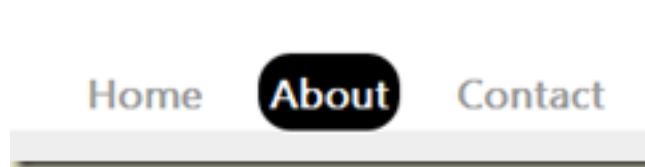


راه حل‌های مختلفی جهت Highlight کردن لینک صفحه جاری وجود دارد و مهم‌ترین کاربرد آن در منوی اصلی سایت است.



در این مطلب سعی داریم با ارائه یک Helper راه حل مناسبی را برای این موضوع ارائه کنیم. مسئولیت این Helper ایجاد لینک است با در نظر گرفتن یک شرط: آیا لینک ایجاد شده به Action جاری اشاره دارد؟ اگر بله یک CSS Class با عنوان `currentMenuItem` به آن اضافه کن.

```
public static MvcHtmlString MenuLink(this HtmlHelper helper, string text, string action, string controller)
{
    var routeData = helper.ViewContext.RouteData.Values;
    var currentController = routeData["controller"];
    var currentAction = routeData["action"];

    if(String.Equals(action, currentAction as string,
        StringComparison.OrdinalIgnoreCase)
        &&
        String.Equals(controller, currentController as string,
            StringComparison.OrdinalIgnoreCase))
    {
        return helper.ActionLink(
            text, action, controller, null,
            new { @class="currentMenuItem" }
        );
    }
    return helper.ActionLink(text, action, controller);
}
```

نحوه استفاده:

```
<li>@Html.MenuLink("Contact", "Contact", "Home")</li>
```

و البته کمی تغییرات در فایل CSS خود را فراموش نکنید:

```
ul#menu li a {
    background: none;
    color: #999;
    padding: 5px;
    border-radius: 15px;
    text-decoration: none;
}

ul#menu li a.currentMenuItem {
    background-color: black;
    color: white;
}
```

همچنین دوستانی که از Bootstrap و البته Navbar آن استفاده می‌کنند می‌توانند با کمی تغییرات در این Helper استفاده بهینه‌ای از آن داشته باشند:

```
public static MvcHtmlString MenuLinkBootstrap(this HtmlHelper helper, string text, string action,
string controller)
{
    var routeData = helper.ViewContext.RouteData.Values;
    var currentController = routeData["controller"];
    var currentAction = routeData["action"];

    if (String.Equals(action, currentAction as string, StringComparison.OrdinalIgnoreCase) &&
String.Equals(controller, currentController as string, StringComparison.OrdinalIgnoreCase))
    {
        return new MvcHtmlString("<li class=\"active\">" + helper.ActionLink(text, action, controller)
+ "</li>");
    }

    return new MvcHtmlString("<li>" + helper.ActionLink(text, action, controller) + "</li>");
}
```

نظرات خوانندگان

نویسنده:

امین

تاریخ:

۹:۲۶ ۱۳۹۱/۰۶/۲۰

مطلب عالی بود مخصوصا بخش آخر که برای bootstrap رو هم گفتین.

نویسنده:

وحید نصیری

تاریخ:

۱۲:۳۷ ۱۳۹۱/۰۷/۰۶

ممنون. مطلب جالبی است. یک راه حل عمومی دیگر مبتنی بر jQuery :

```
//-----جاری صفحه به ازای لینک های بالای صفحه
$(document).ready(function () {
    $("#headermenu a").each(function () {
        var $a = $(this);
        var href = $a.attr("href");
        if (href && (location.pathname.toLowerCase() == href.toLowerCase())) {
            //صفحه جاری را یافتیم
            $a.css({
                "color": "Yellow",
                "border-bottom": "1px solid"
            });
        }
    });
});
```

این روش بر اساس آدرس صفحه جاری و یافتن آن در ناحیه headermenu و سپس رنگی کردن آن کار می کند.

نویسنده:

امیرحسین مرجانی

تاریخ:

۱۶:۲۳ ۱۳۹۱/۰۷/۲۷

ممنونم

نکته خوبی گفتید ... چندوقتی بود دنبال مطلب شما می گشتم که الان به چشمم خورد.

نویسنده:

صابر فتح الهی

تاریخ:

۱۲:۴۸ ۱۳۹۲/۰۱/۰۷

سلام مهندس (تبریک سال نو با تاخیر)

روشی که گفتین به نظر من یه ایراد کوچک داره و اونم اینه که در صورتی لینک رنگی می شه که حتما آدرس مورد نظر با آدرس لینک برابر باشه، در صورتی که لینک مورد نظر یک صفحه فرعی باشه و ان لینک در لیست منو وجود نداشته باشه هیچ لینکی رنگی نمی شه، مثلا در سایت [دات نت تیپس](#) در صورتی که کاربر روی لینک جزئیات یک پروژه کلیک کنه لینک پروژه ها رنگی میشه در صورتی که ادرسش توی منوها موجود نیست، من وقتی کدهای سایت مذکور بررسی کردم فهمیدم به جای اون شما از تابع startsWith (تعریف شده توسط شما) استفاده کردین، در این صورت لینکهایی که با قسمتی از آدرس مذکور شروع بشه تغییرات رنگ (استایل مورد نظر) رو اون اعمال میشه. اگر اشتباه می کنم لطفا تصحیح بفرمایید.

نویسنده:

وحید نصیری

تاریخ:

۱۲:۵۸ ۱۳۹۲/۰۱/۰۷

بله. هدف این بوده تا حد ممکن شخص بدونه در چه قسمتی از سایت است. مثلا اگر به جزئیات یک پروژه مراجعه کرده، باید بدونه در قسمت پروژه ها هست. این کد جدید و اصلاح شده است:

```

if (typeof String.prototype.startsWith != 'function') {
    // see below for better implementation!
    String.prototype.startsWith = function (str) {
        return this.indexOf(str) == 0;
    };
}

$(document).ready(function () {
    $("#headermenu a").each(function () {
        var $a = $(this);
        var href = $a.attr("href");

        var path = $.trim(location.pathname.toLowerCase());
        var menuHref = $.trim(href.toLowerCase());

        if (href && (path == menuHref || (menuHref != '/' && path.startsWith(menuHref)))) {
            // صفحه جاری را یافتیم
            $a.css({
                "color": "Yellow",
                "border-bottom": "2px solid"
            });
        }
    });
});

```

نویسنده:

مهدی موسوی

تاریخ:

۱۵:۱۱ ۱۳۹۲/۰۱/۰۸

سلام.

به گمانم می‌تونید با یه تغییر کوچک، کدتون رو بهینه کنید. شما می‌تونید پس از یافتن صفحه جاری، return false کنید تا each به سراغ مابقی anchorهای موجود نره (چون لینک مورد نظر پیدا شده و CSS مربوطه هم اعمال شده، بنابراین دیگه نیازی نیست این حلقه به کارش ادامه بده؛ مگر اینکه بگیم بیش از یک لینک در صفحه وجود داره که در اون صورت باید location.pathname خارج each یکبار cache کنیم و ...).

البته، می‌تونستیم ابتدا anchorهای مورد نظر رو filter کنیم، بعد با اضافه کردن کلاس (فرضا) highlight همون کارو انجام بدیم:

```

$(function(){
    var loc = location.pathname.toLowerCase();
    $('#headermenu a').filter(function(){
        var href = this.getAttribute('href');
        return href && href.toLowerCase() == loc;
    }).addClass('highlight');
});

```

طبیعتا اگر فرض کنیم که کلیه URLها از سمت سرور بصورت Lowercase رندر میشن، اونوقت می‌تونیم از کد فوق نیز پرهیز کنیم:

```

$('#headermenu a[href="" + location.pathname + "']").addClass('highlight');

```

موفق باشید.

پ.ن.: البته می‌تونیم از is هم استفاده کنیم که مقایسه رشته‌ها بطور خودکار case-insensitive نیز باشه.

نویسنده:

Mehdi

تاریخ:

۰۳:۳۳ ۱۳۹۲/۰۳/۱۸

منظور از ناحیه headermenu چیست؟

نویسنده:

وحید نصیری

تاریخ:

۱۷:۱۴ ۱۳۹۲/۰۳/۱۹

یک مثال فرضی بود. مثلاً یک div با id مساوی headermenu.