

## 第 3 章

# 日記帳を作ろう

Webアプリケーションの具体例として、日記帳を作ります。この章は、そのスタート地点で、日記を書くためのページを作ります。ページの作り方、入力したデータの扱い方の基本を学びましょう。

- 3.1 日記帳のためのWebサイトを作ろう
- 3.2 日記帳の土台を作ろう
- 3.3 日記を書くページを作ろう
- 3.4 確認画面を作ろう
- 3.5 2つのページを行き来しよう
- 3.6 書いた内容を共有しよう
- 3.7 写真をアップロードしよう
- 3.8 プログラムの動作を確認しよう



## この章で学ぶこと

この章では4つのことを学びます。

- 複数ページの共通部分をマスタページにまとめる方法
- ボタンをクリックしてページを移動する方法
- 複数ページで情報を共有する方法
- 画像ファイルをアップロードする方法

複数ページの共通部分をマスタページにまとめる方法

ボタンをクリックしてページを移動する方法

画像ファイルをアップロードする方法

複数ページで情報を共有する方法

# 3.1

## 日記帳のためのWebサイトを作ろう

日記帳のためのWebサイトを作ります。日記帳に必要なファイルなどは、すべてこのWebサイトの中に保存されることになります。

### 新しいWebサイトを作ろう

「Diary」という名前のWebサイト（プロジェクト）を作ります。このWebサイトが日記帳のすべてを管理します。Webサイトの作り方は第2章の3で説明しましたが、復習のためにもう一度やってみましょう。

**1** [ファイル] メニューの [新しいWebサイト] をクリックする。

**結果** [新しいWebサイト] ダイアログボックスが表示される。

**2** [テンプレート] ボックスで [ASP.NET Webサイト] が選択されている（背景が灰色になっている）ことを確かめる。

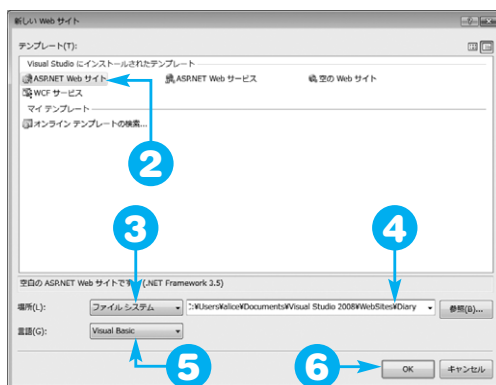
**3** [場所] ボックスで [ファイルシステム] が選択されていることを確かめる。

**4** Webサイトの場所「C:\¥Users¥<ユーザー名>\Documents¥Visual Studio 2008¥WebSites¥WebSite1」の最後の「WebSite1」をDiaryに変更する。

**5** [言語] ボックスが [Visual Basic] になっていることを確かめる。

**6** [OK] をクリックする。

**結果** Diaryという名前のWebサイトが作成される。ソリューションエクスプローラの「C:\¥...¥Diary¥」に、手順④で付けた名前が反映されている。



# 3.2

## 日記帳の土台を作ろう

日記帳は複数の画面からなりますが、タイトルのように、画面間で共通の要素がいくつかあります。これらの共通要素のみを含んだページであるマスタページを作ります。

### マスタページを作ろう

日記帳のすべてのページにタイトルやサイトマップ（後述）がありますが、これらをページ毎に作る必要はありません。マスタページという特殊なページ上に一度だけ作ればいいのです。個々のページを作る前に、マスタページを作っておきましょう。

**1** ソリューションエクスプローラの [C:\¥...¥Diary¥] を右クリックし、[新しい項目の追加] をクリックする。

**結果** [新しい項目の追加] ダイアログボックスが表示される。

**2** [テンプレート] で [マスタページ] を選択する。

**3** [名前] ボックスに [MasterPage.master] と入力されていることを確かめる。

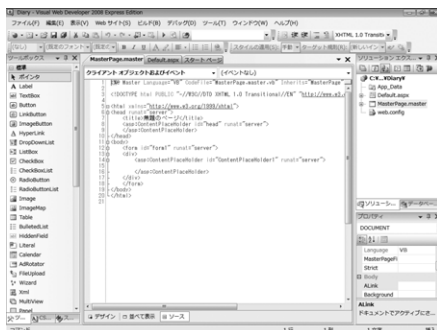
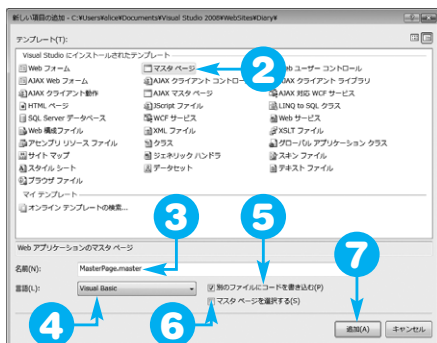
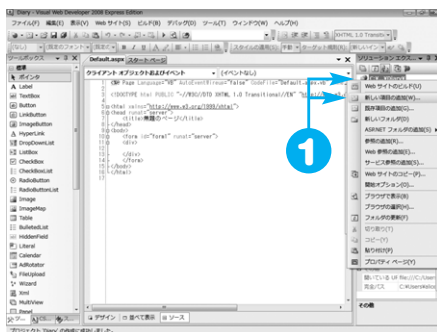
**4** [言語] ボックスで [Visual Basic] が選択されていることを確かめる。

**5** [別のファイルにコードを書き込む] チェックボックスがオンであることを確認する。

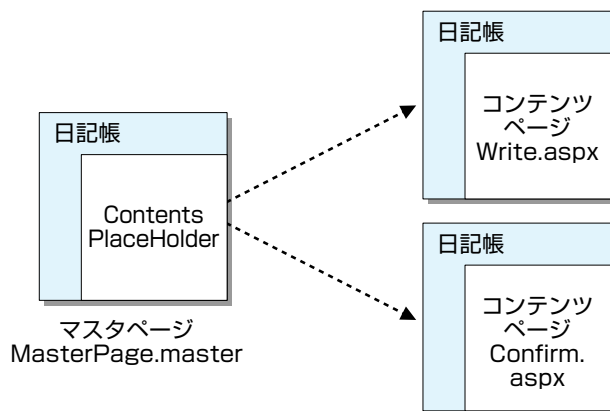
**6** [マスタページを選択する] チェックボックスがオフであることを確認する。

**7** [追加] をクリックする。

**結果** ソリューションエクスプローラに MasterPage.master ができ、ドキュメントウィンドウが MasterPage.master のソースビューになる。



## マスタページとは何か



Webアプリケーションは複数のページからできていることがふつうですが、それらのページはまったく違うものというわけではなく、タイトルやサイドバーなど、共通の要素を含んでいるものです（本書で作成する日記帳もそうです）。それらの共通要素をページごとに作るのは大変です。そこで、共通要素はマスタページ上に一度だけ作っておいて、他のページを使うときにもそれを利用するようにするのです。

マスタページを利用するページ（コンテンツページ）の内容は、マスタページのContent Placeholderの部分に埋め込まれます。

## タイトルを追加しよう

マスタページに「日記帳」というタイトルを追加します。できあがった画像を見ながら作業するとわかりやすいでしょう。

1

「デザイン」タブをクリックする。

結果

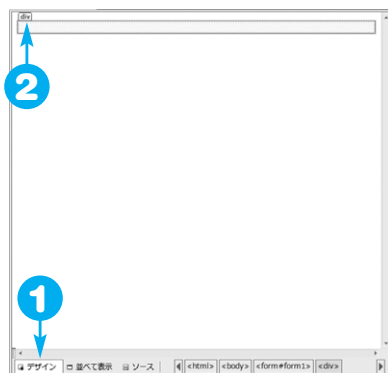
MasterPage.masterのデザインビューになる。

2

画面の枠の中をクリックし、**Enter**キーを押す。

結果

枠の左肩の表記が「div」になる。



**3** [ツールボックス] の [標準] を展開し、[Panel] をダブルクリックする。

**結果** パネルが挿入される。

**4** パネルをクリックする。

**結果** パネルの中でカーソルが点滅する。

**5** [ツールボックス] の [Label] をダブルクリックする。

**結果** パネルの中にラベルが挿入される。

**6** 左上の [(なし)] と書かれたボックスで、[見出し 1 <h1>] を選択する。

**結果** 「Label」という文字列が大きくなる。

**7** プロパティウィンドウで、Textの右をクリックし、文字列のLabelを削除し、日記帳と入力する。

**結果** ラベル上の文字列が「日記帳」になる。

**ヒント**

パネルを配置した理由

ラベルだけでタイトルを作ってもいいのですが、ウィンドウと同じ幅の帯を作るためにパネルを配置しました。第4章で見た目を整えます。

**用語**

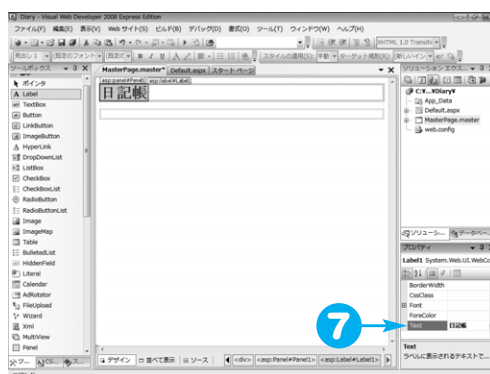
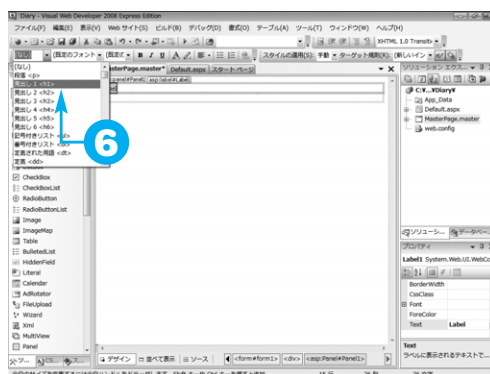
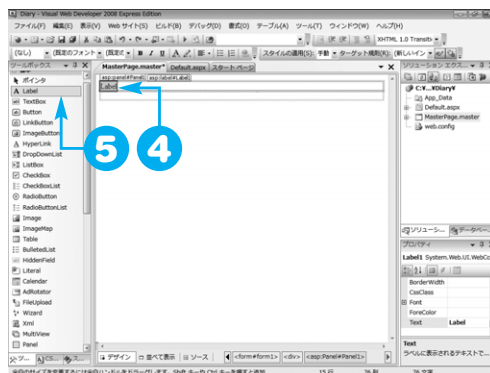
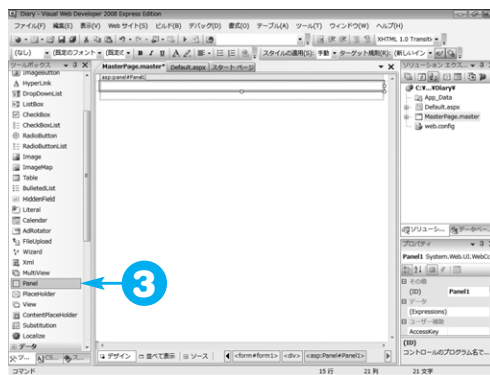
<h1>

<h1>は「最上位の見出し」であることを意味します。見出しは<h1>から<h6>まであり、数字が大きいものは下位の見出しです。

**参照**

デザインビューとソースビュー

第2章の4



## ソースビューでの確認

手順③が少しわかりにくいので、マスタページが正しくできたかどうかをソースビューで確認しましょう。「MasterPage.master」のソースビューの [<body>] から [</body>] までが、次のようになっていれば大丈夫です。間違っているときは、正しく修正してください。

```
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Panel ID="Panel1" runat="server">
        <h1>
          <asp:Label ID="Label1" runat="server" Text="日記帳"></asp:Label>
        </h1>
      </asp:Panel>
      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
    </div>
  </form>
</body>
```

## ページのタイトル

ページには2種類のタイトルがあります。1つは本節で作成したもので、HTMLでは「<h1>タイトル</h1>」と書かれるものです。h1は最上位の見出しを意味するタグで、このタグが付いた文字列は他より大きく表示されるのが一般的なので、ページのタイトルだと言っていいでしょう。

もう1つは次節で紹介するTitleプロパティの値で、HTMLでは「<title>タイトル</title>」と書かれるものです。Webブラウザのウィンドウタイトルやお気に入り（ブックマーク）に登録する際のデフォルトタイトルにはTitleプロパティが使われます。

つまり、見た目のタイトルを決定するのがh1タグで、意味的なタイトルを決定するのがTitleプロパティです。この2つは連動しません。特に、Titleプロパティを設定しないと、ページの（意味的な）タイトルが「無題のページ」になってしまうので注意してください。

# 3.3

## 日記を書くページを作ろう

日記を書くためのページを作ります。新しいページを作る方法は第2章の4で紹介しましたが、ここで作るページはマスタページを使うので、作り方が少し違います。

### マスタページを利用するページを作ろう

「admin」（管理、administrationの略）という名前の新しいフォルダを作り、その中に日記を書くページWrite.aspxを置きます。Write.aspxは前節で作ったマスタページを利用します。

**1** ソリューションエクスプローラで [C:\¥...¥Diary¥] を右クリックし、[新しいフォルダ] をクリックします。

**結果** [新しいフォルダ1] が作られる。

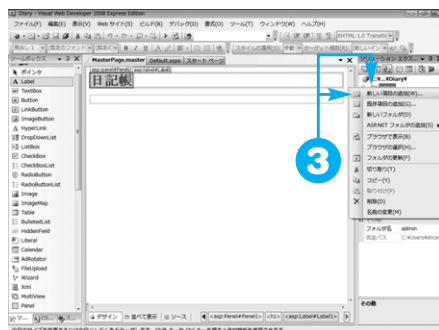
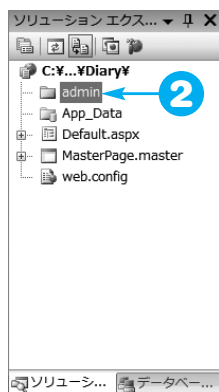
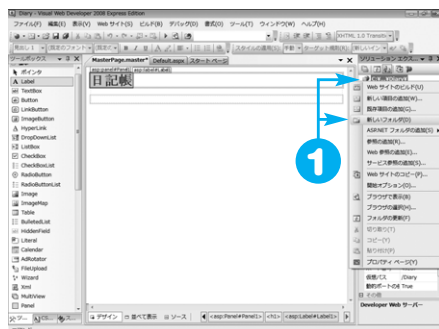
**2** フォルダ名を **admin** と入力する。

**3** [admin] フォルダを右クリックし、[新しい項目の追加] をクリックする。

**結果** [新しい項目の追加] ダイアログボックスが現れる。

**4** [テンプレート] ボックスで、[Webフォーム] を選択する。

**5** [名前] を **Write.aspx** とする。





6 [言語] が [Visual Basic] であることを確かめる。

7 [別のファイルにコードを書き込む] チェックボックスがオンになっていることを確かめる。

8 [マスタページを選択する] チェックボックスをオンにする。

9 [追加] をクリックする。

結果 [マスタページを選択してください] ダイアログボックスが表示される。

10 [MasterPage.master] をクリックして選択する。

11 [OK] をクリックする。

結果 Write.aspx のソースビューになる。

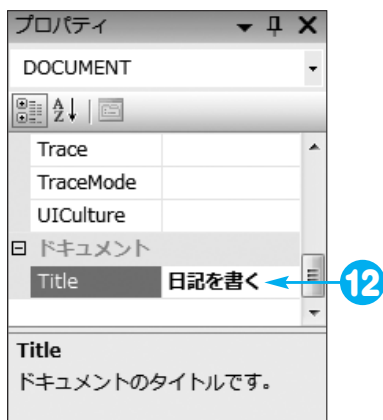
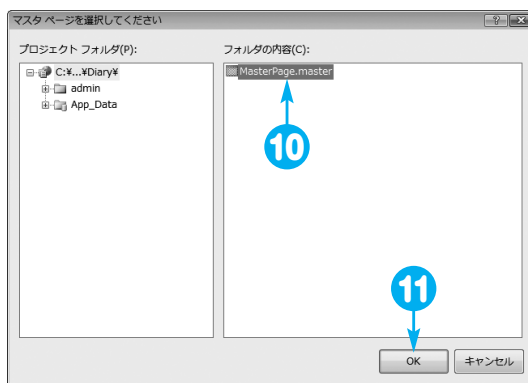
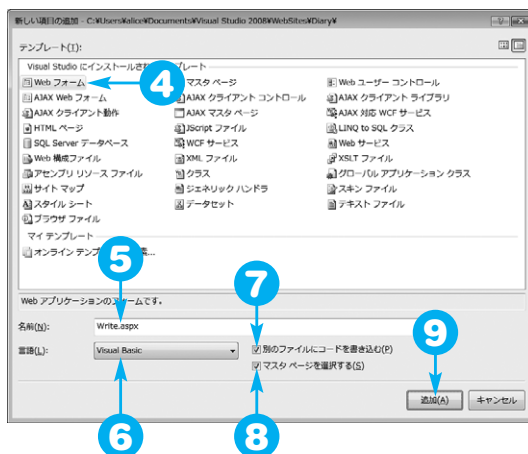
12 プロパティウィンドウで [Title] の右に **日記を書く** と入力して **Enter** キーを押す。



#### ヒント

##### [admin] フォルダを作った理由

新たに [admin] というフォルダを作ってその中に Write.aspx を置いたのは、第8章で日記を書けるユーザーを制限するための準備です。アクセス制限はフォルダ単位で行うのが簡単なのです。



## 入力のためのコントロールを配置しよう

入力のためのコントロールを配置します。まず、場所のことは考えずに、必要なものをダブルクリックで置いていきます。最後の画面を見ながらやるとわかりやすいでしょう。

**1** [デザイン] タブをクリックする。

**結果** Write.aspxのデザインビューになる。

**2** [ツールボックス] の [Label] をダブルクリックする。

**結果** ラベルが置かれる。

**3** [ツールボックス] の [TextBox] をダブルクリックする。

**結果** 入力ボックスが置かれる。

**4** [ツールボックス] の [Label] をダブルクリックする。

**結果** ラベルが置かれる。

**5** [ツールボックス] の [TextBox] をダブルクリックする。

**結果** 入力ボックスが置かれる。

**6** [ツールボックス] の [Label] をダブルクリックする。

**結果** ラベルが置かれる。

**7** [ツールボックス] の [TextBox] をダブルクリックする。

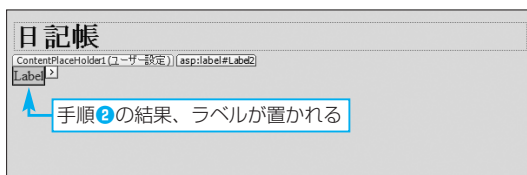
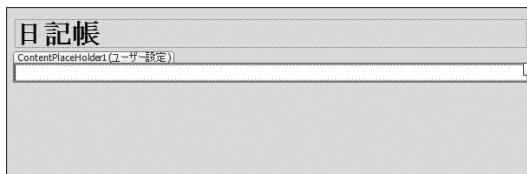
**結果** 入力ボックスが置かれる。

**8** [ツールボックス] の [Image] をダブルクリックする。

**結果** イメージコントロールが置かれる。

**9** [ツールボックス] の [Label] をダブルクリックする。

**結果** ラベルが置かれる。



#### ヒント

##### コントロールの消し方

コントロールを間違えて置いた時は、そのコントロールを選択し（青くして）、**Delete**キーを押してください。



#### 用語

##### イメージコントロール

イメージコントロールは画像を表示するためのコントロールです。表示するファイルはImageUrlプロパティで指定します。

##### FileUploadコントロール

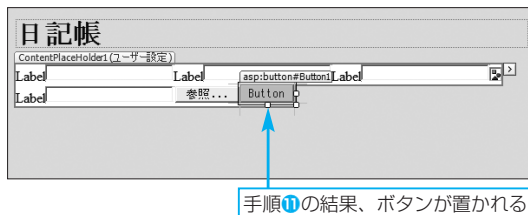
FileUploadコントロールはファイルをアップロードするためのコントロールです。アップロードされたファイルの処理方法は第3章の7で説明します。

**10** | [ツールボックス] の [FileUpload] をダブルクリックする。

**結果** ▶ ファイルアップロードコントロールが置かれる。

**11** | [ツールボックス] の [Button] をダブルクリックする。

**結果** ▶ ボタンが置かれる。



## コントロールの位置を決めよう

配置したコントロールの位置を調整します。できあがりの結果を頭に描きながらカーソルを動かして、改行や空白を挿入してください。コントロールが選択された（青くなっている）状態で`Enter`キーを押さないように注意しましょう。

**1** | デザインビュー上で矢印キーを使ってカーソルを最初のラベルの後に移動する。

**2** | `Enter`キーを押す。

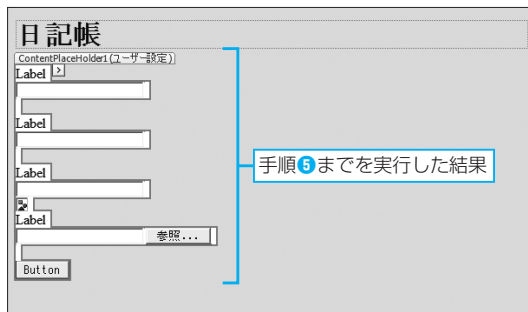
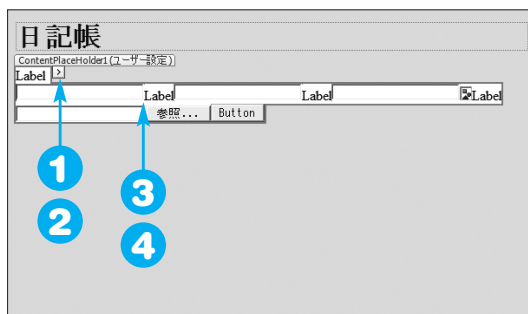
**結果** ▶ ラベルの後ろで改行される。

**3** | カーソルを入力ボックスの後に移動する。

**4** | `Enter`キーを2回押す。

**結果** ▶ 入力ボックスの下に空行が挿入される。


**5** | 手順①～④を繰り返し、コントロールの位置を調整する。



### ヒント

#### admin/Write.aspx.vbのソースビューになったら

ラベルや入力ボックスが選択されている（青になっている）状態で`Enter`キーを押すと、admin/Write.aspx.vbのソースビューになります。このファイルを

編集する必要は今はないので、 [閉じる] ボタンをクリックして、ファイルを閉じてください。改行を挿入するには、コントロールを選択していない状態で`Enter`キーを押します。

## コントロールのプロパティを設定しよう

ラベルやボタンのTextプロパティを変更して、見た目をわかりやすくしましょう。入力ボックスは3つもあるので、区別しやすいように、IDをわかりやすいものに変えておきます。

- 1 | ドキュメントウィンドウで日付用の入力ボックスをクリックして選択する。

**結果** 入力ボックスが選択され、青くなる。

- 2 | プロパティウィンドウで [(ID)] の右の値を **TextBoxDate** に変えて、**Enter** キーを押す。

- 3 | タイトル用の入力ボックスをクリックして選択する。

**結果** 入力ボックスが選択され、青くなる。

- 4 | プロパティウィンドウで [(ID)] の右の値を **TextBoxTitle** に変えて、**Enter** キーを押す。

- 5 | プロパティウィンドウで [Width] の右の値を **300px** にして **Enter** キーを押す。

**結果** タイトル用の入力ボックスの幅が広がる。

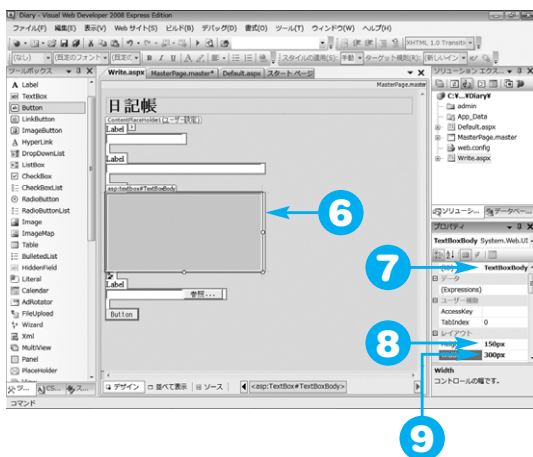
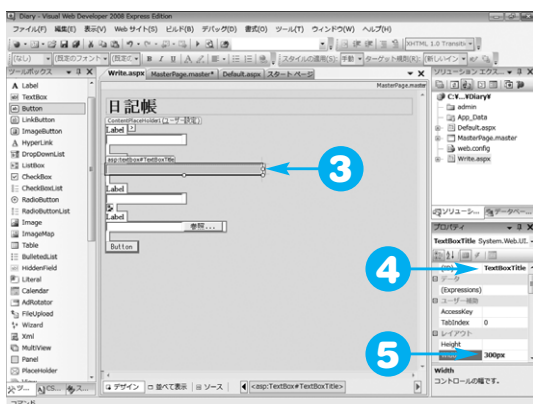
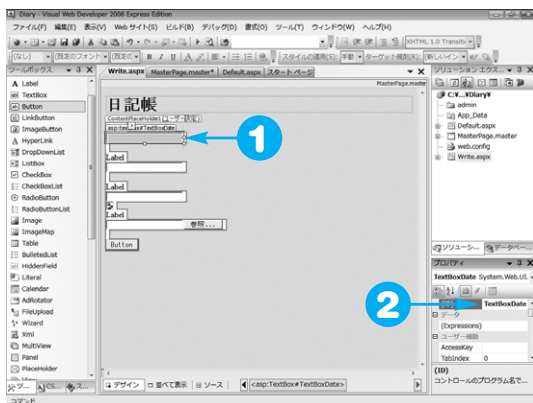
- 6 | 本文用の入力ボックスをクリックして選択する。

**結果** 入力ボックスが選択され、青くなる。

- 7 | プロパティウィンドウで [(ID)] の右の値を **TextBoxBody** に変えて、**Enter** キーを押す。


- 8 | プロパティウィンドウで [Height] の右の値を **150px** にして **Enter** キーを押す。

**結果** タイトル用の入力ボックスが縦に広がる。



**9** プロパティウィンドウで [Width] の右の値を**300px**にして**[Enter]**キーを押す。

**結果** タイトル用の入力ボックスが横に広がる。

**10** プロパティウィンドウで [TextMode] の右の  アイコンをクリックし、プルダウンメニューの [MultiLine] をクリックする。

**結果** 入力ボックスに縦スクロールバーが付く。

**11** 日付用のラベルをクリックして選択する。

**結果** ラベルが選択され、青くなる。


**12** プロパティウィンドウで [Text] プロパティを**日付**にして、**[Enter]**キーを押す。

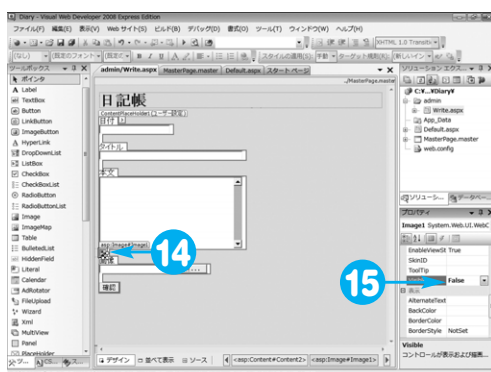
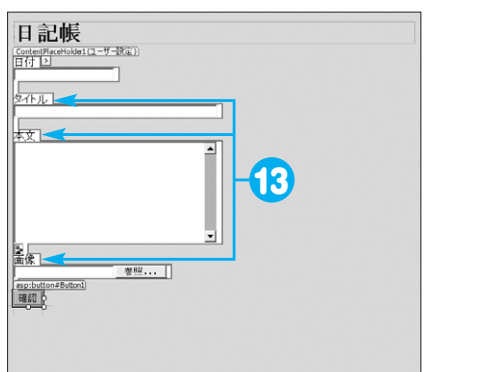
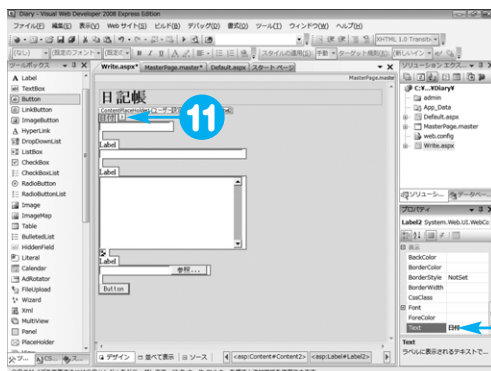
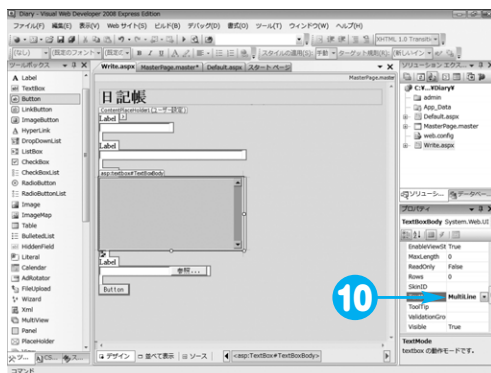
**結果** ドキュメントウィンドウのラベルの表記が「日付」になる。

**13** 同様に、タイトルや本文、画像用のラベルのTextプロパティを設定する。

**14** イメージコントロールをクリックして選択する。

**結果** イメージコントロールが選択され、青くなる。

**15** プロパティウィンドウで Visible の右の  アイコンをクリックし、プルダウンメニューの [False] をクリックする。



## 用語

### IDプロパティ

IDは特別なプロパティです。プログラムからコントロールを操作するときに、IDによってコントロールを特定するからです。IDは必ず変えなければならないというわけではありません。しかし、TextBox2、TextBox3、TextBox4よりもTextBoxDate、TextBoxTitle、TextBoxBodyというIDのほうがわかりやすいでしょう。本書では、種類が同じコントロールがページ内に複数あって、それらをプログラムから操作する場合には、IDをわかりやすいものに変えることにしています。

# 3.4

## 確認画面を作ろう

書いた日記の内容を確認できるようにしましょう。第2章の5で作ったアプリケーションのように書いた内容をその場で表示するのではなく、別のページで表示させます。まずはそのためのページを作ります。

### 新しいページConfirm.aspxを作ろう

前節のWrite.aspxのときと同様に、[admin] フォルダの中にConfirm.aspxを作ります。

**1** ソリューションエクスプローラで [admin] フォルダを右クリックし、[新しい項目の追加] をクリックする。

**結果** [新しい項目の追加] ダイアログボックスが表示される。

**2** [テンプレート] で [Webフォーム] が選択されていることを確かめる。

**3** [名前] ボックスの内容をConfirm.aspx に変える。

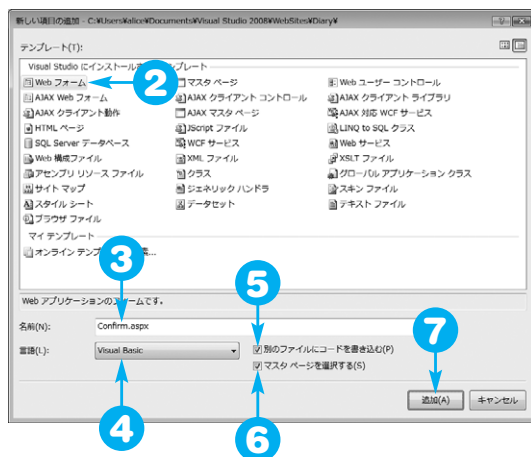
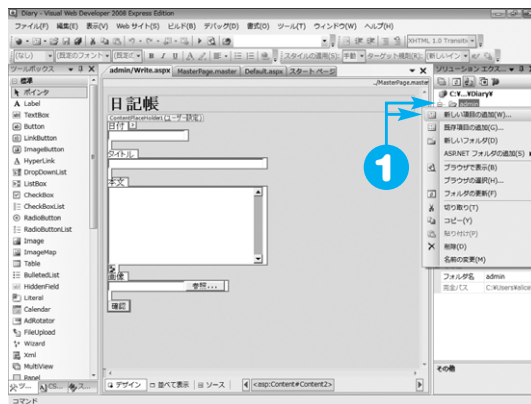
**4** [言語] ボックスで「Visual Basic」が選択されていることを確かめる。

**5** [別のファイルにコードを書き込む] チェックボックスがオンになっていることを確かめる。

**6** [マスタページを選択する] チェックボックスがオンになっていることを確かめる。

**7** [追加] をクリックする。

**結果** [マスタページを選択してください] ダイアログボックスが表示される。

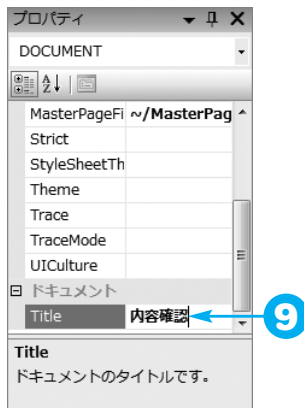




8 [フォルダの内容] ボックスで [Master Page.master] を選択し、[OK] をクリックする。

**結果** Confirm.aspxが作られ、そのコードビューになる。

9 プロパティウィンドウでTitleプロパティの値を内容確認とし、[Enter]キーを押す。



## 用語

### TextModeプロパティ

「コントロールプロパティを設定しよう」の手順10で設定したTextModeプロパティはSingleLine、MultiLine、Passwordのいずれかの値をとるプロパティです。SingleLineにすると、1行の文字列を入力できるものになります。MultiLineにすると、複数行の文字列を入力できるようになります。Passwordにすると、入力した文字列を画面上では読めなくなります（パスワードのための入力ボックスで利用します）。

### Visibleプロパティ

「コントロールプロパティを設定しよう」の手順15で設定したVisibleプロパティでは、そのコントロールを実行時に画面に表示するかどうかを指定します。Trueにすると表示され、Falseにすると表示されません。画像コントロールのVisibleプロパティをFalseにしているのは、表示する画像がまだ決まっていないからです。表示する画像がない状態で画像コントロールを表示させると、画像の枠だけが表示されてしまいます。

## 確認ページにコントロールを配置しよう

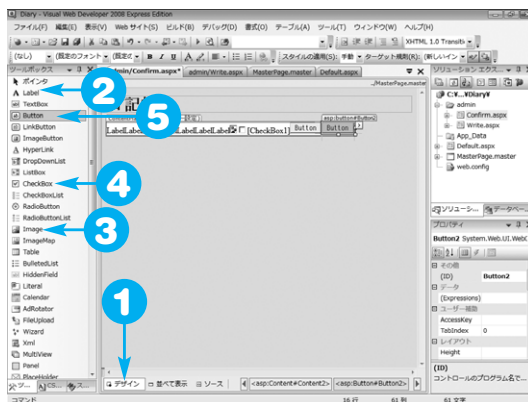
書いた内容を確認するページConfirm.aspxにコントロールを配置します。コントロールの配置方法はWrite.aspxと同様です。完成した画面を参考にしながら作業すると、わかりやすいでしょう。

1 [デザイン] タブをクリックする。

**結果** Confirm.aspxのデザインビューになる。

2 [ツールボックス] の [Label] を7回ダブルクリックする。

**結果** ラベルが7個置かれる。



### 3 [ツールボックス] の [Image] をダブルクリックする。

**結果** イメージコントロールが置かれる。

### 4 [ツールボックス] の [CheckBox] をダブルクリックする。

**結果** チェックボックスコントロールが置かれる。

### 5 [ツールボックス] の [Button] を2回ダブルクリックする。

**結果** ボタンが2個置かれる。

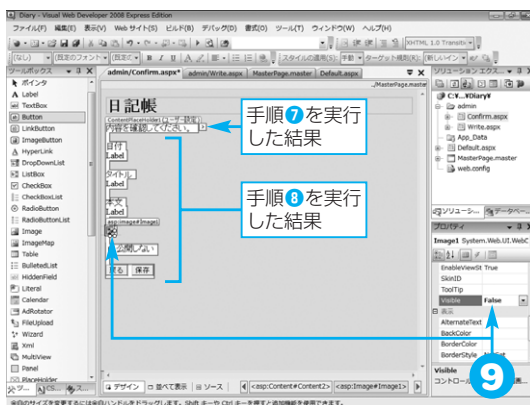
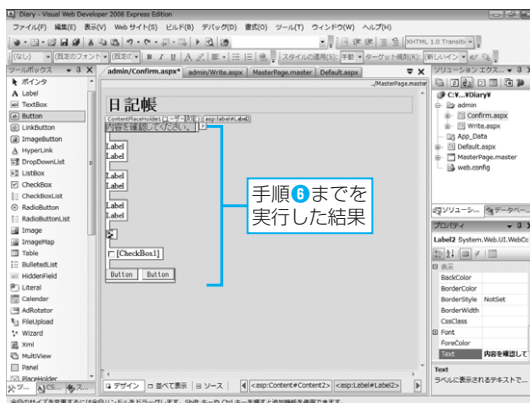
### 6 各コントロールの間にカーソルを移動し、**Enter**キーを押して適当に改行を挿入する。

### 7 一番上の [Label] をクリックして選択し、プロパティウィンドウでTextプロパティを**内容を確認してください。**に変えて**Enter**キーを押す。

**結果** ラベルの表記が「内容を確認してください。」になる。

### 8 手順⑥と同様に各コントロールのTextプロパティを変える。上から2番目のラベルは**日付**、上から4番目のラベルは**タイトル**に、上から6番目のラベルは本文に、チェックボックスは**公開しない**に、左のボタンは**戻る**に、右のボタンは**保存**にする。

### 9 イメージコントロールをクリックして選択し、プロパティウィンドウで [Visible] プロパティの右の ▼ ボタンをクリックし、プルダウンメニューの [False] をクリックする。



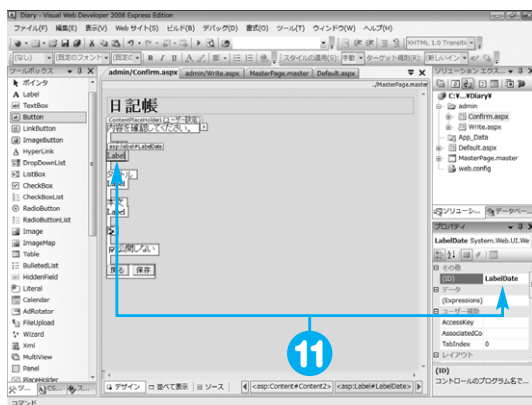
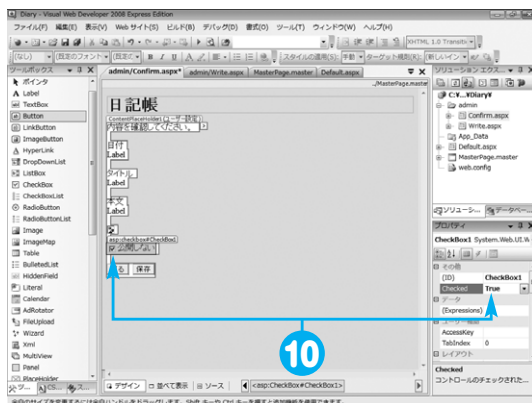


**10** [公開しない] チェックボックスをクリックして選択し、プロパティウィンドウで [Checked] プロパティの右の ▼ ボタンをクリックし、プルダウンメニューの [True] をクリックする。

**結果** チェックボックスがオンになる。

**11** [日付] の下のラベルをクリックして選択し、プロパティウィンドウで (ID) を **LabelDate** 変えて **Enter** キーを押す。

**12** 手順**11**と同様に、[タイトル] の下のラベルのIDを **LabelTitle** に、[本文] の下のラベルのIDを **LabelBody** に変える。



## 注意

### コントロールの上で **Enter** キーを押すと

コントロールが選択された（青になった）状態で **Enter** キーを押すと、admin/Confirm.aspx.vbが開いてしまいます。このファイルを編集する必要は今のないので、**X** [閉じる] ボタンをクリックしてファイルを閉じてください。コントロールを選択していない状態で **Enter** キーを押せば改行を挿入できます。

## ヒント

### チェックボックスコントロール

その名の通りチェックボックスを生成するコントロールです。生成されるチェックボックスのオン・オフは、Checkedプロパティで決まります。値がTrueならオン、Falseならオフです。ここで指定した値がデフォルトになります。つまり、CheckedプロパティをTrueにしておくと、チェックボックスがオンになった状態でページが生成されます。ユーザーがチェックボックスを操作すると、その結果はCheckedプロパティに反映されます。

# 3.5

## 2つのページを行き来しよう

日記を書くページであるWrite.aspxと書いた内容を確認するページであるConfirm.aspxの間を行き来できるようにしましょう。ボタンをクリックするとページを移動するようにします。

### ボタンをクリックしたときの動作をプログラミングしよう

Write.aspxの[確認] ボタンをクリックするとConfirm.aspxに、Confirm.aspxの[戻る] ボタンをクリックするとWrite.aspxに移動するようにします。

- 1 ソリューションエクスプローラで[Write.aspx] をダブルクリックする。

結果 Write.aspxが開く。

- 2 [デザイン] タブをクリックする。

結果 Write.aspxのデザインビューになる。

- 3 [確認] ボタンをダブルクリックする。

結果 Write.aspx.vbが開いて、Button1\_Clickのコードが自動的に作られる。



- 4 以下のコードを書く（色文字部分のみ）。

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Response.Redirect("Confirm.aspx")
End Sub
```

- 5 ソリューションエクスプローラで[Confirm.aspx] をダブルクリックする。

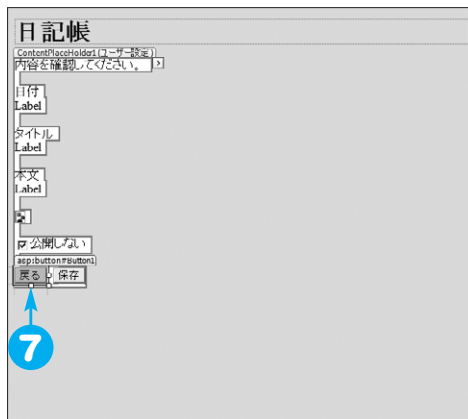
結果 Confirm.aspxが開く。

- 6 [デザイン] タブをクリックする。

結果 Confirm.aspxのデザインビューになる。

## 7 「戻る」ボタンをダブルクリックする。

**結果** Confirm.aspx.vbが開いて、Button1\_Clickのコードが自動的に作られる。



## 8 以下のコードを書く（色文字部分のみ）。

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Response.Redirect("Write.aspx")
End Sub
```

### 参照

デバッグが無効です

第2章の4の手順 5

### 注意

#### Response.Redirect

「Response」は「応答」、「Redirect」は「転送する」ということです。「Response.Redirect」は「このページの応答を転送する」ということになります。

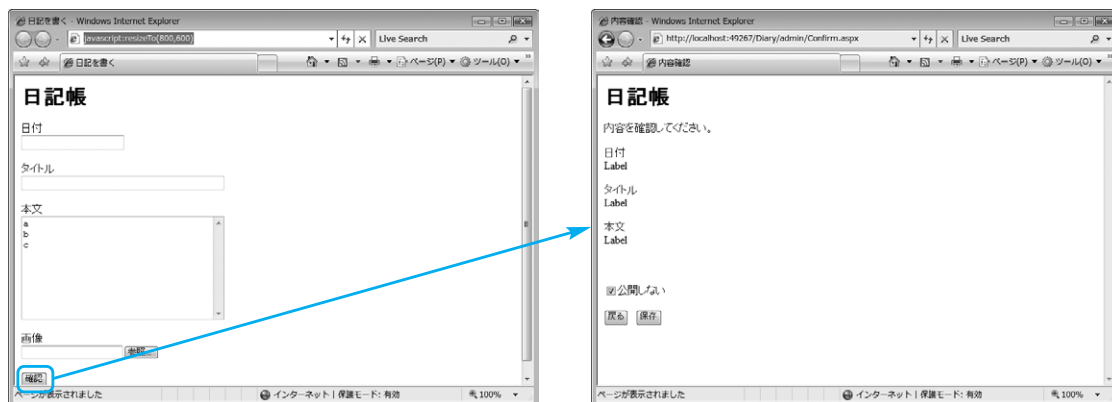
## イベントとイベントハンドラ

ボタンをクリックすると「Click」というイベントが発生します。イベントが発生すると、対応するイベントハンドラがそれを処理（handle）します。手順④や手順⑥で書いたコードの1行目の最後に「Handles Button1.Click」などを書いてあることから、これらがボタンのClickのためのイベントハンドラだということがわかります。ボタンから発生するイベントはClickの他にもいくつかありますが、Clickがボタンの代表的なイベントなので、手順③や⑥でボタンをダブルクリックしたときに、Clickイベント用のイベントハンドラが自動的に作られたのです（代表的でないイベントのイベントハンドラの書き方は次節手順④で紹介します）。ボタンのクリックに限らず、ASP.NETのアプリケーションは「イベントの発生とその処理」の繰り返しで動いています。このようなプログラムは「イベント駆動型」と呼ばれます。

## 実行して確かめてみよう

▶ [デバッグ開始] アイコンをクリックしてWebアプリを実行してみましょう。Write.aspxの[確認] ボタンを押すとConfirm.aspxに移動すること、Confirm.aspxの[戻る] ボタンを押すとWrite.aspxに移動することを確かめてください。

Write.aspx上で何か入力してからボタンをクリックしてみましょう。[確認] ボタンをクリックしてConfirm.aspxに移動すると、書いた内容は反映されていません。[戻る] ボタンをクリックしてWrite.aspxに戻ると、先に書いた内容は消えています。これでは役に立たないので、次節では書いた内容をページ間で共有できるようにしましょう。




日記を書いて [確認] ボタンをクリックすると、Confirm.aspxに移動するが、書いた日記は反映されない。

## 移動先のページの指定方法

Write.aspxからの移動先のページ表記を「Confirm.aspx」ではなく、「/Diary/admin/Confirm.aspx」と書いても正しく動作します。しかしながら、この書き方はお勧めできません。開発の途中でWebサイトの名前 (Diary) やフォルダ名 (admin) が変わったとき、「Confirm.aspx」ならプログラムを修正する必要はありませんが、「Diary/admin/Confirm.aspx」は修正しなければならないからです。このようにページは、現在のフォルダ、つまりWrite.aspxのあるフォルダ (admin) を基準に指定します。このような指定方法を「相対パス」といいます。Webアプリなので、相対パスは「相対URL」になります。これに対して、「/」からのパスを全部指定する書き方は「絶対パス」です。

さらに言えば、移動先のページを「http://localhost:49267/Diary/admin/Confirm.aspx」のように書くこともできます。この書き方は絶対にしないでください（ちなみにこのような書き方を完全修飾URLといいます）。開発サーバーのポート番号（この例では49267）はWebアプリを起動するたびに変わる可能性がありますし、サーバー名（localhost）も運用時には変わるからです（localhostは今使っているコンピュータのことです。「127.0.0.1」と書くこともできます）。完全修飾URLを使っていると、頻繁にプログラムを修正しなければならなくなります。

## Webアプリケーションのスタートアクション

Write.aspxとConfirm.aspxというように複数のページがあるとき、 [デバッグ開始] アイコンをクリックしたときに表示されるのは、ドキュメントウィンドウに開いているページです。いつも決まったページから始めたいという場合には、そのように設定することもできます。

[Webサイト] メニューの [開始オプション] をクリックします。[開始動作] が [現在のページを使用する] になっているので、Webアプリケーションの開始時には、その時開いていたページが実行されます。ここを [ページを指定する] を選択し、「admin/Write.aspx」を指定すれば、「Write.aspx」が実行されるようになります。今はこの設定を変えないでください。これ以降も、[現在のページを使用する] になっているとして話を進めます。

