Objects, Properties and Methods

Part 2 Methods

Let's take a look at some examples using Methods:

We're going to be using get-process and the process named notepad

- First let's open notepad, open your search bar and press notepad, minimize to the taskbar
- **Type get-process, press return, press return again. Scroll up, and we see that notepad is running. Press return and cls.**
- **Type get-process -name notepad** press return, now press return again

  In our boat and car **illustration** we showed you that **objects** contain **properties** and **methods,** but when we take a look at get-process there are no methods or properties listed.

  So, the question is, how can I display the **property or methods** for **get-process?**

- By using a special command called **get-member**. We'll use the **alias gm for get-member.**
- Type get-process -name notepad | gm. (Use pipe operator) That's the symbol right above the enter key. Press your shift key and your pipe operator.
  If you recall, a pipeline takes the output of one command and pushes it through to the input of the second command.

  So, get-process has a process called notepad, and you're piping the **output of notepad** into the input of get-member. Press return

- There is some very important information, when using **get-member.**

  It's **TypeName**. Typename tells us **what kind of object that is being sent across the pipeline.** In this case the **TypeName** is **System.Diagnostics.Process** but we'll just use the **last part** which is **process.** Just keep that in mind, you'll need it later when we get to the **pipeline lecture.**

- Notice the **method** called **kill**. Kill is definitely an **action.**
  Go ahead and type this out then we'll explain it
- **Type (Get-Process Notepad).kill()**

- To use a **method** of an object**, Place the cmdlet name and the argument in parenthesis, then type a dot (.), then the method name, and a set of parentheses "()"**.
  The parentheses are **required** for every **method call,** even when there are no arguments.

The point is that we don't need a separate cmdlet to close the process notepad we have the method kill. Press return
And we see that **notepad has been closed.**

Let's try another example of using Methods

- **Let's say** we want to copy files from one location to another. In this case we'll use **the get-childitem** command.

- **Let open** windows explorer, **for demonstration purposes** I've created a **folder named content**.
  Double click on content and we have a text file named **computers.** click that file and displayed is the contents of the text file.

- **Now we want to copy this file from the C:\content folder to the c:\test folder using a method called copyto.**

  Type get-childitem | GM – (Pipe operator) press return

  Notice the get-childitem TypeName: is **FileInfo**

  We'll use the copyto method to copy a file from one location to another

- **We don't need a cmdlet, not when we have the copyto method, press return**

  Let's go ahead and type the command then I'll explain it.

  Type **(Get-childitem C:\content\computers.txt). copyto("C:\test\computers.txt")**

- **Put the cmdlet name and the argument in parenthesis then type a dot (.), the method name, in this case is copyto. and a set of parentheses "()"**.
  If the method has arguments, place the argument values inside the parentheses.
  In this case we also added the quotation marks, and press return
- And we see that the copyto method was successful. Go out to explorer and the test folder, and there is our text file.

  This was just two examples of using methods, but it gives you an idea of how to use methods to accomplish an everyday task.