

ASP.NET Fondamental

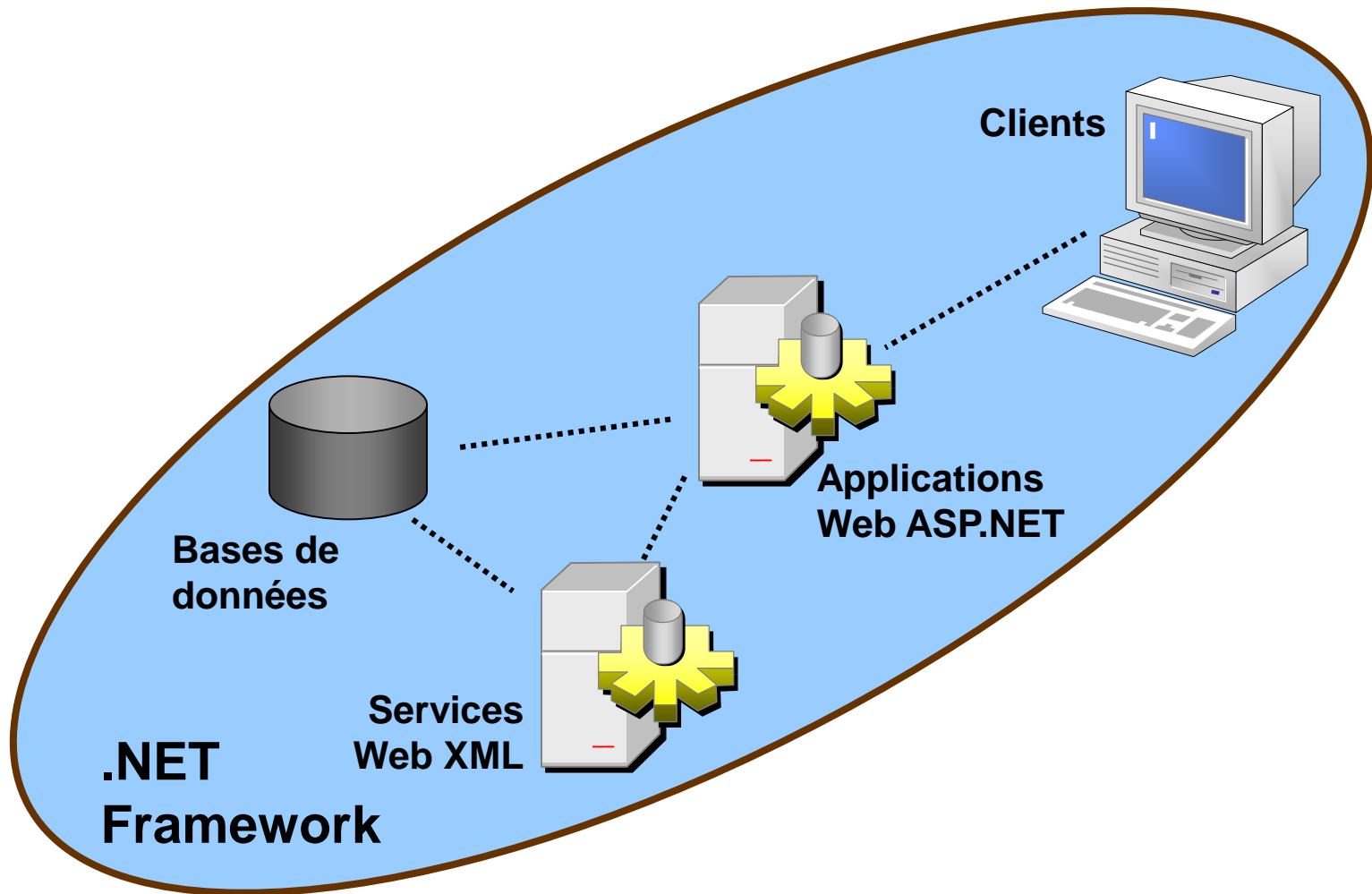
1^{ère} approche

Table des matières

- I. Introduction au Framework Microsoft.Net et à ASP.Net
 - I.1.Présentation de Microsoft .NET Framework*
 - I.2. Composants du .NET Framework*
 - I.3.Points essentiels du .NET Framework*
 - I.4.Qu'est ce qu' ASP.NET?*
 - I.5.Les avantages du modèle ASP.Net*
 - I.6. Installation de l'environnement de travail*
 - I.7.L 'espace de travail*
- II. Présentation des principes d'une architecture Web
 - II.1.Le formulaire Web*
 - II.2.Compilation et exécution*
 - II.3.Application WEB ASP.Net*
- III. Création d'un formulaire simple
- IV. Les Webs Controls

I. Introduction au Framework Microsoft.Net et à ASP.Net

Présentation de Microsoft .NET Framework



Qu'est ce qu' ASP.NET?

- **ASP.NET** est un ensemble de technologies de programmation web créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites web dynamiques, des applications web ou encore des web services.
- **ASP.NET** fait partie de la plateforme Microsoft.Net et est le successeur de la technologie Active Server Page ou ASP
- Les programmeurs peuvent écrire du code ASP.NET en utilisant n'importe lequel des langages de programmation supportés par le Framework.Net, généralement **C#, VB.Net ou Jscript.Net**, mais aussi d'autres langages "indépendants" de Microsoft tels que **Perl et Python**.

Qu'est ce qu' ASP.NET?

- s'appuie sur le Framework .NET (côté serveur)
- basé sur un modèle composants (côté serveur)
- prend en compte les capacités respectives des différents navigateurs (support JavaScript, DHTML, ...)
- ne nécessite rien de particulier sur le client
- moteur de rendu HTML

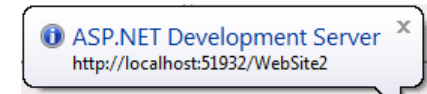
Les avantages du modèle ASP.Net

- Puissance du Framework .Net
- Composants plus riches que les contrôles HTML
- Simplification du développement d'applications WEB
- Programmation orientée objet
- Adaptation : le composant adapte son rendu au type de navigateur client

Installation de l'environnement de travail

- Le serveur Web sous Windows s'appelle IIS : *Internet Information Server*

RMQ: Il n'est pas nécessaire d'installer un serveur Web pour développer. Un serveur Web dit « de développement » est intégré à l'environnement de travail. Nous le verrons en activité, lors de l'exécution de l'application grâce à une petite bulle informative en bas à droite de l'écran :



- Microsoft .NET Framework

Le package redistribuable de Microsoft .NET Framework 2.0 installe le runtime .NET Framework et les fichiers associés requis pour l'exécution d'applications développées pour le .NET Framework 2.0

- IDE : Microsoft Visual Studio 2012, Visual Web Developer 2005, Express Edition



L'espace de travail

Barre d'outils

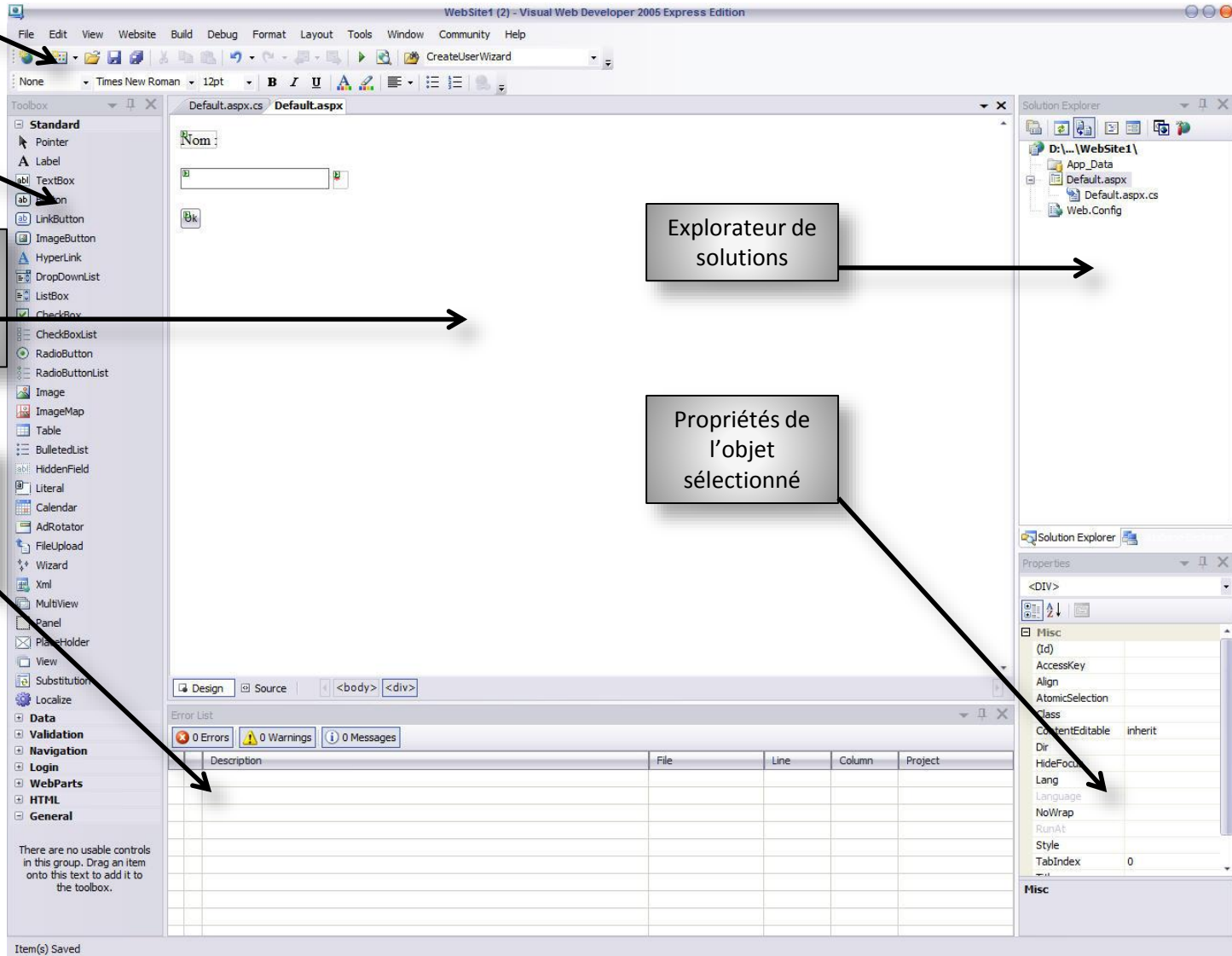
Boîte à outils

Zone de travail
(design, source et code views)

Zone de messages
d'erreur et des alertes

Explorateur de solutions

Propriétés de l'objet sélectionné



II. Présentation des principes d'une architecture Web

Le formulaire Web

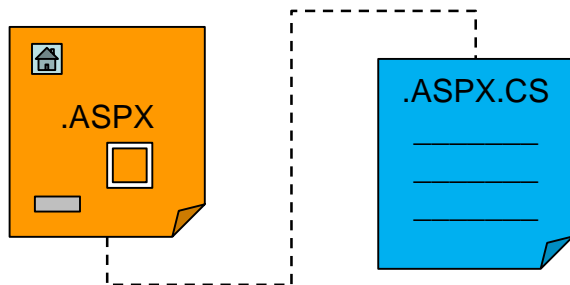
- Les **WebForms** sont le cœur et l'âme d'une application en ASP.Net
- Elles constituent l'interface avec laquelle l'utilisateur va « dialoguer »
- Les **WebForms** sont semblables aux **WindowsForms** dans le sens où elles fournissent des propriétés, des méthodes et des événements pour les contrôles qui y sont déposés (en ASP.net, on parlera de **WebControls**).

Une WebForm est composée de deux éléments :

1 - une partie “**visuelle**” (le fichier avec l'extension **.aspx**)

2 - une partie “**traitement**” (le fichier avec l'extension **.aspx.cs** ou **aspx.vb** en fonction du langage de programmation choisi)

Ce fichier contient le code qui sera exécuté côté serveur.



Le formulaire Web

- Le composant principal d'une WebForm, c'est :

```
<form runat="server">  
    ...  
</form>
```

- C'est grâce à ce formulaire qu'ASP.NET est capable de faire transiter des informations de page en page.

Le formulaire Web

Le fichier .aspx tel qu'il est, juste après sa création

Directive de page

**Fichier de code
behind associé**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="MyWebForm.aspx.cs" Inherits="MyWebForm" %>

<html >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>

    </div>
  </form>
</body>
</html>
```

Balises HTML

Formulaire

**Nom de la classe
dont hérite la page**

Le formulaire Web

Le fichier de code behind associé à la WebForm

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
```

Classe C#

Hérite de la classe Page

```
public partial class MyWebForm : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

**Le code de cette méthode
sera exécuté au
chargement de la page**

**Classe
partielle**

Les scriptlets – expressions systématiques (<%= expression%>)

Les scriptlets sont des extraits de code (C#) figurant dans une page aspx. Ils sont délimités par des marqueurs spécifiques.

Le scriptlet `<%= instructions %>` permet d'évaluer une expression lorsqu'elle apparaît dans la page, dans le contexte d'exécution. Il peut s'agir de valeurs littérales, de variables, ou d'appels de méthodes.

Exemple:

```
<span>Nous sommes le : <%=DateTime.Now.ToShortDateString() %> </span>
```


Les scriptlets – expressions systématiques (<%= expression%>)

Les scriptlets sont des extraits de code (C#) figurant dans une page aspx. Ils sont délimités par des marqueurs spécifiques.

Le scriptlet <%# instructions %> permet d'évaluer une expression lorsque la méthode de page ou de contrôle **databind ()** est appelée. L'utilisation est quasiment la même que pour les expressions systématiques.

Exemple:

```
<asp:Label runat="server" Text="<%#DateDuJour %>" ID="MonLabel" />
```

Sachant que dans le code behind, nous aurons (par exemple) :

```
protected string DateDuJour { get { return string.Format("Nous sommes le {0}", DateTime.Now.ToShortDateString()); } }
```

```
protected override void OnLoad(EventArgs e) { if (!IsPostBack) MonLabel.DataBind(); base.OnLoad(e); }
```

Les scriptlets – expressions systématiques (<%= expression%>)

Le scriptlet <%\$ instructions %> permet d'analyser une expression à la compilation de la page, évaluée à chaque requête.

Ces expressions conviennent bien aux accès aux bases de données par exemple et sont évaluées au moment du **databind()**.

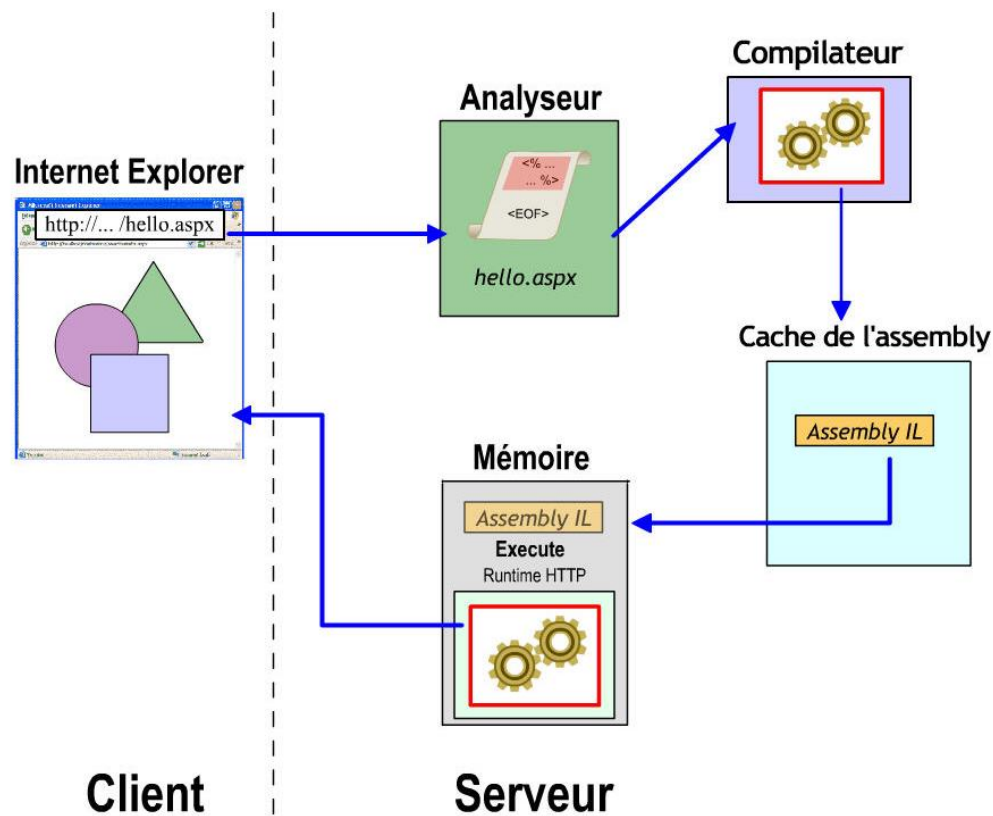
ASP.net connaît 3 types d'expressions :

ConnectionString, qui va lire dans la section **ConnectionStrings** du **web.config**
AppSettings, qui va lire dans la section **AppSettings** du **web.config**
Resources, qui lit une entrée dans un fichier de ressources

Exemple:

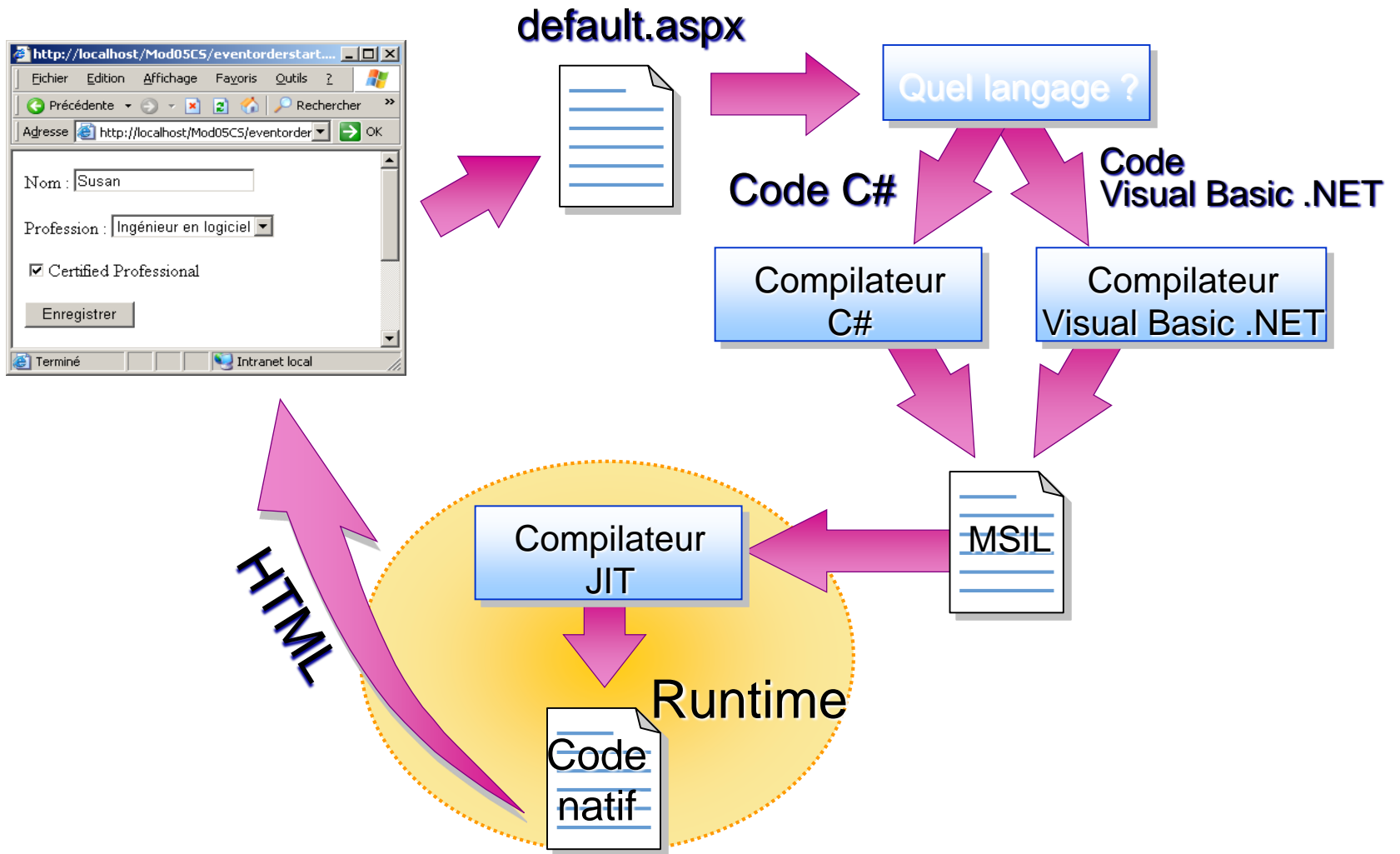
```
<asp:Label runat="server" Text="<%$AppSettings:version%" ID="MonAppSettingsExpression"/>
```

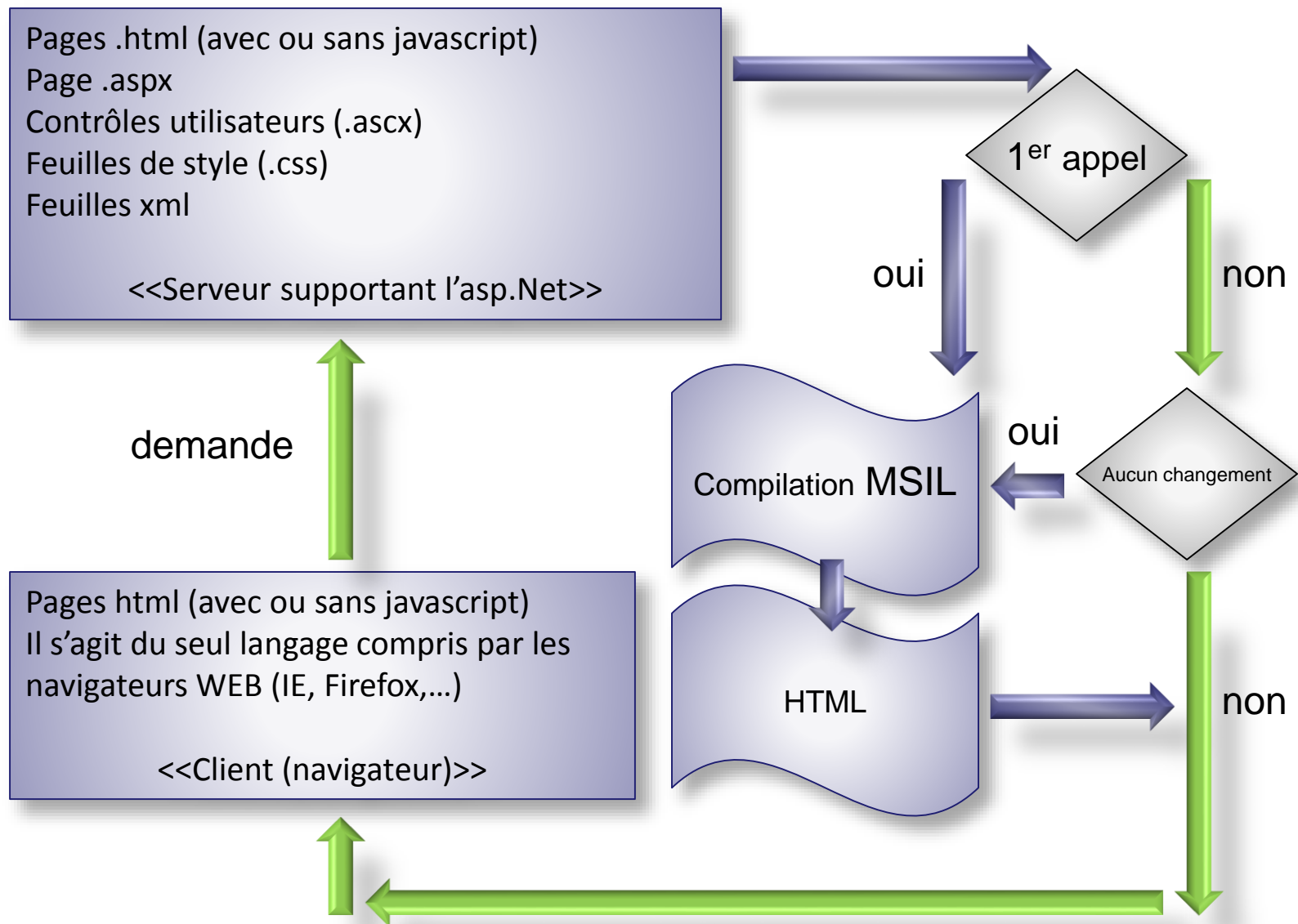
Le schéma suivant montre comment ASP.NET envoie des informations à un client qui les demande



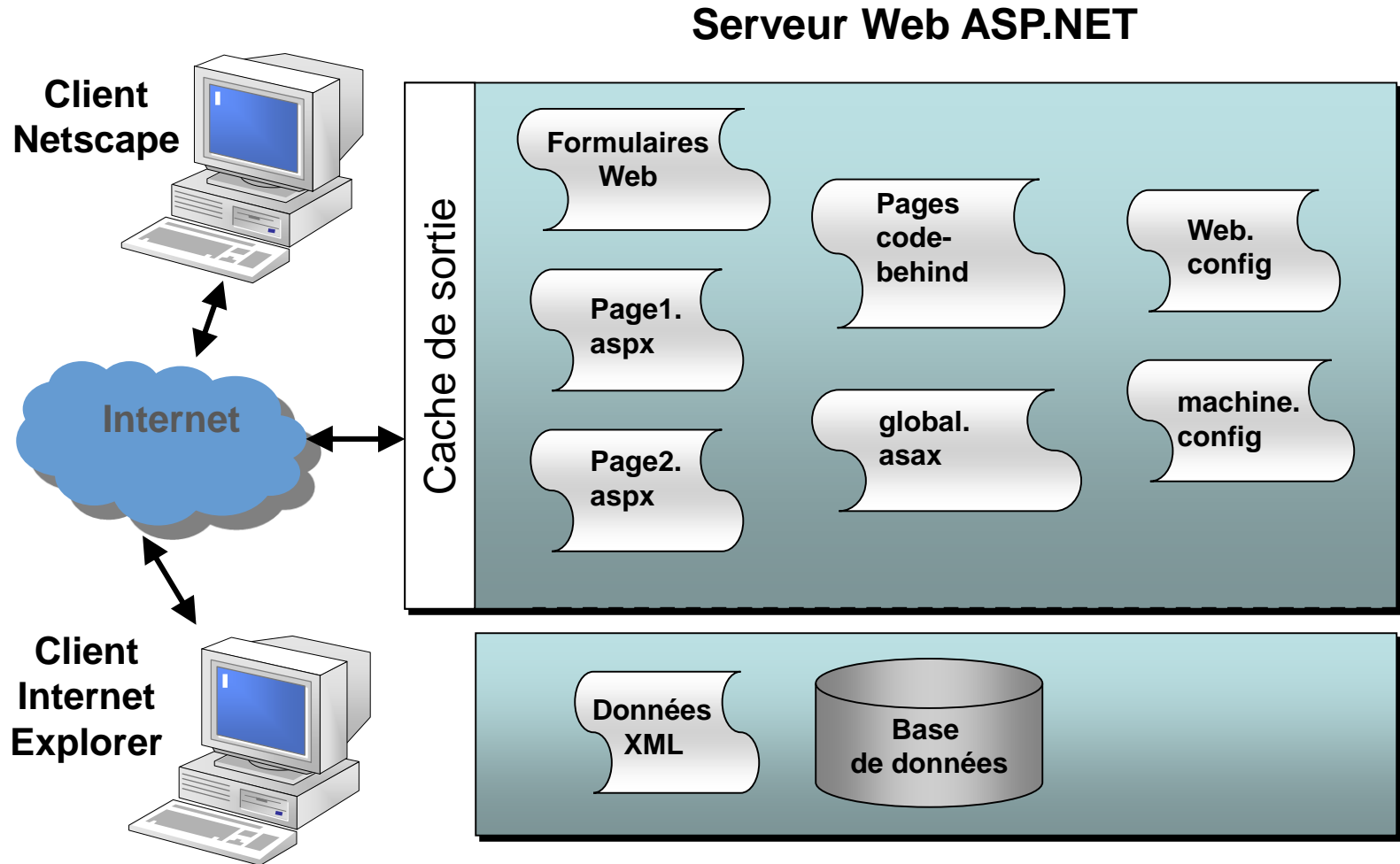
- Le navigateur client émet une demande HTTP GET au serveur
- L'analyseur ASP.NET interprète le code source
- ASP.NET appelle le compilateur
- Le runtime charge et exécute le code du langage intermédiaire

Compilation et exécution





Application Web en ASP .NET



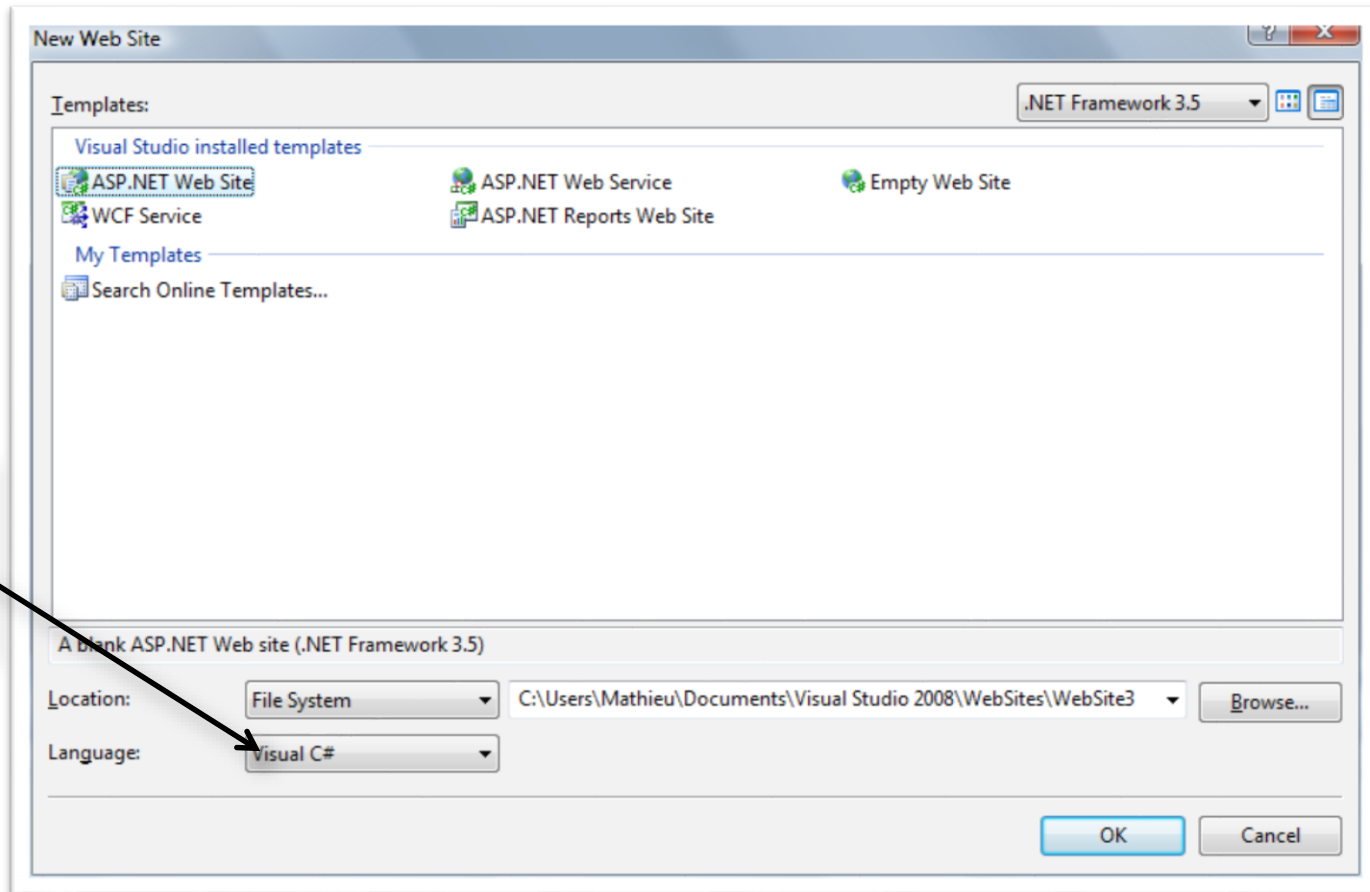
Application Web ASP.Net

- Les formulaires Web ou pages .aspx fournissent l'interface utilisateur de l'application Web.
- Les pages **code-behind** (code sous-jacent) contiennent le code côté serveur correspondant au formulaire Web auquel elles sont associées.
- Le fichier **global.asax** contient le code nécessaire pour répondre aux événements déclenchés au niveau de l'application.
- Les fichiers de configuration sont des fichiers XML qui définissent les paramètres par défaut de l'application et du serveur Web :
- Application Web → **web.config**
- Serveur Web → **machine.config**

III. Création d'un formulaire simple

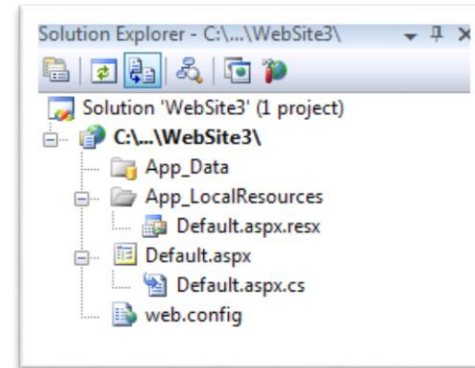
Création d'un formulaire simple

Démarrer Microsoft Visual Studio 2010 et dans le menu **File** cliquer sur **New** et ensuite **Web Site**. Vous voyez alors apparaître la fenêtre ci-dessous:



Création d'un formulaire simple

- Une première page Web (formulaire Web) est créée par défaut, il s'agit de la page **default.aspx** à laquelle est associée sa page de code behind : **default.aspx.cs**
- L'explorateur de solutions permet de visualiser l'ensemble des fichiers constituant un site Web :



Création d'un formulaire simple

Voici le code HTML par défaut d'une page aspx :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!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" >

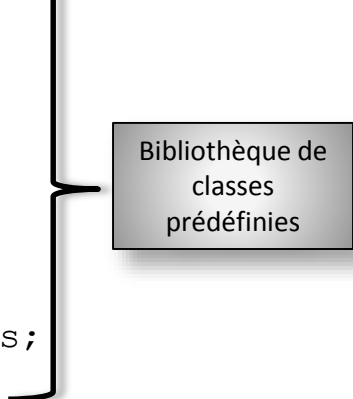
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            </div>
        </form>
    </body>
</html>
```

Création d'un formulaire simple

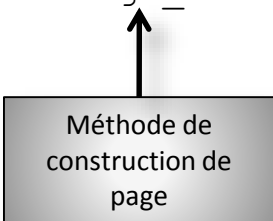
La page *code-behind* default.aspx.cs est la suivante :

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```



Bibliothèque de classes prédéfinies

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```



Méthode de construction de page

IV. Les Webs Controls

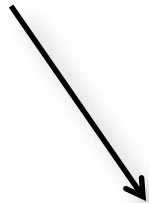
Les Web Controls

- Les **Web Controls** sont des éléments d'une page Web qui permettent une interaction avec l'utilisateur (un bouton, une liste de choix, une liste déroulante)
- Il existe beaucoup de **Web Controls** prédéfinis et vous pouvez en créer vous-mêmes ou en réutiliser d'autres créés par d'autres personnes
- Les **Web Controls** se retrouvent dans la boîte à outils située à gauche de l'IDE dans la section **Standard**

Les Web Controls

```
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>  
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
```

Remarquons la présence du *tag* asp qui a été généré lors du placement des contrôles sur la page. Nous retrouvons certaines propriétés, comme le texte de l'étiquette : `Text="Label"` et l'événement qui sera exécuté lors du clic du bouton : `OnClick="Button1_Click"`

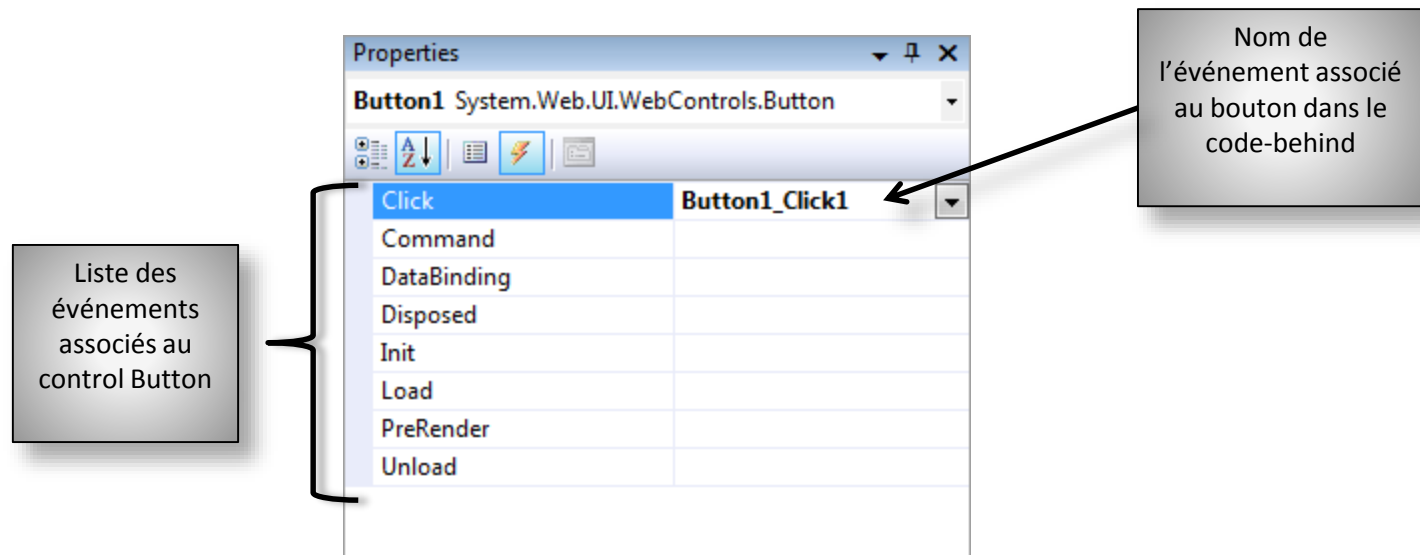


```
protected void Button1_Click(object sender, EventArgs e)  
{  
    Label1.Text = "Bonjour tout le monde.";  
}
```

La gestion des événements

Dans le début de ce chapitre, nous avons déjà évoqué le travail avec l'événement **OnClick** d'un objet de type **Button**.

L'ensemble des événements d'un objet est accessible dans la zone **Properties**, en cliquant sur le bouton **éclair**.



La gestion des événements

AutoPostBack : lorsqu'un contrôle doit fournir de l'information au serveur, sa propriété **AutoPostBack** peut être activée afin d'envoyer l'information dès qu'il y a une modification de l'objet.

```
<asp:DropDownList AutoPostBack="true" ID="DropDownList1" runat="server"
onselectedindexchanged="DropDownList1SelectionChanged">
```

PostBack : peut se définir comme un retour au serveur.
Il s'agit d'un retour d'informations générées par les contrôles.
Le serveur doit alors exécuter un traitement
et renvoyer un nouveau contenu de page Web au client.

Page_load: exécuté à chaque chargement de la page, la propriété **IsPostBack** de l'objet *Page* est essentielle pour savoir s'il s'agit de la première demande de la page Web ou non.

La gestion des événements

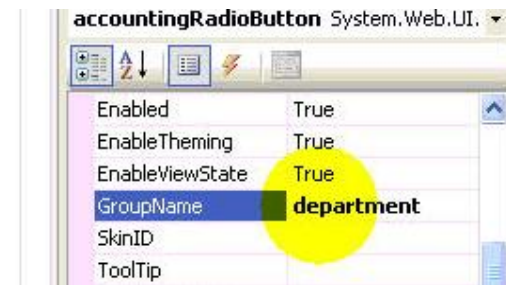
```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        // s'exécute uniquement lors du premier chargement de la page
        TextBox1.Text = "Bonjour";
    }
    // ce code s'exécute à chaque demande
}
```

Quelques contrôles essentiels

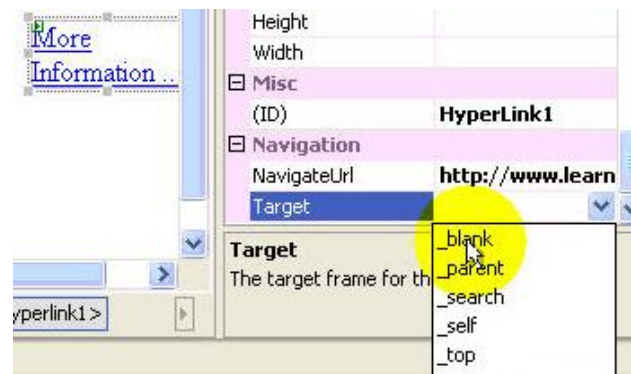
- DropDownList



- RadioButton
- CheckBox

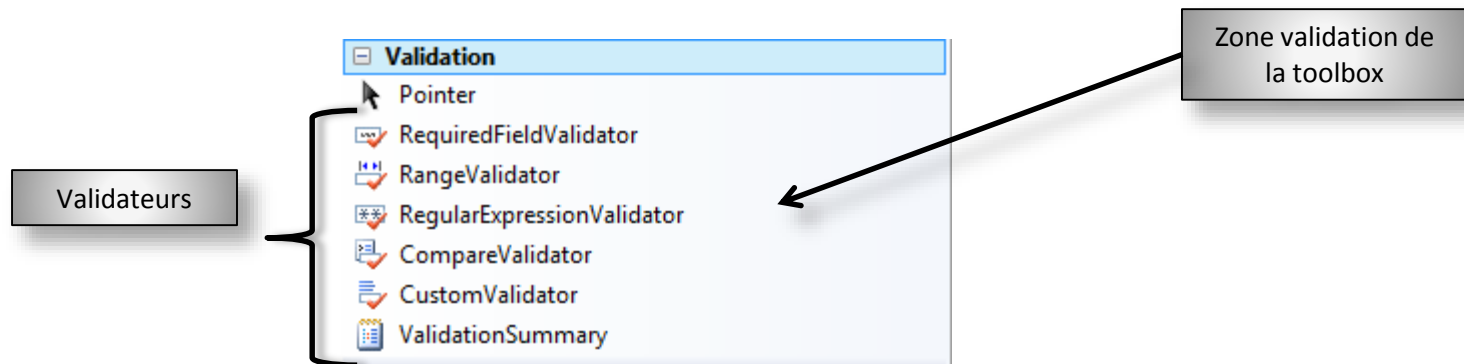


- HyperLink
- Label/Literal



La validation des données

Une série de contrôles (**Web Controls**) existent pour faciliter la vérification des encodages lors de la saisie de données par l'utilisateur dans un formulaire, les **validateurs** que l'on retrouve dans la section Validation de la boîte à outils.



La validation des données

RequiredFieldValidator

valide un champ obligatoire ;

RangeValidator

valide une plage de valeurs ;

RegularExpressionValidator

valide une expression régulière ;

CompareValidator

comparaison de valeurs (ex:mot de passe) ;

CustomValidator

validateur personnalisé (ex:nombre impair) ;

ValidationSummary

permet de générer une liste automatique de tous les messages d'erreur des différents contrôles de validation présents sur le formulaire Web.

Exercice : Réaliser un formulaire d'inscription avec validation

Login	
Libelle	Valeur
Login :	<input type="text" value="kbi"/>
Email :	<input type="text" value="k.biebuyck@wavenet.be"/>
Mot de passe :	<input type="password" value="..."/>
Mot de passe 2 :	<input type="password" value="..."/>

Sexe / Age																																																									
Sexe :	<input type="radio"/> Homme <input checked="" type="radio"/> Femme																																																								
Age :	<input type="text" value="24"/>																																																								
Date de naissance :	<table border="1"><thead><tr><th colspan="7">≤ janvier 1980 ≥</th></tr><tr><th>lun.</th><th>mar.</th><th>mer.</th><th>jeu.</th><th>ven.</th><th>sam.</th><th>dim.</th></tr></thead><tbody><tr><td><u>31</u></td><td><u>1</u></td><td><u>2</u></td><td><u>3</u></td><td><u>4</u></td><td><u>5</u></td><td><u>6</u></td></tr><tr><td><u>7</u></td><td><u>8</u></td><td><u>9</u></td><td><u>10</u></td><td><u>11</u></td><td><u>12</u></td><td><u>13</u></td></tr><tr><td><u>14</u></td><td><u>15</u></td><td><u>16</u></td><td><u>17</u></td><td><u>18</u></td><td><u>19</u></td><td><u>20</u></td></tr><tr><td><u>21</u></td><td><u>22</u></td><td><u>23</u></td><td><u>24</u></td><td><u>25</u></td><td><u>26</u></td><td><u>27</u></td></tr><tr><td><u>28</u></td><td><u>29</u></td><td><u>30</u></td><td><u>31</u></td><td><u>1</u></td><td><u>2</u></td><td><u>3</u></td></tr><tr><td><u>4</u></td><td><u>5</u></td><td><u>6</u></td><td><u>7</u></td><td><u>8</u></td><td><u>9</u></td><td><u>10</u></td></tr></tbody></table>	≤ janvier 1980 ≥							lun.	mar.	mer.	jeu.	ven.	sam.	dim.	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
≤ janvier 1980 ≥																																																									
lun.	mar.	mer.	jeu.	ven.	sam.	dim.																																																			
<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>																																																			
<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>																																																			
<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>																																																			
<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>																																																			
<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>																																																			
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>																																																			

Connaissance	
Langue :	<input checked="" type="checkbox"/> C# <input type="checkbox"/> JavaScript <input type="checkbox"/> Java
Qui êtes-vous ?	<input type="text" value="Formateur"/> ▼
Numéro de compte :	<input type="text" value="aucun"/>

Validation	
<input type="button" value="Valider"/>	
<input type="button" value="Valider sans validation"/>	
c'est validé	