

## Задания на производственную практику «Проектирование и разработка веб-приложений»

**Цель:** научиться созданию страниц веб-форм ASP.NET в среде разработки программного обеспечения Microsoft Visual Studio 2019.

**Задача:** выполнить отчёт согласно рекомендациям к оформлению по нижеприведённому заданию.

### ЗАДАНИЕ 1. Создание страницы веб-форм ASP.NET в Visual Studio 2019

В задании описывается создание простой страницы веб-форм ASP.NET и демонстрируются основные методы создания новой страницы, добавления элементов управления и написания кода.

В задании представлены следующие задачи:

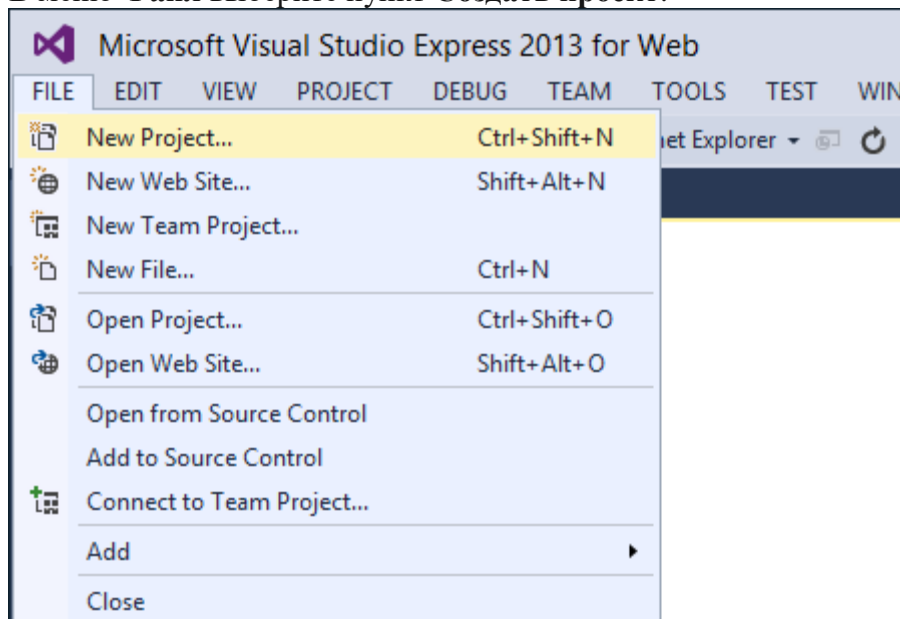
- Создание проекта приложения веб-форм файловой системы.
- Ознакомление с Visual Studio.
- Создание страницы ASP.NET.
- Добавление элементов управления.
- Добавление обработчиков событий.
- Запуск и тестирование страницы из Visual Studio.

#### Предварительные требования

Microsoft Visual Studio 2019.

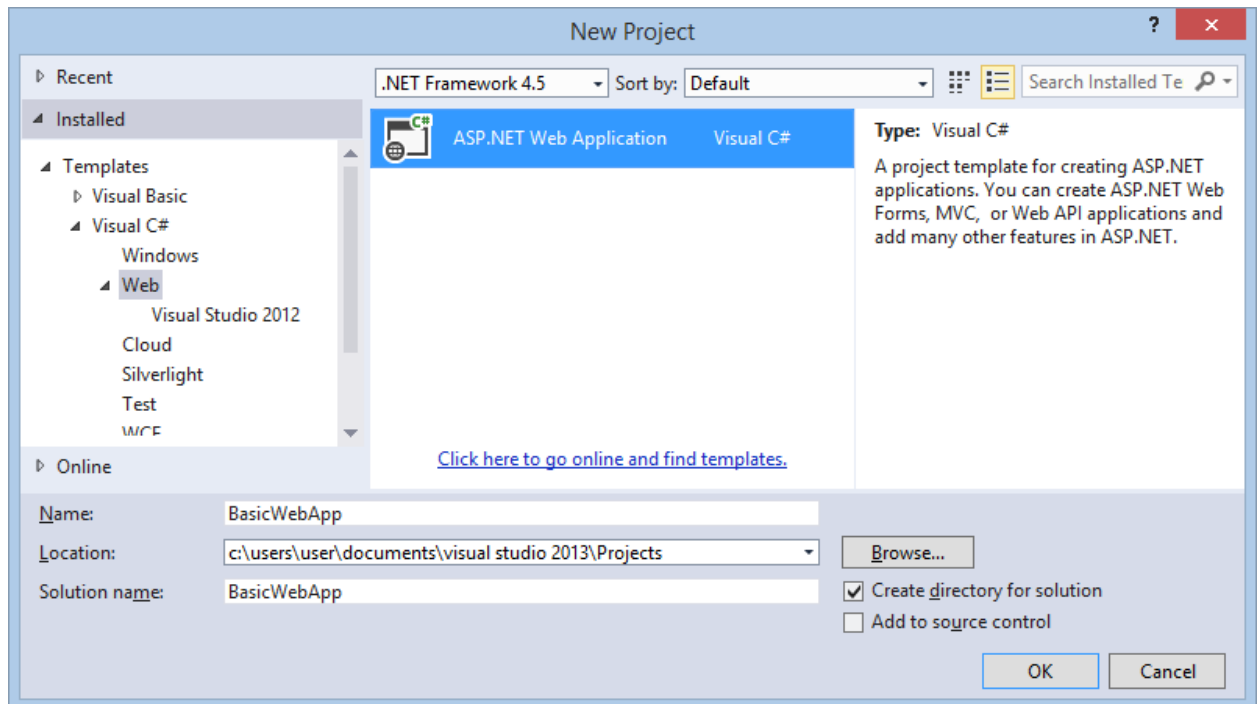
#### Создание проекта веб-приложения

1. Откройте среду Microsoft Visual Studio.
2. В меню **Файл** выберите пункт **Создать проект**.

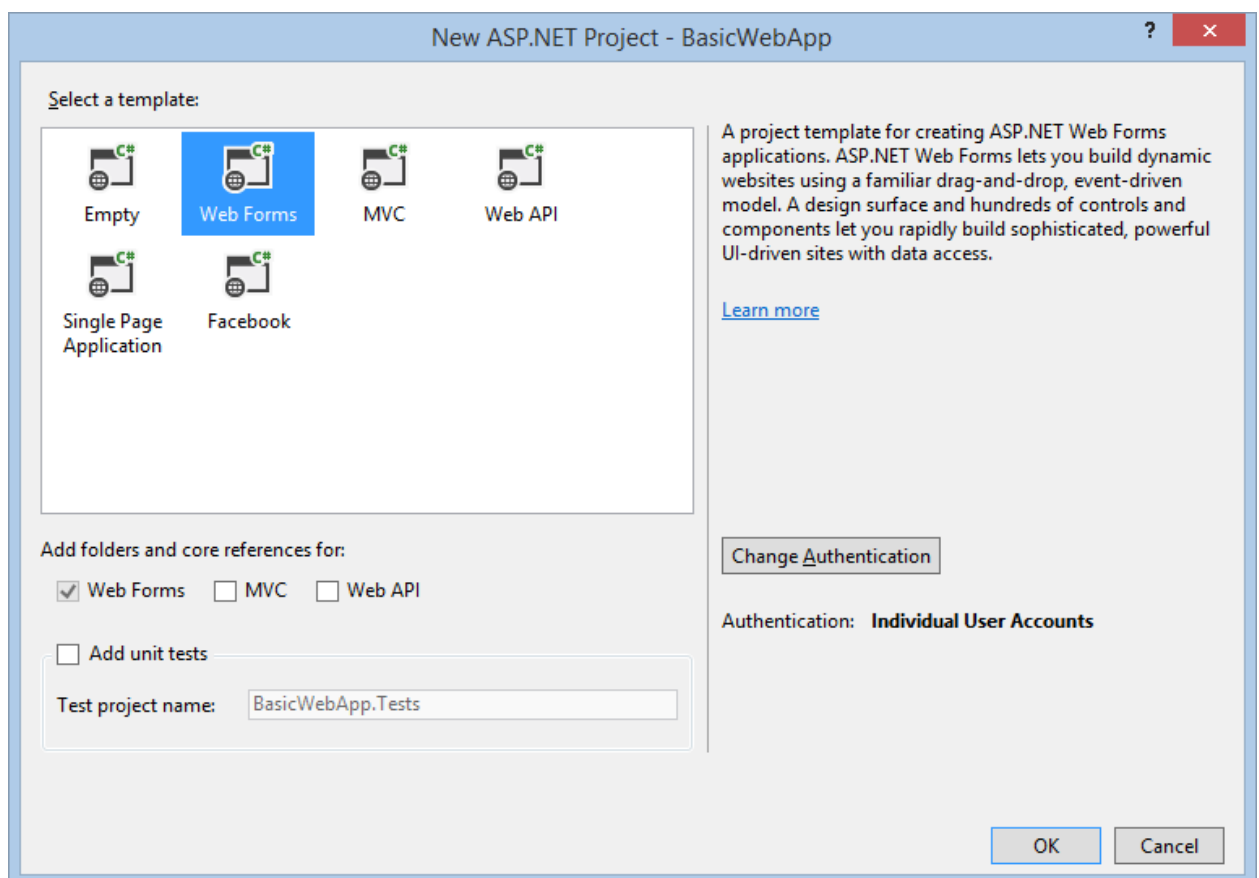


Откроется диалоговое окно **Создание проекта** .

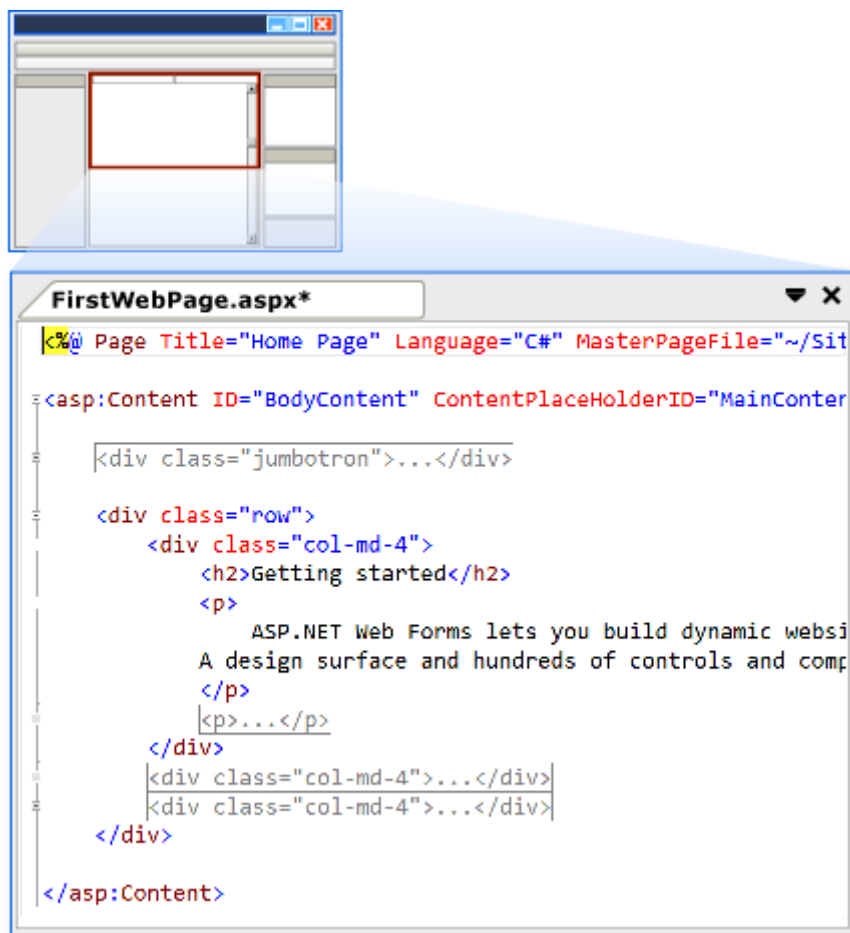
3. В левой части выберите **шаблоны** -> **C# Visual** -> группы **веб-** шаблонов.
4. Выберите шаблон **Веб-приложения ASP.NET** в центральном столбце.
5. Присвойте проекту имя *басиквебapp* и нажмите кнопку **ОК**.



6. Затем выберите шаблон **веб-формы** и нажмите кнопку **ОК**, чтобы создать проект.



Visual Studio создает новый проект, содержащий предварительно созданные функции на основе шаблона веб-форм. Она не только предоставляет вам страницу *Home.aspx*, страницу *About.aspx*, страницу *Contact.aspx*, но также включает функции членства, которые регистрируют пользователей и сохраняют свои учетные данные, чтобы они могли входить на веб-сайт. При создании страницы по умолчанию Visual Studio отображает страницу в представлении **исходного кода**, где можно видеть элементы HTML страницы. На следующем рисунке показано, что вы видите в представлении **исходного кода**, если вы создали новую веб-страницу с именем *басиквебанн.aspx*.



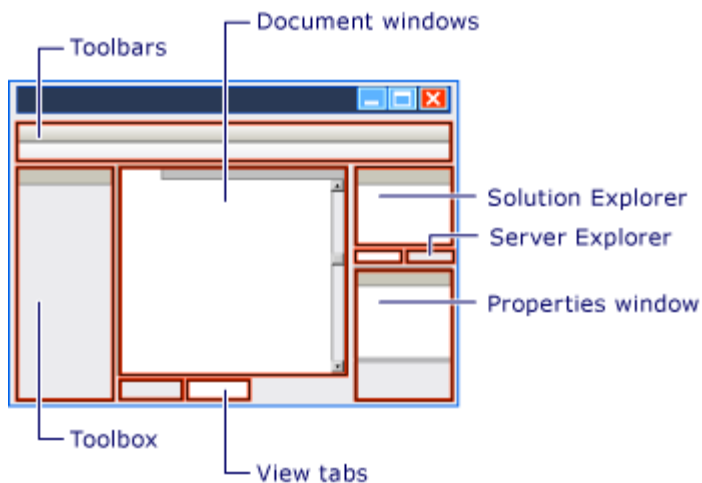
## Обзор среды веб-разработки Visual Studio

Прежде чем продолжить изменение страницы, полезно ознакомиться с средой разработки Visual Studio. На следующем рисунке показаны окна и инструменты, доступные в Visual Studio, и Visual Studio Express для Web.

### Примечание

На этой диаграмме показаны окна и расположения окон по умолчанию. Меню **вид** позволяет отображать дополнительные окна, а также изменять порядок окон и изменять их размер в соответствии с вашими предпочтениями. Если изменения уже внесены в расположение окон, то отображаемое содержимое не будет соответствовать иллюстрации.

Среда Visual Studio



## Ознакомьтесь с веб-конструктор

Изучите приведенную выше иллюстрацию и сопоставьте текст со следующим списком, в котором описываются наиболее часто используемые окна и средства. (Здесь перечислены не все отображаемые окна и средства, а только те, которые отмечены на предыдущем рисунке.)

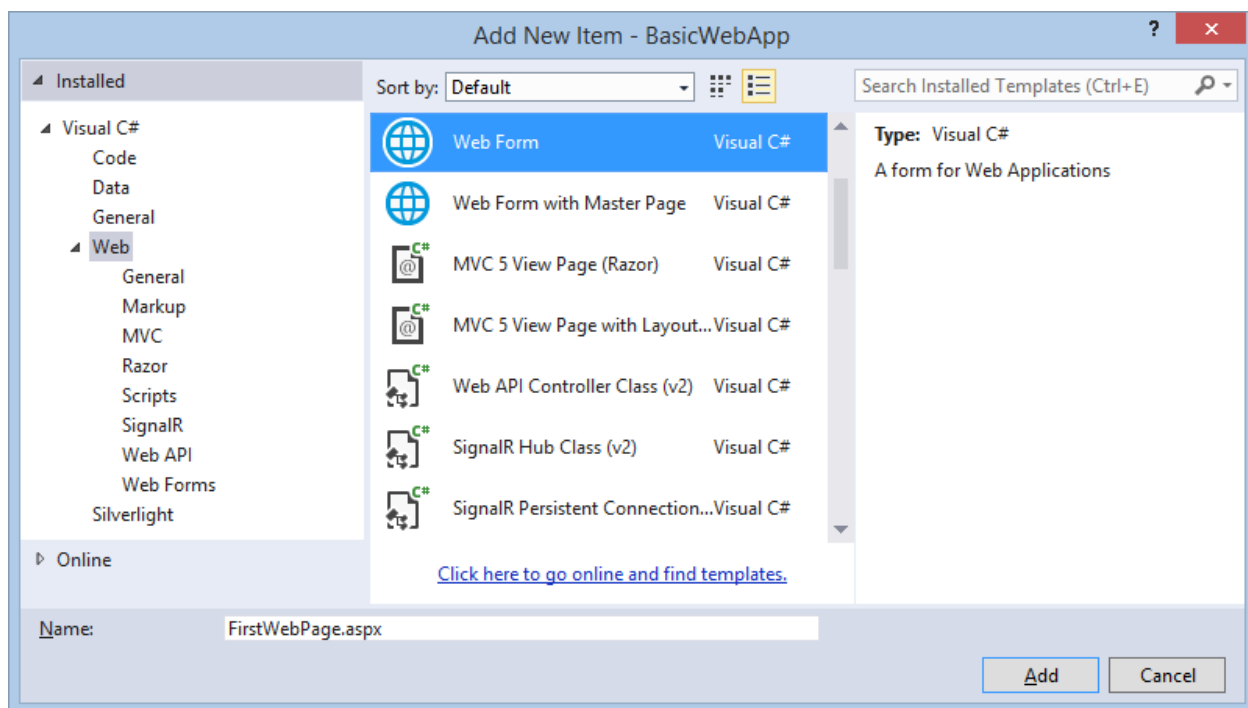
- **Toolbar.** Предоставляют команды для форматирования текста, поиска текста и т. д. Некоторые панели инструментов доступны только при работе в режиме **конструктора**.
- **Обозреватель решений** окно. Отображает файлы и папки в веб-приложении.
- **Окно документа.** Отображает документы, над которыми вы работаете, в окнах с вкладками. Можно переключаться между документами, щелкая вкладки.
- **Окно "Свойства".** Позволяет изменять параметры страницы, элементы HTML, элементы управления и другие объекты.
- **Вкладки представлений.** Представьте себе различные представления одного документа. Представление **конструктора** — это область редактирования практически в режиме WYSIWYG. Представление **исходного кода** — это HTML-редактор для страницы. Представление с **разделением** отображает как представление **конструктора**, так и представление **исходного кода** для документа. Далее в этом пошаговом руководстве вы будете работать с представлениями **конструирования** и **источника**. Если вы предпочитаете открывать веб-страницы в режиме **конструктора**, в меню **Сервис** выберите пункт **Параметры**, выберите узел **Конструктор HTML** и измените параметр **начальные страницы**.
- **Панель элементов.** Предоставляет элементы управления и элементы HTML, которые можно перетащить на страницу. Элементы **панели элементов** группируются по общей функции.
- **Ever обозреватель.** Отображает подключения к базе данных. Если обозреватель сервера не отображается, в меню Вид выберите пункт обозреватель сервера.

## Создание новой страницы веб-форм ASP.NET

При создании нового приложения Web Forms с помощью шаблона проекта **веб-приложения ASP.NET** в Visual Studio добавляется страница ASP.NET (страница веб-форм) с именем *Default.aspx*, а также несколько других файлов и папок. Страницу *Default.aspx* можно использовать в качестве домашней страницы веб-приложения.

## Добавление страницы в веб-приложение

1. Закройте страницу *Default.aspx*. Для этого перейдите на вкладку с именем файла и нажмите кнопку Close (заккрыть).
2. В **Обозреватель решений** щелкните правой кнопкой мыши имя веб-приложения (в этом руководстве имя приложения — **басиквебсите**), а затем щелкните **добавить - > новый элемент**.  
Откроется диалоговое окно **Добавление нового элемента**.
3. Выберите в левой части группы **веб-** шаблоны **Visual C#** ->. Затем выберите **веб-форму** из среднего списка и назовите ее *фирствебпаже.aspx*.



4. Нажмите кнопку **Добавить**, чтобы добавить веб-страницу в проект. Visual Studio создаст новую страницу и откроет ее.

## Добавление HTML на страницу

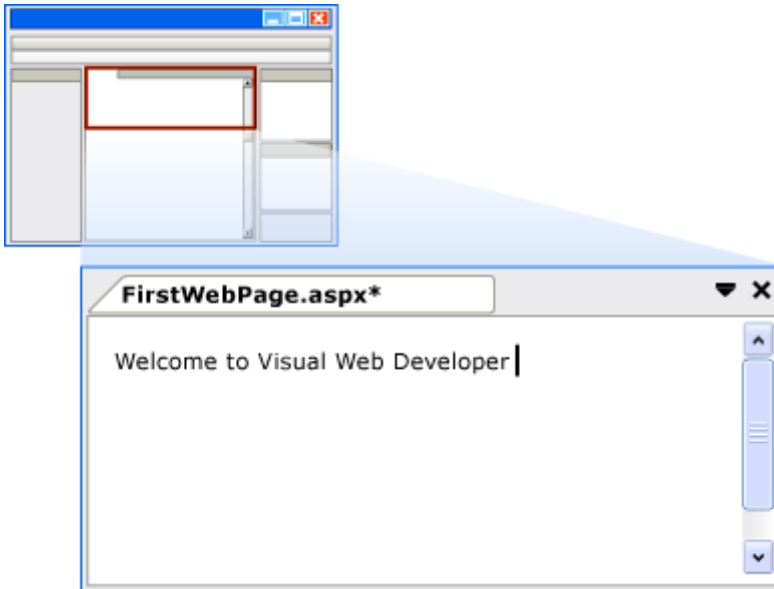
### Добавление текста на страницу

1. В нижней части окна документа перейдите на вкладку **конструктор**, чтобы переключиться в режим **конструктора**.

Представление конструирования отображает текущую страницу в режиме, аналогичном WYSIWYG. На этом этапе у вас нет текста или элементов управления на странице, поэтому страница пуста, за исключением пунктирной линии, которая обозначает прямоугольник. Этот прямоугольник представляет элемент **div** на странице.

2. Щелкните внутри прямоугольника, выделенного пунктирной линией.
3. Введите **Добро пожаловать в Visual Web Developer** и дважды нажмите клавишу **Ввод**.

На следующем рисунке показан текст, введенный в режиме **конструктора**.



4. Переключитесь в представление **исходного кода**.

Вы видите HTML в представлении **исходного кода**, созданном при вводе в режиме **конструктора**.

```
<%@ Page Language="C#" AutoEventWireup="true"
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      Welcome to Visual Web Developer<br />
      <br />
    </div>
  </form>
</body>
</html>
```

100 % Design Split Source <html> <body> <form#form1> <div>

Веб-страница

### Запуск страницы

Прежде чем приступать к добавлению элементов управления на страницу, можно сначала запустить его.

## Запуск страницы

1. В **Обозреватель решений** щелкните правой кнопкой мыши *фирствебпаже.aspx* и выберите **задать в качестве начальной страницы**.
2. Нажмите клавиши **CTRL + F5**, чтобы запустить страницу.

Страница отобразится в браузере. Несмотря на то, что созданная страница имеет расширение имени файла *.aspx*, она в настоящее время работает как любая HTML-страница.

Чтобы отобразить страницу в браузере, можно также щелкнуть страницу правой кнопкой мыши в **Обозреватель решений** и выбрать пункт **Просмотр в браузере**.

3. Закройте браузер, чтобы завершить работу веб-приложения.

## Добавление и программирование элементов управления

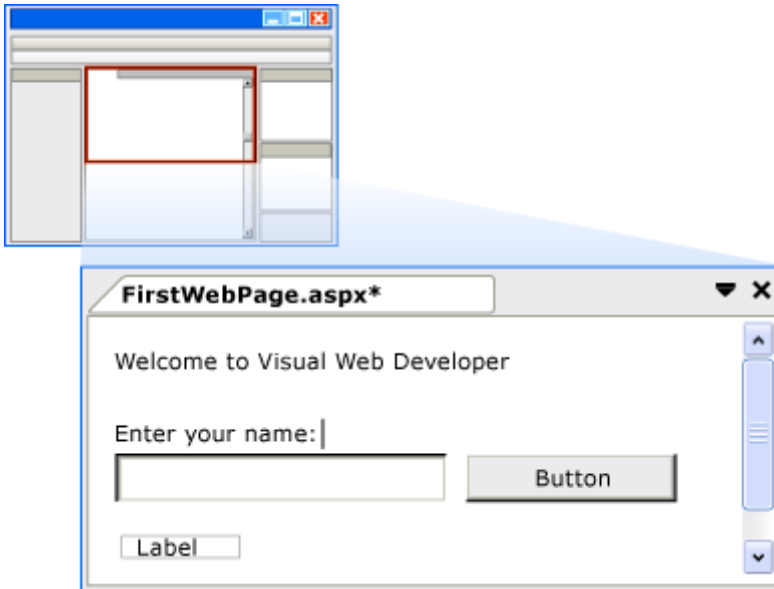
Теперь на страницу будут добавляться серверные элементы управления. Серверные элементы управления, такие как кнопки, метки, текстовые поля и другие привычные элементы управления, предоставляют стандартные возможности обработки форм для страниц веб-форм. Однако можно программировать элементы управления с помощью кода, выполняемого на сервере, а не на клиенте.

На страницу будет добавлен элемент управления [Button](#), элемент управления [TextBox](#) и элемент управления [Label](#), а также написан код для работы с событием [Click](#) для элемента управления [Button](#).

## Добавление элементов управления на страницу

1. Перейдите на вкладку **конструктор**, чтобы переключиться в режим **конструктора**.
2. Поместите точку вставки в конец текста **Добро пожаловать в текст Visual Web Developer** и нажмите клавишу **Ввод** пять или более раз, чтобы освободить место в поле элемента **div**.
3. В **области элементов** разверните группу "**стандартные**", если она еще не развернута.  
Обратите внимание, что может потребоваться развернуть окно **панели элементов** слева, чтобы просмотреть его.
4. Перетащите элемент управления [TextBox](#) на страницу и поместите его в середину поля элемента **div**, которая содержит **Вас приветствует Visual Web Developer** в первой строке.
5. Перетащите элемент управления [Button](#) на страницу и поместите его справа от элемента управления [TextBox](#).
6. Перетащите элемент управления [Label](#) на страницу и поместите его на отдельную строку под элементом управления "**Кнопка**".
7. Поместите точку вставки над элементом управления [TextBox](#), а затем введите **введите свое имя:**.

Этот статический HTML-текст является заголовком элемента управления [TextBox](#). На одной странице можно смешивать статические HTML-и серверные элементы управления. На следующем рисунке показано, как три элемента управления отображаются в режиме **конструктора**.



## Настройка свойств элемента управления

Visual Studio предлагает различные способы задания свойств элементов управления на странице. В этой части пошагового руководства будут заданы свойства как в представлении **конструктора**, так и в представлении **исходного кода**.

### Задание свойств элемента управления

1. Сначала откройте окно **Свойства**, выбрав в меню **вид пункт> другие - Windows > Свойства**. Можно также нажать **клавишу F4**, чтобы открыть окно **свойства**.
2. Выберите элемент управления **Button**, а затем в окне **Свойства** задайте для параметра **текст** значение **Отображаемое имя**. Введенный текст появится на кнопке в конструкторе, как показано на следующем рисунке.



3. Переключитесь в представление **исходного кода**.

Представление **исходного кода** отображает HTML для страницы, включая элементы, созданные Visual Studio для серверных элементов управления. Элементы управления объявляются с помощью синтаксиса, аналогичного HTML, за исключением того, что теги используют префикс **ASP:** и включают атрибут **runat = "Server"** .

Свойства элемента управления объявляются как атрибуты. Например, при задании свойства **Text** для элемента управления **Button** на шаге 1 фактически задается атрибут **Text** в разметке элемента управления.

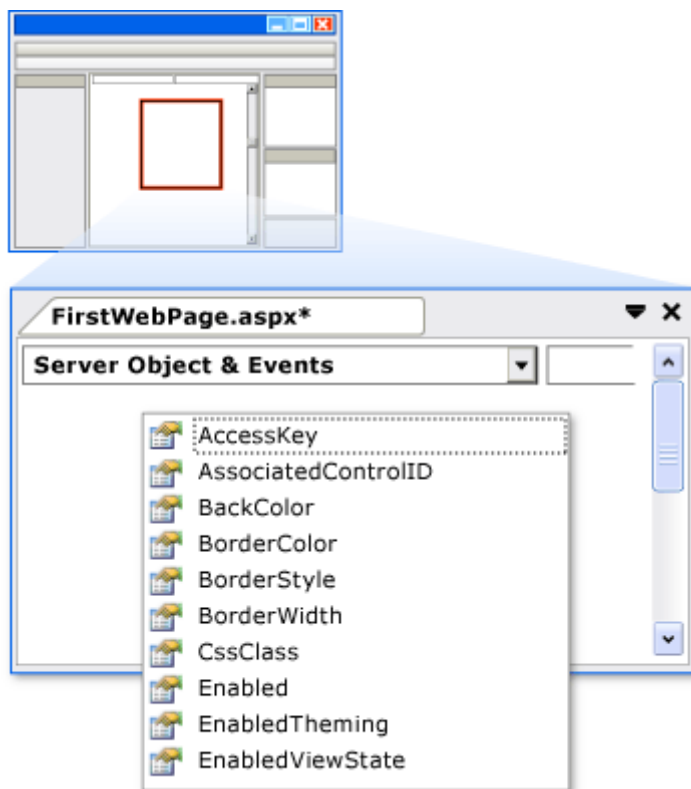
### Примечание

Все элементы управления находятся внутри элемента **Form** , в котором также имеется атрибут **runat = "Server"** . Атрибут **runat = "server"** и тег **ASP:** prefix для

управляющих тегов отмечают элементы управления, чтобы они обрабатывались с помощью ASP.NET на сервере при выполнении страницы. Код за пределами `<Form runat = "server">` и `<script runat = "Server">` элементы отправляются в браузер без изменений, поэтому код ASP.NET должен находиться внутри элемента, открывающий тег которого содержит атрибут `runat = "Server"`.

4. Далее в элемент управления [Label](#) будет добавлено дополнительное свойство. Поместите точку вставки непосредственно после кода `ASP: Label` в тег `<asp: Label>` и нажмите клавишу **пробел**.

Появится раскрывающийся список, в котором отображается список доступных свойств, которые можно задать для элемента управления [Label](#). Эта функция, называемая **IntelliSense**, помогает в представлении **исходного кода** с использованием синтаксиса серверных элементов управления, элементов HTML и других элементов на странице. На следующем рисунке показан раскрывающийся список **IntelliSense** для элемента управления [Label](#).



5. Выберите **ForeColor** и введите знак равенства.

IntelliSense отображает список цветов.

### Примечание

Раскрывающийся список **IntelliSense** можно отобразить в любое время, нажав клавиши **CTRL + J** при просмотре кода.

6. Выберите цвет для текста элемента управления "[Метка](#)". Убедитесь, что выбран цвет, достаточно темный для чтения на белом фоне.

Атрибут **ForeColor** заполняется выбранным цветом, включая закрывающие кавычки.

## Программирование элемента управления "Кнопка"

В этом пошаговом руководстве будет написан код, который считывает имя, введенное пользователем в текстовое поле, и отображает имя в элементе управления [Label](#) .

### Добавление обработчика событий кнопки по умолчанию

1. Переключитесь в режим **конструктора**.
2. Дважды щелкните элемент управления "[Кнопка](#)".

По умолчанию Visual Studio переключается на файл кода программной части и создает каркас обработчика событий для события по умолчанию элемента управления [Button](#), события [Click](#). Файл кода программной части отделяет разметку пользовательского интерфейса (например, HTML) от серверного кода (например, C#).

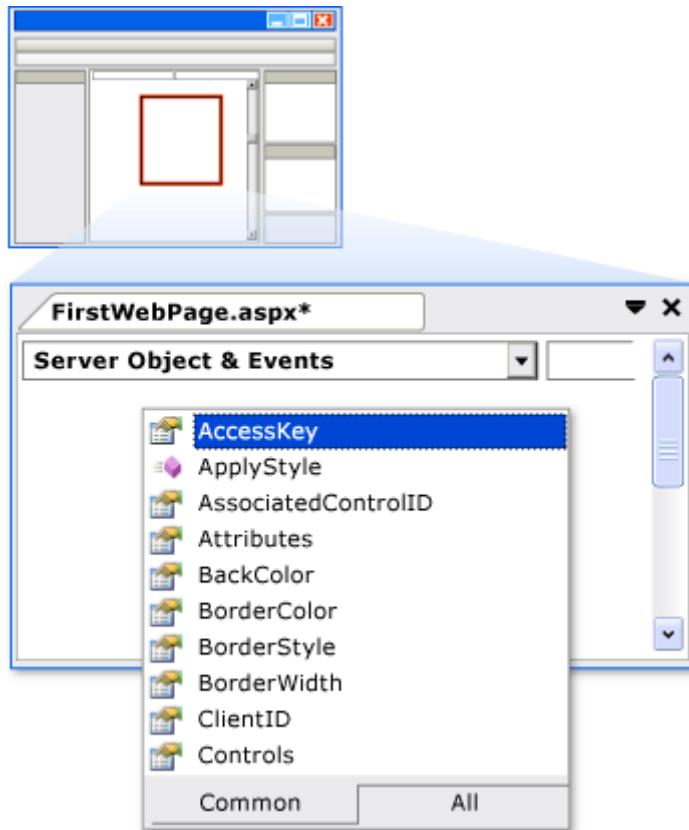
Курсор позиционируется на добавленный код для этого обработчика событий.

### Примечание

Двойной щелчок элемента управления в представлении **конструктора** — это лишь один из нескольких способов создания обработчиков событий.

3. Внутри обработчика событий **Button1\_Click** введите **Label1**, за которым следует точка ( . ).

При вводе точки после **идентификатора** метки (**Label1**) в Visual Studio отображается список доступных элементов для элемента управления [Label](#), как показано на следующем рисунке. Элемент обычно является свойством, методом или событием.



4. Завершите обработчик события **нажатия** для кнопки, чтобы он был считан, как показано в следующем примере кода.

C#Копировать

```
protected void Button1_Click(object sender, System.EventArgs e)
{
    Label1.Text = TextBox1.Text + ", welcome to Visual Studio!";
}
```

VBКопировать

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Label1.Text = Textbox1.Text & ", welcome to Visual Studio!"
End Sub
```

5. Вернитесь к просмотру **исходного** кода HTML-разметки, щелкнув правой кнопкой мыши *фирствебпаже. aspx* в **Обозреватель решений** и выбрав пункт **Просмотреть разметку**.
6. Перейдите к элементу **<ASP: Button>**. Обратите внимание, что элемент **<ASP: Button>** теперь имеет атрибут **onclick = "Button1\_щелкните"**.

Этот атрибут привязывает событие [нажатия](#) кнопки к методу обработчика, который вы написали на предыдущем шаге.

Методы обработчика событий могут иметь любое имя; отображаемое имя является именем по умолчанию, созданным Visual Studio. Важно отметить, что имя, используемое для атрибута **OnClick** в HTML, должно соответствовать имени метода, определенного в коде программной части.

## Запуск страницы

Теперь можно протестировать серверные элементы управления на странице.

## Запуск страницы

1. Нажмите клавиши **CTRL + F5**, чтобы запустить страницу в браузере. Если возникает ошибка, выполните описанные выше действия.
2. Введите имя в текстовое поле и нажмите кнопку **Отображаемое имя**.

Введенное имя отображается в элементе управления "[Метка](#)". Обратите внимание, что при нажатии кнопки страница отправляется на веб-сервер. Затем ASP.NET повторно создает страницу, выполняет код (в данном случае запускается обработчик событий [Click](#) элемента управления [Button](#)), а затем отправляет новую страницу в браузер. Если просмотреть строку состояния в браузере, можно увидеть, что страница осуществляет циклический проход на веб-сервер при каждом нажатии кнопки.

3. В браузере просмотрите источник страницы, которую вы используете, щелкнув правой кнопкой мыши страницу и выбрав пункт **Просмотреть источник**.

В исходном коде страницы отображается HTML без какого-либо кода сервера. В частности, не отображаются элементы `<ASP:>`, с которыми вы работали в представлении **исходного кода**. При запуске страницы ASP.NET обрабатывает серверные элементы управления и отображает HTML-элементы на страницу, которая выполняет функции, представляющие элемент управления. Например, элемент управления `<ASP: Button>` отображается как элемент HTML `<input type = "Submit">`.

4. Закройте браузер.

## Работа с дополнительными элементами управления

В этой части пошагового руководства вы будете работать с элементом управления [Calendar](#), который отображает даты в месяц за раз. Элемент управления "[Календарь](#)" является более сложным элементом управления, чем кнопка, текстовое поле и метка, с которыми вы работали, и демонстрирует некоторые дополнительные возможности серверных элементов управления.

В этом разделе вы добавите элемент управления [System. Web. UI. Controls. Calendar](#) на страницу и отформатируете его.

## Добавление элемента управления Calendar

1. В Visual Studio переключитесь в режим **конструктора**.
2. Из **стандартного** раздела панели элементов перетащите элемент управления [Calendar](#) на страницу и поместите его под элементом **div**, который содержит другие элементы управления.

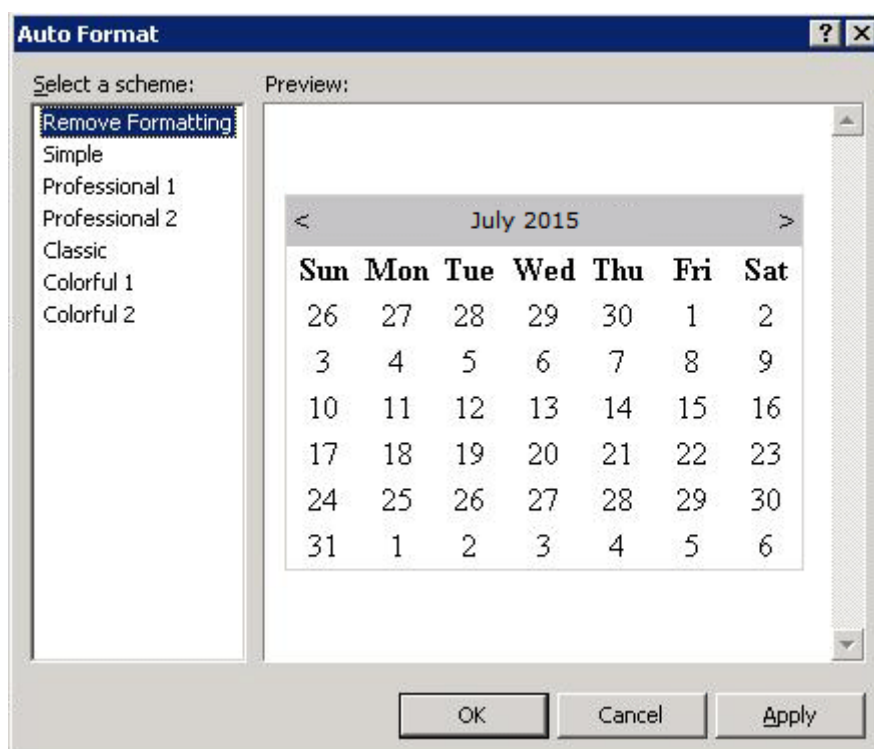
Отобразится панель смарт-тегов календаря. На панели отображаются команды, облегчающие выполнение наиболее распространенных задач для выбранного

элемента управления. На следующем рисунке показан элемент управления [Calendar](#), отображаемый в режиме **конструктора** .

![Элемент управления Calendar в представлении конструирования](creating-a-basic-web-forms-page/\_static/image13.png "Элемент управления "Календарь" в конструкторе")

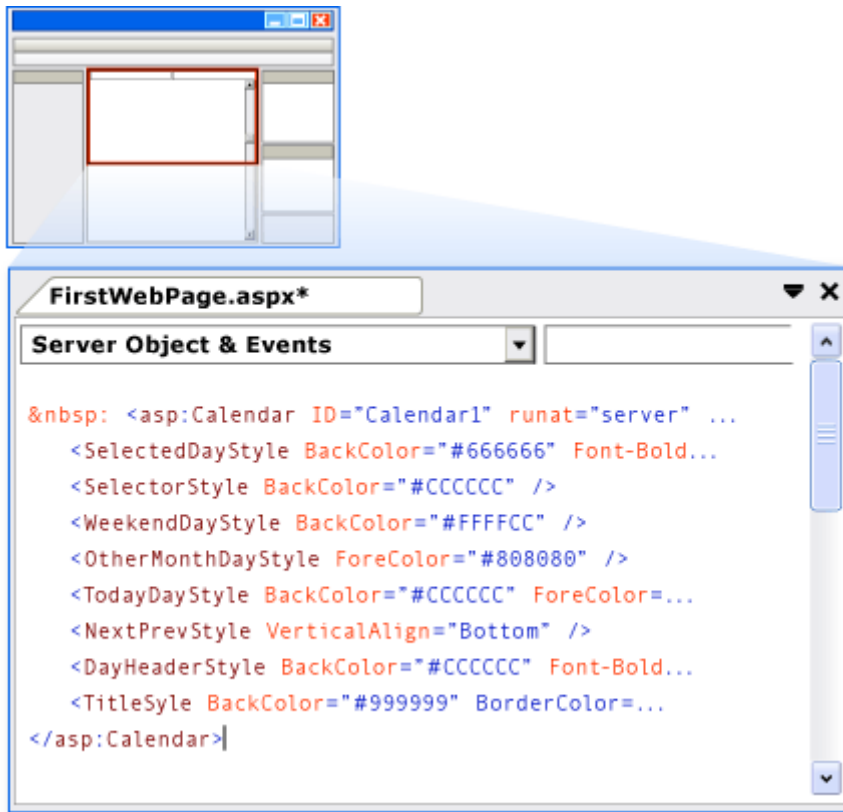
3. На панели смарт-тегов выберите **Автоматический формат**.

Откроется диалоговое окно **Автоформат**, в котором можно выбрать схему форматирования для календаря. На следующем рисунке показано диалоговое окно **Автоформат** для элемента управления [Calendar](#).



4. В списке **выберите схему** выберите **простой** и нажмите кнопку **ОК**.
5. Переключитесь в представление **исходного кода**.

Вы видите элемент `<ASP: Calendar>`. Этот элемент намного больше, чем элементы для простых элементов управления, созданных ранее. Он также включает подэлементы, такие как `<викендайстиле>`, которые представляют различные параметры форматирования. На следующем рисунке показан элемент управления [Calendar](#) в представлении **исходного кода**. (Точная разметка, отображаемая в представлении **исходного кода**, может немного отличаться от иллюстрации.)



## Программирование элемента управления Calendar

В этом разделе будет запрограммирован элемент управления [Calendar](#) для показа текущей выбранной даты.

## Программирование элемента управления Calendar

1. В режиме **конструктора** дважды щелкните элемент управления [Calendar](#) .

Новый обработчик событий создается и отображается в файле кода программной части с именем *FirstWebPage.aspx.CS*.

2. Завершите обработчик событий [SelectionChanged](#) , используя следующий код.

C#Копировать

```
protected void Calendar1_SelectionChanged(object sender, System.EventArgs e)
{
    Label1.Text = Calendar1.SelectedDate.ToLongDateString();
}
```

VBКопировать

```
Protected Sub Calendar1_SelectionChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    Label1.Text = Calendar1.SelectedDate.ToLongDateString()
End Sub
```

Приведенный выше код задает текст элемента управления Label равным выбранной дате элемента управления Calendar.

## Запуск страницы

Теперь можно протестировать календарь.

## Запуск страницы

1. Нажмите клавиши **CTRL + F5**, чтобы запустить страницу в браузере.
2. Щелкните дату в календаре.

Выбранная дата отображается в элементе управления " [Метка](#) ".

3. В браузере просмотрите исходный код страницы.

Обратите внимание, что элемент управления [Calendar](#) отображается на странице в виде **таблицы**, каждый день в виде элемента **TD** .

4. Закройте браузер.

Подробное описание и дополнительные статьи можно найти по ссылке:

<https://docs.microsoft.com/ru-ru/aspnet/web-forms/overview/getting-started/creating-a-basic-web-forms-page>