

Lesson 3-3: Use Breakpoints

So far, you've been able to tell that your code is working properly by running it and seeing the results. When things don't work the way you expect them to, it is helpful to be able to get a closer look at what is going wrong.

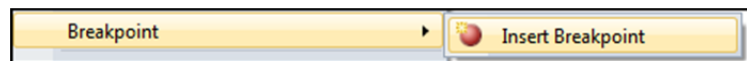
Breakpoints let you pause your code and examine the current state of the application.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open *default.aspx.cs* (The code-behind file of *default.aspx*).
- 3 Add a breakpoint to the *CalendarSelect_SelectionChanged* event handler.

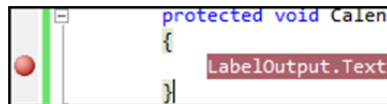
1. Click on the line of code that sets *LabelOutput*'s *Text* property in the *CalendarSelect_SelectionChanged* event handler.

```
protected void CalendarSelect_SelectionChanged(object sender, EventArgs e)
{
    LabelOutput.Text = CalendarSelect.SelectedDate.ToString();
}
```

2. Right-click on the line and then click Breakpoint→Insert Breakpoint from the shortcut menu.

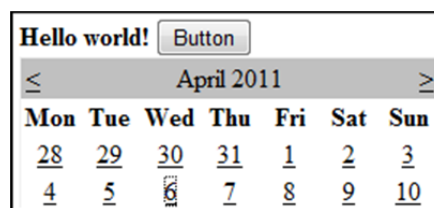


The line will be highlighted in red, and a red circle will appear in the bar to the left. This indicates that a breakpoint is present.



- 4 Inspect properties while code is running.

1. Run *default.aspx* in Debug mode (see sidebar).
2. Click on a date in the calendar.



You should be automatically brought back to Visual Studio with the line you added a breakpoint to highlighted in yellow.

```
protected void CalendarSelect_SelectionChanged(object sender, EventArgs e)
{
    LabelOutput.Text = CalendarSelect.SelectedDate.ToString();
}
```

Your web application is now paused at the point where you placed the breakpoint.

3. Examine the current *Text* of *LabelOutput*.

tip

Setting breakpoints by clicking the left bar

As well as right-clicking and clicking *Insert Breakpoint*, you can quickly add and remove breakpoints by clicking in the gray bar on the left where the breakpoint indicator circles appear.



If you click the breakpoint indicator circle the breakpoint will be removed.

important

The difference between Debug mode and View in Browser

So far you've been able to view your pages either by starting debugging or right-clicking on them in the *Solution Explorer* and clicking *View in Browser* from the shortcut menu.

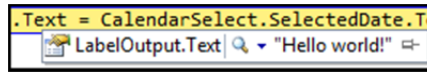
At this point, an important difference between the two emerges.

If you choose *View in Browser* from the *Solution Explorer*, Visual Studio won't process your breakpoints.

For breakpoints to work properly, you must always use Debug mode by clicking *Debug→Start Debugging* or by clicking the 'play' button on the toolbar:



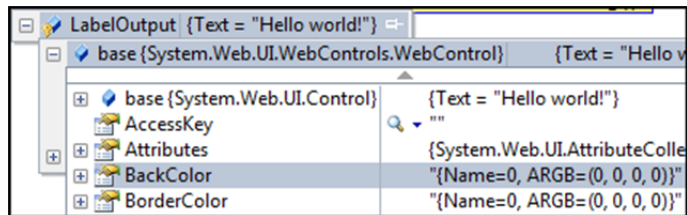
You can quickly see the current values of properties by hovering your mouse cursor over them. Hover your cursor over the `.Text` part of `LabelOutput.Text`.



You can see that the current value of `LabelOutput.Text` is: *Hello world!*

- View the rest of the properties of the `LabelOutput` control.

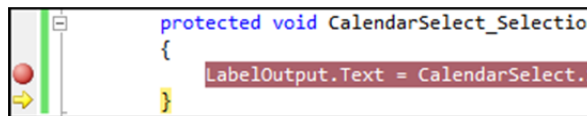
By hovering your mouse cursor over `LabelOutput` itself and expanding the categories by clicking the + symbol, you can see the rest of `LabelOutput`'s properties.



5 Step through code using the *Debug* tools.

Although it's useful to see the properties at your breakpoint, it's even more useful to be able to step through your code line by line and see what is affected.

- Click `Debug` → `Step Over`.



You will see that the next line is highlighted in yellow and has a small yellow arrow in the bar to the left. You have now stepped through the last line and onto this one.

Because you stepped through the last line, it will have updated the label's text.

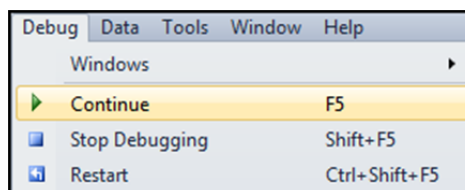
- Check the `Text` property of `LabelOutput`.

As you did before, move your mouse cursor over `.Text` in the `LabelOutput.Text` code.

You'll see that the text has been updated.



- Click `Debug` → `Continue`.



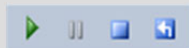
You will be sent back to your web browser and the site will continue running as normal.

You can also continue running by clicking the 'play' button on the toolbar (see sidebar).

note

The Debug Toolbar

As well as using the *Debug* menu, you can use the icons on the Debug toolbar.



These allow you to continue, pause, stop and restart debugging.



These correspond to *Step Into*, *Step Over* and *Step Out*.

The difference between these isn't obvious at the moment, but will make more sense in: *Session Six: C# Classes, Namespaces and Methods*.

For now you only need to use *Step Over*.