Asynchronous Programming in C# 5

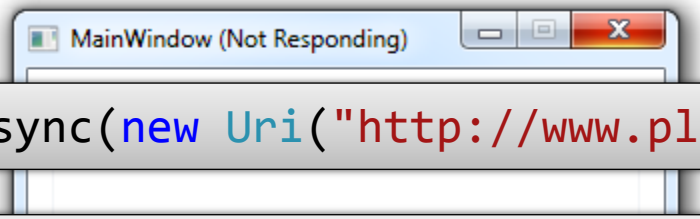
What's New in C# 5

http://msdn.microsoft.com/vstudio/async

pluralsight
see what you can learn

Synchronous vs Asynchronous

```
WebClient w = new WebClient();
...
string txt = w.DownloadString("http://www.pluralsight.com/");
dataTextBox.Text = txt;
```




```
w.DownloadStringAsync(new Uri("http://www.pluralsight.com/"));
```

```
void w_DownloadStringCompleted(object sender,
          DownloadStringCompletedEventArgs e)
{
    dataTextBox.Text = e.Result;
}
```

```
w.DownloadStringAsync(new Uri("http://www.pluralsight.com/"));
```

```
void w_DownloadStringCompleted(object sender,
          DownloadStringCompletedEventArgs e)
{
    dataTextBox.Text = e.Result;
}
```

C# 5: async and await

```
private async void DoDownload()
{
  WebClient w = new WebClient();

  string txt = await w.DownloadStringTaskAsync("http://192.168.27.77/");
  dataTextBox.Text = txt;
}
```

returns

```
System.Threading.Task<string>
```

Inside await

```
var result = await someTask;
// ...rest of method
```


```
var awaiter = someTask.GetAwaiter();
Action callback = delegate
{
    var result = awaiter.EndAwait();

    // ...rest of method
};

if (awaiter.BeginAwait(callback)) { return; }
else { callback(); }
```

Returning Tasks

```
private async Task<string> GetHeaders()
{
    var req = (HttpWebRequest)
            WebRequest.Create("http://www.pluralsight.com/");
    req.Method = "HEAD";
    var getResponseTask = Task.Factory.FromAsync<WebResponse>(
            req.BeginGetResponse, req.EndGetResponse, null);
    var resp = (HttpWebResponse) await getResponseTask;

    string result = FormatHeaders(resp.Headers);
    return result;
}
```

Exceptions

```
try
{
  string txt = await w.DownloadStringTaskAsync(url);
  dataTextBox.Text = txt;
}
catch (WebException x)
{
  ... Handle as usual
}
```

Concurrent Work

```
var task1 = wc1.DownloadStringTaskAsync(url1);
var task2 = wc2.DownloadStringTaskAsync(url2);

string txt1 = await task1;
string txt2 = await task2;
```

```
var task1 = wc1.DownloadStringTaskAsync(url1);
var task2 = wc2.DownloadStringTaskAsync(url2);

string[] results = await TaskEx.WhenAll(task1, task2);
```

Asynchronous Work
!=
Threads

For more in-depth online developer training visit

on-demand content from authors you trust