Creating Consistent Looking Web Sites

Asst. Prof. Dr. Özgü Can

Master Pages

- You only need to modify the master page and the pages based on this master will pick up the changes automatically.

- Looks like a normal ASPX page
  - Not a true ASPX page
    - Can not be requested in the browser.
    - It only serves as the template that real web pages (*content pages)* are based on.

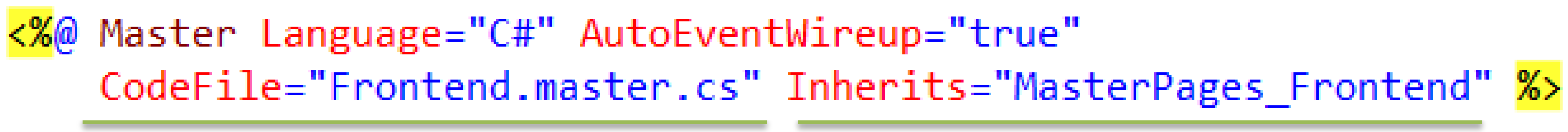
Master Pages

- Instead of the `@ Page` directive, a master page uses an `@ Master` directive that identifies the file as a master page:

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

Master Pages

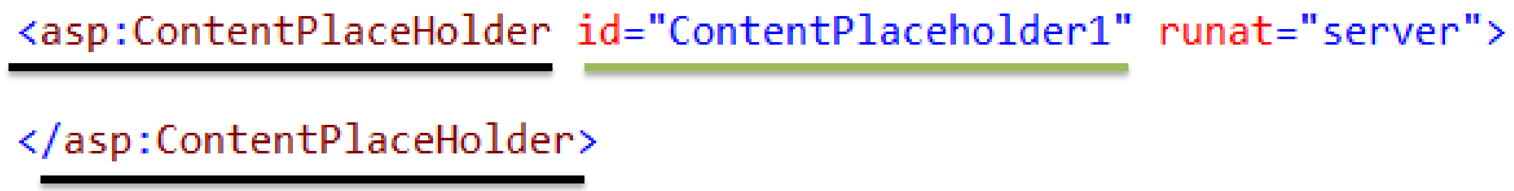
- Just like a normal ASPX page, a master page can have a Code Behind file, identified by its `CodeFile` and `Inherits` attributes:



```
<%@ Master Language="C#" AutoEventWireup="true"
    CodeFile="Frontend.master.cs" Inherits="MasterPages_Frontend" %>
```

Master Pages

- To create regions that content pages can fill in, you need to define `ContentPlaceHolder` controls in your page like this:

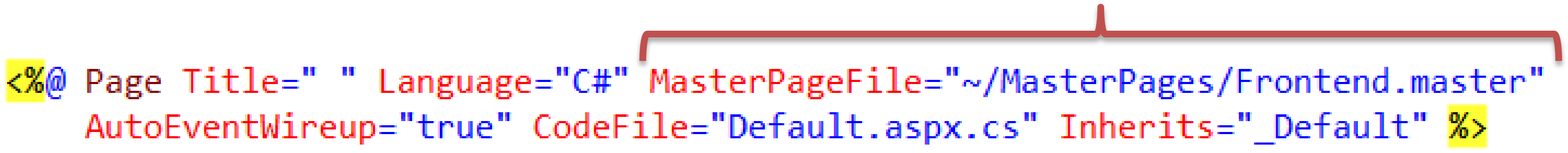


```
<asp:ContentPlaceHolder id="ContentPlaceholder1" runat="server">
</asp:ContentPlaceHolder>
```

- You can create as many placeholders as you like.

Content Pages

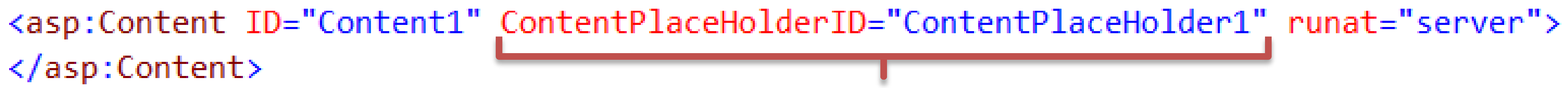
- Normal ASPX files
- Connected to a master page using the `MasterPageFile` attribute of the `Page` directive:



```
<%@ Page Title=" " Language="C#" MasterPageFile="~/MasterPages/Frontend.master"
    AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
```

Content Pages

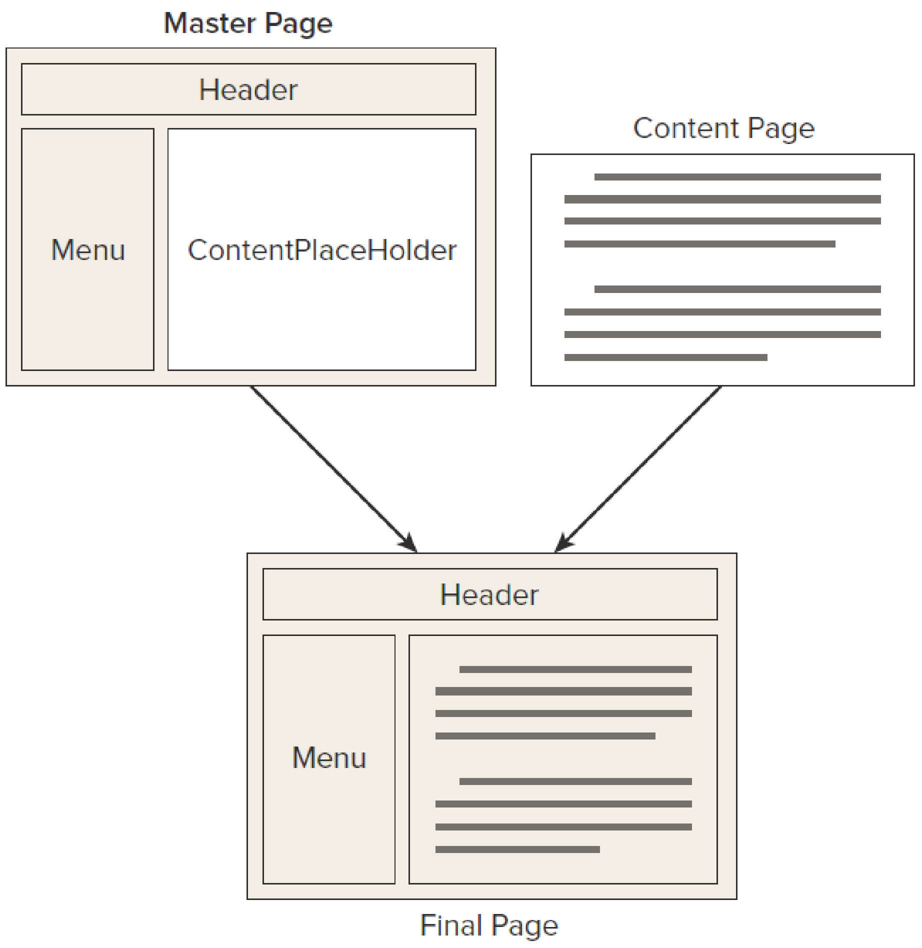
- The page-specific content is then put inside a `Content` control that points to the relevant `ContentPlaceHolder`:



```
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>
```

Points to the `ContentPlaceHolder` that is defined in the master page.

Runtime


Master Page
Header
Menu
ContentPlaceHolder
Content Page
Header
Menu
Final Page

Master Page Caveats

- In a normal ASPX page:

```
<asp:Button ID="Button1" runat="server" Text="Click Me" />
```

- Inside an `<asp:Content>` control:

Auto generated `ID` of the master page

```
<input type="submit" name="ctl00$cpMainContent$Button1" value="Click Me"
        id="cpMainContent_Button1" />
```

ID of the `ContentPlaceHolder` control

Master Page Caveats

- Nested master pages:

```
<input type="submit" name="ctl00$ctl00$cpMainContent$ContentPlaceHolder1$Button1"
      value="Click Me" id="cpMainContent_ContentPlaceHolder1_Button1" />
```

- You should keep the IDs of your `ContentPlaceHolder` and `Content` controls as short as possible.

Base Page

- By default, pages in your web site inherit from the `Page` class defined in the `System.Web.UI` namespace.
  - However, in some circumstances this behavior is not enough and you need to add your own stuff.
- You can easily insert your own `base page` class between a web page and the standard `Page` class.

Page Life Cycle

1. Page Request
2. Start
3. Page Initialization
4. Load
5. Validation
6. Postback Event Handling
7. Rendering
8. Unload

Please READ http://msdn.microsoft.com/en-us/library/ms178472(v=vs.100).aspx