

# Pages maîtres et thèmes

# Page Maître

- Une page maître est une page Asp.Net qui possède les caractéristiques suivantes:
  - Classe de base : MasterPage
  - Extension: master
  - Possède la directive Master au lieu de la directive Page
- Création d'une page maître
  - Ajouter un nouvel élément de type : Page maître
  - Les zones modifiables sont définies par des éléments de type **ContentPlaceholder**
  - Le titre des pages de contenu peut être défini dans la directive page.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="page.master.cs"
Inherits="WebApplication1.page" %>
<!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>
  <asp:ContentPlaceholder ID="head" runat="server">
  </asp:ContentPlaceholder>
</head>
<body><form id="form1" runat="server"><div>
  <asp:ContentPlaceholder ID="ContentPlaceholder1" runat="server">

  </asp:ContentPlaceholder>
</div></form></body></html>
```

# Pages de contenu

- Création
  - Ajouter une nouvelle page Web de type « Web Form avec page maître »
  - Définir
    - Title
    - Les zones Content
- Remarque
  - Les propriétés définies dans la page maître sont prioritaires sur les propriétés définies dans les page de contenu
  - La page de contenu est attachée à la page maître à l'aide de la propriété `MasterPageFile`
  - Le lien entre les pages maîtres et les pages de contenu peut être aussi effectué dans le fichier `web.config` :

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

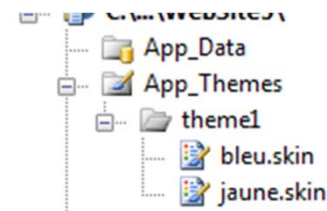
```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs" Inherits="WebApplication1.WebForm3" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>
```

# Thèmes

- Un thème est un ensemble de définitions de propriétés, de styles et d'images.
- Un thème peut être appliqué à une page ou bien à toutes les pages d'un site

## Définition d'un thème

- Un thème est un ensemble de fichiers .skin enregistré dans un sous dossier nommé App\_Themes. (Ajouter le dossier ASP.NET/Thème)
- Un thème peut être constitué de:
  - Fichiers d'apparence ( skin ) contenant les valeurs par défaut des propriétés d'un ou de plusieurs contrôles.
  - Feuilles de styles css
  - Images et autres ressources



## Création d'un thème

1. Ajouter le dossier ASP.Net App\_Themes à l'application web
2. Créer un dossier pour chaque thème de votre application (le nom du dossier représente le nom du thème).
3. Ajouter les fichiers d'apparence, feuilles de style et images dans le dossier du thème.

## Application d'un thème

Pour appliquer un thème il faut définir l'attribut Theme ou bien l'attribut StyleSheetTheme.

- A une page: `<%@ Page ...Theme="theme1" ...%>`
- Appliquer le thème à toutes les pages dans le fichier web.config, :

```
<configuration> <system.web> <pages theme="theme1"> </ pages >< /system.web>
</configuration>.
```

## Remarque:

- l'attribut StyleSheetTheme applique le thème avant l'exécution des propriétés Set des Contrôles, alors que l'utilisation de la propriété Theme permet d'appliquer le thème après exécution des propriétés Set des contrôles.

## Création d'un thème global

- Un thème global peut être utilisé par toute application sur le même serveur web, et doit être stocké dans l'un des dossiers suivants:
  - `<iis_default_root>\Aspnet_client\System_web\<version>\Themes.`
  - `%SystemRoot%\Microsoft.NET\Framework\<version>\ASP.NETClientFiles\Themes`

# Fichiers d'apparence (fichiers skins)

- Un fichier skin peut être créé pour un ou plusieurs types de contrôles, il existe deux types de fichiers d'apparence:
  - Fichiers d'apparence par défaut: ils sont appliqués sur tous les contrôles de même type, exemple:

```
<asp:TextBox BackColor="Blue" ForeColor="White" Runat="Server" />  
<asp:Label BackColor="White" ForeColor="Blue" Runat="Server" Font-  
Bold="True" />
```

- Les fichiers d'apparence nommés: le skin est appliqué aux contrôles ayant un SkinId .

```
<asp:Label SkinId="skin1" BackColor="White" ForeColor="Blue"  
Runat="Server" Font-Bold="True" />
```

- Ajouter une image à un thème

Dans le fichier d'apparence:

```
<asp:Image runat="server" ImageUrl="~/App_Themes/theme1/logo.gif"  
SkinId="logo" />
```

Dans la page de contenu:

```
<asp:Image ID="Image1" runat="server" SkinID="logo"/>
```

## Ordre de précedence des definitions d'attributs

1. l'attribut Theme dans la directive Page
2. `<pages Theme="nomTheme">` dans la section `<system.web>` du fichier Web.config
3. Les attributs definis dans les controles
4. L'attribut StyleSheetTheme dans la directive Page
5. `<pages StyleSheetTheme="nomTheme">` dans le fichier Web.config.