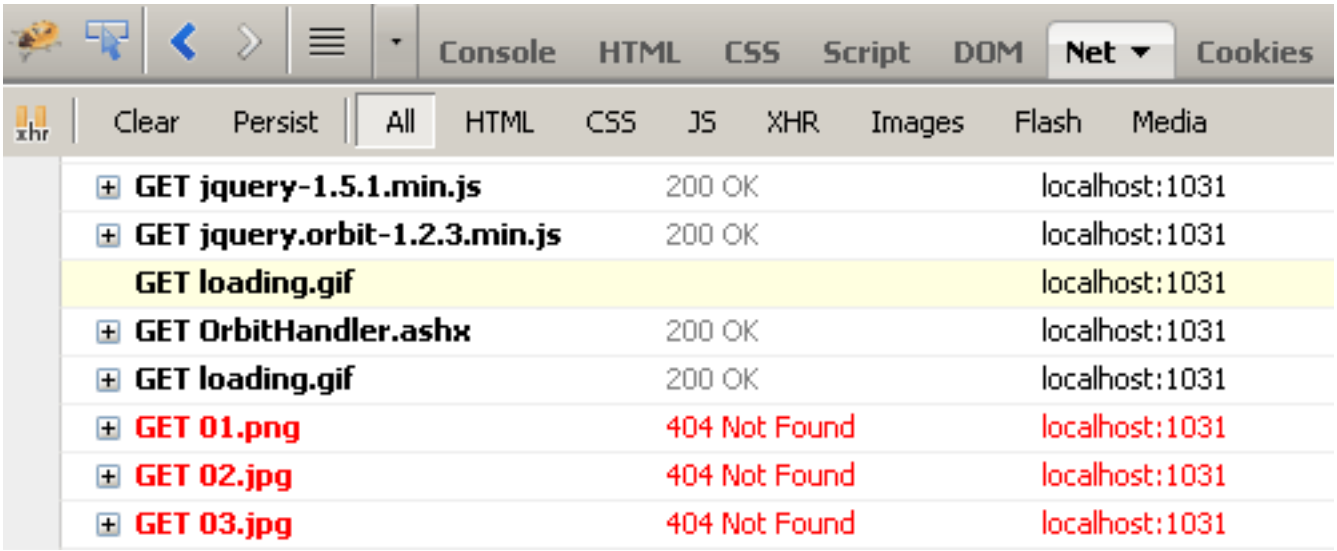


با اجرای مجدد سایت، تبادل اطلاعات صحیحی با فایل OrbitHandler.ashx برقرار شده است، اما خروجی خاصی قابل مشاهده نیست. بنابراین بازهم سایت کار نمی‌کند.

برای رفع این مشکل، متد ProcessRequest را به نحو ذیل تغییر خواهیم داد:

```
public void ProcessRequest(HttpContext context)
        {
            var items = PicturesDataSource();
            var json = new JavaScriptSerializer().Serialize(items);
            context.Response.ContentType = "text/plain";
            context.Response.Write(json);
        }
```

برنامه را کامپایل کرده و اجرا می‌کنیم. برنامه اجرا می‌شود، اما باز هم کار نمی‌کند. مشکل از کجاست؟



بله. تمام تنظیمات به نظر درست هستند، اما در برگه شبکه فایرباگ تعدادی خطای 404 و یا «یافت نشد»، مشاهده می‌شوند.

مشکل اینجا است که مسیرهای بازگشت داده شده توسط متد Directory.GetFiles، مسیرهای مطلقی هستند؛ مانند c:\path\images\01.jpg و جهت نمایش در یک وب سایت مناسب نمی‌باشند. برای تبدیل آن‌ها به مسیرهای نسبی، اینبار کدهای متد تهیه منبع داده را به نحو ذیل ویرایش می‌کنیم:

```
IList<Picture> PicturesDataSource()
        {
            var results = new List<Picture>();
```

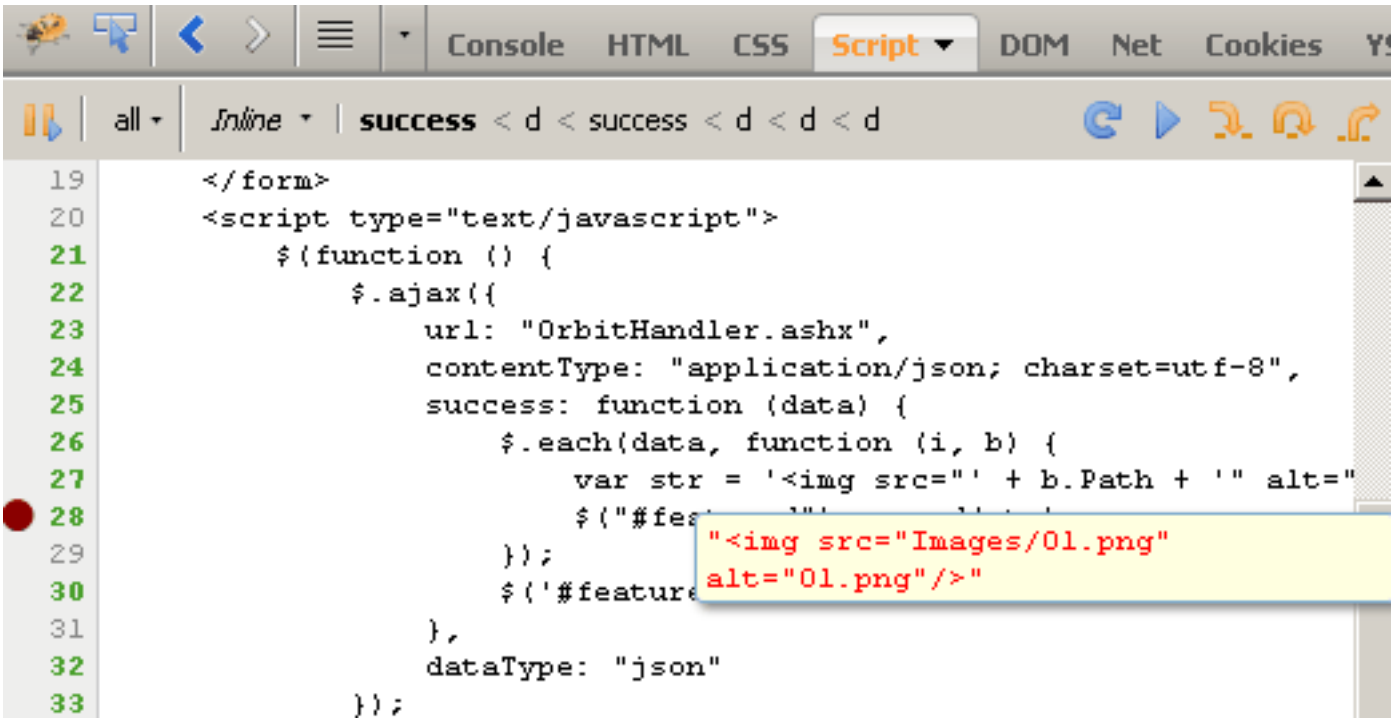
```
            var path = HttpContext.Current.Server.MapPath("~/Images");

            foreach (var item in Directory.GetFiles(path, "*.*"))
            {
                var name = Path.GetFileName(item);
                results.Add(new Picture
                {
                    Path = "Images/" + name,
                    Title = name
                });
            }

            return results;
        }
```

در این کدها فقط قسمت Path ویرایش شده است تا به مسیر پوشه Images واقع در ریشه سایت اشاره کند.
اینبار اگر برنامه را اجرا کنیم، بدون مشکل کار خواهد کرد.

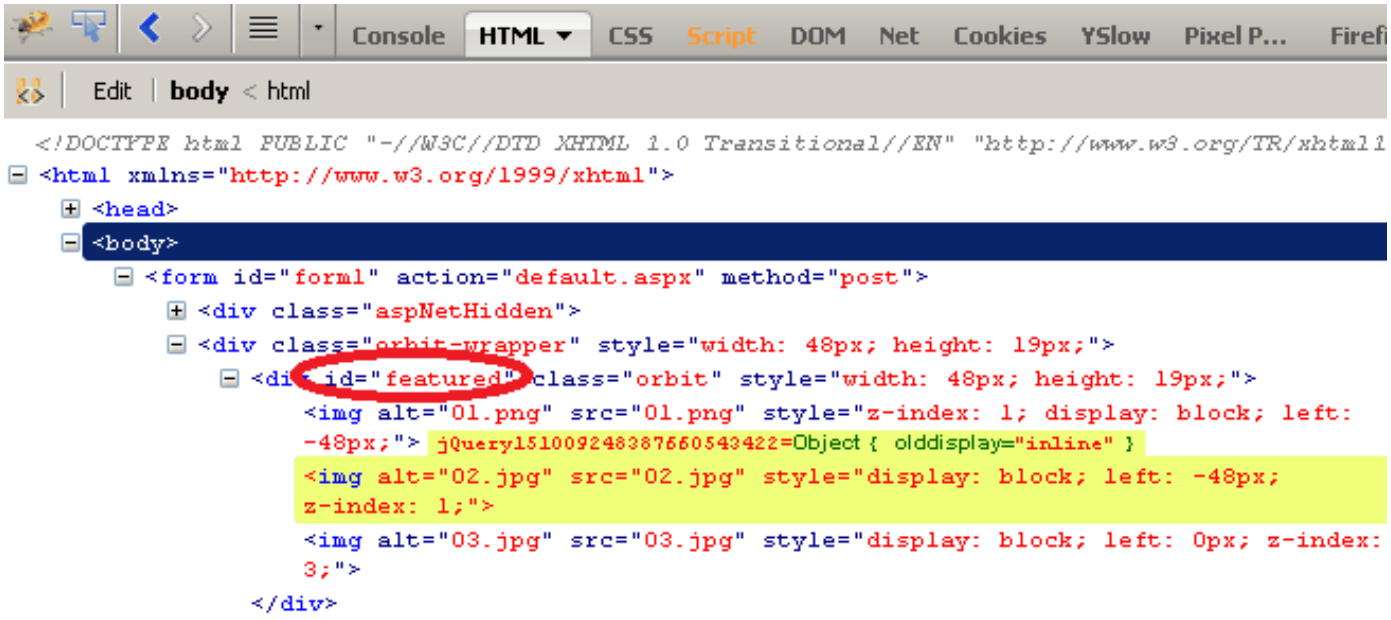
بنابراین در اینجا مشاهده کردیم که اگر «برنامه‌ای مبتنی بر jQuery کار نمی‌کند»، چگونه باید قدم به قدم با استفاده از فایرباگ و امکانات آن، به خطاهایی که گزارش می‌دهد و یا مسیرهایی را که یافت نشد بیان می‌کند، دقت کرد تا بتوان برنامه را عیب یابی نمود.

سؤال مهم: اجرای کدهای jQuery Ajax فوق، چه تغییری را در صفحه سبب می‌شوند؟

اگر به برگه اسکریپت‌ها در کنسول فایرباگ مراجعه کنیم، امکان قرار دادن breakpoint بر روی سطرهای کدهای جاوا اسکریپتی نمایش داده شده نیز وجود دارد:



در اینجا همانند VS.NET می‌توان برنامه را در مرورگر اجرا کرده و تگ‌های تصویر پویای تولید شده را پیش از اضافه شدن به صفحه، مرحله به مرحله بررسی کرد. به این ترتیب بهتر می‌توان دریافت که آیا src بازگشت داده شده از سرور فرمت صحیحی دارد یا خیر و آیا به محل مناسبی اشاره می‌کند یا نه. همچنین در برگه HTML آن، عناصر پویای اضافه شده به صفحه نیز بهتر مشخص هستند:


Console
HTML
CSS
Script
DOM
Net
Cookies
YSlow
Pixel P...
Edit
body < html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<body>
<form id="form1" action="default.aspx" method="post">

<img alt="01.png" src="01.png" style="z-index: 1; display: block; left: -48px;">
jQuery15100924838760543422=Object { olddisplay="inline" }
<img alt="02.jpg" src="02.jpg" style="display: block; left: -48px; z-index: 1;">
<img alt="03.jpg" src="03.jpg" style="display: block; left: 0px; z-index: 3;">

نظرات خوانندگان

نویسنده: صادق
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۱

ممنون آقای نصیری - برای کروم چطور، آیا developer tools کفایت میکند یا ابزار بهتری سراغ دارید؟
البته من گاهی که نیاز به دستکاری ریسپانس و ریکوئست باشه از فیدلر استفاده میکنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۵

تمام مواردی که در بحث جاری با فایرباگ مطرح شدن به صورت توکار در developer tools مربوط به کروم هم وجود دارند و مباحث آن تقریبا یکی است.

نویسنده: محسن جمشیدی
تاریخ: ۱۳۹۲/۰۲/۰۶ ۱۰:۴۹

یکی از مواردی هم که ممکن است آزار دهنده باشد خطای سینتکسی است که در FireFox با زدن Ctrl+Shift+J قابل مشاهده خواهد بود.

نویسنده: Abolfazl
تاریخ: ۱۳۹۲/۰۲/۰۸ ۲۳:۴۸

سلام آقای نصیری ..ممنونم ..خیلی عالی بود
من قبلا هم این مشکل را داشتم و نمی‌تونستم برطرفش کنم ..الان درست شد..مرسی

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۴۲

```
$(document).ready(function () {
     $(':radio').click(function () {
         debugger; //فعال سازی دیباگر ویژوال استودیو
      });
});
```

یک نکته جانبی است برای فعال سازی دیباگر خود ویژوال استودیو در حین کار با جی‌کوئری

نویسنده: ahmad.valipour
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۱۹

با سلام
سوالی برام پیش اومده.اگر ما 10 تا عکس تو پوشه image داشته باشیم ولی فقط 5 تا اونها رو در صفحه ارجاع بهشون داشته باشیم، فقط اون 5 تا به طرف کلاینت میرن.درسته؟
پس چرا وقتی با firebug مسیر عکس رو تغییر میدیم به عکس دیگه، میشناسه اون عکس رو ،و نشون میده؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۲۵

همکاری متقابل موتور مرورگر و فایرباگ. فایرباگ درخواست می‌کنه، مرورگر دریافت و ارائه.
چیزی مثل دیباگر در VS.NET. زمانیکه مثلا در کدهای کار با Entity framework روی سطری break point قرار می‌دید و خروجی

یک کوئری را بررسی می‌کنید، این دیباگر قابلیت دریافت مقادیر بررسی شده و حتی نشده را هم از بانک اطلاعاتی دارا است (حتی اگر این مقادیر، در کوئری اولیه درخواست نشده باشند؛ نوعی lazy loading در اینجا صورت می‌گیرد)

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۵

سلام
من اولین بار که اجرا می‌کنم تصویر را نشون نمی‌ده به این صورت نمایش میده :

بعد از لود صفحه یک بار که صفحه را Refresh کنم همه چیز درست میشه ؟
در قسمت Console هم خطایی وجود ندارد ....
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۲۹

حداقل دو علت می‌تونه داشته باشه:
الف) تصاویر رو نمی‌تونه پیدا کنه، یا صفحه کش شده بیش از حد. قسمت «اجرای کدهای jQuery Ajax فوق، چه تغییری را در صفحه سبب می‌شوند؟» را بررسی کنید که چه آدرسی توسط کدهای جی‌کوئری در حال پردازش است.
همچنین کش شدن نتایج قبلی رو هم می‌شود غیرفعال کرد:

```
$.ajax({
  cache: false /* گاهی از اوقات خصوصا برای آی ایی نیاز است */
});
```

ب) چند وقت قبل در یکی از بحث‌های سایت دیدم که مورد زیر رعایت نشده بود و کدهای جی‌کوئری کار نمی‌کردند:

```
<script type="text/javascript">
        $(function () {
            // کدهای جی‌کوئری در اینجا
        });
</script>
```

اجرای کدهای جی‌کوئری نیازی به DOM حاضر و آماده دارند که توسط متد document ready آن مانند کدهای فوق باید تدارک دیده شود. نیازی به این کد نخواهد بود اگر اسکریپت‌ها در آخر صفحه و پیش از بسته شدن تگ body اضافه بشن.

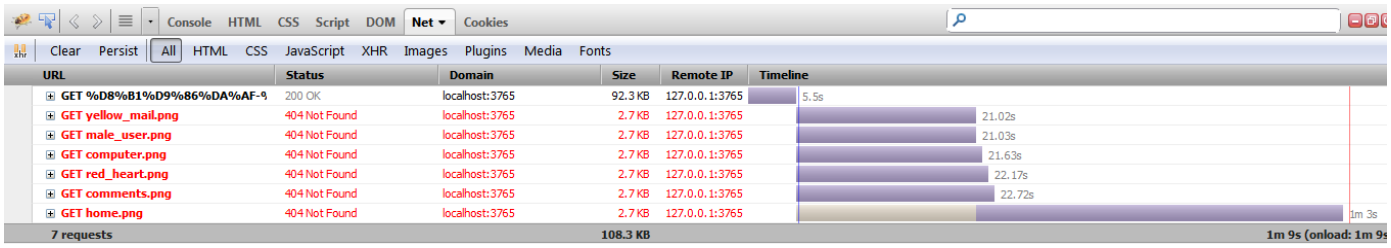
نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۷/۰۷ ۱۰:۵۸

با تشکر از مطلب بسیار کاربردیتون ، در هنگام استفاده از URL Routing همانطور که قبلا راهنماییم کرده بودید برای آدرس فایل‌های جاوا اسکریپت از

```
<%=ResolveUrl("~/App_Themes/MainTheme/jquery.js")%>
```

استفاده کردم و مشکل حل شد ولی برای یو آر ال این تصاویر آیکون که در Jquery تعریف شدند میتونید کمکم کنید

```
<script type="text/javascript">
    $(document).ready(function () {

        $('#exampleMenu').sweetMenu({
            top: 200,
            padding: 8,
            iconSize: 48,
            easing: 'easeOutBounce',
            duration: 500,
            icons: [
'images/home.png',
'images/comments.png',
'images/red_heart.png',
'images/computer.png',
'images/male_user.png',
'images/yellow_mail.png'
]
        });
    });
</script>
```



اگر آدرس آیکون‌ها را به صورت

'http://site.ir/images/home.png'

تعریف کنم مشکل حل میشه ولی فکر کنم راه حل درستی نباشه . بسیار ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۷ ۱۱:۲۸

امکان قرار دادن کدهای سمت سرور داخل اسکریپت‌ها هم هست؛ مثلا:

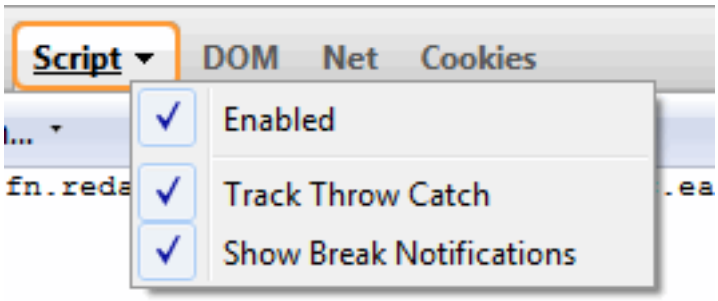
```
data: "{'username': '" + $('#<%= TextBox1.ClientID %>').val() + "'}",
```

به این شرط که این اسکریپت داخل صفحه runat=server دار باشد یا داخل head ایی با این مشخصات.

عنوان: **آموزش فایرباگ - #8 - Script Panel**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۳/۳۰ ۲۲:۰
آدرس: www.dotnettips.info
برچسب‌ها: Debugging, JavaScript, FireBug

تب Script در FireBug مخصوص دیباگ کردن کدهای JavaScript است. امکاناتی که در این قسمت گنجانده شده بسیار کاربردی بوده و همچنین در بیشتر قسمت‌ها با ابزارهای خطایابی دیگر مشابه است. برای مثال اگر با Visual Studio کار کرده باشید، با نحوه‌ی ایجاد Break Point ، قسمت‌های Watch,Stack و ... آشنا خواهید بود.

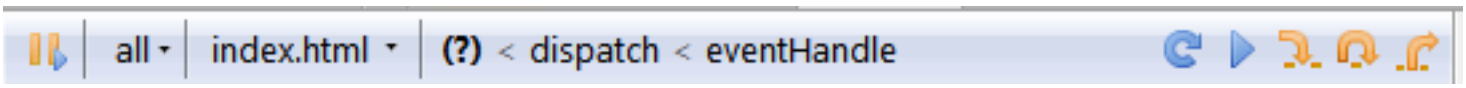
این پنل هم مشابه پنل‌های دیگر فایرباگ دارای یک بخش با عنوان Options Menu هست که با راست کلیک کردن بروی عناون یا کلیک بروی مثلث کنار عنوان پنل قابل دسترسی است. تنظیماتی که در اینجا قابل تعیین است عبارتند از:



Enabled/Disabled : برای فعال/غیرفعال کردن پنل است. فعال بودن این قسمت ممکن است موجب کاهش سرعت بارگزاری صفحات شود.
Track Throw/Catch : در صورت فعال بودن این گزینه، در صورت رویدادن خطا در بلاک try/catch ، دیباگر در آن نقطه از برنامه متوقف شده و کنترل برنامه به شما داده می‌شود. (البته بنده موفق بررسی کامل این قابلیت نشدم. ظاهرا ورژن‌های آخری باگ دارند. مثل غیرفعال شدن کامل همین پنل، "Debugger not activated"! اگر کسی موفقیتی در کار با این مورد داشت، سپاسگذار میشوم اطلاع بدهد.)
Show Break Notifications : در صورت فعال بودن این گزینه، هنگام توقف کدی در صفحه، اطلاعاتی در مورد علت توقف آن در بالای پنل ارائه خواهد شد.

توجه داشته باشید که ممکن است در ورژن‌های مختلف، تعداد این گزینه‌ها بیشتر یا کمتر باشد.

Panel Toolbar



نواری است که ابزارهای پنل بروی آن قرار دارند و به ترتیب عبارتند از:
Break On Next : این دکمه که مشابه آن در اکثر پنل‌ها وجود دارد، هنگام اجرای یک دستور JavaScript آن را متوقف کرده و شما می‌توانید به بررسی آن بپردازید.
Script Type Menu : با انتخاب یکی از چهار گزینه موجود می‌توانید نتیجه اسکریپت‌های اضافه شده به صفحه که در قسمت Script Location Menu نمایش داده شده اند را فیلتر کنید. (متاسفانه این گزینه هم مشابه گزینه Track Throw/Catch برای بنده،

ورژن 1.11.4 نتیجه ای نداشته است.)

Script Location Menu : در این قسمت اسکریپت هایی که در صفحه وجود دارند نمایش داده می‌شوند. مشابه پنل HTML می‌توانید هنگام بازبودن این لیست، با تایپ کردن نتایج را فیلتر کنید.

همچنین با راست کلیک کردن بروی این قسمت می‌توانید آیتم یا فایل انتخاب شده را در ادیتور مورد نظر باز کنید، در پنل DOM بررسی کنید، فایل را در یک تب جدید باز کنید، آدرس فایل را در حافظه موقت کپی کنید.

Execution Control Buttons : این دکمه‌ها زمانی که دیباگر به یک Break Point برسد یا به دلیل فعال بودن دکمه‌های Break On ... متوقف شود، فعال می‌شوند و با کمک آن‌ها می‌توانید عملیات خطایابی را ادامه دهید.

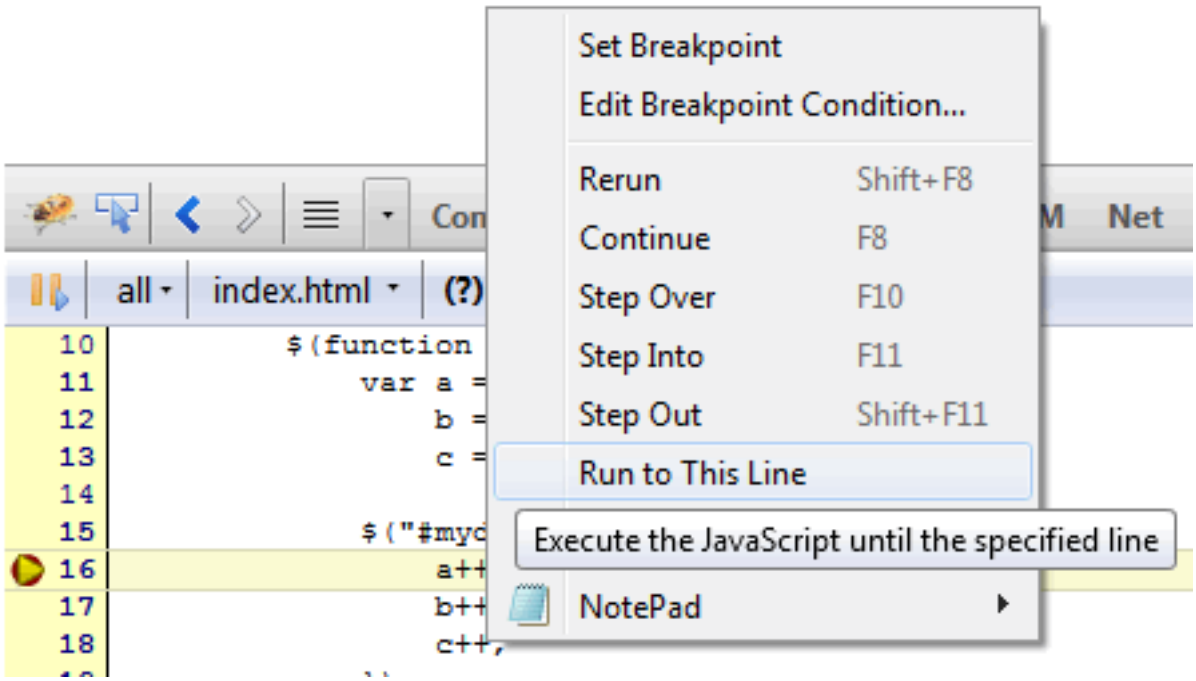
در جدول زیر این دکمه‌ها و توضیحات تکمیلی هریک را مشاهده می‌کنید:

نوع	دکمه	Shortcut	توضیحات
اجرای مجدد		Shift + F18	Stack فعلی را مجدد اجرا می‌کند. (Stack: لیستی از توابع به ترتیبی که با فراخوانی آنها تابع جاری فراخونی شده است. در مقاله بعدی توضیحات بیشتری ارائه خواهد شد.)
ادامه		F8	اجرای برنامه را (تا Break Point بعدی) ادامه می‌دهد.
وارد شدن		F11	در صورت رسیدن به یک تابع، با زدن این دکمه دیباگر وارد بنده‌ی آن تابع می‌شود.
رد شدن		F10	اجرای برنامه را در سطح (Scope) فعلی ادامه می‌دهد. مشابه قبلی وارد تابع نمی‌شود.
خارج شدن		Shift + F11	کنترل به تابعی که تابع جاری را فراخوانی کرده است بازمیگردد. روشی سودمند زمانی که هنگام خطایابی وارد کدهای jQuery یا مثل آن می‌شوید :)

بعد از Script Location Menu قسمتی وجود دارد که وضعیت فعلی Stack را نشان می‌دهد و با کلیک بروی هرکدام، به کد آن قسمت هدایت می‌شوید. (البته اگر فایرباگ شما در این قسمت باگ نداشته باشد!)

Context Menu

تنها قابلیت جدیدی که در این منو وجود دارد، Run To This Line است. هنگامی که پنل در حالت دیباگ است، با راست کلیک کردن بروی خط مورد نظر و انتخاب گزینه‌ی Run to This Line می‌توانید خطوط میانی را رد کرده و به خط مورد نظر بروید. البته خطوطی که رد می‌شوند ، اجرا می‌شوند.
این کار معادل این است که درخط مورد نظر یک Break Point اضافه کنید و دکمه‌ی F8 را بزنید.



Breakpoints

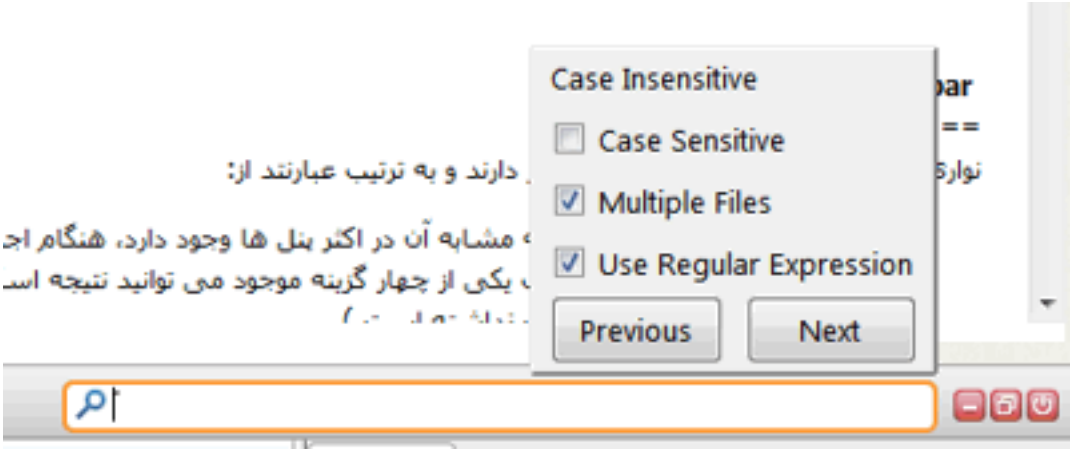
توسط Break Pointها می‌توانید خطوطی از برنامه را برای خطایابی مشخص کنید. زمانی که دیباگر به نقطه‌ی مورد نظر برسد متوقف شده و می‌توانید به بررسی برنامه بپردازید. می‌توانید با بردن موس بروی متغییرها مقدار آن هارا بررسی کنید یا با کمک قسمت‌های Watch و Stack در پنل جانبی اطلاعات بیشتری در مورد اجرای برنامه در نقطه‌ی جاری بدست آورید.(در مورد پنل‌های جانبی در قسمت بعدی توضیح خواهم داد.) همچنین با کمک دکمه هایی که در جدول فوق توضیح داده شده اند روند اجرای برنامه را خط به خط ادامه دهید.
برای ایجاد یک Break Point، بروی شماره‌ی خط مورد نظر کلیک کنید. نقطه ای قرمز رنگ نمایش داده خواهد شد. البته دقت کنید که همه‌ی خطوط برنامه اجرا نمی‌شوند و نمی‌توانید در آن خطوط از این امکان بهره ببرید. شماره‌ی خطوط فعال با رنگ سبز مشخص شده اند.

امکان مفید دیگری که همراه با این قابلیت ارائه شده است (که در محیطهای دیگر هم وجود دارد)، عبارت شرطی است. به این صورت که ضمن قرار دادن Break Point در یک نقطه از برنامه، می‌توانید یک شرط هم برای توقف برنامه تعیین کنید.
فرض کنید یک حلقه دارید که 300 بار تکرار می‌شود و مثلا در اجرای 250ام آن مشکلی وجود دارد. در این حالت می‌توانید از این قابلیت استفاده کنید و شرط توقف را *i == 250* قرار بدهید.
برای تعیین یک شرط برای یک Break Point، بروی خط مورد نظر راست کلیک کنید و شرط را در قسمت مشخص شده وارد کنید.

امکان مفید دیگری که وجود دارد، Break Point خودکار است. اگر از مقالات قبلی دکمه‌ی Break On All Errors در پنل Console و Break On Mutate در پنل HTML را بخاطر داشته باشید می‌دانید که در هر یک هنگام اجرای یک رخداد مورد نظر، دیباگر در خطی که موجب آن رخداد شده است متوقف شده و می‌توانید کنترل برنامه را بدست بگیرید. در این حالت نیازی به ایجاد Break Point نیست و FireBug بصورت خودکار این کار را انجام می‌دهد.

Search

این بخش در همه پنل‌ها وجود دارد با این تفاوت که در پنل Script و CSS دو گزینه‌ی *Multiple Files* و *Use Regular Expression* وجود دارد که به ترتیب امکان جستجو در فایل‌های js و css اضافه شده به صفحه هستند. قابلیت دیگری هم که فقط در پنل Script وجود دارد، پرش به یک خط در برنامه است به این صورت که با وارد کردن # و یک عدد به عنوان شماره‌ی خط، همان خط نمایش داده می‌شود.



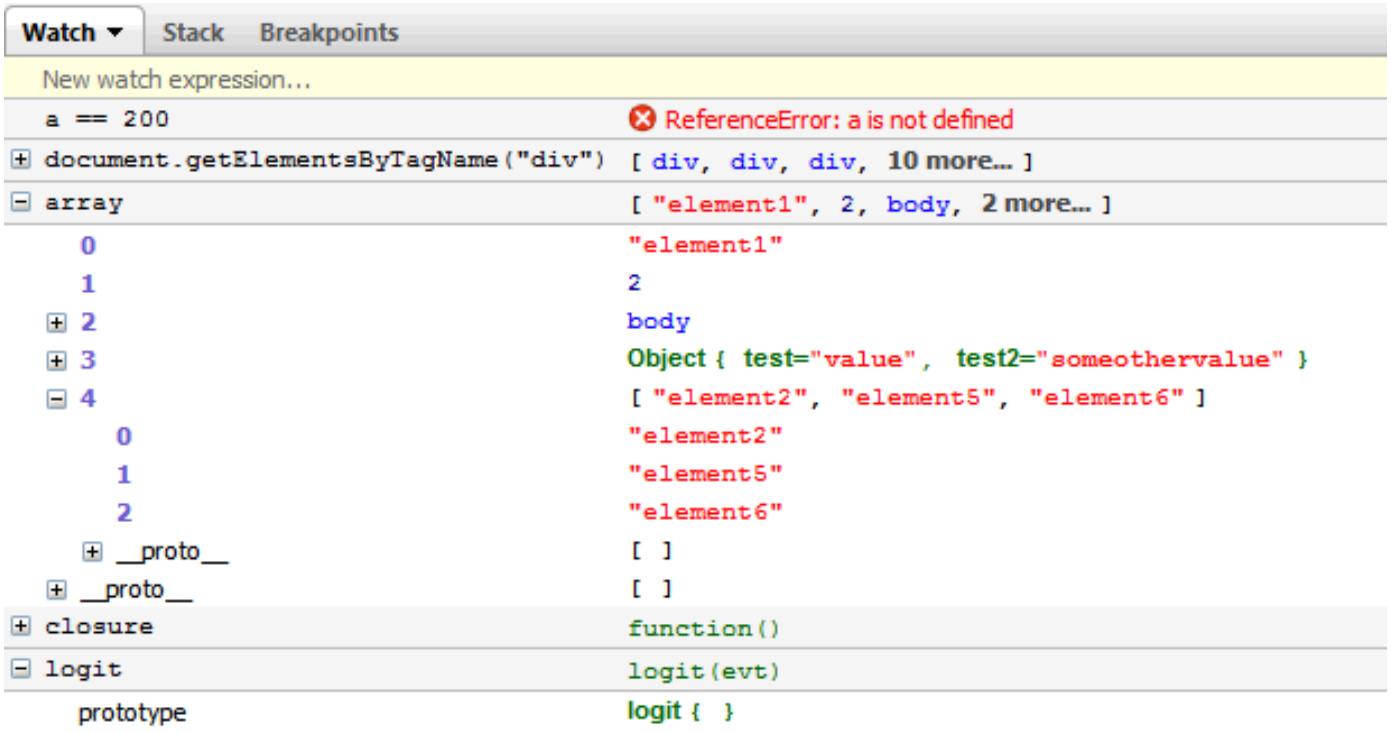
در قسمت بعدی پنل جانبی که شامل سه بخش Watch، Stack و Breakpoints است را بررسی خواهیم کرد.

عنوان: **آموزش فایرباگ - #9 - Script Panel - Side Panels**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۴/۰۴ ۲۲:۰
آدرس: www.dotnettips.info
برچسب‌ها: Debugging, JavaScript, FireBug

سه پنل Watch، Stack و Breakpoints پنل‌های جانبی پنل Script هستند که امکانات و اطلاعات مفیدی در ارتباط با قطعه کدی که در حال دیباگ آن هستیم ارائه می‌کنند.

Watch

این پنل متغییرها و تغییرات مقادیر آن‌ها در هنگام دیباگ را نمایش می‌دهد. بصورت پیشفرض متغییرهایی که در بلاک فعلی (که دیباگر در آن توقف کرده) ایجاد شده اند در این لیست قرار دارند. اما می‌توانید متغییرها و عبارات مورد نظر را در این قسمت وارد کنید. برای مثال می‌توانید مساوی بودن یک متغییر با یک مقدار را در این لیست وارد کنید و نتیجه را در هر اجرا مشاهده کنید. یا حتی یک متد را در این لیست وارد کنید تا در هر اجرا ، اجرا شود و نتیجه نمایش داده شود.
برای اضافه کردن یک عبارت جدید به این لیست، بروی عبارت *New watch expression...* در بالای پنل کلیک کرده و عبارت مورد نظر را وارد کنید و در نهایت کلید Enter را بزنید.
برای حذف عبارتی که وارد کردید، بروی علامت X کلیک کنید.
توجه کنید که هر عبارتی که در این قسمت وارد می‌کنید در اجرای هر خط از برنامه در حالت دیباگ اجرا می‌شود و اگر عبارتی که وارد کرده اید مقادیر داخل برنامه را تغییر دهند، برنامه‌ی شما متفاوت از حالت عادی عمل خواهد کرد.



Options Menu و Context Menu در این پنل همان موارد پنل DOM هستند و از آنجایی که همه‌ی این آیتم‌ها در این پنل به درستی عمل نمی‌کنند، توضیح آن را در مقاله‌ی مربوط به پنل DOM ارائه می‌کنیم.

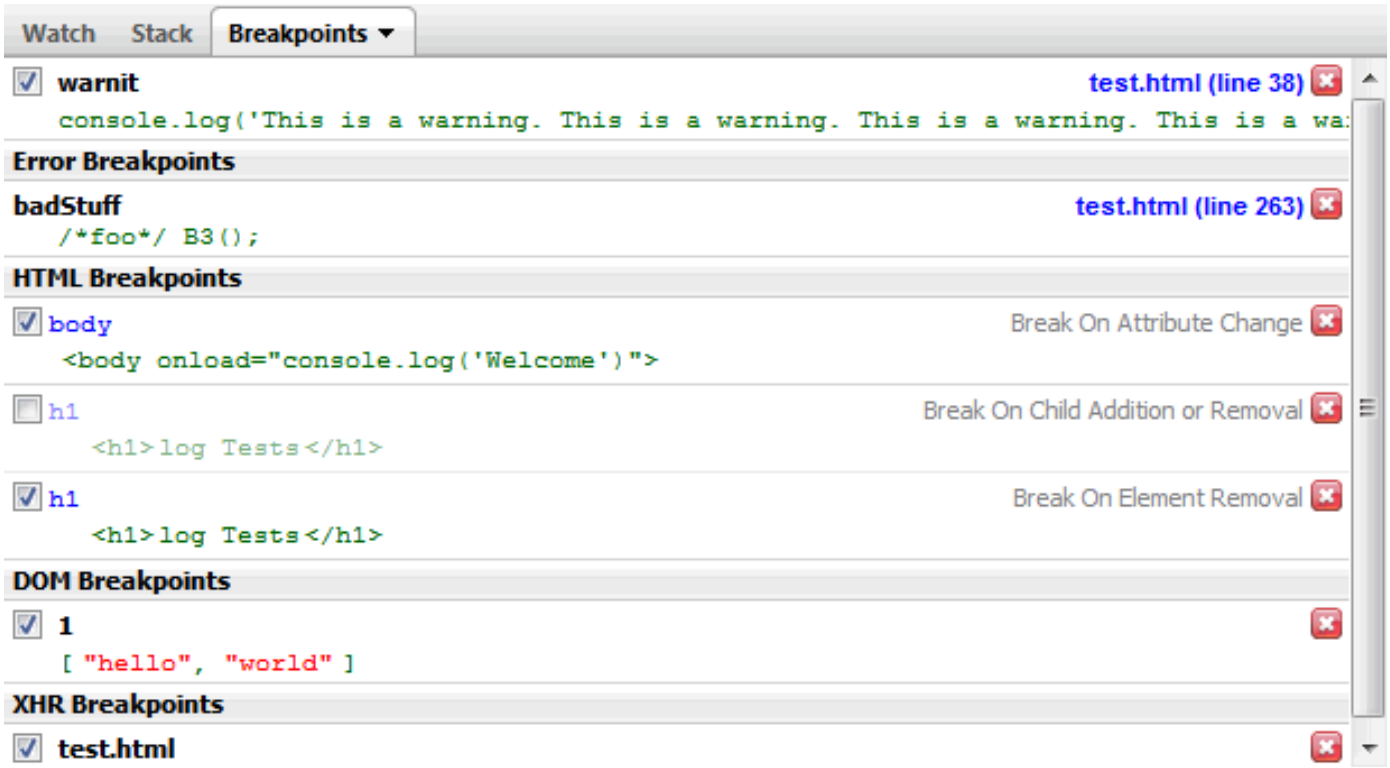
Stack

در این پنل، زمانی که دیباگر متوقف شده است،لیستی از توابع که با فراخوانی یکدیگر به تابع جاری رسیده اند، نمایش داده می‌شوند. برای هر تابع هم پارامترها و مقادیر آن‌ها هنگام فراخوانی قابل مشاهده است. با کلیک بروی هر کدام از آن ها، خطی که

تابع در آن قرار دارد نمایش داده می‌شود و همچنین می‌توانید مقادیر متغییرهای محلی آن تابع هنگامی که فرخوانی شده است را هم بررسی کنید. (جالب نیست؟)
با راست کلیک کردن بروی نام این پنل یا باز کردن فلش آن، می‌توانید نمایش یا عدم نمایش این لیست در Toolbar را مشخص کنید.

Breakpoints

در این پنل لیست Break Pointهای اضافه شده نمایش داده می‌شود و می‌توانید آن هارا فعال، غیرفعال یا حذف نمایید.
برای فعال، غیرفعال و حذف کردن یک یا همه‌ی Break Pointها بروی قسمت از پنل راست کلیک کرده و یکی از گزینه‌های مورد نظر را انتخاب کنید.


Watch
Stack
Breakpoints
warnit
test.html (line 38)
console.log('This is a warning. This is a warning. This is a warning. This is a wa
Error Breakpoints
badStuff
test.html (line 263)
/*foo*/ B3();
HTML Breakpoints
body
Break On Attribute Change
<body onload="console.log('Welcome')">
h1
Break On Child Addition or Removal
<h1>log Tests</h1>
h1
Break On Element Removal
<h1>log Tests</h1>
DOM Breakpoints
1
["hello", "world"]
XHR Breakpoints
test.html

عنوان: **آموزش فایرباگ - #10 - DOM Panel**
نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۸/۱۰ ۲۲:۰
آدرس: www.dotnettips.info
برچسب‌ها: JavaScript, FireBug, DOM

در پنل DOM توابع و متغییرهایی که در صفحه وجود دارند بصورت درختی نمایش داده می‌شوند. objectها و arrayها قابل باز شدن هستند و بصورت درختی می‌توانید محتویات آن‌ها را مشاهده کنید. توابع هم بصورت لینک هستند که با کلیک برون آن ها، کد مربوط در پنل Script نمایش داده می‌شود. توجه کنید که محتویاتی که مشاهده می‌کنید برای همان لحظه ای است که پنل را باز کردید و برای مشاهده تغییرات ثانویه باید محتویات پنل را بروزرسانی کرد.

قبل از بررسی این پنل اجازه دهید نگاهی به تعریف DOM بیندازیم.

DOM چیست؟

مدل شیءگرای سند یا دام (DOM - Document Object Model) عنوان یکی از دو ساختوارهٔ (architecture) اصلی است (در کنار اس‌اِی‌اکس) که بر اساس آن سندهای اکس‌ام‌ال را به اشیایی که در بردارندهٔ آن است، تجزیه نموده، و آن‌ها را به‌صورت یک ساختار درختی داده‌ها در فضای حافظه اصلی پهن می‌کند. ساختوارهٔ دام، نه به زبان برنامه‌نویسی خاصّی وابستگی دارد و نه به سکّوی برنامه‌نویسی ویژه‌ای، بلکه، به منظور اجراء و پیاده‌سازی آن باید از یک زبان برنامه‌نویسی بلندتراز همچون جاوا، سی‌شارپ، جاوااسکریپت یا مشابه آن‌ها سود بجوییم. آنسوی رابط کاربر سند با مدلی شیءگرا نمایانده می‌شود.

Options Menu

این منو با راست کلیک کردن بروی نام پنل یا کلیک کردن بروی مثلثی که روی پنل قرار دارد، نمایش داده می‌شود.

Show User-defined Properties

در صورت فعال بودن، پراپرتی هایی که توسط کاربر به صفحه اضافه شده اند را نمایش می‌دهد.

Show User-defined Functions

در صورت فعال بودن، توابعی که توسط کاربر به صفحه اضافه شده اند را نمایش می‌دهد.

Show DOM Properties

در صورت فعال بودن، پراپرتی هایی که بصورت پیشفرض در DOM وجود دارند را نمایش می‌دهد.

Show DOM Functions

در صورت فعال بودن، توابعی که بصورت پیشفرض در DOM وجود دارند را نمایش می‌دهد.

Show DOM Constants

در صورت فعال بودن، const هایی که بصورت پیشفرض در DOM وجود دارند را نمایش می‌دهد.

Show Inline Event Handlers

در صورت فعال بودن، رویدادهایی که بصورت خطی در تگ‌ها تعریف شده اند را نمایش می‌دهد.

Show Closures

در صورت فعال بودن، Closure ها را نمایش می‌دهد.

Show Own Properties Only

در صورت فعال بودن، فقط پراپرتی هایی که بروی خود شئ تعریف شده اند را نمایش می‌دهد.

Show Enumerable Properties Only

در صورت فعال بودن، فقط پراپرتی‌های شمارشی را نمایش می‌دهد.

Refresh

محتویات پنل را بروزرسانی می‌کند.

Property Path

این قسمت در بالاترین بخش پنل قرار دارد و وظیفه‌ی آن نمایش مسیر شئ از خود شئ تا window است.

همچنین با راست کلیک کردن بروی این قسمت دو گزینه نمایش داده می‌شود. Refresh برای بروزسازی آدرس نمایش داده شده و Use in Command Line هم برای استفاده از شئ در خط فرمان است. پس از راست کلیک کردن بروی یک شئ و انتخاب گزینه‌ی Use in Command Line فایرباگ تمرکز برنامه را به خط فرمان منتقل می‌کند و شئ را تحت متغییری به نام $p در خط فرمان کپی می‌کند.

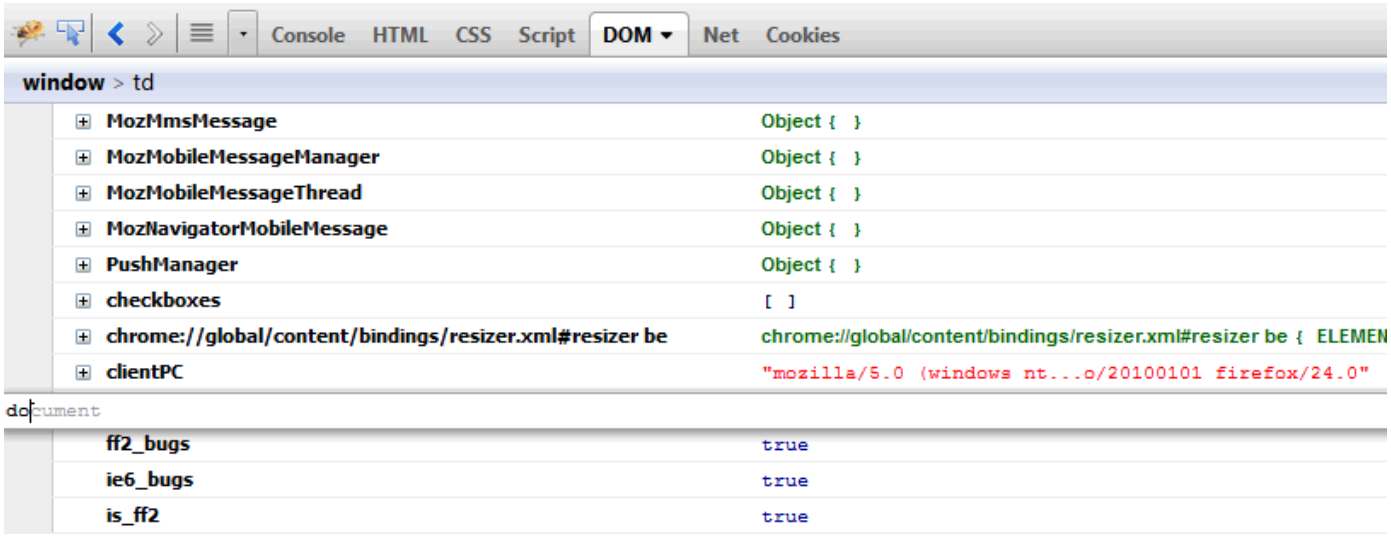
رنگ ها

برای مشخص کردن نوع متغییرهای این پنل، فایرباگ برای هر نوع متغییر از یک رنگ استفاده می‌کند.

اشیاء ، اشیاء DOM ، توابع Getter ، **توابع تعریفی کاربر** ، توابع DOM ، **توابع Constructor** ، پراپرتی‌های Read-only

Auto-Completion

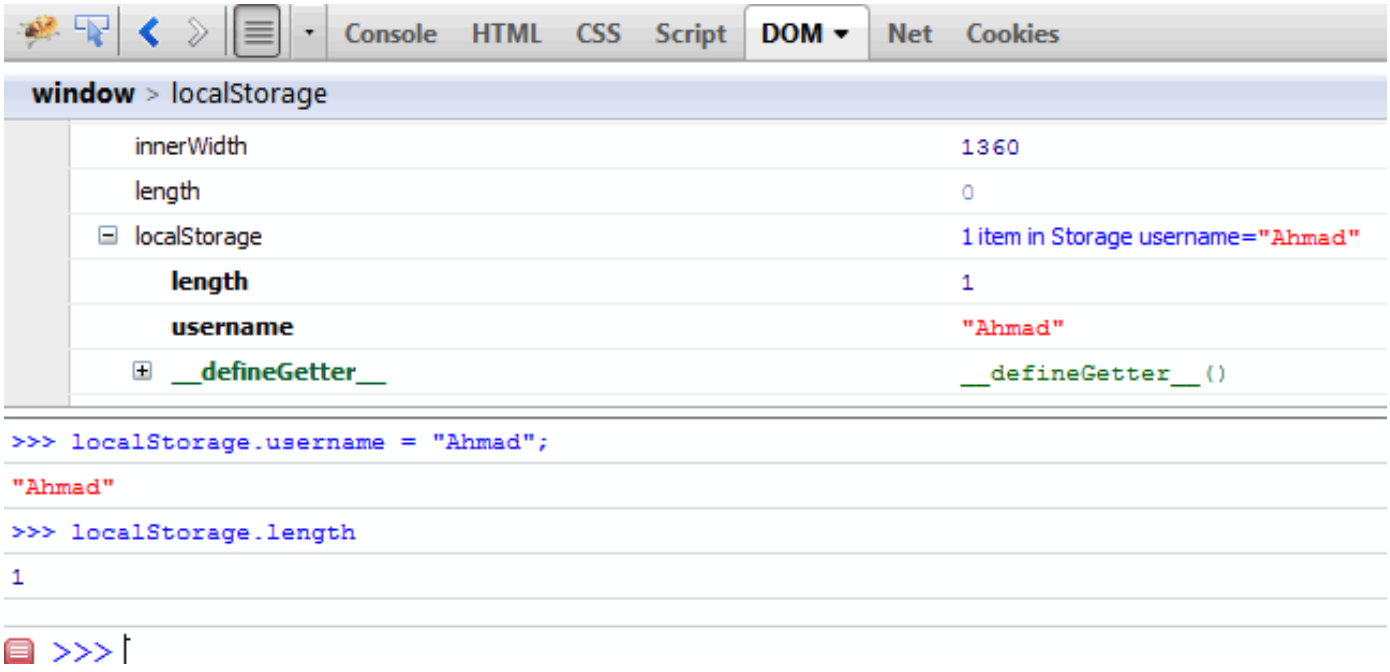
مشابه پنل‌های HTML , CSS, Console در این پنل هم امکان اعمال تغییرات همراه با قابلیت تکمیل خودکار وجود دارد.



localStorage

در HTML5 سیستمی برای ذخیره مقادیر در سمت کاربر، به نام localStorage معرفی شد. در این پنل می‌توانید محتویات آن را بررسی/ویرایش کنید. برای کار با توابع آن هم می‌توانید از پنل Console استفاده کنید. (همچنین می‌توانید از پنل Console بصورت Popup در این پنل و پنل‌های دیگر هم استفاده کنید. به تصویر زیر توجه فرمایید.)

توجه کنید که برای مشاهده‌ی این شئ، باید گزینه‌ی Show DOM Properties فعال باشد و همچنین در Property Path، شئ window فعال باشد.



Breakpoint Column

شما می‌توانید با کلیک بروی ستون سمت چپ پراپرتی ها، آن پراپرتی را تحت نظر گرفته و در صورت تغییر یافتن مقدار آن پراپرتی، کنترل روند اجرای برنامه از همان نقطه را بدست بگیرید. به این صورت که زمانی که کدی پراپرتی موردنظر را تغییر دهد، پنل Script روند اجرای کد را در همان قسمت متوقف می‌کند.

(ممکن است فایرباگ بصورت خودکار به پنل Script سوئیچ نکند و بعد از متوقف شدن برنامه هم اگر به پنل Script سوئیچ کنید نتیجه را نبینید. پس بهتر است قبل از تغییر یافتن پراپرتی مورد نظر و بعد از قرار دادن Breakpoint به پنل Script بروید.)

Context Menu

با راست کلیک کردن در قسمت‌های مختلف پنل، منوهای متفاوتی را خواهید دید. همچنین با راست کلیک کردن بروی مقادیر پراپرتی ها، منوی متناسب با آن مقدار را خواهید دید. مثلا اگر بروی یک تگ HTML در این پنل راست کلیک کنید، منویی که خواهید دید همان منویی است که در پنل HTML مشاهده می‌کردید.

گزینه	Context	توضیحات
Copy Name	Property List	نام پراپرتی را در حافظه کپی می‌کند.
Copy Path	Property List	آدرس پراپرتی را در حافظه کپی می‌کند.
Copy Value	String and Number values	محتوای پراپرتی را در حافظه کپی می‌کند.
Edit Property...	Property List (پراپرتی و توابع کاربر)	پراپرتی را به حالت ویرایش می‌آورد.
Delete Property	Property List (پراپرتی و توابع کاربر)	پراپرتی را حذف می‌کند.
Break On Property Change	Property List (پراپرتی و توابع کاربر)	مشابه پاراگرف قبلی.
Refresh	Property List, Property Path	محتویات پنل را بروزرسانی می‌کند.

برای توابع هم دو منوی اضافی وجود دارد:

"<Log Calls to "<function name

فراخوانی‌های تابع مورد نظر را Log می‌کند. (برای توضیحات بیشتر دستور monitor که از توابع خط فرمان است را ملاحظه بفرمایید.)

Copy Function

نام و بنده‌ی تابع را در حافظه کپی می‌کند.