

عنوان: آموزش فایر باگ - 2# - تب Console

نویسنده: احمد احمدی

تاریخ: ۱۳۹۱/۰۴/۰۵ ۱:۴

آدرس: www.dotnettips.info

برچسب‌ها: FireBug

در [قسمت قبلی](#) با تب‌های فایر باگ 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 فایر باگ 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 کنیم ؟ کلید میانبر (HotKey) اجرای کدها چیست ؟

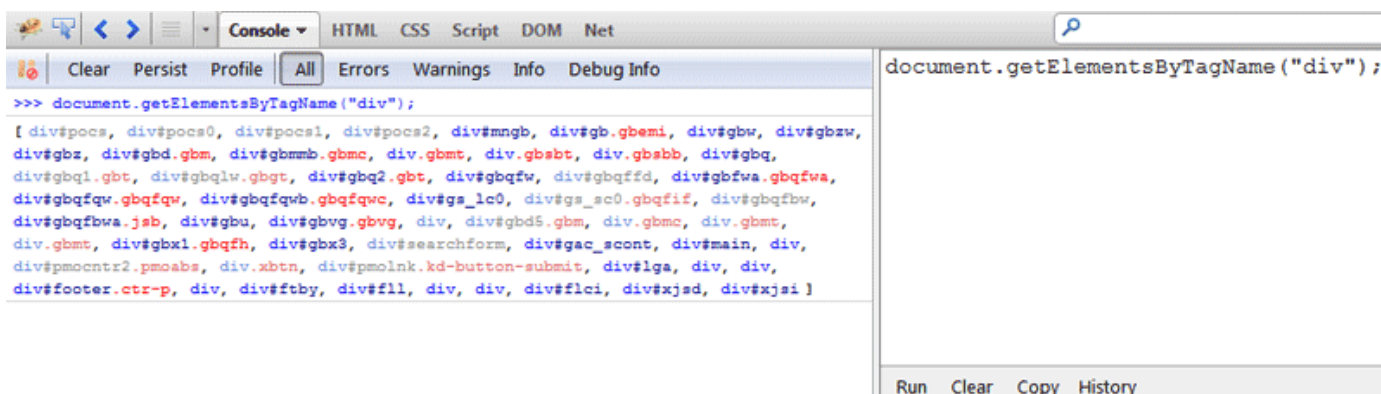
پاسخ : فشردن کلید CTRL + Enter

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 ، کلاس های CSS ای هستند که المنت دارد .
 قسمت هایی که با رنگ آبی تیره و یک شارپ (#) بعد از نام تگ قرار گرفته اند ، ID تگ ها 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 می شود .

دکمه ها و حالت های 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

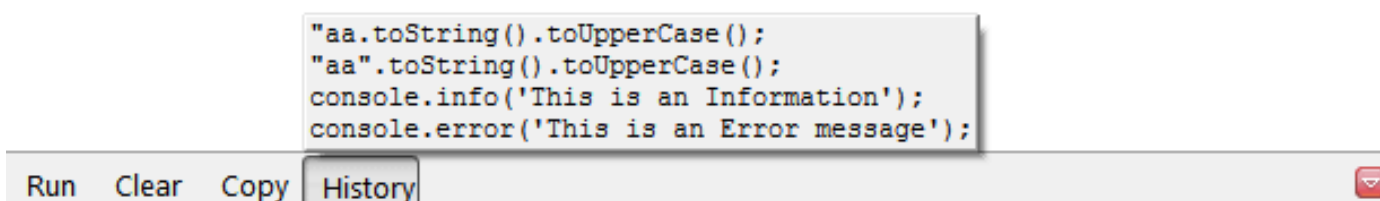
بخش کد نویسی می تواند به 2 شکل نمایش داده شود :

بصورت جعبه چند خطی (Command Editor)

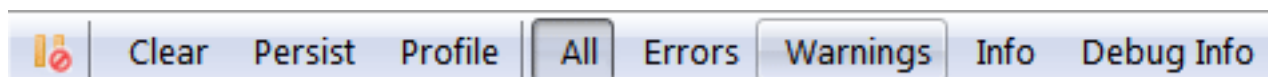
بصورت تک خطی (Command Line)

با دکمه ی قرمز رنگ موجود در پایین - چپ می توانید 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)

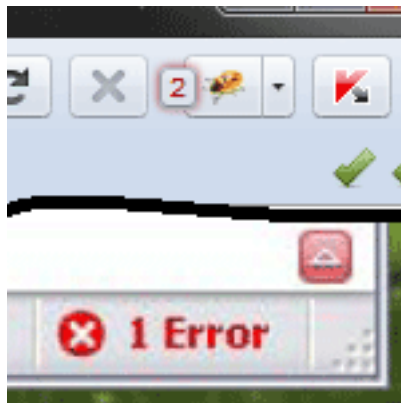
Function	Calls	Percent	Own Time	Time	Avg	Min	Max	File
getCaretDescriptor\$1	93	31.35%	5.54ms	6.313ms	0.068ms	0.039ms	0.305ms	rs=Alt...HRj0V7g (line 234)
o	95	10.28%	1.816ms	5.315ms	0.056ms	0.031ms	0.161ms	rs=Alt...HRj0V7g (line 271)
S	95	9.06%	1.601ms	13.677ms	0.144ms	0.093ms	0.379ms	rs=Alt...HRj0V7g (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 !

