

## فهرست مطالب

۲.....	صفحه اصلی در ASP.NET (در یک نگاه)
۲.....	نحوه کار صفحه اصلی
۲.....	• صفحه های اصلی
۳.....	• مکان نگهدارنده محتوا با قابلیت جایگزینی
۴.....	• صفحه های محتوا
۶.....	فوائد استفاده از صفحه های اصلی
۶.....	رفتار زمان اجرای یک صفحه اصلی
۸.....	صفحه اصلی و مسیر صفحه محتوا
۹.....	• ارجاع دادن به منابع خارجی
۹.....	• کنترل های سرور
۱۰.....	• دیگر عناصر
۱۰.....	صفحه اصلی و تم ها
۱۰.....	حوزه صفحه اصلی

## صفحه اصلی در ASP.NET (در یک نگاه)

صفحه اصلی<sup>۱</sup> به شما امکان اضافه نمودن یک چارچوب برای سایت را می دهد. با استفاده از یک صفحه اصلی این امکان را خواهید داشت که برای کلیه یا برخی از صفحه های برنامه یک چارچوب استاندارد تهیه نمایید. پس از آن فقط محتویات صفحه های را ساخته و طراحی می کنید. وقتی که کاربر درخواست یک صفحه را می کند، محتویات صفحه همراه با صفحه اصلی ادغام شده و یک صفحه کامل را تشکیل می دهد و سپس برای کاربر به عنوان یک صفحه ارسال می گردد.

### نحوه کار صفحه اصلی

صفحه اصلی واقعا دو تکه است، یکی خود صفحه اصلی و دیگری محتویات صفحه. هرچند که امکان ترکیب صفحه اصلی ها به شکل تو در تو نیز هست.

### صفحه های اصلی

صفحه اصلی یک فایل asp.net با پسوند master. (برای مثال MySite.master) با چارچوبی است که از قبل توسط تگهای html و متون و عناصر سروری ایجاد شده است. صفحه های اصلی با علامت @master در ابتدای تعریف فایل معرفی می شوند که بعداً بصورت مستقیم با @page جایگزین می گردد و در نتیجه یک صفحه.aspx را ایجاد می نماید. به مثالهایی که در زیر آمده است دقت نمایید:

**C#**

```
<%@ Master Language="C#" %>
```

در تعریف @master می توانید از تعاریفات دیگری مانند، @control و ... استفاده نمایید. در مثال زیر تعرف master page حاوی معرفی فایل کد و معرفی کلاسی برای صفحه است.

**C#**

```
<%@ Master Language="C#"
CodeFile="MasterPage.master.cs" Inherits="MasterPage"
%>
```

باید اضافه نماییم که، یک صفحه اصلی همچنین می تواند شامل همه عناصر ابتدایی یک صفحه مانند `html`, `head` و `form` باشد. برای مثال در یک صفحه اصلی ممکن است بخواهید از یک جدول `html` برای ساخت چارچوب مورد نظر، یک `img` برای علامت تجاری شرکت، یک متن ساده برای قانون کپی رایت و کنترل‌های سرور به عنوان لینک‌های سایت خود استفاده نمایید. شما می توانید از هر عنصر `HTML` و هر عنصر `ASP.NET` به عنوان قسمتی از صفحه اصلی استفاده نمایید.

### مکان نگهدارنده محتوا<sup>۱۱</sup> با قابلیت جایگزینی

به اضافه متون ثابت و کنترل‌هایی که در همه صفحه ها دیده می شوند، یک صفحه اصلی می تواند شامل چندین کنترل `ContentPlaceholder` باشد. این کنترل‌های نگهدارنده، محدوده ای را تعریف می کنند که محتویات قابل جایگزین باید در آنجا دیده شوند. محتوا قابل جایگزین در صفحه ها تعریف می شوند. پس از تعریف کنترل‌های `ContentPlaceholder` یک صفحه اصلی باید مانند زیر ایجاد شود:

**C#**

```
<%@ Master Language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server" >
    <title>Master page title</title>
</head>
<body>
    <form id="form1" runat="server">
```

```
<table>
  <tr>
    <td><asp:contentplaceholder
id="Main" runat="server" /></td>
    <td><asp:contentplaceholder
id="Footer" runat="server" /></td>
  </tr>
</table>
</form>
</body>
</html>
```

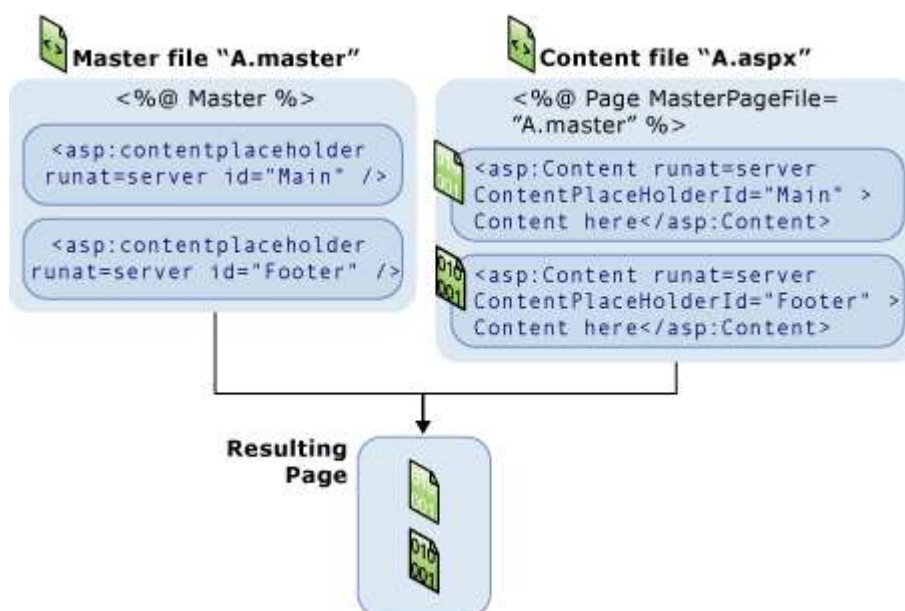
### صفحه های محتوا<sup>iii</sup>

شما محتوا صفحه ها را توسط کنترل های نگهدارنده مکان با ایجاد صفحه ای مستقل برای صفحه ها اصلی تعریف می کنید، که هر کدام یک صفحه ASP.NET است. (فایل های .aspx و فایل های اختیاری code-behind) که به یک صفحه اصلی مرتبط شده اند. این اتصال در صفحه محتوایی دارای راهنمای @Page با استفاده از خصوصیت MasterPageFile صورت می گیرد. برای مثال یک صفحه محتوا باید راهنمای @Page را دارا باشد تا به صفحه Master1.master اتصال برقرار کند.

#### C#

```
<%@ Page Language="C#"
MasterPageFile="~/MasterPages/Master1.master"
Title="Content Page"%>
```

در صفحه محتوا، شما باید یک کنترل محتوا ایجاد کرده و آن را به کنترل ContentPlaceHolder صفحه اصلی ارتباط دهید. برای مثال صفحه اصلی باید دارای نگهدارنده مکان با نامهای Main و Footer باشد. در صفحه محتوا، شما می توانید دو کنترل Content ایجاد نمایید. یکی از آنها به کنترل ContentPlaceHolder با نام Main اشاره نموده و دیگری به کنترل ContentPlaceHolder با نام Footer همانند شکل زیر اشاره می کند:



پس از ساخت کنترل‌های Content شما باید متن و کنترل‌ها را به آن اضافه کنید. در صفحه محتوا هر چیزی نمی‌تواند داخل کنترل‌های Content قرار بگیرد (مگر بلاک‌های کد سمت سرور) و این باعث تولید خطا خواهد شد. شما می‌توانید هر کاری را که در یک صفحه ASP.NET انجام می‌دادید در یک صفحه محتوا نیز انجام دهید. برای مثال می‌توانید محتوایی مورد نظر را برای کنترل Content با استفاده از کنترل‌های سروری و دستورات بانک اطلاعاتی یا دیگر مکانیزم‌های پویا ایجاد نمایید.

یک صفحه محتوا باید شبیه به دستورات زیر باشد:

**C#**

```

<% @ Page Language="C#"
MasterPageFile="~/Master.master" Title="Content Page
1" %>
<asp:Content ID="Content1"
ContentPlaceHolderID="Main" Runat="Server">
    Main content.
</asp:Content>

<asp:Content ID="Content2"
ContentPlaceHolderID="Footer" Runat="Server" >
    Footer content.
</asp:content>
  
```

راهنمای @Page صفحه محتوا را به صفحه اصلی خاصی اتصال می دهد. و یک عنوان برای صفحه که می خواهد با صفحه اصلی ادغام شود را تعریف می کند. دقت شود که صفحه محتوا شامل هیچ دستور html خارج از کنترل‌های Content نباشد. (صفحه اصلی باید عنصر head را با خصوصیت runat="server" دارا باشد که این می تواند باعث تنظیم ادغام در زمان اجرا شود).

شما می توانید چندین صفحه اصلی را با چارچوبهای مختلفی برای قسمتهای مختلف سایت تعریف نمایید. و دسته های مختلف صفحه های محتوا را برای هر صفحه اصلی.

### فواید استفاده از صفحه های اصلی

صفحه های اصلی این امکان را فراهم می کند که برنامه نویسان، کدهای قدیمی ایجاد شده را با کپی متون، کنترلها، استفاده از frameset ها، استفاده از فایل های include برای عناصر مشترک، استفاده از کنترل های asp.net کاربر و غیره مورد استفاده قرار دهند. فواید استفاده از صفحه های اصلی شامل موارد زیر است:

- آنها به شما اجازه متمرکز کردن امکانات مشترک صفحه ها را می دهند که در این حالت می توانید بروزرسانی سایت را فقط با اعمال تغییرات در یک مکان انجام دهید.
- آنها به ما امکان ساخت آسان مجموعه ای از کنترلها و کدها را می دهند که نتیجه آن را می توان به مجموعه ای از صفحه ها اعمال نمود. برای مثال، شما می توانید از کنترلها در صفحه های اصلی برای ساخت منو استفاده کنید که به همه صفحه ها اعمال می شود.
- آنها یک مدل شی گرا را فراهم می کنند و به شما اجازه تنظیم دلخواه صفحه اصلی، از روی یک صفحه محتوا منحصر بفرد را می دهند.

### رفتار زمان اجرای یک صفحه اصلی

در زمان اجرا، صفحه های اصلی در موارد زیر بترتیب قرار می گیرند:

کاربر درخواست یک صفحه را با تایپ آدرس صفحه محتوا می دهد.

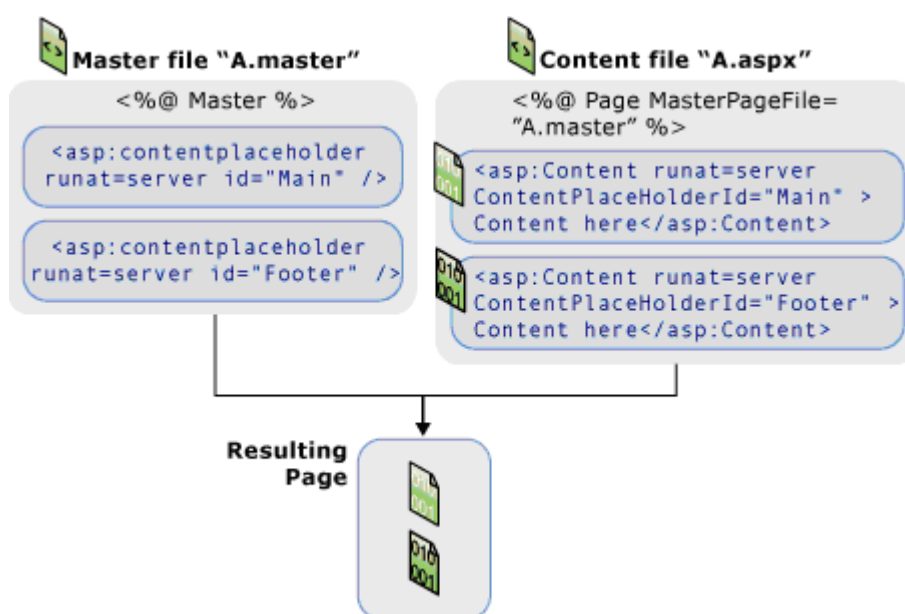
زمانی که صفحه بدست می آید، راهنمای @Page خوانده می شود، اگر راهنما به یک صفحه اصلی اشاره کند، صفحه اصلی نیز خوانده خواهد شد. اگر این اولین مرتبه درخواست صفحه باشد، هر دو صفحه کامپایل خواهند شد.

صفحه اصلی با محتوا بروز شده در صفحه محتوا ادغام می شود.

محتوا منحصر بفرد کنترل Content با کنترل ContentPlaceHolder در صفحه اصلی ادغام می شود.

صفحه ادغام شده بدست آمده برای مرورگر ترجمه می شود.

فرآیند فوق در دیاگرام زیر قابل دیدن است:



از دیدگاه کاربر، صفحه اصلی و صفحه محتوا ترکیب شده یک صفحه هستند. و آدرس صفحه مربوط به صفحه محتوا است.

از دیدگاه برنامه نویس دو صفحه در نقش ظروف مجزا برای کنترل های شان عمل می کنند. صفحه محتوا نقش ظرف را برای صفحه اصلی دارد. هر چند، شما می توانید به عنصرهای عمومی صفحه اصلی (کد) در صفحه محتوا ارجاع داشته باشید.

نکته اینکه صفحه اصلی به عنوان بخشی از صفحه محتوا تبدیل می شود. در این حالت، صفحه اصلی نقشی مانند نقش کنترل های کاربر به عنوان فرزند صفحه محتوا و به عنوان ظرف در صفحه ای که قرار گرفته است را دارد. که در این حالت صفحه اصلی ظرفی است برای همه کنترل های سمت سرور که برای مرورگر ترجمه می شوند. کنترل درختی برای یک صفحه اصلی و محتوا ادغام شده به کل زیر شبیه خواهد بود:

**Pseudo code**

```
Page
    Master Page
        (Master page markup and controls)
        ContentPlaceHolder
            Content page markup and server
controls
    (Master page markup and controls)
    ContentPlaceHolder
        Content page markup and server
controls
    (Master page markup and controls)
```

این یک دیاگرام ساده شده است. اگر صفحه محتوا دارای کنترل Content نباشد، صفحه اصلی باید دارای کنترل ها و کد در کنترل های ContentPlaceHolder باشد. در حالت عمومی این ساختار تاثیری بر نحوه ساخت صفحه شما یا برنامه شما نخواهد داشت. در بسیاری حالات، اگر شما خصوصیت Page-Wide صفحه اصلی را تنظیم کرده باشید، می تواند بر نحوه رفتار صفحه محتوا تاثیر بگذارد، زیرا صفحه اصلی یک کنترل بسته برای کنترل های داخل آن است. برای مثال اگر خصوصیت EnableViewState در صفحه محتوا true انتخاب شده باشد، اما این خصوصیت در صفحه اصلی مقدار false انتخاب شده باشد، view state غیر فعال خواهد شد زیرا تنظیمات صفحه اصلی دارای اولویت بیشتری است.

**صفحه اصلی و مسیر صفحه محتوا**

وقتی که یک صفحه محتوا درخواست می شود، محتوا با صفحه اصلی ترکیب می شود و صفحه در قسمت صفحه محتوا اجرا می شود. برای مثال، اگر شما خصوصیت CurrentExecutionFilePath از شی HttpRequest بگیرید، چه در کد صفحه اصلی یا در صفحه محتوا، مسیر صفحه محتوا را نمایش می دهد.



صفحه اصلی و صفحه محتوا می توانند در یک پوشه قرار نگرفته باشند. با استفاده از خصوصیت MasterPageFile در راهنمای دستور @Page یک فایل master. با صفحه محتوا توسط ASP.NET رندر شده و بصورت یک صفحه رندر شده مجزا در می آید.

### ارجاع دادن به منابع خارجی

هر دوی صفحه های اصلی و محتوا می توانند دارای عناصر و کنترل هایی باشند که به منابع خارجی ارجاع داده اند. برای مثال، هر دو می توانند دارای یک کنترل تصویر که به یک فایل تصویری ارجاع شده یا یک پیوند برای ارجاع به یک صفحه دیگر باشند.

زمینه برای محتوا ترکیب شده با صفحه اصلی، صفحه محتوا است. و این می تواند بروی نحوه مشخص نمودن URL ها برای منابع تاثیر گذار باشد، مانند فایل های تصویری و صفحه های مقصد برای پیوندها.

### کنترل های سرور

ASP.NET بصورت پویا آدرسهای URL کنترل های سروری صفحه اصلی که به منابع خارجی ارجاع می کنند را تغییر می دهد. برای مثال ممکن است شما یک کنترل تصویر را در صفحه اصلی قرار دهید و خصوصیت imageUrl را نسبت به صفحه اصلی تنظیم کنید. در زمان اجرا، ASP.NET مسیر را نسبت به صفحه محتوا اصلاح کرده و آدرس اصلاح شده مورد استفاده قرار می گیرد.

ASP.NET می تواند آدرسهای URL را در شرایط زیر اصلاح نماید:

- URL خصوصیت یک کنترل سروری ASP.NET باشد
- خصوصیت آدرس بصورت داخلی در کنترل تعیین شده باشد. (خصوصیت توسط صفت UrlPropertyAttribute علامت گذاری شده است.) در اصطلاح عملی، خصوصیهایی کنترل های سروری ASP.NET که عموماً برای ارجاع به منابع خارجی استفاده شده اند، به این روش علامت گذاری شده اند.

### دیگر عناصر

ASP.NET نمی تواند آدرس URL کنترل‌های غیر سروری را تغییر دهد. برای مثال، اگر شما از عنصر `img` در یک صفحه اصلی استفاده کرده باشید و صفت `src` را با یک URL تنظیم کرده باشید، ASP.NET آدرس را تغییر نخواهد داد. در این حالت آدرس در زمینه صفحه محتوا ایجاد می شود.

در حالت کلی، زمانی که با یک عنصر در صفحه اصلی کار می شود بهتر است که شما از کنترل‌های سروری استفاده کنید. برای مثال در صورت استفاده از عنصر `img` می توانید از کنترل سرور `image` استفاده کنید. در این حالت ASP.NET می تواند آدرس URL را بدرستی بسازد.

### صفحه اصلی و تم‌ها<sup>۱۷</sup>

شما نمی توانید تم‌ها را بطور مستقیم به صفحه‌های اصلی اعمال کنید. اگر صفت `theme` را به راهنمای `@master` اضافه کنید، در زمان اجرا یک خطا ایجاد خواهد شد.

هرچند که تم‌ها به صفحه‌های اصلی در موقعیتهای زیر اعمال می شوند:

- اگر یک تم در صفحه محتوا تعریف شده باشد، تم محتوا صفحه پس از ترکیب با صفحه اصلی به صفحه اصلی نیز اعمال می شود.
- اگر برای کل سایت یک تم تعریف شده باشد.

### حوزه صفحه اصلی

شما می توانید صفحه محتوا را به صفحه اصلی در سه سطح اضافه کنید:

- در سطح صفحه: شما می توانید از راهنمای صفحه در هر صفحه محتوا برای اتصال به صفحه اصلی مانند مثال زیر، استفاده کنید:

**C#**

```
<%@ Page Language="C#"
MasterPageFile="MySite.Master" %>
```

- در سطح برنامه: با ساخت تنظیمات در عنصر pages فایل تنظیمات برنامه ( Web.Config ) می توانید بصورت اتوماتیک صفحه اصلی را به همه صفحه های ( .aspx ) اضافه کنید. عنصر مورد نظر می تواند شبیه مثال زیر باشد:

```
<pages masterPageFile="MySite.Master" />
```

- اگر از استراتژی استفاده شده باشد، همه صفحه های asp.net در برنامه که دارای کنترل Content باشد با صفحه اصلی مشخص شده ادغام می شوند. (اگر صفحه asp.net دارای کنترل Content نباشد، صفحه اصلی به آن اعمال نمی شود)
- در سطح پوشه: این استراتژی مانند اتصال در سطح برنامه است، به جز اینکه شما تنظیماتی را فقط در فایل Web.Config در یک پوشه قرار داده باشید. اتصال صفحه اصلی و محتوا صفحه فقط در این پوشه اعمال می شود.

---

Master page<sup>i</sup>  
ContentPlaceHolder<sup>ii</sup>  
Content Page<sup>iii</sup>  
Theme<sup>iv</sup>