

WINDOWS POWERSHELL

PRIMEROS PASOS

Contenido



1. Qué es Windows PowerShell
2. La consola PowerShell

Contenido



1. Qué es Windows PowerShell

2. La consola PowerShell

Windows PowerShell™

1. Qué es Windows PowerShell

- ❑ Es una interfaz de línea de órdenes (o *shell*) y un **lenguaje** de scripting específicamente diseñados para tareas de administración
- ❑ Está construido sobre la **infraestructura .NET**
- ❑ Se incluye por defecto desde Windows Server 2008 R2 y Windows 7
- ❑ Actualmente, la versión para Windows Server 2012 R2 es la 4.0

Características

1. Qué es Windows PowerShell

- En lugar de procesar texto, procesa **objetos** basados en la infraestructura .NET
- Sus órdenes nativas se denominan ***cmdlets***
 - ▣ Nomenclatura/sintaxis coherentes y sistemáticas
 - ▣ Interfaz amplia y extensible
- Proporciona un **lenguaje de *scripting*** completo
- **Navegación** por objetos del sistema idéntica a la del sistema de archivos (*providers*)

Contenido



1. Qué es Windows PowerShell

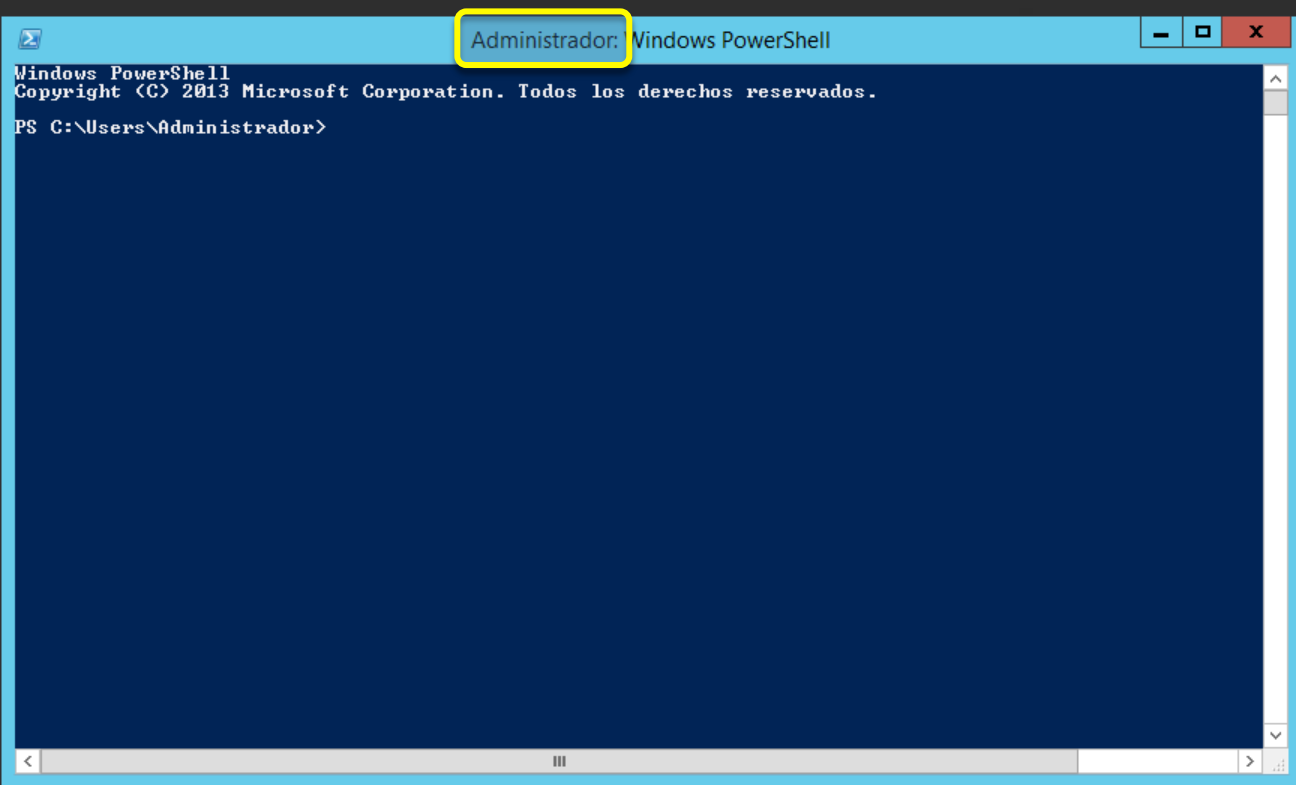
2. La consola PowerShell



 Windows Server 2012 R2



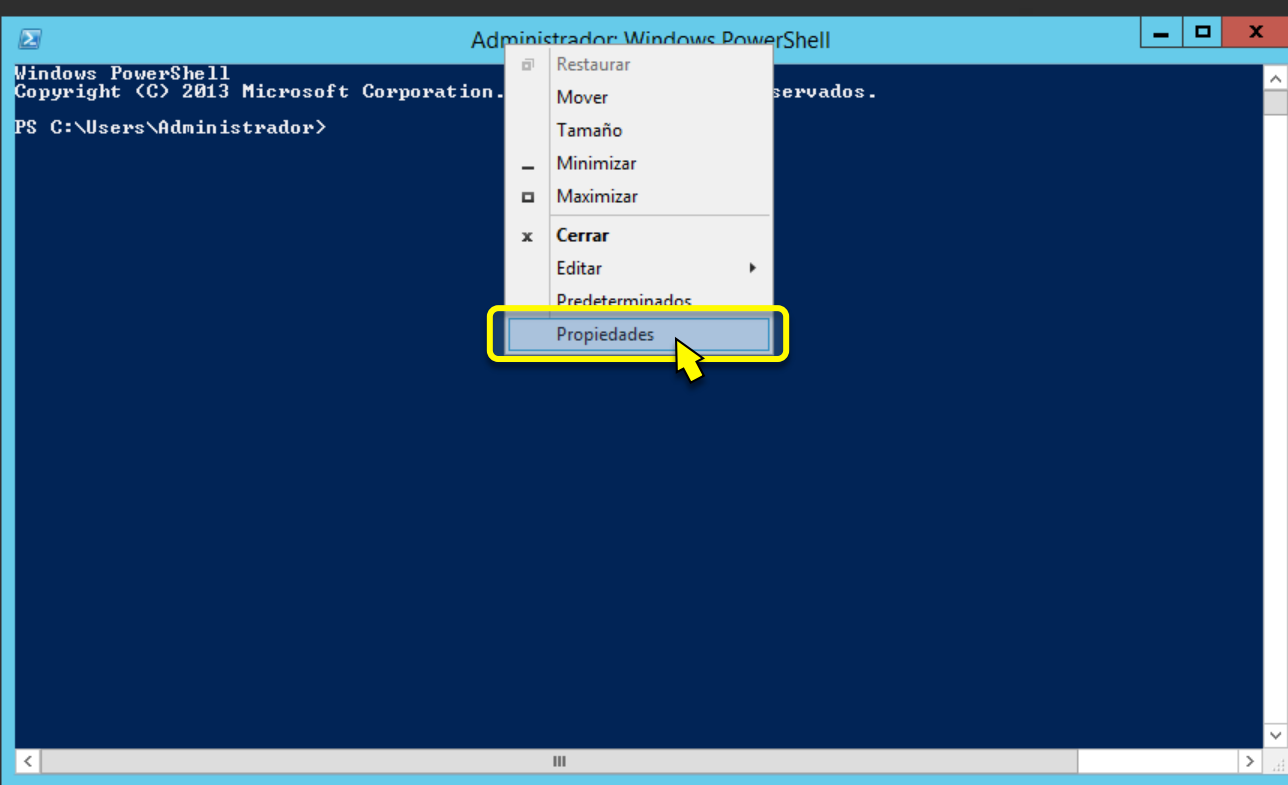
Cómo ejecutar la consola



Botón derecho + "Ejecutar como administrador"

012 R2

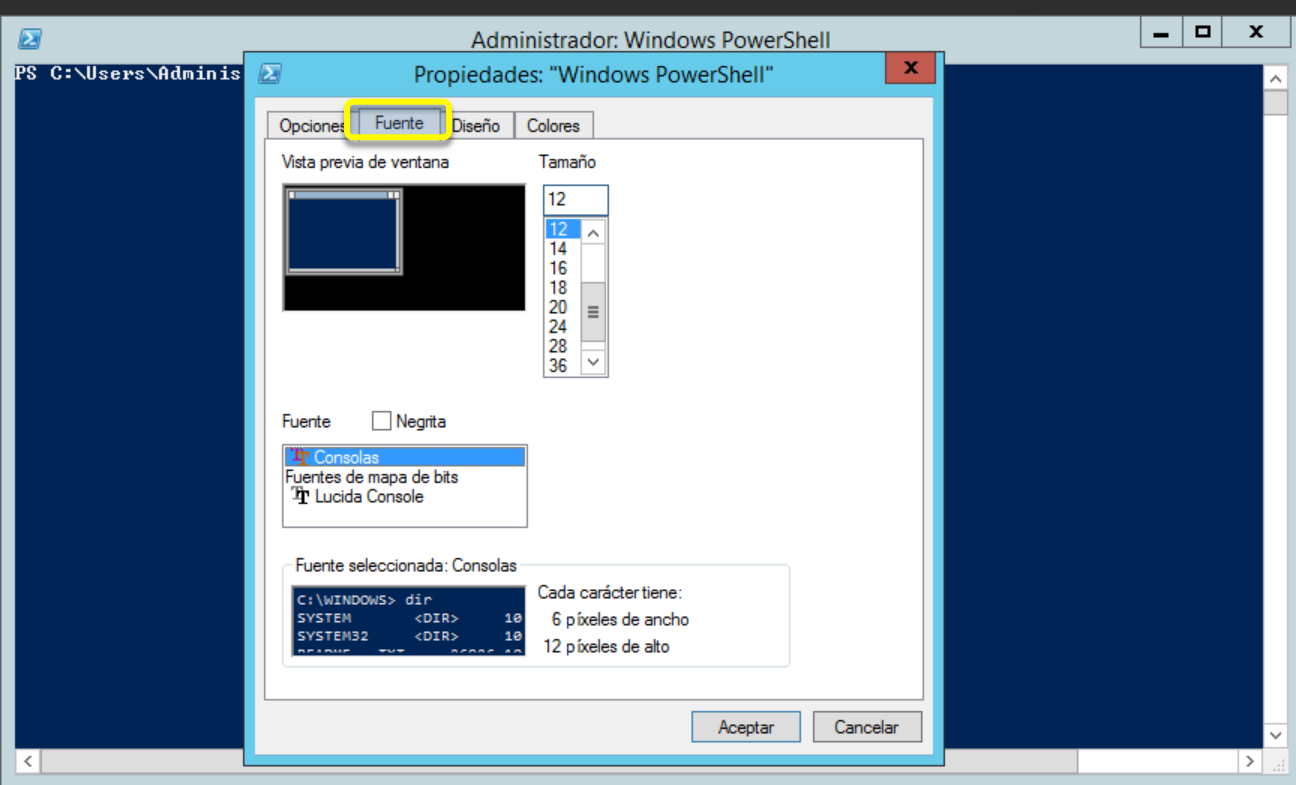
Cómo ejecutar la consola



Windows Server 2012 R2

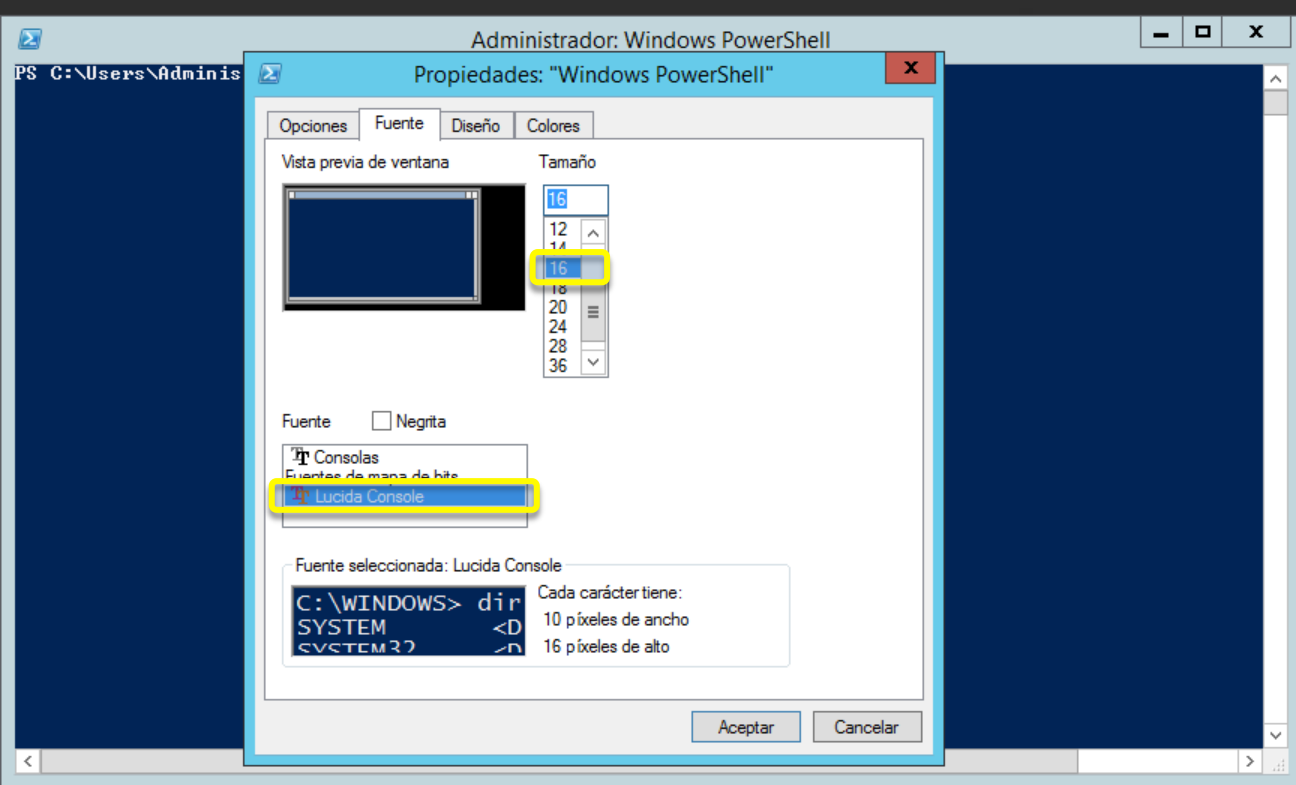


Propiedades de la consola



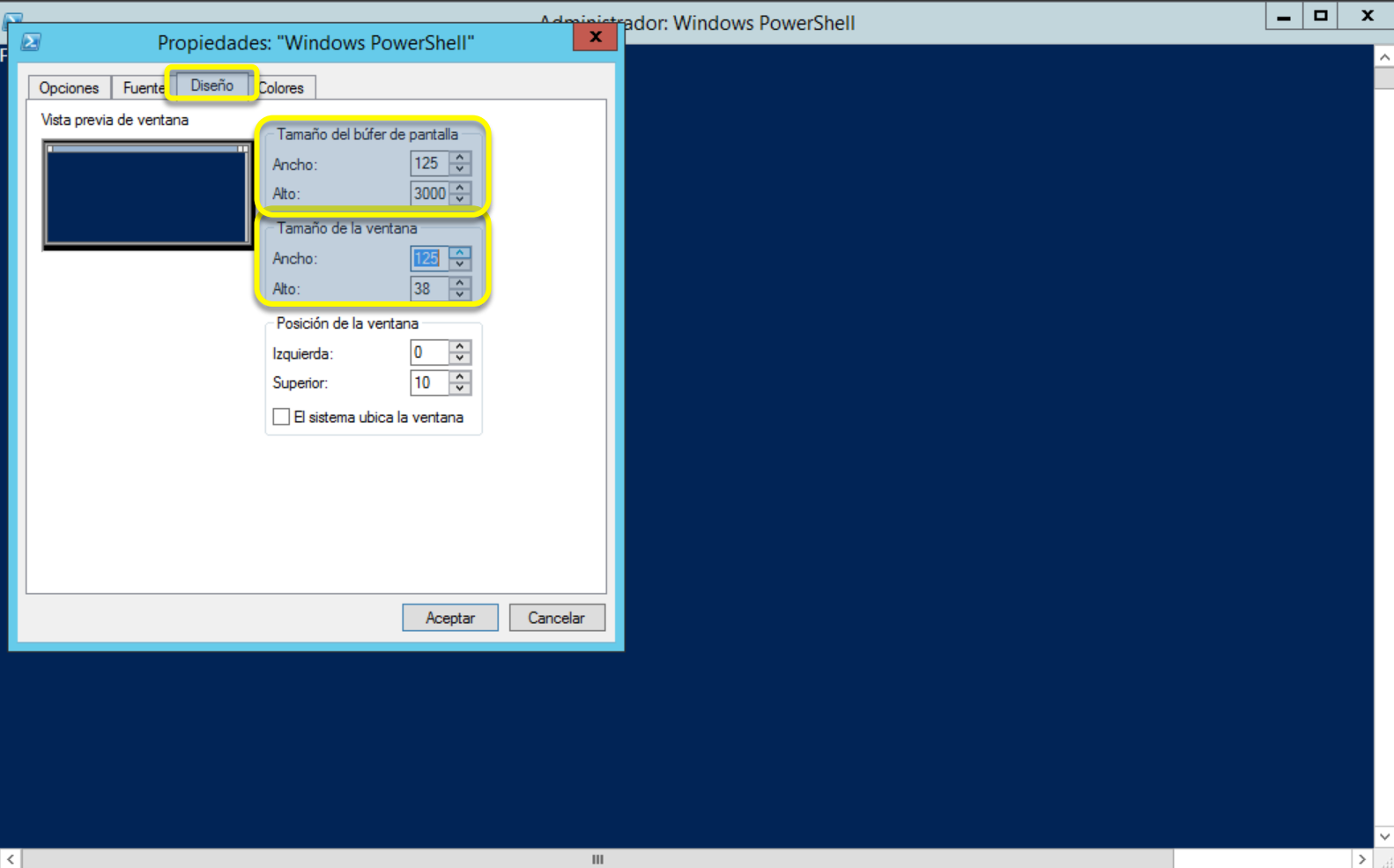
Windows Server 2012 R2

Propiedades de la consola



Windows Server 2012 R2

Propiedades de la consola



```
PS C:\Users\Administrador> holamundo
holamundo : El término 'holamundo' no se reconoce como nombre de un cmdlet, función, archivo de script o programa
ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es
correcta e inténtelo de nuevo.
En línea: 1 Carácter: 1
+ holamundo
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (holamundo:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Administrador> get-date

viernes, 9 de enero de 2015 12:39:34

PS C:\Users\Administrador> get-location

Path
----
C:\Users\Administrador

PS C:\Users\Administrador> set-location c:\
PS C:\>
PS C:\> get-location

Path
----
C:\

PS C:\> ■
```

```
+ CategoryInfo          : ObjectNotFound: (holamundo:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

```
PS C:\Users\Administrador> get-date
```

```
viernes, 9 de enero de 2015 12:39:34
```

```
PS C:\Users\Administrador> get-location
```

```
Path
```

```
----
```

```
C:\Users\Administrador
```

```
PS C:\Users\Administrador> set-location c:\
```

```
PS C:\>
```

```
PS C:\> get-location
```

```
Path
```

```
----
```

```
C:\
```

```
PS C:\> get-childitem
```

```
Directorio: C:\
```

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
d----	22/08/2013	17:52		PerfLogs
d-r--	19/09/2014	14:09		Program Files
d----	19/09/2014	14:09		Program Files (x86)
d-r--	17/09/2014	14:38		Users
d----	19/09/2014	14:09		Windows

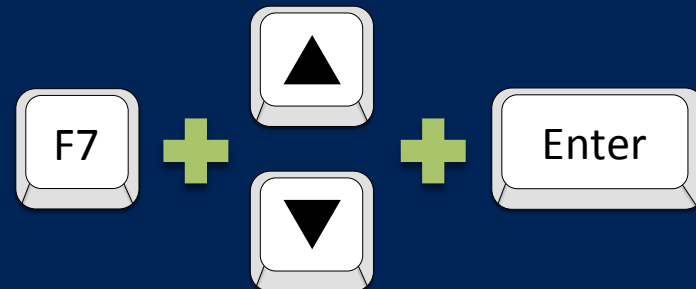
```
PS C:\> clear-host
```



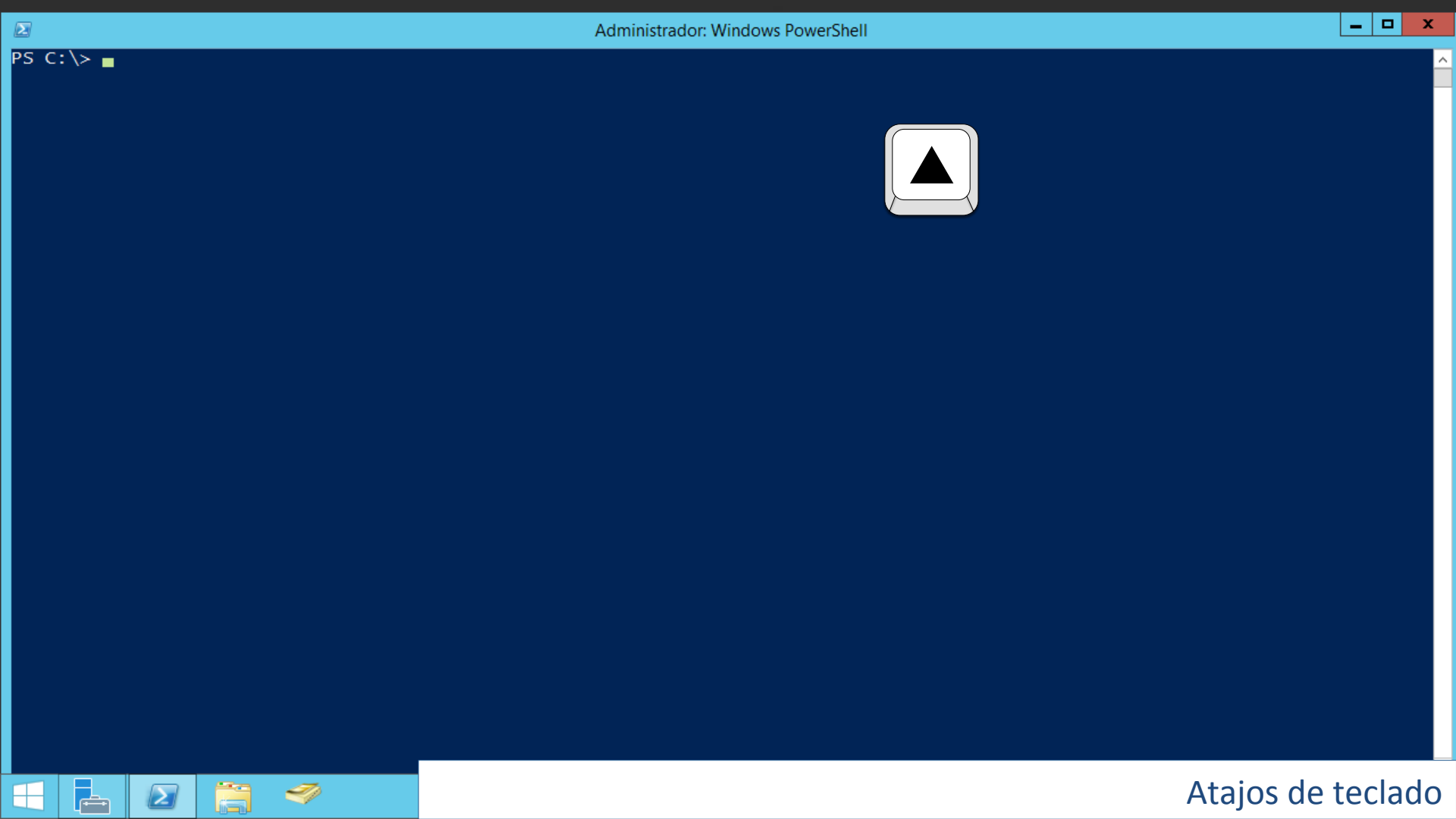
PS C:\>



PS C:\>

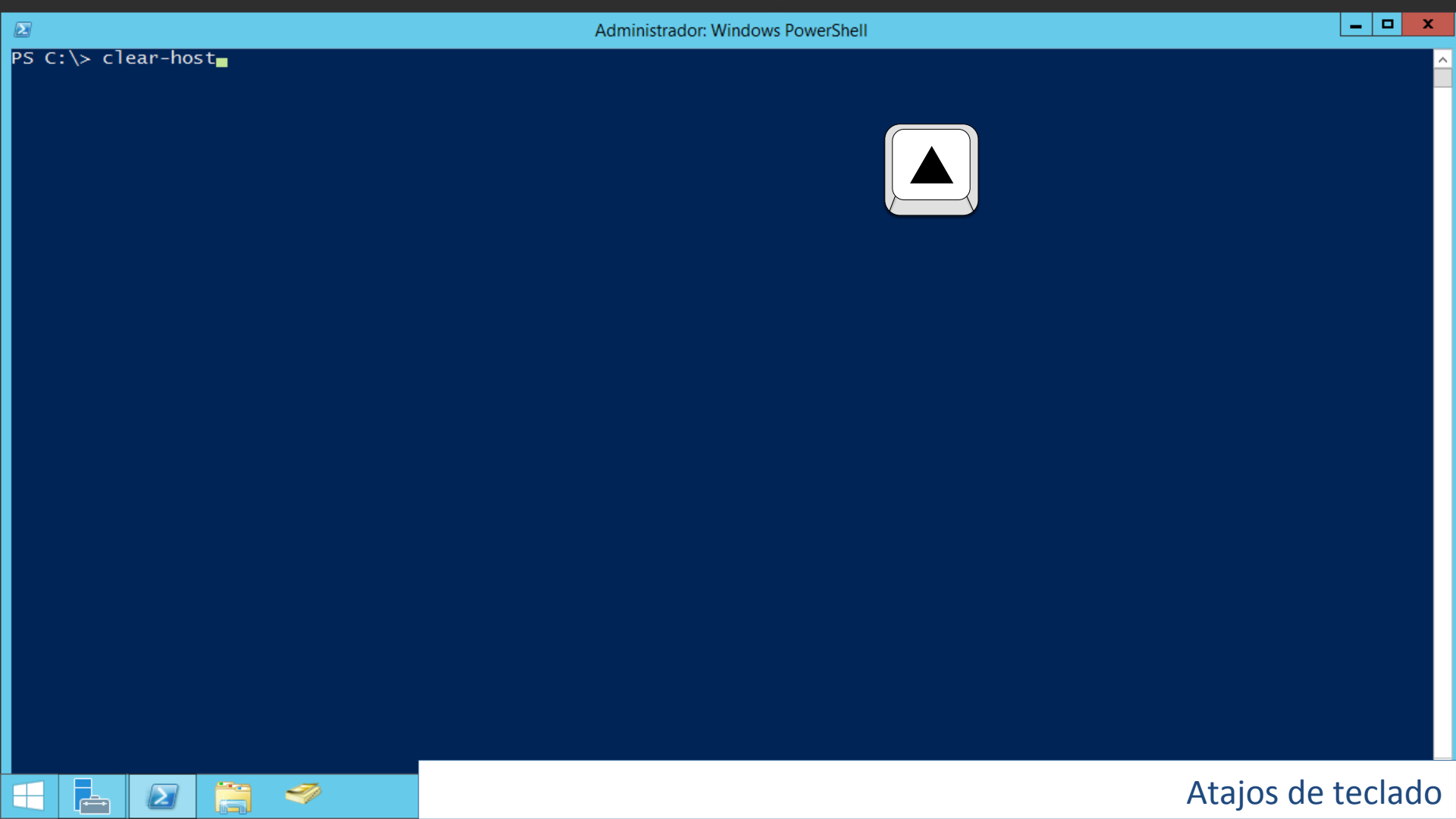


```
1: clear
2: dir
3: clear
4: holamundo
5: get-date
6: get-location
7: set-location c:\
8: get-location
9: get-childitem
10: clear-host
```

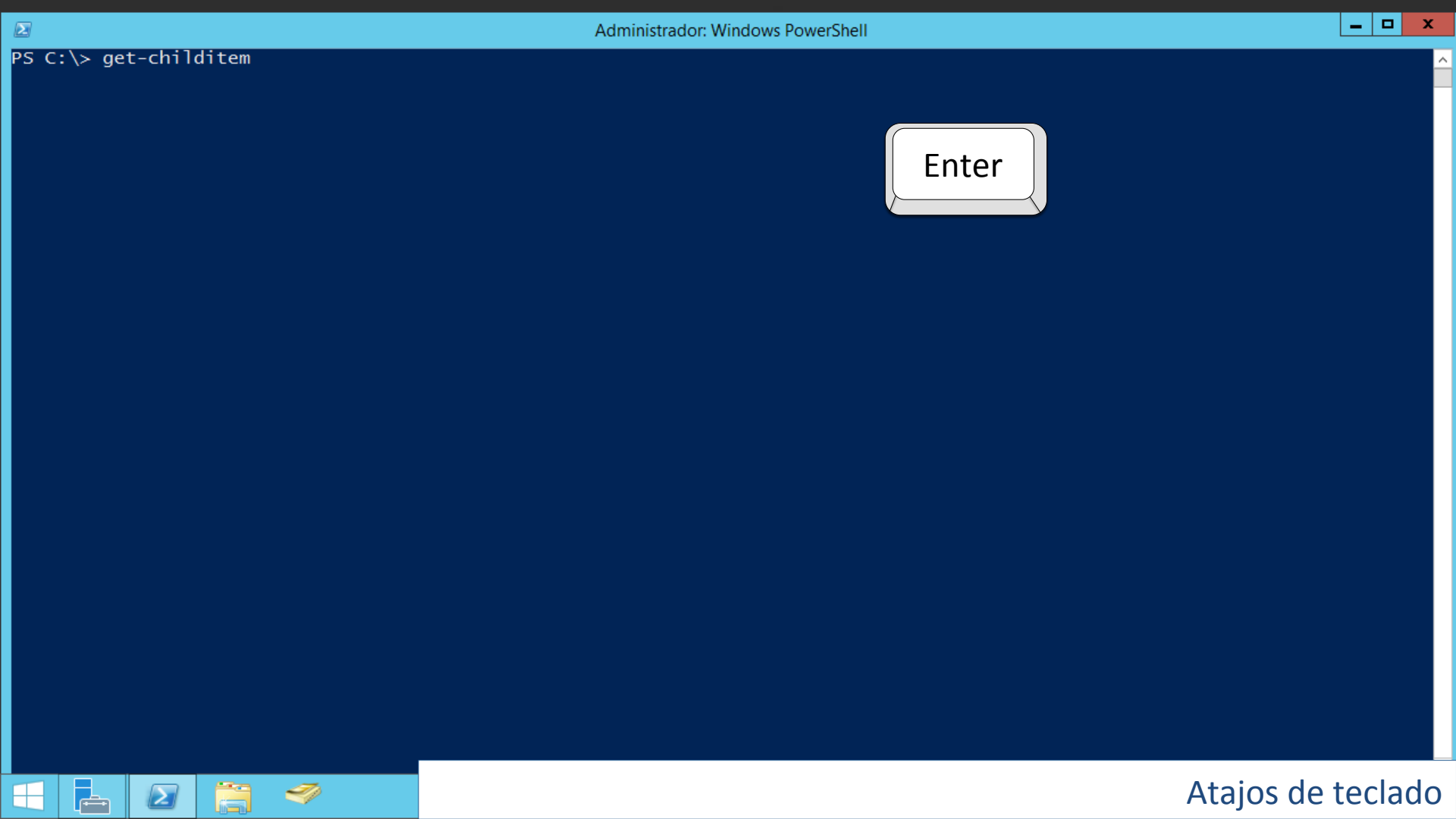
PS C:\>





PS C:\> clear-host





PS C:\> get-childitem



PS C:\> get-childitem

Directorio: C:\

Mode	LastWriteTime		Length	Name
-----	-----		-----	-----
d----	22/08/2013	17:52		PerfLogs
d-r--	19/09/2014	14:09		Program Files
d----	19/09/2014	14:09		Program Files (x86)
d-r--	17/09/2014	14:38		Users
d----	19/09/2014	14:09		Windows

PS C:\> █

```
PS C:\> get-childitem
```

Directorio: C:\

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	-----
d----	22/08/2013	17:52		PerfLogs
d-r--	19/09/2014	14:09		Program Files
d----	19/09/2014	14:09		Program Files (x86)
d-r--	17/09/2014	14:38		Users
d----	19/09/2014	14:09		Windows

Tab

```
PS C:\> get-l■
```

PS C:\> get-childitem

Directorio: C:\

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	-----
d----	22/08/2013	17:52		PerfLogs
d-r--	19/09/2014	14:09		Program Files
d----	19/09/2014	14:09		Program Files (x86)
d-r--	17/09/2014	14:38		Users
d----	19/09/2014	14:09		Windows

Enter

PS C:\> Get-Location

PS C:\> get-childitem

Directorio: C:\

Mode	LastWriteTime		Length	Name
d----	22/08/2013	17:52		PerfLogs
d-r--	19/09/2014	14:09		Program Files
d----	19/09/2014	14:09		Program Files (x86)
d-r--	17/09/2014	14:38		Users
d----	19/09/2014	14:09		Windows

PS C:\> Get-Location

Path

C:\

PS C:\> █

```
PS C:\> Write-Host "Hola Mundo"
Hola Mundo
PS C:\>
PS C:\>
PS C:\>
PS C:\> Write-Host "Hola
```

```
>>
```



```
PS C:\> Write-Host "Hola Mundo"
Hola Mundo
PS C:\>
PS C:\>
PS C:\>
PS C:\> Write-Host "Hola
>> mundo
>> "
>>
>>
Hola
mundo
PS C:\> ■
```



```
PS C:\> Write-Host "Hola Mundo"
Hola Mundo
PS C:\>
PS C:\>
PS C:\>
PS C:\> Write-Host "Hola
>> mundo
>> "
>>
Hola
mundo
PS C:\> Set-Location `
>>
```



```
PS C:\> Write-Host "Hola Mundo"
Hola Mundo
PS C:\>
PS C:\>
PS C:\>
PS C:\> Write-Host "Hola
>> mundo
>>
>> "
>>
Hola
mundo
PS C:\> Set-Location `
>>     c:\users
>>
PS C:\users>
```

Y para terminar...

- Ajustar las **propiedades** de la consola a vuestro gusto
- Practicar con los **atajos de teclado** que hemos visto:
 - F7
 - Cursores
 - Tabulador



1 Inicia sesión como Administrador y ejecuta la consola de PowerShell mediante el acceso directo de la barra de tareas. A continuación, configura algunas de las propiedades de la consola. Comienza por seleccionar el estilo y tamaño de letra que prefiera. A continuación, ajusta el tamaño del búfer de pantalla y la ventana para que ocupen exactamente el escritorio, sin solaparse con la barra de tareas. Cierra la consola y vuelve a abrirla, y contesta: ¿las propiedades que has ajustado se mantienen entre sesiones de la consola o no?.

2 Ejecuta los siguientes cmdlets en la consola de PowerShell, en orden, y marca con una "X" aquellos que provocan un error.

1	get-location	
2	getlocation	
3	Get-Location	
4	set-location	
5	clearhost	
6	get-childitem c:\	
7	get-childitem c:/	
8	get-childitem abcdefghijk	

3 ¿Cómo has sabido cuáles de los cmdlets del ejercicio anterior eran erróneos? Explica brevemente el motivo por el cual los erróneos han fallado.

4 Vuelve a ejecutar los cmdlets no erróneos del ejercicio 2, utilizando los atajos de teclado mencionados en el vídeo. ¿Cuáles son esos atajos?

5 Intenta ejecutar correctamente el siguiente cmdlet

```
get-childitem c:\Windows
```

en dos líneas, ubicando el nombre del cmdlet (`get-childitem`) en la primera, y el argumento (`c:\Windows`) en la segunda. Explica brevemente cómo has podido conseguirlo.

WINDOWS POWERSHELL

CMDLETS Y ALIAS

Contenido



1. *Cmdlets*
2. Alias
3. Algunos *cmdlets* sencillos

Contenido



1. *Cmdlets*

2. Alias

3. Algunos *cmdlets* sencillos

Cmdlets

1. Cmdlets

- Un *cmdlet* es una orden propia o nativa de Windows Powershell
- Se **nombra** siempre mediante la convención

<Verbo>-<Nombre>

- como por ejemplo:

- ▣ Get-Date
- ▣ Set-Location
- ▣ Write-Host

PowerShell no distingue entre mayúsculas y minúsculas

(Get-Date = get-date = GET-DATE)

```
PS C:\> get-command
```

CommandType	Name	ModuleName
Alias	Add-ProvisionedAppxPackage	Dism
Alias	Add-WindowsFeature	ServerManager
Alias	Apply-WindowsUnattend	Dism
Alias	Expand-IscsiVirtualDisk	IscsiTarget
Alias	Export-DnsServerTrustAnchor	DnsServer
Alias	Flush-Volume	Storage
Alias	Get-GPPermissions	GroupPolicy
Alias	Get-ProvisionedAppxPackage	Dism
Alias	Initialize-Volume	Storage
Alias	Move-SmbClient	SmbWitness
Alias	Reconcile-DhcpServerv4IPRecord	DhcpServer
Alias	Remove-ProvisionedAppxPackage	Dism
Alias	Remove-WindowsFeature	ServerManager
Alias	Set-GPPermissions	GroupPolicy
Alias	Write-FileSystemCache	Storage
Function	A:	
Function	Add-BCDataCacheExtension	BranchCache
Function	Add-DhcpServerInDC	DhcpServer
Function	Add-DhcpServerSecurityGroup	DhcpServer
Function	Add-DhcpServerv4Class	DhcpServer
Function	Add-DhcpServerv4ExclusionRange	DhcpServer
Function	Add-DhcpServerv4Failover	DhcpServer
Function	Add-DhcpServerv4FailoverScope	DhcpServer
Function	Add-DhcpServerv4Filter	DhcpServer
Function	Add-DhcpServerv4Lease	DhcpServer
Function	Add-DhcpServerv4MulticastExclusionRange	DhcpServer
Function	Add-DhcpServerv4MulticastScope	DhcpServer
Function	Add-DhcpServerv4OptionDefinition	DhcpServer
Function	Add-DhcpServerv4Policy	DhcpServer
Function	Add-DhcpServerv4PolicyIPRange	DhcpServer
Function	Add-DhcpServerv4Reservation	DhcpServer
Function	Add-DhcpServerv4Scope	DhcpServer
Function	Add-DhcpServerv4Superscope	DhcpServer
Function	Add-DhcpServerv6Class	DhcpServer
Function	Add-DhcpServerv6ExclusionRange	DhcpServer
Function	Add-DhcpServerv6Lease	DhcpServer

Get-Command: Consulta de todos los *cmdlets*

Cmdlet	Test-Connection	Microsoft.PowerShell.Management
Cmdlet	Test-KdsRootKey	Kds
Cmdlet	Test-ModuleManifest	Microsoft.PowerShell.Core
Cmdlet	Test-NfsMappedIdentity	NFS
Cmdlet	Test-Path	Microsoft.PowerShell.Management
Cmdlet	Test-PSSessionConfigurationFile	Microsoft.PowerShell.Core
Cmdlet	Test-WSMan	Microsoft.WSMan.Management
Cmdlet	Trace-Command	Microsoft.PowerShell.Utility
Cmdlet	Unblock-File	Microsoft.PowerShell.Utility
Cmdlet	Unblock-Tpm	TrustedPlatformModule
Cmdlet	Undo-DtcDiagnosticTransaction	MsDtc
Cmdlet	Undo-Transaction	Microsoft.PowerShell.Management
Cmdlet	Uninstall-ADDSDomainController	ADDSDeployment
Cmdlet	Uninstall-ADServiceAccount	ActiveDirectory
Cmdlet	Unlock-ADAccount	ActiveDirectory
Cmdlet	Unregister-Event	Microsoft.PowerShell.Utility
Cmdlet	Unregister-PSSessionConfiguration	Microsoft.PowerShell.Core
Cmdlet	Unregister-ScheduledJob	PSScheduledJob
Cmdlet	Update-FormatData	Microsoft.PowerShell.Utility
Cmdlet	Update-Help	Microsoft.PowerShell.Core
Cmdlet	Update-List	Microsoft.PowerShell.Utility
Cmdlet	Update-TypeData	Microsoft.PowerShell.Utility
Cmdlet	Update-WIMBootEntry	Dism
Cmdlet	Use-Transaction	Microsoft.PowerShell.Management
Cmdlet	Use-WindowsUnattend	Dism
Cmdlet	Wait-Event	Microsoft.PowerShell.Utility
Cmdlet	Wait-Job	Microsoft.PowerShell.Core
Cmdlet	Wait-Process	Microsoft.PowerShell.Management
Cmdlet	Where-Object	Microsoft.PowerShell.Core
Cmdlet	Write-Debug	Microsoft.PowerShell.Utility
Cmdlet	Write-Error	Microsoft.PowerShell.Utility
Cmdlet	Write-EventLog	Microsoft.PowerShell.Management
Cmdlet	Write-Host	Microsoft.PowerShell.Utility
Cmdlet	Write-Output	Microsoft.PowerShell.Utility
Cmdlet	Write-Progress	Microsoft.PowerShell.Utility
Cmdlet	Write-Verbose	Microsoft.PowerShell.Utility
Cmdlet	Write-Warning	Microsoft.PowerShell.Utility

PS C:\>

PS C:\> get-command -verb write

Si queremos conocer los cmdlets con verbo "write"

CommandType	Name	ModuleName
Alias	Write-FileSystemCache	Storage
Function	Write-DtcTransactionsTraceSession	MsDtc
Function	Write-PrinterNfcTag	PrintManagement
Function	Write-VolumeCache	Storage
Cmdlet	Write-Debug	Microsoft.PowerShell.Utility
Cmdlet	Write-Error	Microsoft.PowerShell.Utility
Cmdlet	Write-EventLog	Microsoft.PowerShell.Management
Cmdlet	Write-Host	Microsoft.PowerShell.Utility
Cmdlet	Write-Output	Microsoft.PowerShell.Utility
Cmdlet	Write-Progress	Microsoft.PowerShell.Utility
Cmdlet	Write-Verbose	Microsoft.PowerShell.Utility
Cmdlet	Write-Warning	Microsoft.PowerShell.Utility

PS C:\> get-command -noun process

Si queremos conocer los cmdlets con nombre 'process'

CommandType	Name	ModuleName
Cmdlet	Debug-Process	Microsoft.PowerShell.Management
Cmdlet	Get-Process	Microsoft.PowerShell.Management
Cmdlet	Start-Process	Microsoft.PowerShell.Management
Cmdlet	Stop-Process	Microsoft.PowerShell.Management
Cmdlet	Wait-Process	Microsoft.PowerShell.Management

PS C:\> get-command get-lo*

Con wildcards o comodines

CommandType	Name	ModuleName
Function	Get-LogProperties	PSDiagnostics
Cmdlet	Get-Location	Microsoft.PowerShell.Management

PS C:\>

Contenido



1. *Cmdlets*

2. **Alias**

3. Algunos *cmdlets* sencillos

Alias

2. Alias

- Son **nombres alternativos** para los **cmdlets**
- Existen **muchos alias integrados**
- Sus **nombres suelen atender a dos motivos:**
 - ▣ **Comodidad:** nombres más cortos (acrónimos)
`gci -> Get-Childitem`
 - ▣ **Histórico:** órdenes equivalentes en DOS, shell de UNIX, ...
`dir -> Get-Childitem`
`ls -> Get-Childitem`

```
PS C:\> get-alias
```

CommandType	Name	ModuleName
-----	----	-----
Alias	% -> ForEach-Object	
Alias	? -> Where-Object	
Alias	ac -> Add-Content	
Alias	asnp -> Add-PSSnapin	
Alias	cat -> Get-Content	
Alias	cd -> Set-Location	
Alias	chdir -> Set-Location	
Alias	clc -> Clear-Content	
Alias	clear -> Clear-Host	
Alias	clhy -> Clear-History	
Alias	cli -> Clear-Item	
Alias	clp -> Clear-ItemProperty	
Alias	cls -> Clear-Host	
Alias	clv -> Clear-Variable	
Alias	cnsn -> Connect-PSSession	
Alias	compare -> Compare-Object	
Alias	copy -> Copy-Item	
Alias	cp -> Copy-Item	
Alias	cpi -> Copy-Item	
Alias	cpp -> Copy-ItemProperty	
Alias	curl -> Invoke-WebRequest	
Alias	cvpa -> Convert-Path	
Alias	dbp -> Disable-PSBreakpoint	
Alias	del -> Remove-Item	
Alias	diff -> Compare-Object	
Alias	dir -> Get-ChildItem	
Alias	dnsn -> Disconnect-PSSession	
Alias	ebp -> Enable-PSBreakpoint	
Alias	echo -> Write-Output	
Alias	epal -> Export-Alias	
Alias	epcsv -> Export-Csv	
Alias	epsn -> Export-PSSession	
Alias	erase -> Remove-Item	
Alias	etsn -> Enter-PSSession	
Alias	exsn -> Exit-PSSession	
Alias	fc -> Format-Custom	

```
Alias      rnp -> Rename-ItemProperty
Alias      rp -> Remove-ItemProperty
Alias      rsnp -> Remove-PSSession
Alias      rsn -> Remove-PSSnapin
Alias      rujb -> Resume-Job
Alias      rv -> Remove-Variable
Alias      rvpa -> Resolve-Path
Alias      rwmi -> Remove-WmiObject
Alias      sajb -> Start-Job
Alias      sal -> Set-Alias
Alias      saps -> Start-Process
Alias      sasv -> Start-Service
Alias      sbp -> Set-PSBreakpoint
Alias      sc -> Set-Content
Alias      select -> Select-Object
Alias      set -> Set-Variable
Alias      shcm -> Show-Command
Alias      si -> Set-Item
Alias      sl -> Set-Location
Alias      sleep -> Start-Sleep
Alias      sls -> Select-String
Alias      sort -> Sort-Object
Alias      sp -> Set-ItemProperty
Alias      spjb -> Stop-Job
Alias      spps -> Stop-Process
Alias      spsv -> Stop-Service
Alias      start -> Start-Process
Alias      sujb -> Suspend-Job
Alias      sv -> Set-Variable
Alias      swmi -> Set-WmiInstance
Alias      tee -> Tee-Object
Alias      trcm -> Trace-Command
Alias      type -> Get-Content
Alias      wget -> Invoke-WebRequest
Alias      where -> Where-Object
Alias      wjb -> Wait-Job
Alias      write -> Write-Output
```

PS C:\>


```
PS C:\> get-alias cd
```

CommandType	Name	ModuleName
Alias	cd -> Set-Location	

```
PS C:\> get-alias pwd
```

CommandType	Name	ModuleName
Alias	pwd -> Get-Location	

```
PS C:\> get-alias cat
```

CommandType	Name	ModuleName
Alias	cat -> Get-Content	

```
PS C:\>
```

```
PS C:\>
```

```
PS C:\> get-alias -Definition get-childitem
```

CommandType	Name	ModuleName
Alias	dir -> Get-ChildItem	
Alias	gci -> Get-ChildItem	
Alias	ls -> Get-ChildItem	

```
PS C:\>
```

Consultar todos los alias que apuntan al cmdlet "get-childitem"



Contenido



1. *Cmdlets*

2. Alias

3. Algunos *cmdlets* sencillos

PS C:\> new-item -itemtype directory C:\pruebas

Crear directorio

Directorio: C:\

Mode	LastWriteTime	Length	Name
d----	10/01/2015 18:45		pruebas

PS C:\> Get-ChildItem c:\

dir / ls

Directorio: C:\

Mode	LastWriteTime	Length	Name
d----	22/08/2013 17:52		PerfLogs
d-r--	19/09/2014 14:09		Program Files
d----	19/09/2014 14:09		Program Files (x86)
d----	10/01/2015 18:45		pruebas
d-r--	17/09/2014 14:38		Users
d----	19/09/2014 14:09		Windows

PS C:\> set-location c:\pruebas

PS C:\pruebas> new-item -itemtype file .\prueba1.txt

cd directorio

Directorio: C:\pruebas

Mode	LastWriteTime	Length	Name
-a---	10/01/2015 18:46	0	prueba1.txt

PS C:\pruebas>

```
PS C:\pruebas> copy-item .\prueba1.txt .\prueba2.txt
PS C:\pruebas> Get-ChildItem .
```

copiar ficheros

Directorio: C:\pruebas

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
-a---	10/01/2015	18:46	0	prueba1.txt
-a---	10/01/2015	18:46	0	prueba2.txt

```
PS C:\pruebas> rename-item prueba2.txt nueva_prueba2.txt
PS C:\pruebas> Get-ChildItem .
```

renombrar ficheros

Directorio: C:\pruebas

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
-a---	10/01/2015	18:46	0	nueva_prueba2.txt
-a---	10/01/2015	18:46	0	prueba1.txt

```
PS C:\pruebas> Invoke-Item prueba1.txt
PS C:\pruebas> █
```

Abrir fichero

Invoke-Item: Ejecutar la acción predeterminada para un ítem

```
PS C:\pruebas> copy-item .\prueba1.txt .\prueba2.txt
PS C:\pruebas> Get-ChildItem .
```

Directorio: C:\pruebas

Mode

```
-----
-a---      10
-a---      10
```

```
PS C:\pruebas>
PS C:\pruebas>
```

Directorio:

Mode

```
-----
-a---      10
-a---      10
```

```
PS C:\pruebas>
PS C:\pruebas>
PS C:\pruebas>
```

prueba1: Bloc de notas

Archivo Edición Formato Ver Ayuda

Invoke-Item: Ejecutar la acción predeterminada para un ítem

```
PS C:\pruebas> remove-item .\nueva_prueba2.txt  
PS C:\pruebas> Get-ChildItem .
```

**Borrar item**

Directorio: C:\pruebas

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a---	10/01/2015 18:46	0	prueba1.txt

```
PS C:\pruebas> █
```

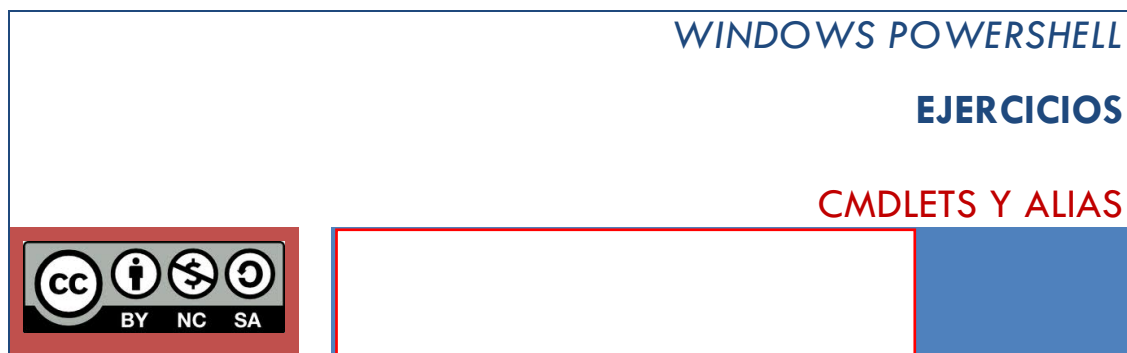
```
PS C:\pruebas> get-content .\prueba1.txt
PS C:\pruebas> add-content prueba1.txt "Lorem ipsum dolor sit amet"
PS C:\pruebas> get-content prueba1.txt
Lorem ipsum dolor sit amet
PS C:\pruebas> Clear-Content prueba1.txt
PS C:\pruebas>
```

Mostrar por pantalla contenido de fichero y luego añadir contenido para luego borrar el contenido del mismo

Y para terminar...

- Practicar con los **cmdlets** y **alias** vistos en la presentación
- Para cualquier duda sobre un cmdlet, o para ampliar información:

```
get-help <cmdlet>
```

1 Utilizando `Get-Command`, escribe un cmdlet que resuelva cada uno de los casos siguientes en la consola:

- a) Consultar todos los comandos PowerShell disponibles en el sistema.
- b) Consultar los comandos que comiencen por “update”.
- c) Consultar los comandos cuyo verbo de acción sea “set”.
- d) Consultar los comandos cuyo nombre de objeto sea “process”.

2 Uno de los parámetros de `Get-Command` es “-CommandType”, que permite filtrar en la consulta los diferentes tipos de comandos de PowerShell (`cmdlet`, `alias`, `function`). Utiliza este parámetro para repetir las dos primeras búsquedas del ejercicio anterior, pero de forma que sólo se recuperen los cmdlets.

3 Utilizando `Get-Alias`, escribe un cmdlet que resuelva cada uno de los casos siguientes en la consola:

- a) Consultar todos los alias disponibles en el sistema.
- b) Consultar los alias que comiencen por “c”.
- c) Consultar todos los alias del cmdlet “Clear-Host”.
- d) Consultar el cmdlet cuyo alias es “copy”

4 Escribe una secuencia de cmdlets que resuelva la siguiente secuencia de acciones:

- a) Situar en “c:\”
- b) Crear una carpeta denominada “c:\Datos”.
- c) Situar en “c:\Datos”.
- d) Crear un fichero vacío denominado “inf1.txt”.
- e) Añadir al fichero anterior el contenido “IES Arcipreste de Hita - ASIR1”.
- f) Copiar el fichero anterior a otro denominado “inf2.txt”.
- g) Borrar el fichero “inf1.txt”.

- h) Visualizar por pantalla el contenido del fichero "inf2.txt".
- i) Ejecutar la acción por defecto sobre el fichero "inf2.txt" (abrir el bloc de notas), y modificar el contenido del fichero como prefiera.
- j) Renombrar el fichero "inf2.txt" como "inf.txt".
- k) Visualizar por pantalla el fichero "inf.txt".
- l) Borrar el fichero "inf.txt".
- m) Situar en "c: \"
- n) Eliminar la carpeta "c:\Datos".

5 Averigua posibles alias para los cmdlets utilizados en el ejercicio anterior, y repite la misma secuencia de acciones utilizando esos alias en lugar de los cmdlets originales.

WINDOWS POWERSHELL

EL SISTEMA DE AYUDA

Contenido



1. El sistema de ayuda de Powershell
2. Usando la ayuda

Contenido



1. El sistema de ayuda de Powershell

2. Usando la ayuda

Características

1. El sistema de ayuda de Powershell

- ❑ Sistema nativo ofrecido desde la **consola**
 - ❑ Interfaz **consistente**
 - ❑ Basado en el ***cmdlet* Get-HeIp**
- ❑ Información **completa**, incluyendo
 - ❑ Interfaz (API): ***cmdlets*, alias, funciones,...**
 - ❑ **Conceptos y características**
- ❑ Ayuda **local o en línea, siempre actualizada**

Contenido



1. El sistema de ayuda de Powershell

2. Usando la ayuda

PS C:\> update-help

Actualizando Ayuda para el módulo Microsoft.PowerShell.Utility

Buscando contenido de Ayuda...

[oo]

Lo primero, actualizar los archivos locales con Update-Help


```
PS C:\> get-help
```

TEMA

Sistema de ayuda de Windows PowerShell

DESCRIPCIÓN BREVE

Muestra ayuda acerca de los cmdlets y los conceptos de Windows PowerShell.

DESCRIPCIÓN LARGA

En la ayuda de Windows PowerShell se describen los cmdlets, las funciones, los scripts y los módulos de Windows PowerShell y se explican conceptos, incluidos los elementos del lenguaje Windows PowerShell.

Windows PowerShell no incluye archivos de ayuda, pero puede leer los temas de ayuda en línea o usar el cmdlet Update-Help para descargar archivos de ayuda en el equipo y, a continuación, usar el cmdlet Get-Help para mostrar los temas de ayuda en la línea de comandos.

También puede usar el cmdlet Update-Help para descargar archivos de ayuda actualizados a medida que se publiquen, para que el contenido de ayuda local nunca se quede obsoleto.

Sin archivos de ayuda, Get-Help obtiene ayuda generada automáticamente para los cmdlets, las funciones y los scripts.

AYUDA EN LÍNEA

Encontrará ayuda en línea para Windows PowerShell en la Biblioteca de TechNet, a partir de <http://go.microsoft.com/fwlink/?LinkID=108518>.

Para abrir la ayuda en línea de cualquier cmdlet o función, escriba:

```
Get-Help <nombreDelCmdlet> -Online
```

UPDATE-HELP

Para descargar e instalar los archivos de ayuda en el equipo:

1. Inicie Windows PowerShell con la opción "Ejecutar como administrador".
2. Escriba:

Update-Help

Una vez instalados los archivos de ayuda, puede usar el cmdlet Get-Help para mostrar los temas de ayuda. También puede usar el cmdlet Update-Help para descargar archivos de ayuda actualizados, de modo que los archivos de ayuda locales siempre estén actualizados.

Para obtener más información acerca del cmdlet Update-Help, escriba:

```
Get-Help Update-Help -Online
```

o vaya a: <http://go.microsoft.com/fwlink/?LinkID=210614>

GET-HELP

El cmdlet Get-Help muestra ayuda en la línea de comandos del contenido de los archivos de ayuda del equipo. Sin archivos de ayuda, Get-Help muestra ayuda básica acerca de los cmdlets y las funciones. También puede usar Get-Help para mostrar ayuda en línea sobre los cmdlets y las funciones.

Para obtener ayuda sobre un cmdlet, escriba:

```
Get-Help <nombreDelCmdlet>
```

Para obtener ayuda en línea, escriba:

```
Get-Help <nombreDelCmdlet> -Online
```

Los títulos de los temas conceptuales empiezan con "About_".
Para obtener ayuda sobre un concepto o elemento del lenguaje, escriba:

```
Get-Help About_ <nombreDelTema>
```

Para buscar una palabra o frase en todos los archivos de ayuda, escriba:

```
Get-Help <términoDeBúsqueda>
```

Para obtener más información acerca del cmdlet Get-Help, escriba:

```
Get-Help Get-Help -Online
```

```
o vaya a: http://go.microsoft.com/fwlink/?LinkID=113316
```

EJEMPLOS:

```
Save-Help           : descarga archivos de ayuda de Internet y los guarda  
                     : en un recurso compartido de archivos.  
Update-Help         : descarga e instala archivos de ayuda de  
                     : Internet en un recurso compartido de archivos.  
Get-Help Get-Process : muestra ayuda acerca del cmdlet Get-Process.  
Get-Help Get-Process -Online  
                     : abre la ayuda en línea del cmdlet Get-Process.  
Help Get-Process    : muestra ayuda acerca de Get-Process, página por página.  
Get-Process -?       : muestra ayuda acerca del cmdlet Get-Process.  
Get-Help About_Modules : muestra ayuda acerca de los módulos de Windows PowerShell.  
Get-Help remoting    : busca la palabra "remoting" en los temas de ayuda.
```

VÉASE TAMBIÉN:

```
about_Updatable_Help  
Get-Help  
Save-Help  
Update-Help
```

```
PS C:\> █
```

```
PS C:\> get-help about_history
```

Los conceptos siempre son
términos que comienzan por
“about_”

Get-**Help** para recuperar información sobre un **concepto**

```
PS C:\> get-help about_history
```

```
TOPIC
    about_History
```

SHORT DESCRIPTION

Describes how to get and run commands in the command history.

LONG DESCRIPTION

When you enter a command at the command prompt, Windows PowerShell saves the command in the command history. You can use the commands in the history as a record of your work. And, you can recall and run the commands from the command history.

History Cmdlets

Windows PowerShell has a set of cmdlets that manage the command history.

Cmdlet (Alias)	Description
Get-History (h)	Gets the command history.
Invoke-History (r)	Runs a command in the command history.
Add-History	Adds a command to the command history.
Clear-History (clh)	Deletes commands from the command history.

Keyboard Shortcuts for Managing History

In the Windows PowerShell console, you can use the following shortcuts to manage the command history.

For other host applications, see the product documentation.

Use this key	To perform this action
UP ARROW	Displays the previous command

DOWN ARROW	Displays the next command.
F7	Displays the command history. To hide the history, press ESC.
F8	Finds a command. Type one or more characters, and then press F8. For the next instance, press F8 again.
F9	Find a command by history ID. Type the history ID, and then press F9. To find the ID, press F7.

MaximumHistoryCount

The `$MaximumHistoryCount` preference variable determines the maximum number of commands that Windows PowerShell saves in the command history. The default value is 4096, meaning that Windows PowerShell saves the 4096 most recent commands, but you can change the value of the variable.

For example, the following command lowers the `$MaximumHistoryCount` to 100 commands:

```
$MaximumHistoryCount = 100
```

To apply the setting, restart Windows PowerShell.

To save the new variable value for all your Windows PowerShell sessions, add the assignment statement to a Windows PowerShell profile. For more information about profiles, see `about_Profiles` (<http://go.microsoft.com/fwlink/?LinkID=113729>).

For more information about the `$MaximumHistoryCount` preference variable, see `about_Preference_Variables` (<http://go.microsoft.com/fwlink/?LinkID=113248>).

NOTE: In Windows PowerShell 2.0, the default value of the `$MaximumHistoryCount` preference variable is 64.

Order of Commands in the History

Commands are added to the history when the command finishes executing

To save the new variable value for all your Windows PowerShell sessions, add the assignment statement to a Windows PowerShell profile. For more information about profiles, see `about_Profiles` (<http://go.microsoft.com/fwlink/?LinkID=113729>).

For more information about the `$MaximumHistoryCount` preference variable, see `about_Preference_Variables` (<http://go.microsoft.com/fwlink/?LinkID=113248>).

NOTE: In Windows PowerShell 2.0, the default value of the `$MaximumHistoryCount` preference variable is 64.

Order of Commands in the History

Commands are added to the history when the command finishes executing, not when the command is entered. If commands take some time to be completed, or if the commands are executing in a nested prompt, the commands might appear to be out of order in the history. (Commands that are executing in a nested prompt are completed only when you exit the prompt level.)

SEE ALSO

- `about_Line_Editing`
- `about_Preference_Variables`
- `about_Profiles`
- `about_Variables`

PS C:\>

```
PS C:\> get-help about_*
```

Name	Category	Module	Synopsis
-----	-----	-----	-----
about_Aliases	HelpFile		Describes how to use alternate names for cmdlets an...
about_Arithmetic_Operators	HelpFile		Describes the operators that perform arithmetic in ...
about_Arrays	HelpFile		Describes arrays, which are data structures designe...
about_Assignment_Operators	HelpFile		Describes how to use operators to assign values to ...
about_Automatic_Variables	HelpFile		Describes variables that store state information fo...
about_Break	HelpFile		Describes a statement you can use to immediately ex...
about_Classes	HelpFile		Describes how you can use classes to develop in Win...
about_Command_Precedence	HelpFile		Describes how Windows PowerShell determines which c...
about_Command_Syntax	HelpFile		Describes the syntax diagrams that are used in Wind...
about_Comment_Based_Help	HelpFile		Describes how to write comment-based help topics fo...
about_CommonParameters	HelpFile		Describes the parameters that can be used with any ...
about_Comparison_Operators	HelpFile		Describes the operators that compare values in Wind...
about_Continue	HelpFile		Describes how the Continue statement immediately re...
about_Core_Commands	HelpFile		Lists the cmdlets that are designed for use with Wi...
about_Data_Sections	HelpFile		Explains Data sections, which isolate text strings ...
about_Debuggers	HelpFile		Describes the Windows PowerShell debugger.
about_DesiredStateConfiguration	HelpFile		Provides a brief introduction to the Windows
about_Do	HelpFile		Runs a statement list one or more times, subject to...
about_Environment_Variables	HelpFile		Describes how to access Windows environment variabl...
about_Escape_Characters	HelpFile		Introduces the escape character in Windows PowerShe...
about_Eventlogs	HelpFile		Windows PowerShell creates a Windows event log that is
about_Execution_Policies	HelpFile		Describes the Windows PowerShell execution policies...
about_For	HelpFile		Describes a language command you can use to run sta...
about_Foreach	HelpFile		Describes a language command you can use to travers...
about_Format.pslxml	HelpFile		The Format.pslxml files in Windows PowerShell defin...
about_Functions	HelpFile		Describes how to create and use functions in Window...
about_Functions_Advanced	HelpFile		Introduces advanced functions that act similar to c...
about_Functions_Advanced_Methods	HelpFile		Describes how functions that specify the CmdletBind...
about_Functions_Advanced_Param...	HelpFile		Explains how to add parameters to advanced functions.
about_Functions_CmdletBindingA...	HelpFile		Describes the attribute that makes a function work ...
about_Functions_OutputTypeAttr...	HelpFile		Describes an attribute that reports the type of obj...
about_Group_Policy_Settings	HelpFile		Describes the Group Policy settings for Windows Pow...
about_Hash_Tables	HelpFile		Describes how to create, use, and sort hash tables ...
about_History	HelpFile		Describes how to get and run commands in the comman...
about_If	HelpFile		Describes a language command you can use to run sta...
about_Jobs	HelpFile		Provides information about how Windows PowerShell h


```
PS C:\> get-help get-childitem
```

```
more
```

NOMBRE

```
Get-ChildItem
```

SINOPSIS

```
Gets the items and child items in one or more specified locations.
```

SINTAXIS

```
Get-ChildItem [[-Path] <String[]>] [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] [-UseTransaction [<SwitchParameter>]] [<CommonParameters>]
```

```
Get-ChildItem [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] -LiteralPath <String[]> [-UseTransaction [<SwitchParameter>]] [<CommonParameters>]
```

DESCRIPCIÓN

The Get-ChildItem cmdlet gets the items in one or more specified locations. If the item is a container, it gets the items inside the container, known as child items. You can use the Recurse parameter to get items in all child containers.

A location can be a file system location, such as a directory, or a location exposed by a different Windows PowerShell provider, such as a registry hive or a certificate store.

VÍNCULOS RELACIONADOS

Online Version: <http://go.microsoft.com/fwlink/p/?linkid=290488>

Get-Item

Get-Location

Get-Process

about_Providers

NOTAS

Para ver los ejemplos, escriba: "get-help Get-ChildItem -examples".

Para obtener más información, escriba: "get-help Get-ChildItem -detailed".

Para obtener información técnica, escriba: "get-help Get-ChildItem -full".

Para obtener ayuda disponible en línea, escriba: "get-help Get-ChildItem -online"

Si no cabe en una pantalla...
"he1p", "man"

```
PS C:\> get-help get-childitem -examples
```

NOMBRE

Get-ChildItem

SINOPSIS

Gets the items and child items in one or more specified locations.

----- EXAMPLE 1 -----

```
PS C:\>Get-ChildItem
```

This command gets the child items in the current location. If the location is a file system directory, it gets the files and sub-directories in the current directory. If the item does not have child items, this command returns to the command prompt without displaying anything.

The default display lists the mode (attributes), last write time, file size (length), and the name of the file. The valid values for mode are d (directory), a (archive), r (read-only), h (hidden), and s (system).

----- EXAMPLE 2 -----

```
PS C:\>Get-ChildItem -Path *.txt -Recurse -Force
```

This command gets all of the .txt files in the current directory and its subdirectories. The Recurse parameter directs Windows PowerShell to get objects recursively, and it indicates that the subject of the command is the specified directory and its contents. The Force parameter adds hidden files to the display.

To use the Recurse parameter on Windows PowerShell 2.0 and earlier versions of Windows PowerShell, the value use the Path parameter must be a container. Use the Include parameter to specify the .txt file type. For example, Get-ChildItem -Path .* -Include *.txt -Recurse

----- EXAMPLE 3 -----

```
PS C:\>Get-ChildItem -Path C:\Windows\Logs\* -Include *.txt -Exclude A*
```



```
PS C:\> get-help get-childitem -parameter path
```

```
-Path <String[]>
```

Specifies a path to one or more locations. Wildcards are permitted. The default location is the current directory (.).

¿Requerido?	false
¿Posición?	1
Valor predeterminado	Current directory
¿Aceptar canalización?	true (ByValue, ByPropertyName)
¿Aceptar caracteres comodín?	true

```
PS C:\>
```

```
PS C:\> get-help get-childitem -detailed
```

NOMBRE
Get-ChildItem

SINOPSIS
Gets the items and child items in one or more specified locations.

SINTAXIS

```
Get-ChildItem [[-Path] <String[]>] [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] [-UseTransaction [<SwitchParameter>]] [<CommonParameters>]
```

```
Get-ChildItem [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] -LiteralPath <String[]> [-UseTransaction [<SwitchParameter>]] [<CommonParameters>]
```

DESCRIPCIÓN

The Get-ChildItem cmdlet gets the items in one or more specified locations. If the item is a container, it gets the items inside the container, known as child items. You can use the Recurse parameter to get items in all child containers.

A location can be a file system location, such as a directory, or a location exposed by a different Windows PowerShell provider, such as a registry hive or a certificate store.

PARÁMETROS

-Exclude <String[]>
Omits the specified items. The value of this parameter qualifies the Path parameter. Enter a path element or pattern, such as "*.txt". Wildcards are permitted.

-Filter <String>
Specifies a filter in the provider's format or language. The value of this parameter qualifies the Path parameter. The syntax of the filter, including the use of wildcards, depends on the provider. Filters are more efficient than other parameters, because the provider applies them when retrieving the objects, rather than having Windows PowerShell filter the objects after they are retrieved.

-Force [<SwitchParameter>
Allows the cmdlet to get items that cannot otherwise not be accessed by the user, such as hidden or system files. Implementation varies among providers. For more information, see about Providers

Si aún queremos más...
Get-Help <cmdlet> -Full

```
PS C:\> get-help get-childitem -showwindow
```

```
PS C:\>
```

Ayuda de Get-ChildItem

Buscar: filterAnteriorSiguiente7 coincidenciasConfiguración

Sinopsis

Gets the items and child items in one or more specified locations.

Sintaxis

```
Get-ChildItem [[-Path] <String[]>] [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] [-UseTransaction <SwitchParameter>] [<Parámetros comunes>]
```

```
Get-ChildItem [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] [-LiteralPath <String[]>] [-UseTransaction <SwitchParameter>] [<Parámetros comunes>]
```

Descripción

The Get-ChildItem cmdlet gets the items in one or more specified locations. If the item is a container, it gets the items inside the container, known as child items. You can use the Recurse parameter to get items in all child containers.

A location can be a file system location, such as a directory, or a location exposed by a different Windows PowerShell provider, such as a registry hive or a certificate store.

Parámetros

-Exclude <String[]>

140 %

```
PS C:\> get-help get-childitem -online  
PS C:\>
```

The screenshot shows a web browser window displaying the TechNet website. The address bar shows the URL: <http://technet.microsoft.com/library/75cf79bb-4db6-4a67-8c36-3d2>. The page title is "TN Get-ChildItem". The TechNet logo is visible, along with navigation links: Inicio, Biblioteca (highlighted), Aprendizaje, Descarga, Soporte, Comunidad, and Foros. A search bar is present. The main content area is titled "Biblioteca TechNet" and lists various categories: Deployment, Forefront, optimización de infraestructuras, Internet Explorer, Microsoft Dynamics Products and Technologies, Office Products, Online Services, Seguridad y actualizaciones, Productos y tecnologías de servidor, and Guía de soluciones para profesionales de TI. The right sidebar shows the language "EN" and a message: "Este contenido no se encuentra disponible en su idioma, pero aquí tiene la versión en inglés." The main article is titled "Get-ChildItem" and is for "Windows PowerShell 4.0". It includes the text "Updated: December 3, 2014" and "Applies To: Windows PowerShell 4.0". The description of the command is: "Gets the items and child items in one or more specified locations." The word "Alias" is also visible.

TN <http://technet.microsoft.com/library/75cf79bb-4db6-4a67-8c36-3d2> TN Get-ChildItem

TechNet Productos Recursos de TI Descargas Entrenamiento Soporte técnico

Microsoft España (Español) Iniciar sesión

TechNet

Inicio **Biblioteca** Aprendizaje Descarga Soporte Comunidad Foros

Expandir todo Imprimir

Biblioteca TechNet

- Deployment
- Forefront
- optimización de infraestructuras
- Internet Explorer
- Microsoft Dynamics Products and Technologies
- Office Products
- Online Services
- Seguridad y actualizaciones
- Productos y tecnologías de servidor
- Guía de soluciones para profesionales de TI

EN Este contenido no se encuentra disponible en su idioma, pero aquí tiene la versión en inglés.

Get-ChildItem

Windows PowerShell 4.0 | [Otras versiones](#)

Updated: December 3, 2014

Applies To: Windows PowerShell 4.0

Get-ChildItem

Gets the items and child items in one or more specified locations.

Alias

Y para terminar...

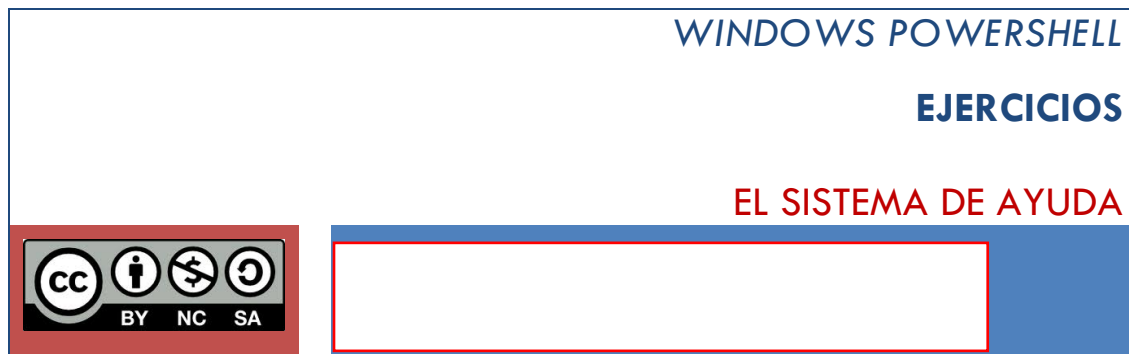
- Practicar con el **sistema de ayuda**:

```
get-help <cmdlet>
```

```
get-help about_*
```

- Como por ejemplo:

```
get-help about_updatable_help
```



1 Ejecuta `Get-Update`, para instalar los ficheros actualizados de ayuda. A partir de ahí, utiliza los cmdlets `Get-Help` y `Get-Command` para averiguar la información necesaria para contestar el resto de las siguientes cuestiones.

2 En PowerShell, los cmdlets que permiten interactuar con los procesos en ejecución en el sistema se nombran con el nombre de objeto "process". Mediante `Get-Command`, averigua el nombre de todos esos cmdlets.

A partir de ahí, utiliza dichos cmdlets para resolver los siguientes ejercicios, consultando la ayuda disponible sobre ellos para saber cuál utilizar en cada caso, y cómo:

- Lanzar a ejecución un proceso que ejecute el programa `notepad.exe` (bloc de notas).
- Obtener el listado de los procesos en ejecución en el sistema.
- Listar exclusivamente el proceso que ejecuta el bloc de notas, utilizando su nombre de proceso (columna "ProcessName" en el listado anterior).
- Finalizar el proceso que ejecuta el bloc de notas, utilizando su identificador interno ("Id").
- Volver a listar todos los procesos en ejecución del sistema, y localizar el que más CPU ha consumido y aquel que está utilizando más memoria virtual (NOTA: consulta la ayuda completa del cmdlet para averiguar el significado de las columnas del listado).

3 Consulta la ayuda sobre el concepto "about_History", y contesta a las siguientes preguntas:

- ¿Qué hace el cmdlet `Get-History`?
- ¿Qué hace el cmdlet `Invoke-History`?
- Averigua los alias, si existen, de los dos cmdlets anteriores.
- Utilizando ambos cmdlets (y/o sus alias), vuelve a ejecutar los cmdlets que has utilizado para resolver el ejercicio 2.
- Indica cuáles son los atajos de teclado que se mencionan en la ayuda del concepto "about_History", y practica brevemente con ellos.

4 Consulta la ayuda sobre el concepto “`about_line_editing`”, y contesta brevemente qué tipo de ayuda incorpora este concepto. ¿Qué se consigue exactamente con las siguientes combinaciones de teclas?

- a) CTRL + Cursor izquierdo.
- b) CTRL + cursor derecho.
- c) CTRL + Fin.
- d) Tabulador

5 Utiliza la ayuda sobre el cmdlet “`New-Alias`” (y en particular, la ayuda que incluye ejemplos de uso) para averiguar cómo crear nuevos alias para un cmdlet existente. A continuación, crea un alias del cmdlet “`Get-Childitem`” denominado “`l`”. Crea otros alias a tu gusto para alguno de los cmdlets que hayas utilizado hasta ahora.

WINDOWS POWERSHELL

SINTAXIS DE LOS MANDATOS

Contenido



1. Sintaxis básica
2. Notación
3. Conjuntos de parámetros

Sintaxis básica

Cmdlet

Parámetro(s): -Param <Valor> -Conmutador

```
New-Alias [-Name] <String> [-Value] <String>  
          [-Description <String>] [-Scope <String>]  
          [-Option {None|ReadOnly|Constant|Private|AllScope}]  
          [-Force] [-PassThru] [-Confirm] [-WhatIf]  
          [<CommonParameters>]
```

-Debug, -ErrorAction, -Verbose, ...
(get-help about_common_parameters)

Notación

{i1|i2|..} = ENUMERACIÓN

[] = OPCIONAL

[-Param <Valor>]

[-Param] <Valor>

```
New-Alias [-Name] <String> [-Value] <String>  
          [-Description <String>] [-Scope <String>]  
          [-Option {None|ReadOnly|Constant|Private|AllScope}]  
          [-Force] [-PassThru] [-Confirm] [-WhatIf]  
          [<CommonParameters>]
```

```
New-Alias ll Get-Childitem -Description "My own dir"  
New-Alias -Description "My own dir" -Value Get-Childitem -Name ll
```

Notación

{i1|i2|..} = ENUMERACIÓN

[] = OPCIONAL

<Valor[]> = COLECCIÓN DE “Valor”

[-Param <Valor>]

[-Param] <Valor>

[[-Param] <Valor>]

```
Get-Alias [-Name] <String[]> [-Exclude <String[]>]  
[-Scope <String>] [<CommonParameters>]
```

```
Get-Alias -Name gci  
Get-Alias gci
```

```
Get-Alias -Name gci, cat, pwd  
Get-Alias gci, cat, pwd
```

Conjuntos de parámetros

Parameter Set: Default

```
Get-Alias [[-Name] <String[]>] [-Exclude <String[]>]  
          [-Scope <String>] [<CommonParameters>]
```

Parameter Set: Definition

```
Get-Alias [-Definition <String[]>] [-Exclude <String[]>]  
          [-Scope <String>] [<CommonParameters>]
```

```
Get-Alias gci
```

```
Get-Alias -Definition Get-ChildItem
```

Y para terminar...

get-help

<cmdlet> -Parameter <param>

about_command_syntax

about_parameters

about_common_parameters

SINTAXIS DE LOS MANDATOS



1 A continuación se presenta la sintaxis del cmdlet `Clear-History`, tal como se muestra en la ayuda en línea de PowerShell:

```
Parameter Set: IDParameter
Clear-History [[-Id] <Int32[]> ] [[-Count] <Int32> ]
[-Newest] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: CommandLineParameter
Clear-History [[-Count] <Int32> ] [-CommandLine <String[]> ]
[-Newest] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Consultando la ayuda siempre que lo necesites, contesta las siguientes preguntas:

- ¿Qué acción realiza este cmdlet en el sistema? Describe brevemente.
- Explica brevemente en qué se diferencia el funcionamiento del cmdlet en ambos conjuntos de parámetros, y pon un ejemplo en cada caso.
- ¿Existe algún parámetro obligatorio en alguno de ambos conjuntos de parámetros? En caso afirmativo, identifícalos. En caso negativo, explica qué consecuencia tiene este hecho.
- Identifica los parámetros considerados conmutadores.
- Clasifica cada uno de los siguientes parámetros en obligatorio u opcional, y en posicional o no posicional: `-Id`, `-Count`, `-CommandLine`, `-Newest`.
- Si invocamos el cmdlet de la siguiente forma:

```
Clear-History 15 5
```

¿Se trata de una invocación correcta o errónea? ¿A qué parámetros corresponderían los valores 15 y 5? Justifica tus respuestas.

- Si invocamos el cmdlet de la siguiente forma:

```
Clear-History 10 -CommandLine get*
```

¿Se trata de una invocación correcta o errónea? ¿A qué parámetro correspondería el valor 10? Justifica tus respuestas.

- Si invocamos el cmdlet de la siguiente forma:

```
Clear-History 20 8 -CommandLine get*
```

¿Se trata de una invocación correcta o errónea? ¿A qué parámetros corresponderían los valores 20 y 8? Justifica tus respuestas.

2 A continuación se presenta la sintaxis del cmdlet `Clear-Content`, tal como se muestra en la ayuda en línea de PowerShell (NOTA: Para simplificar la pregunta, se han eliminado algunos parámetros no relevantes):

```
Parameter Set: Path
Clear-Content [-Path] <String[]> [-Credential <PSCredential> ]
[-Exclude <String[]> ] [-Filter <String> ] [-Force]
[-Include <String[]> ] [-Confirm] [-WhatIf] [-UseTransaction]
[<CommonParameters>]

Parameter Set: LiteralPath
Clear-Content -LiteralPath <String[]> [-Credential <PSCredential> ]
[-Exclude <String[]> ] [-Filter <String> ] [-Force]
[-Include <String[]> ] [-Confirm] [-WhatIf] [-UseTransaction]
[ <CommonParameters>]
```

Consultando la ayuda siempre que lo necesites, contesta las siguientes preguntas:

- ¿Qué acción realiza este cmdlet en el sistema? Describe brevemente.
- Explica brevemente en qué se diferencia el funcionamiento del cmdlet en ambos conjuntos de parámetros, y pon un ejemplo en cada caso.
- ¿Existe algún parámetro obligatorio en alguno de ambos conjuntos de parámetros? En caso afirmativo, identifícalos. En caso negativo, explica qué consecuencia tiene este hecho.
- Identifica los parámetros considerados conmutadores.
- Clasifica cada uno de los siguientes parámetros en obligatorio u opcional, y en posicional o no posicional: `-Path`, `-Credential`, `-LiteralPath`, `-Confirm`.
- Si invocamos el cmdlet de la siguiente forma:

```
Clear-Content c:\Datos\Dato1.txt
```

¿Se trata de una invocación correcta o errónea? ¿A qué parámetro correspondería el valor `"c:\Datos\Dato1.txt"`? Justifica tus respuestas.