

Konfektionierung

Eine Homepage hat üblicherweise ein einheitliches Aussehen. Das Programm `mail.aspx` ist aber ganz roh und ohne Layout-Elemente und enthält auch keinen sonstigen Inhalt wie er üblicherweise auf einer Webseite verwendet wird.

Man entwirft also zunächst die Seite, die später das Kontaktformular enthalten soll, zum Beispiel `kontakt.html` mit dem eigenen HTML-Designer, zum Beispiel Expression Web und lässt den Bereich des Formulars zunächst frei.

Integration als iFrame

Die einfachste Integration erfolgt über ein `iframe`. Dazu genügt es, im Dokument `kontakt.html` an der gewünschten Stelle einzufügen:

Damit die Formatierung der Formular-Felder korrekt erfolgt, ist es auch zusätzlich nötig, in der Datei `mail.aspx` im head-Abschnitt auf den Style-Sheet der Seite `kontakt.html` zu verweisen, also zum Beispiel

```
<link rel="stylesheet" type="text/css" href="mystyle.css" media="screen">
```

In bestehende Webseite integrieren

Wer kein iFrame verwenden will, kann auch die Datei `kontakt.html` editieren, und zwar so:

Man benennt die Seite in `kontakt.aspx` um und fügt folgende Kopfzeilen ein

```
<@ Page Language="C#" Debug="true" %>
<@ Import Namespace="System.Net.Mail" %>
<!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 id="Head1" runat="server">
```

Innerhalb des Header-Tag fügt man den folgenden Abschnitt aus `mail.aspx` ein.

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

Unmittelbar nach dem `body`-Tag fügt man ein

```
<form id="form1" runat="server">
```

Und vor dem `/body`-Tag

```
</form>
```

Den Rest zwischen

```
<asp:Panel ID="Panel_Message" runat="server">
<asp:Label ID="Label_Message" runat="server"></asp:Label>
</asp:Panel>
<asp:Panel ID="Panel_MailDialog" runat="server">
</asp:Panel>
```

fügt man an der für das Formular vorgesehenen Stelle ein.

Dann muss man noch den Link/die Links, die auf `kontakt.html` gezeigt haben, auf `kontakt.aspx` ändern.

Zum Programm

Eine ASPX-Datei enthält im HTML-Teil neben den üblichen HTML-Tags (in `mail.aspx` ist es eine Tabelle, siehe Code rechts oben) auch so genannte Webserver-Steuerelemente, die sehr ähnlich wie Html-Elemente aufgebaut sind aber Objekte mit vielen Eigenschaften und Methoden sind, über eine `Id` und über das spezielle Tag `runat="server"` durch ein Programm im Kopfteil angesprochen werden können (hier `TextBox`, `Label` und `Button`).

Das Programm im Kopfteil (vorige Seite) ist kein Skript sondern ein echtes C#-Programm. Dieses Programm wird beim ersten Aufruf am Server kompiliert und für den Anwender unsichtbar im Betriebssystem abgelegt. Beim Aufruf der Datei wird mit großem Geschwindigkeitsvorteil nur mehr das fertige Binärprogramm gerufen. Eine automatische serverseitige Neukompilierung findet immer dann statt, wenn die Datei in einer neuen Version auf den Server geladen wird.

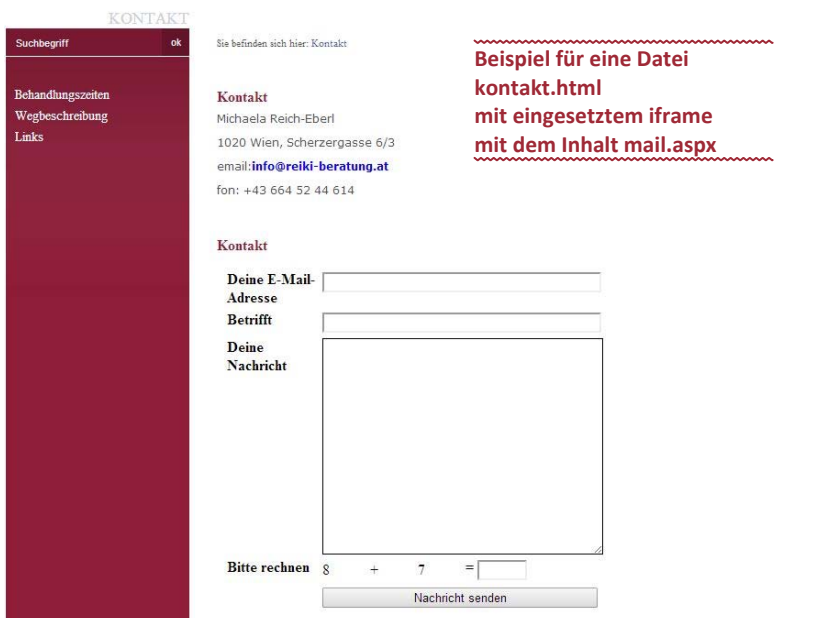
```
<body>
<form id="form1" runat="server">
<asp:Panel ID="Panel_Message" runat="server">
<asp:Label ID="Label_Message" runat="server"></asp:Label>
</asp:Panel>
<asp:Panel ID="Panel_MailDialog" runat="server">
<asp:Label ID="Label_SendError" runat="server"></asp:Label>
<table>
<tr>
<td valign="top" width="100">
<b>Deine E-Mail-Adresse</b>
<asp:TextBox ID="from" Width="300px" runat="server"></asp:TextBox>
</td>
<td valign="top" width="100">
<b>Betreff</b>
<asp:TextBox ID="subj" runat="server" Width="300px"></asp:TextBox>
</td>
<td valign="top" width="100">
<b>Deine Nachricht</b>
<asp:TextBox ID="body" runat="server" Height="230px" TextMode="MultiLine" Width="300px"></asp:TextBox>
</td>
<td valign="top" width="100">
<b>Bitte rechnen</b>
<asp:Label ID="Label_x" runat="server" Width="48px"></asp:Label>
<asp:Label ID="Label_op" runat="server" Width="48px"></asp:Label>
<asp:Label ID="Label_y" runat="server" Width="48px">=
<asp:TextBox ID="Text_x" runat="server" Width="50px"></asp:TextBox>
</td>
<td valign="top" width="100">
<b>
<asp:Button ID="Button_Send" Width="300px" runat="server" OnClick="Button_Send_Click" Text="Nachricht senden" />
</b>
</td>
</tr>
</table>
</asp:Panel>
</body>
</html>
```

Das Programm prüft zunächst, ob die Konfigurations-Strings angegeben worden sind und danach wird getestet, ob der Aufruf vom lokalen Server erfolgt oder von einer erlaubten IP-Adresse. Wenn es ein Erstauftritt ist (kein Post-Back, also ein nochmaliger Aufruf derselben Seite durch eine Benutzerinteraktion auf der Seite), wird die Rechenaufgabe initialisiert. Wenn es aber ein PostBack ist, wird die Lösung der Rechenaufgabe geprüft. Wenn dieser Test erfolgreich war (CheckOK), werden die Inhalte aus den Text-Boxen gelesen (From, Subj, Body) und nur wenn sie ausgefüllt wurden, wird die Sendung veranlasst. Das Sendeprogramm prüft das Format der beteiligten E-Mail-Adressen.

Die Darstellung besteht aus den zwei Panels (werden als `div` gerendert), die bei Programmstart zunächst als unsichtbar geschaltet werden und von denen jeweils eines, je nach Programmverlauf, sichtbar gemacht werden. Das `Panel_MailDialog` enthält das Formular, das Rechenbeispiel und den Sendebutton. Das `Panel_Message` enthält nur den Text, der angezeigt wird, wenn die Mail erfolgreich gesendet wurde. In diesem Panel könnten auch mehr Informationen verpackt werden, etwa eine Zusammenfassung der soeben versendeten Mail.

Das Bild unten zeigt das Formular auf der Webseite eines Clubmitglieds.

Der Code kann auch bei der Webversion dieser Ausgabe downgeloadet werden.



Beispiel für eine Datei `kontakt.html` mit eingesetztem `iframe` mit dem Inhalt `mail.aspx`