

ASP.NET の特徴

- 開発者の生産性向上
 - .NET 言語のサポート
 - デザインの簡易化
 - 多数の高機能コントロール
 - 管理機能の向上
 - 容易なインストール
 - ツールによる Web.Config の作成
 - パフォーマンスの向上
 - キャッシュ機能の提供
- ...etc

■ ASP.NET

ASP.NET は、.NET Framework を構成する要素の 1 つで、Web アプリケーションや Web サービスを開発および実行するための基盤技術です。Web サーバーにインストールされた .NET Framework によって、ASP.NET Web アプリケーションは多彩な機能を実現しています。

■ 開発者の生産性向上

● .NET 言語のサポート

.NET Framework 上で動作する .NET アプリケーションは特定のプログラミング言語に依存しません。Microsoft では、Visual Basic、C#、J#、C++ といった言語をサポートしています。また、サードパーティ製のものを含めると、20 以上の言語を利用することができます。

● デザインの簡易化

Web アプリケーションの作成における画面デザインは、Windows アプリケーション作成の場合と同じように、コントロールの配置によって簡単におこなうことができます。配置されたコントロールは、Web アプリケーション実行時に HTML タグとして出力されます。また ASP.NET では、レイアウトを定義し、一元管理するためのしくみが多く準備されています。

● 多数の高機能コントロール

ASP.NET では、Web アプリケーション作成のためのコントロールが多数提供されています。入力検証用のコントロールやナビゲーションコントロール、データベース用コントロールなどの高度なコントロールも存在します。これらのコントロールを利用することによって、アプリケーション作成の際の工数を大幅に削減することができます。

■ 管理機能の向上

● 容易なインストール

Visual Studio では、Web アプリケーション配置機能が提供されています。この機能によって Web アプリケーションをバージョンアップした際、前回の配置から変更されたファイルのみをコピーできるので、管理が容易になります。

● ツールによる Web.Config の作成

Web アプリケーションの環境設定で利用される Web.Config ファイルを、Visual Studio ではツールを使って作成することができます。これにより、Web.Config ファイルの設定ミスによるエラーを回避することができます。

■ パフォーマンスの向上

● キャッシュ機能の提供

ASP.NET は、出力キャッシュや Cache オブジェクトといった、さまざまなキャッシュ（保存）機能を提供しています。出力キャッシュはページの出力結果をキャッシュする機能です。この機能によって、同じページが呼び出された場合のレスポンスが向上します。一方、Cache オブジェクトは、アプリケーション上で利用されるさまざまなデータをキャッシュできるオブジェクトです。何度も利用するデータであれば、Cache オブジェクトに格納しておくことでアクセススピードを向上させることができます。

Visual Studio による ASP.NET アプリケーション開発

■ 作成できる Web プログラム

- ASP.NET Web サイト
 - コードを実装可能な Web ページ
 - Web ブラウザーベースのユーザーインターフェースアプリケーションの構築
- XML Web サービスと WCF サービス
 - ネットワークを介したメソッドの提供
 - 分散型 Web アプリケーションの構築

■ ASP.NET Web サイト

ASP.NET Web サイトでは、Web フォームとサーバーコントロールを利用することによって、Windows アプリケーションを作成する場合と同じように、Web アプリケーションを作成することができます。作成された Web アプリケーションは、Web ブラウザーからのリクエストに対して、Web サーバー上で処理をした後、HTML を生成して Web ブラウザーに送信します。

■ XML Web サービスと WCF サービス⁷

XML Web サービスと WCF サービスは、ネットワーク上のコンピュータでサービスを提供する機能を持つ Web プログラムです。XML Web サービスや WCF サービスでは、要求された処理を実行した後、処理結果を要求元に送信することができます。Windows アプリケーションおよび Web アプリケーション作成の際、XML Web サービスや WCF サービスを利用することで、処理の一部を別の Web サーバーでおこなわせる、分散アプリケーションとして構築することができます。

⁷WCF: .NET Framework 3.0 から追加されたクラスライブラリ。分散アプリケーション開発において、統合されたプログラミングモデルを提供する。

ASP.NET アプリケーションの配置

■ Web サイトの作成場所

● HTTP

- ☐ Web (HTTP) サーバーの仮想ディレクトリに作成
- ☐ IIS が必要

● ファイルシステム

- ☐ ローカルコンピュータ上に作成
- ☐ Visual Web Developer Web Server を利用
- ☐ IIS が不要

■ Web サイトの作成場所

ASP.NET Web アプリケーションは、IIS と .NET Framework がインストールされた Web サーバー上で動作します。ただし開発時には、必ずしも IIS は必要とされません。Visual Studio で Web アプリケーションを開発する場合、ファイルの作成・保存場所とし「HTTP」、「ファイルシステム」、「FTP」のいずれかを選ぶことができます。

● HTTP

開発環境またはネットワーク上に IIS による Web サーバーが存在する場合は、作成場所として「HTTP」を選択することができます。このとき作成場所は URL で指定し、ファイルは Web サーバー (IIS) の仮想ディレクトリに保存されます。

● ファイルシステム

「ファイルシステム」では、ファイルはローカルコンピュータまたは共有フォルダ上に保存され、作成場所はディスクのディレクトリのパスで指定します。開発実行時には、開発用仮想サーバーである IIS Express が利用されます。IIS Express は Web アプリケーションのテストやデバッグ用に提供される Web サーバーであり、既定ではリモートホストからのリクエストを受け付けません。また、アプリケーションの配置に管理者権限を必要としません。

■Webフォームの開発■

- Web フォーム
- Web フォームのデザイン
- サーバーコントロール
- Web アプリケーションのイベント処理
- コードインライン / コードビハインド

Web フォーム

- Web アプリケーション開発の基盤
 - 1 つの Web フォームが、Web サイトの1ページになる
 - ASP タグによるコントロール情報の記述
- GUI ツールを利用した目視的なデザインが可能
 - コントロールの配置による画面デザイン
- イベントドリブン プログラミング
 - Web サーバーで処理
- コードビハインドによる処理コードの分離
 - 開発作業の分担容易化

■ Web フォーム

Visual Studio で、新規に Web サイトを作成すると、既定で 1 つの Web フォーム(拡張子.aspx) が作成されます。Web アプリケーションの開発では、この Web フォームが基盤になり、1 つの Web フォームが、Web サイトの 1 ページになります。Visual Studio を利用して Web サイトを作成すると、Windows アプリケーションの開発と同様、Web フォームにコントロールを配置することによって、Web ページの UI を目視的にデザインすることができます。

Web フォームのデザイン

■ ASP タグによる、Web フォーム UI 情報の記述

■ 主なタグ

- @Page ディレクティブ <%@ Page %>
 - ページ固有の様々な属性を定義
 - ページの先頭にひとつだけ記述
- body 要素 <body>
 - ページの外観を定義
- form 要素 <form>
 - Web フォーム上のコントロールをまとめ、データの送受信を定義



■ Web フォームのデザイン

ASP.NET Web アプリケーションでは、Web フォーム（拡張子.aspx）を利用して、Web ページの UI を作成します。Visual Studio では、Web フォームを「デザインビュー」と「ソースビュー」で表示することができます。デザインビューでは、実際にブラウザに表示されるデザインを確認することができ、ソースビューではタグを確認することができます。また、Visual Studio からは、デザインとソースの両方が表示できる「並べて表示」ビューも追加されています。どちらのビューにおいても、コントロールをドラッグ&ドロップして配置することができます。コントロールを配置すると、それぞれのコントロールに対応した ASP タグが自動的に追加されます。

Web フォームは、「ASP.NET Web ページ」とも呼ばれ、ASP.NET Web ページの主要要素には、以下のものがあります。

● @Page ディレクティブ<%@Page %>

ASP.NET Web ページの先頭に 1 つだけ記述することができます。そのページに関する固有の設定をおこないます。例えば、@Page ディレクティブの Language 属性では、ページ内のプログラムコードをコンパイルするときの言語を指定します。

【VB】

```
<% @Page Language="VB" %>
```

【C#】

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

● body 要素<body>

ブラウザに表示するページの外観を定義する要素です。

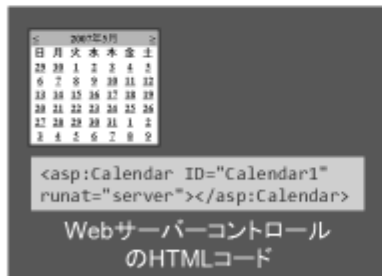
● form 要素<form>

Web フォームに、Web サーバー上で実行されるサーバーコントロールを配置する場合には、form 要素が必要となります。この際、form 要素の属性に `runat="server"` と指定します。HTML ドキュメントでは、form 要素は複数配置可能ですが、Web フォームにおいて配置できる form 要素は 1 つだけです。また、Web フォームに配置するサーバーコントロールは、form 要素の内側に配置する必要があります。(サーバーコントロールについては、後述)

サーバーコントロール

■ Web サーバー コントロール

- ASP.NET 用に準備されたタグで記述される
- Web フォーム上に UI を利用して配置可
- Web アプリケーション実行時、HTML コードを生成



■ Web サーバーコントロール

Web サーバー上で実行されるコントロールのことを「サーバーコントロール」と呼びます。また、ASP.NET によって提供される高機能なサーバーコントロールのことを「(ASP.NET) Web サーバーコントロール」と呼んでいます。Web サーバーコントロールを配置するには、ツールボックスからコントロールを Web フォームにドラッグ&ドロップするか、デザインビューで<asp:~>タグを直接記述します。Web サーバーコントロールを表す<asp:~>タグは、HTML で定義されている要素ではなく、ASP.NET オリジナルの要素です。また、<asp:~>タグは、そのままブラウザに送信されるのではなく、Web サーバーで Web サーバーコントロールが実行されると、処理結果が HTML として生成され、ブラウザに送信されます。

Web アプリケーションのイベント処理

■ Web アプリケーションでもイベントドリブンプログラミングを実現

- Web ブラウザー上でイベント発生
- Web サーバーで処理
- 初期設定で、Web ブラウザーから Web サーバーにポストされるのはクリックイベントのみ

■ イベントドリブンプログラミング

イベントとは、フォームやコントロールといったオブジェクトが認識することができるアクションを指します。個々のサーバーコントロールは、それぞれのイベントを持っています。Web ブラウザー上で Button のクリックなどのイベントが発生すると、イベント情報が Web サーバーに送られ、Web サーバー上でイベントハンドラとして記述された処理が実行されます。イベントの発生をトリガー（きっかけ）として処理を実行させるプログラミング形態のことをイベントドリブンプログラミングと呼びます。

■ Web アプリケーションのイベント処理

Web アプリケーションでは、Web ブラウザー側で発生したイベント情報が Web サーバー側に送られると、イベントに関連付けられた Web サーバー側の処理が実行されます。しかし、全てのイベントが発生するたびに、情報が Web サーバーに送信（ポスト）されると、ネットワークトラフィックが多くなり、パフォーマンスが低下します。そのため、既定でサーバーにポストされるのは、クリックイベントのみです。その他のイベントに関しては、次のポストバックが発生したタイミングでまとめて処理されます（ポストバックの詳細は後述）。