

웹 폼과 응용 프로그램의 이해

* 학습목표

- 웹 폼의 동작 원리를 이해할 수 있다.
- 웹 응용 프로그램에서 Page 클래스의 속성과 이벤트를 이용할 수 있다.
 - ASP.NET 응용 프로그램의 구조를 이해할 수 있다.
 - ASP.NET 구성 파일의 특징을 이해할 수 있다.

01. 웹 폼의 동작 원리

02. Page 클래스 속성

03. Page 클래스 이벤트

04. ASP.NET 응용 프로그램

05. ASP.NET 구성 파일

요약

연습문제

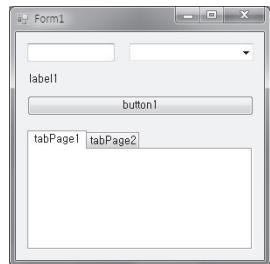
• Preview

이 장에서는 ASP.NET 개발자들이 반드시 익혀야 할 핵심 주제에 대해 다룬다. 먼저 웹 품의 빌딩 블록인 서버 컨트롤에 대해 알아보고, 정적 HTML을 동적 ASP.NET 페이지로 바꿔본다. 이 과정을 통해 ASP.NET 페이지의 동작 원리와 ASP.NET 응용 프로그램을 살펴본다. 그런 다음 웹 사이트를 구성하는 웹 페이지의 핵심 클래스인 Page 클래스를 시작으로 응용 프로그램의 구조, 애플리케이션 이벤트, ASP.NET 응용 프로그램 폴더 등을 알아보고, 마지막으로 ASP.NET의 구성 파일 모델을 살펴본다.



1 웹 폼의 동작 원리

윈도우(Windows) 응용 프로그램 개발에서 사용자 인터페이스(User Interface, UI)를 구성하는 모든 종류의 창 또는 대화 상자를 ‘윈도우 폼(Windows Form)’, 줄여서 ‘원 폼(Win Form)’이라고 한다. 레고 블록(Lego Block)으로 다양한 모양을 만들듯이 도구 상자에서 글상자, 버튼, 탭 등과 같은 빌딩 블록인 컨트롤을 드래그하여 다양한 원 폼을 구성할 수 있다. [그림 4-1]은 글상자, 콤보 상자, 버튼, 탭 컨트롤 등을 이용하여 구성한 원 폼의 예다.



[그림 4-1] 원도우 폼

ASP.NET 웹 페이지에서도 사용자 인터페이스를 구성하는 블록인 서버 컨트롤을 웹 페이지에 배치하여 웹 응용 프로그램을 작성할 수 있다. 원 폼을 이용하여 Windows 응용 프로그램을 작성하듯 웹 응용 프로그램을 작성한다는 의미에서 ASP.NET 웹 페이지를 ‘웹 폼(Web Form)’이라고 한다.

웹 폼을 이해하는 데 필요한 서버 컨트롤, GET 방식, POST 방식, 뷰, HTML 서버 컨트롤 클래스에 대해 알아보자.

1. 서버 컨트롤 소개

웹 폼은 주목할 만한 새로운 웹 개발 모델이다. 예전 방식의 웹 개발에서는 프로그래머들이 동적 웹 페이지를 디자인하기 전에 HTML의 세세한 부분까지 익혀야 했다. ASP.NET은 서버 컨트롤을 이용하여 이 문제를 해결했다. 서버 컨트롤은 서버에서 운영되는 객체로, 다른 컨트롤들과 함께 HTML 페이지를 생성한다. 더 나아가 서버 컨트롤은 자신의 상태를 유지하고 사용자가 발생시킨 이벤트에 응답하는 등 마치 윈도우에 있는 컨트롤처럼 동작한다.

ASP.NET은 웹 폼에 포함할 수 있는 두 가지 유형의 서버 컨트롤을 제공하는데, 각각의 역할은 다음과 같다.

- **HTML 서버 컨트롤** : 표준 HTML 태그와 상응하는 서버 기반 컨트롤이다. HTML로 작업하는 것이 익숙한 경우에는 HTML 서버 컨트롤을 쉽게 이용할 수 있을 것이다. 최소의 변경으로 기존의 HTML 페이지나 ASP 페이지를 ASP.NET 페이지로 바꿀 수 있다.
- **웹 컨트롤** : 윈도우 응용 프로그램을 작성할 때 사용하는 컨트롤이 웹 폼에서 사용될 경우, 이를 ‘웹 컨트롤’이라고 한다. 웹 컨트롤은 HTML 서버 컨트롤과 유사하지만 스타일이나 서식을 위한 속성들을 더욱 풍부하게 제공한다. GridView, Calendar, Validation 웹 컨트롤과 같이 HTML에는 없는 요소들도 포함하고 있다.

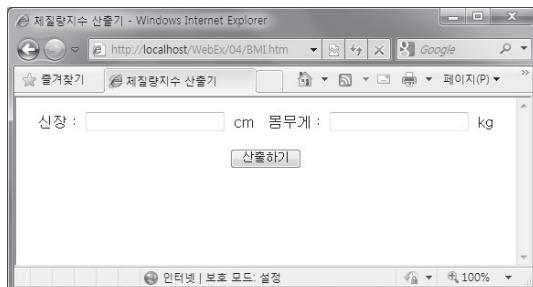
서버 컨트롤에 대한 자세한 내용은 5장에서 다룬다.

실습 4-1 HTML 페이지를 ASP.NET 페이지로 바꾸기

이번 실습에서는 디자이너가 디자인한 HTML 문서를 받았다고 가정하고, 서버 컨트롤을 이용하여 HTML 페이지를 ASP.NET 페이지로 바꿔보자. 다음에 소개하는 방법은 정적 웹 페이지를 빠르게 ASP.NET 페이지로 바꾸는 방법이기도 하다.

1 HTML 페이지 만들기

[그림 4-2]는 체질량지수(BMI, Body Mass Index)를 산출해주는 웹 페이지다. 사용자가 입력한 신장과 몸무게로 체질량지수를 구하여 보여준다.



[그림 4-2] 간단한 체질량지수 산출기

이 페이지의 HTML 코드는 다음과 같다. 메모장에서 다음 HTML 코드를 작성한 후 “C:\ASPNET_Example” 폴더 아래에 “04” 폴더를 만들어 ‘BMI.htm’으로 저장한다.

```

<html>
  <head><title>체질량지수 산출기</title></head>
  <body>
    <form>
      <div align="center">
        신장 :&nbsp;
        <input type="text" />
        &nbsp;cm&nbsp;&nbsp;&nbsp;몸무게 :&nbsp;
        <input type="text" />
        &nbsp;kg
        <br /><br />
        <input type="submit" value="산출하기" />
      </div>
    </form>
  </body>
</html>

```

알아두기

HTML에서의 모든 입력 컨트롤은 `<input>` 엘리먼트로 표현하며, 컨트롤형은 `type` 속성의 값에 의해 결정된다. `<input type="text">`는 글자를, `<input type="submit">`은 사용자 입력 정보를 웹 서버로 보내기 위한 제출 버튼을 만든다.

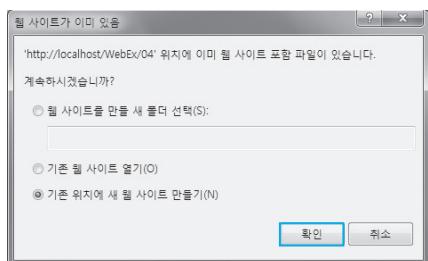
이 페이지는 3개의 `<input>` 엘리먼트(2개의 글자와 1개의 제출 버튼)를 갖고 있다. 버튼을 클릭하면 입력한 정보를 서버에 넘겨주도록 `<form>` 태그를 사용하였다. 현재 페이지는 사용자 인터페이스만을 만들어 놓은 HTML 문서이기 때문에 어떤 기능도 하지 않는다.

HTML 페이지를 ASP.NET 페이지로 바꾸는 가장 빠른 방법은 Visual Web Developer 2010을 이용하여 새로운 웹 폼을 만드는 것이다.

2 새 웹 사이트 및 새 웹 폼 만들기

Visual Web Developer 2010에서 [파일]-[새 웹 사이트]를 선택하여 “`http://localhost/WebEx/04`”에 새 웹 사이트를 만든다. 앞에서 직접 “04” 폴더를 만들고 웹 사이트를 구성하는 웹 문서인 ‘`BMI.htm`’을 만들어 놓았기 때문에 [그림 4-3]과 같이 [웹

사이트가 이미 있음] 대화상자가 열릴 것이다. ‘기존 위치에 새 웹 사이트 만들기’를 선택하고 [확인] 버튼을 클릭한다.



[그림 4-3] [웹 사이트가 이미 있음] 대화상자

Visual Web Developer 2010에서 웹 사이트를 만들면 해당 폴더는 ASP.NET 응용 프로그램 폴더로 표시된다. 응용 프로그램에 대한 내용은 4절에서 다룬다.

새 웹 폼을 만들기 위해 솔루션 탐색기의 루트 노드인 웹 사이트를 선택한 후 마우스 오른쪽 버튼을 클릭하면 나타나는 팝업 메뉴에서 [새 항목 추가]를 선택한다. [새 항목 추가] 창에서 ‘웹 폼’을 선택한 후 새 페이지의 이름을 ‘BMI.aspx’라고 입력한다. ‘다른 파일에 코드 입력’에 체크 표시가 되어 있는지를 확인한 후 [추가] 버튼을 클릭한다.

새 웹 폼의 ‘aspx’ 파일에서 Page 지시자만을 남기고 모든 내용을 지운다. 그리고 앞에서 만든 ‘BMI.htm’ 파일의 모든 내용을 복사하여 웹 폼의 Page 지시자 밑에 붙인다. 결과적으로 웹 폼의 코드는 다음과 같다.

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

<html>
  <head><title>체질량지수 산출기</title></head>
  <body>
    <form>
      <div align="center">
        신장 :&nbsp;
        <input type="text" />
        &nbsp;cm&nbsp;&nbsp;&nbsp;몸무게 :&nbsp;
        <input type="text" />
        &nbsp;kg
        <br /><br />
```

```

        <input type="submit" value="산출하기" />
    </div>
</form>
</body>
</html>

```

3 서버 컨트롤로 바꾸기

HTML 태그를 서버 컨트롤로 바꾸려면 각 태그에 runat="server"라는 속성을 넣으면 된다. 또 코드 숨김 파일(확장자 cs) 파일에서 컨트롤을 식별하기 위해 유일한 이름으로 id 속성 값을 할당한다.

체질량지수 산출기 응용 프로그램에서 입력 글상자 2개와 제출 버튼 하나를 HTML 서버 컨트롤로 변환한다. ASP.NET과 <form> 엘리먼트가 포함하고 있는 컨트롤들이 정보를 주고받으려면 서버로 보내는 정보를 포함하고 있는 <form> 엘리먼트도 서버 컨트롤로 바꿔야 한다. 다음은 완전하게 수정한 페이지다.

```

<html>
<head><title>체질량지수 산출기</title></head>
<body>
<form id="WebForm" runat="server">
    <div align="center">
        신장 :&nbsp;
        <input id="Height" type="text" runat="server" />
        &nbsp;cm&nbsp;&nbsp;&nbsp;몸무게 :&nbsp;
        <input id="Weight" type="text" runat="server" />
        &nbsp;kg
        <br /><br />
        <input id="Calc" type="submit" value="산출하기"
               runat="server" />
    </div>
</form>
</body>
</html>

```

현재 웹 폼에는 어떤 코드도 쓰지 않았기 때문에 어떤 작업도 수행하지 않는다. 그러나 정적 HTML 엘리먼트들이 HTML 서버 컨트롤로 바뀌었다.

HTTP에서 제공하는 대표적인 전송 방식과 서버 컨트롤을 살펴보고 ‘BMI.aspx’가 동적으로 기능하도록 만들어 보자.

2- GET 방식과 POST 방식

HTML에서는 사용자로부터 입력을 받기 위해 `<input>` 태그를 사용하는데, `<form>` 태그에는 하나 이상의 `<input>` 태그들을 포함할 수 있다. 웹 브라우저는 버튼 클릭과 같은 이벤트가 발생하면 `<form>` 태그에 포함된 `<input>` 태그의 내용을 `action` 속성이 가리키는 페이지로 전송한다. HTTP에서 제공하는 대표적인 전송 방식(method)에는 GET 방식과 POST 방식이 있는데, 이들의 특징은 각각 다음과 같다.

① GET 방식의 특징

- 사용자가 `<input>` 엘리먼트에 입력한 정보를 URL을 통해 서버로 전송한다. GET 방식으로 전달되는 URL에서 ‘?’ 뒤에 오는 문자열을 ‘쿼리 스트링(Query String)’이라고 한다. 쿼리 스트링은 ‘매개변수=변수 값’을 한 쌍으로 하며, ‘&’ 문자로 연결되어 있다.

```
http://localhost/WebEx/04/BMI.aspx?Height=175&Weight=70
```

- URL 길이에는 한계가 있기 때문에 많은 양의 데이터를 보내는 것은 적합하지 않다. HTTP 1.1(RFC2616 section 3.2.1)에서는 서버가 255바이트 길이의 URL을 처리할 수 있도록 규정하고 있으며, 대부분의 웹 브라우저는 2,000자 이상의 URL을 처리하지 못한다.

- 사용자 입력이 URL에 노출되어 보안상 문제가 있다.
- URL에 쿼리 정보가 포함되어 있기 때문에 새로 고침을 하거나 URL을 복사하여 다른 곳에서 보아도 동일한 문서를 볼 수 있다.

② POST 방식의 특징

- 웹 브라우저가 서버로 보내지는 페이지 요청(Request)에 사용자가 `<input>` 엘리먼트를 통해 입력한 내용을 포함시켜 전달한다.

- 사용자 입력 정보는 특별한 도구를 사용하지 않는 한 보이지 않는다.
- GET 방식과는 달리 보내는 정보의 양에 한계가 없다.
- 특별하게 정보를 유지하지 않을 경우, 새로 고침에 의해 ‘만료된 페이지’ 메시지가 나타난다.

3- 뷰 상태

Visual Web Developer 2010에서 **Ctrl+F5**를 눌러 웹 폼을 실행한다. 웹 페이지가 웹 브라우저에 나타나면 웹 브라우저에서 [보기]-[소스] 메뉴를 선택하여 ASP.NET의 보낸 HTML 소스를 살펴보자.

```
<html>
  <head><title>체질량지수 산출기</title></head>
  <body>
    <form name="WebForm" method="post" action="BMI.aspx" id="WebForm">
      <div>
        <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
        value="/wEPDwUKMjEzMDY2NzYyNmRkVZBVsaXmukP3Mr26axMJUHsFUg=" />
      </div>

      <div>
        <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
        value="/wEWBALiruvtBALzj/DXAQLUj/DXAQKhwImNC/+Q7DsJ5sPv0qp2Q+INkvQ/QX+9" />
      </div>

      <div align="center">
        신장 :&nbsp;
        <input name="Height" type="text" id="Height" />
        &nbsp;cm&nbsp;&nbsp;&nbsp;몸무게 :&nbsp;
        <input name="Weight" type="text" id="Weight" />
        &nbsp;kg
        <br /><br />
        <input name="Calc" type="submit" id="Calc" value="산출하기" />
      </div>
    </form>
  </body>
</html>
```

```

</form>
</body>
</html>

```

웹 브라우저에 보내진 HTML은 작성한 aspx 파일과 다르다. runat="server"가 벗겨졌지만 폼에 hidden형의 <input> 태그가 추가되었다.

ASP.NET에서 보내는 hidden 필드는 압축된 형태로 페이지에 있는 모든 서버 컨트롤의 상태 정보를 갖고 있는데, 이를 ‘뷰 상태(View State)’라고 한다. 웹 서버는 사용자에게 보이지 않는 hidden 필드를 통해 웹 브라우저와 지속적으로 정보를 교환하며 변화를 관리한다. 이를 통해 페이지 코드에서 컨트롤 속성을 사용할 수 있다.

이것이 웹 폼 프로그래밍 모델의 핵심 부분이다. 뷰 상태 덕분에 상태가 없는 웹의 특성을 극복하고 페이지를 끊임없이 실행되는 응용 프로그램처럼 다룰 수 있는 것이다.

HTML 페이지에서 글상자에 데이터를 넣고 [산출하기] 버튼을 누르면, 다시 열리는 페이지에서는 글상자의 값이 사라진다. 반면 ASP.NET 페이지에서 [산출하기] 버튼을 눌러 서버에 보내면 서버에서 보내는 페이지에는 이전의 값이 그대로 들어 있는 것을 확인할 수 있다. 그 이유는 ASP.NET 컨트롤이 hidden 필드를 통해 자신들의 상태를 유지하기 때문이다.

4- HTML 서버 컨트롤 클래스

ASP.NET에서 각각의 서버 컨트롤은 별도의 클래스를 갖는다. HTML 서버 컨트롤의 경우 해당 클래스는 System.Web.UI.HtmlControls 네임스페이스에 정의되어 있다. [표 4-1]은 <form> 태그에 포함되어 사용자 입력을 받는 데 사용되는 HTML 엘리먼트와 이에 대응되는 HTML 서버 컨트롤 클래스를 나타낸 것이다.

[표 4-1] HTML 서버 컨트롤 클래스

클래스	HTML 엘리먼트	클래스	HTML 엘리먼트
HtmlForm	<form>	HtmlInputCheckBox	<input type="checkbox">
HtmlInputButton	<input type="button">	HtmlInputRadioButton	<input type="radio">
HtmlInputSubmit	<input type="submit">	HtmlInputText	<input type="text">

HtmlInputReset	<code><input type="reset"></code>	HtmlInputPassword	<code><input type="password"></code>
HtmlButton	<code><button></code>	HtmlTextArea	<code><textarea></code>
HtmlInputHidden	<code><input type="hidden"></code>	HtmlInputImage	<code><input type="image"></code>
HtmlSelect	<code><select></code>	HtmlInputFile	<code><input type="file"></code>

HTML 서버 컨트롤을 추가하는 데는 직접 aspx 파일에 입력하는 방법(일반 HTML 태그에 runat="server" 속성을 넣는 것)과 Visual Web Developer 2010에서 도구 상자의 HTML 템에 컨트롤을 드래그하여 웹 폼에 넣는 방법이 있다.

[실습 4-1]에서 만든 체질량지수 산출기에는 하나의 HtmlForm과 2개의 HtmlInputText, 그리고 하나의 HtmlInputButton 클래스를 사용하였다. 컨트롤의 주요 속성들(properties)은 [표 4-2]와 같다.

[표 4-2] 주요 HTML 서버 컨트롤 클래스의 주요 속성

클래스	주요 속성
HtmlInputCheckBox와 HtmlInputRadioButton	Checked
HtmlInputText	Value
HtmlTextArea	Value
HtmlInputImage	Src, Alt, Align, Border
HtmlSelect	Items(collection)
HtmlGenericControl	InnerText와 InnerHtml(System.Web.UI.HtmlControls 네임스페이스에 관련된 클래스가 없는 다양한 HTML 태그들은 이 컨트롤 클래스를 사용)

알아두기

속성(attribute)과 속성(properties)

HTML에서는 태그의 기능을 완성하기 위해 속성(attribute)을 사용한다. C#의 클래스에서 클래스의 특성 정보는 클래스의 멤버 변수에 저장되는데, 안전한 사용을 위해 클래스의 행동 양식인 메서드를 접목시킨 속성(properties)을 통해 멤버 변수를 설정하거나 불러온다. HTML과 C# 분야에서 모두 '속성'으로 사용하고 있으므로 이 용어를 그대로 사용하기로 한다. 문맥으로 속성(attribute)과 속성(properties)을 구분할 수 있을 것이다.

실습 4-2 체질량지수 산출기 코드 쓰기

웹 폼은 이벤트를 기반으로 하여 코드의 각 부분이 특정 이벤트에 반응하여 동작한다. 체질량지수 산출기 페이지 예에서는 [산출하기] 버튼을 클릭할 때 발생하는 이벤트를 사용한다. 예에서 `HtmlInputButton`이 클릭되면 `ServerClick` 이벤트가 발생되는데, 이를 처리하여 사용자 요청에 반응하도록 코드를 만들어 보자.

1 웹 페이지에 이벤트 처리를 하기 위한 속성(attribute)과 결과 출력용 엘리먼트 추가하기

사용자가 [산출하기] 버튼을 클릭하면 페이지의 입력 내용이 웹 서버로 보내진다. ASP .NET은 보내진 정보를 통해 어떤 코드가 실행되어야 하는지를 판단한다. 이를 위해서는 이벤트와 이벤트 처리기를 연결해야 한다. 연결은 컨트롤 태그에 이벤트 관련 속성 (attribute)을 추가하면서 이루어진다.

[산출하기] 버튼의 `ServerClick` 이벤트 처리 메서드를 만들기 위해서는 이벤트 관련 속성인 `OnServerClick`에 사용하고자 하는 이벤트 처리기를 넣어주면 된다.

```
<input id="Calc" type="submit" value="산출하기"
OnServerClick="Calc_ServerClick" runat="server" />
```

ASP.NET HTML 컨트롤은 태그에 이벤트 처리기를 추가할 때, 이벤트 이름에 ‘On’이 붙여진 속성을 사용한다. 예를 들어, `ServerChange` 이벤트에 대한 처리기는 `OnServerChange`라는 속성을 이용한다.

산출한 결과를 보여줄 HTML 서버 컨트롤을 다음과 같이 추가한다. 형식이 있는 텍스트 블록을 웹 페이지에 넣을 때 사용하는 `<p>` 태그를 사용한다.

```
<p id="Result" runat="server"></p>
```



Level Up

웹 페이지 이벤트 처리

웹 페이지를 모델링한 ASP.NET 페이지 객체는 해당 컨트롤 태그가 없기 때문에 속성을 통해 이벤트와 이벤트 처리기를 연결할 수 없다. 대신 ASP.NET은 정해진 메서드 이름으로 페이지 이벤트와 페이지 이벤트 처리기를 연결한다. 예를 들어, 코드의 `Page_Load()` 메서드는 ASP.NET에 의해 모든 컨트롤이 웹 페이지에 탑재되고, 초기화될 때 발생하는 Page Load 이벤트와 자동으로 연결한다. 이를 ‘automatic event wireup’라고 한다.

aspx의 완성된 코드는 다음과 같다.

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

<html>
    <head><title>체질량지수 산출기</title></head>
    <body>
        <form id="WebForm" runat="server">
            <div align="center">
                신장 :&nbsp;
                <input id="Height" type="text" runat="server" />
                &nbsp;cm&nbsp;&nbsp;&nbsp;몸무게 :&nbsp;
                <input id="Weight" type="text" runat="server" />
                &nbsp;kg
                <br /><br />
                <input id="Calc" type="submit" value="산출하기"
                    OnServerClick="Calc_ServerClick" runat="server" />
                <br /><br />
                <p id="Result" runat="server"></p>
            </div>
        </form>
    </body>
</html>
```

2 BMI.aspx.cs 코드 쓰기

HTML 서버 컨트롤은 웹 컨트롤과 달리 Visual Web Developer 2010의 속성 창을 이용하여 이벤트 처리기를 만들 수 없으므로 직접 입력해야 한다. 다음은 [산출하기] 버튼에 대한 이벤트 처리기를 포함하고 있는 ‘BMI.aspx.cs’ 코드로, 이벤트 처리기 내에서 체질량지수를 산출하여 결과를 보여준다.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```

using System.Web.UI;
using System.Web.UI.WebControls;

public partial class BMI : System.Web.UI.Page {
    protected void Calc_ServerClick(object sender, EventArgs e) {
        decimal height = Decimal.Parse(Height.Value) / 100;
        // cm 단위를 m 단위로 환산
        decimal weight = Decimal.Parse(Weight.Value);
        decimal BMI = weight / (height * height);
        Result.InnerText = "체질량지수(BMI) : " + BMI.ToString();
    }
}

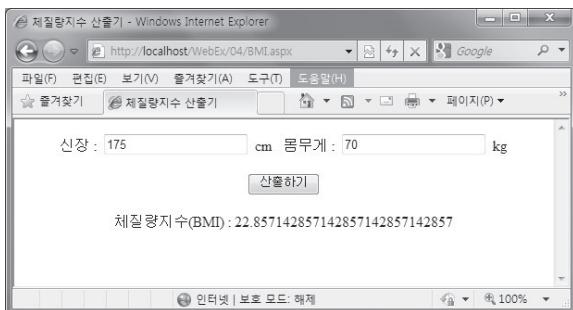
```

BMI.aspx.cs 파일은 전형적인 코드 숨김 클래스의 모습이다.

- using문으로 시작된다. 이를 통해 주요 네임스페이스에 접근할 수 있다.
- BMI 클래스가 partial 키워드로 정의되었다. 보이지는 않지만 다른 코드 파일과 합쳐져서 하나의 클래스가 된다는 것을 의미한다. ASP.NET이 자동으로 만드는 다른 코드 파일에서 id 속성을 따라 서버 컨트롤이 정의되기 때문에 코드 숨김 파일 내에서 id 속성과 같은 컨트롤의 이름을 이용하여 코딩할 수 있다.
- BMI 클래스는 하나의 이벤트 처리기를 갖고 있다. 이벤트 처리기에서 체질량지수를 계산하고 출력하는 과정은 다음과 같다.
 - 글상자의 값을 받아 숫자형으로 변환하고, 변환된 값을 이용하여 체질량지수를 계산 한다.
 - 구한 값을 문자형으로 바꾼 후 <p> HTML 컨트롤 Result의 InnerText 속성을 이용하여 결과를 출력한다.
- 이벤트 처리기는 매개변수 2개를 갖고 있는데, 이는 컨트롤 이벤트의 .NET 표준이다. 매개변수 sender를 통해 이벤트를 보낸 컨트롤을 알 수 있으며, 매개변수 e를 통해 이벤트와 관련된 다른 정보를 얻을 수 있다.

3 실행하기

[그림 4-4]는 완성된 체질량지수 산출기 웹 페이지를 실행시킨 결과이다.



[그림 4-4] 체질량지수 산출기

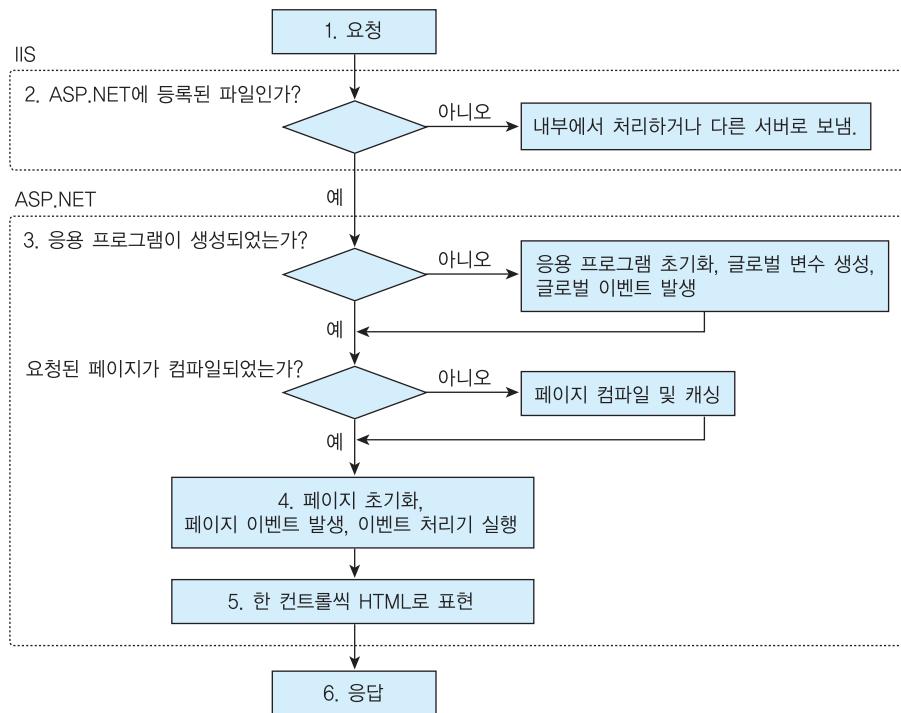
5. ASP.NET에서 웹 폼의 처리 과정

ASP.NET은 BMI.aspx 페이지에 대한 요청을 받으면 웹 서버에서는 다음 과정을 거쳐 사용자에게 응답한다.

- ① 사용자 웹 브라우저가 페이지에 대한 요청을 IIS 웹 서버에 보낸다.
- ② 웹 서버는 요청한 aspx 파일의 확장자가 ASP.NET에 등록되어 있는지를 판단한다. 만일 파일의 확장자가 다른 서비스(asp나 html)에 속해 있다면 ASP.NET은 참여하지 않는다.
- ③ 응용 프로그램에 있는 해당 페이지가 처음 요청된 것이라면 ASP.NET은 자동으로 응용 프로그램 영역을 만든다. 최적화된 실행을 위해 웹 페이지의 모든 코드를 컴파일한 후 컴파일된 파일을 캐싱한다. 이 과정이 이미 이루어졌다면 ASP.NET은 페이지의 컴파일된 버전을 재사용한다.
- ④ 컴파일된 BMI.aspx 페이지는 마치 일반 프로그램처럼 행동하며, Page Load와 같은 이벤트를 발생시킨다. 이 단계에서는 메모리에 있는 .NET 객체와 함께 동작한다.
- ⑤ 코드의 실행이 끝나면 ASP.NET은 웹 페이지에 있는 모든 컨트롤이 적절한 HTML 태그로 표현되었는지를 확인한다.
- ⑥ 최종 페이지는 사용자에게 보내지고 응용 프로그램은 종료된다.

페이지에 있는 버튼을 클릭하면 똑같은 과정이 진행된다. 추가로 4번째 단계에서 PageLoad 이벤트를 처리한 후 ServerClick 이벤트가 처리되어 체질량지수를 구하기 위해 작성한 코드가 처리된다.

[그림 4-5]는 이 과정을 순서대로 나타낸 것이다.



[그림 4-5] 코드 처리 과정



Page 클래스 속성

모든 ASP.NET 웹 페이지는 BMI 클래스와 같이 System.Web.UI.Page를 상속받는 사용자 정의 클래스이다. ASP.NET 웹 페이지는 상속을 통해 System.Web.UI.Page 클래스에서 코드 작성에 필요한 여러 속성과 메서드를 얻는다. 캐싱 기능, 유효성 검사 기능, 추적 기능 등이 이러한 속성에 포함된다. [표 4-3]은 Page 클래스의 주요 속성 중 일부를 나타낸 것이다.

[표 4-3] 기본 Page 클래스의 주요 속성

속성	설명
IsPostBack	논리형 속성으로, 사용자의 요청에 따라 처음 열리는 페이지(false)인지, 아니면 컨트롤 이벤트를 처리한 뒤 상태의 정보를 담고 있는 페이지(true)인지를 나타낸다. 웹 페이지에서 한 번만 초기화가 이루어져야 하는 경우 Page Load 이벤트 처리기에서 이 속성을 이용한다.
EnableViewState	false로 설정되면, 어떤 컨트롤의 상태 정보도 유지하지 않는다.
Application	컬렉션 객체로, 웹 사이트에 있는 사용자 사이에 공유하고 있는 정보를 갖고 있다. 예를 들어 페이지 방문 횟수를 얻기 위해 사용할 수 있다.
Session	컬렉션 객체로, 개인 사용자를 위해 서로 다른 페이지에서 사용할 수 있는 정보를 유지한다. 상거래 웹 사이트에서 사용자의 장바구니에 담겨 있는 물품들을 저장하는 데 사용할 수 있다.
Cache	컬렉션 객체로, 생성에 많은 시간을 소모하는 객체들을 저장하여 다른 페이지나 다른 사용자를 위해 재사용할 수 있다. 이 기술을 통해 웹 페이지의 성능을 향상시킬 수 있다.
Request	사용자 요청에 대한 정보를 포함하고 있는 HttpRequest 객체로, 사용자 웹 브라우저 등의 정보를 얻기 위해 HttpRequest 객체를 사용할 수 있다.
Response	ASP.NET이 사용자 웹 브라우저에 보내는 응답에 대한 정보를 포함하고 있는 HttpResponse 객체로, 쿠키를 만들거나 다른 페이지로 이동할 때 사용할 수 있다.
Server	URL 또는 페이지에 있는 HTML 마크업을 안전하게 전달하기 위해 텍스트를 인코딩하는 것과 같은 다양한 작업을 수행하는 HttpUtility 객체다.
User	사용자 인증(authentication)이 되었다면 이 속성은 사용자의 정보로 초기화된다.

실습 4-3 IsPostBack 속성 이용하기

보통의 체질량지수 산출기에서는 신장에 m 단위를 사용하지만, 미국에서는 feet 단위를 사용한다. 신장의 단위를 사용자가 선택할 수 있도록 만들어 보자. 이를 위해서는 드롭다운 리스트 상자를 사용해야 한다. HTML에서 드롭다운 리스트는 하나 이상의 <option> 태그를 포함하는 <select> 태그로 표현한다. 각각의 <option> 태그에 포함되는 텍스트는 리스트의 항목이 된다.

1 <select> 태그 추가하기

HTML의 양을 줄이기 위해 기존에 ‘cm’ 가 있던 곳에 항목이 없는 빈 <select> 태그를 넣는다. 추가하려는 <select> 태그도 서버 컨트롤이므로 태그에 id와 runat="server" 속성을 추가하여 정의한다.

```
<!-- BMI.aspx 파일 -->

<div align="center">
    신장 :&nbsp;
    <input id="Height" type="text" runat="server" />
    &nbsp;<select id="Measure" runat="server" />&nbsp;&nbsp;몸무게
    :&nbsp;
    <input id="Weight" type="text" runat="server" />
    &nbsp;kg
    <br /><br />
    <input id="Calc" type="submit" value="산출하기"
        OnServerClick="Calc_ServerClick" runat="server" />
    <br /><br />
    <p id="Result" runat="server"></p>
</div>
```

2 드롭다운 리스트에 단위 항목 추가하기

페이지가 만들어질 때 사용할 수 있는 단위를 드롭다운 리스트 항목에 추가하고자 한다. 이를 위해서는 Page 클래스에서 객체가 만들어질 때 가장 먼저 발생하는 Page Load 이벤트에 대한 처리기를 이용해야 한다. BMI 클래스 안에 Page_Load 이벤트 처리기를 다음과 같이 추가한다.