



# PowerShell

## 01- Introduktion

I denna laboration kommer vi att starta PowerShell och börja arbeta med de vanligaste kommandona (cmdlets) samt träna på att hitta kommandon och söka i hjälpen. Orden *cmdlets* och *kommandon* kommer att användas frekvent i laborationen och är synonymt med varandra.

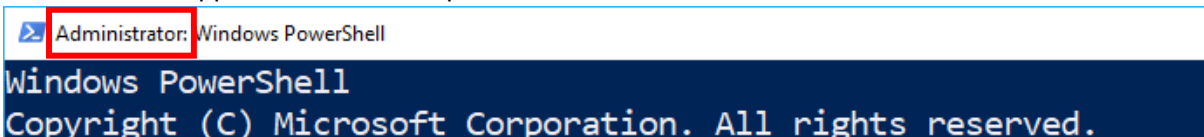
**Antal:** Enskilt

**Material:** En dator med operativsystem Windows 7 (Windows Server 2008) eller senare.

**Tips:** Titta på relevanta genomgångar på webbplatsen <http://itlararen.se/videos.html#video3>

**Utförande:**

1. Starta kommandotolken (**CTRL+R** och skriv **cmd** i kör-dialogrutan). Skriv sedan **powershell** och tryck på **enter**. Observera att prompten ändras och att PowerShell nu körs.
2. Testa valfritt cmdlet, t.ex. **Get-Process**
3. Avsluta PowerShell med kommandot **exit**. Observera att prompten ändras igen. Skriv **exit** och enter igen för att stänga kommandotolken.
4. Prova att starta PowerShell direkt med **CTRL+R** och skriv **powershell** för att starta powershell. Skriv sedan **exit** för att stänga ner PowerShell igen.
5. Många gånger vill man köra PowerShell som administratör. Detta görs enkelt genom att trycka på start-knappen (med muspekaren eller trycka på Windows-tangenten) och sedan söka på *powershell* för att sedan högerklicka på PowerShell-ikonen och välja *Kör som administratör*. Ännu enklare är det i Windows 10: Högerklicka på start-knappen och välj **Windows PowerShell (Administratör)**. Nu startar PowerShell fönstret med lokala administratörs-rättigheter. Detta ser man enkelt då det står *Administrator*: uppe i vänstra hörnet på fönstret.



6. Börja med att uppdatera hjälpfilerna för alla cmdlets och PowerShellmoduler som finns installerat på datorn genom att köra kommandot **Update-Help**  
(Detta kräver Internetåtkomst)
7. Skriv sedan **Get-He** och tryck på **TAB-tangenten**. Observera att TAB-komplettering fungerar bra i PowerShell, både för kommandon (cmdlets) och för parametrar samt ibland även för tillåtna parameter-värden.  
Kör kommandot **Get-Help** och notera resultatet.
8. Kör kommandot **Get-Help Get-Process** för att se hjälpen för kommandot *Get-Process*.
9. Det finns oftast mer hjälp att få om ett kommando än vad som visas i steg 8. Prova kommandot **Get-Help Get-Process -Examples** för att se exempel på hur man kan använda *Get-Process*.
10. Kör kommandot **Get-Help Get-Process -Full** för att se kompletta hjälpen inklusive exempel. Jämför med steg 8.
11. Kör kommandot **Get-Command** som listar alla kommandon som finns tillgängliga bland installerade moduler.



12. Upprepa steg 11 men prova att avbryta kommandot med tangentborskombinationen **CTRL+C**.
13. Kör kommandot **Get-Command \*process\*** för att lista alla kommandon som innehåller ordet *process*
14. Kör kommandot **Get-Command -Verb test** för att visa alla kommandon med verbet *Test*
15. Kör kommandot **Get-Command -Noun \*setting\*** för att visa alla kommandon med ordet *setting* i substantiv-delen.
16. Kör kommandot **Get-Command -Noun \*setting\* -Module NetSecurity** för att visa alla kommandon med ordet *setting* i substantiv-delen i modulen *NetSecurity*
17. Prova att skriva **ipcon** och tryck på **TAB**. Observera att autokompletteringen blir *ipconfig.exe* vilket indikerar att ipconfig är ett gammalt program och inte CMD-kommando eller Powershell-cmdlet. Tryck på **Enter** för att köra ipconfig och kontrollera att det fungerar precis som vanligt. OBS output från äldre program som ipconfig, netstat mm. är i form av *text* och inte objekt.
18. Kör kommandot **dir** och kontrollera att det fungerar.
19. Kör kommandot **ls** och kontrollera att det fungerar.
20. Kör kommandot **pwd** och kontrollera att det fungerar.
21. Som ni ser så fungerar uppenbarligen gamla CMD-kommandon samt vissa Linux-kommandon?  
Kör kommandot **Get-Alias dir** och sedan **Get-Alias ls** så ser vi att även om många gamla CMD-kommandon samt vissa Linux-kommandon är tillgängliga så är de egentligen alias för motsvarande Powershell-cmdlet.  
Kör kommandot **get-alias** för att lista alla alias och studera resultatet.
22. Starta ett notepad-fönster (**CTRL+R** och skriv notepad i kör-dialogrutan).
23. Kör kommandot **Get-Process** och notera resultatet
24. Kör kommandot **Get-Process -Name \*notepad\*** och notera resultatet
25. Kör kommandot **Get-Process -Name \*notepad\* | Stop-Process -Whatif** och notera resultatet
26. Kör kommandot **Get-Process >ps.txt** vad händer? Öppna ps.txt filen och undersök resultatet. Stäng filen igen.
27. Kör kommandot **Get-Process -Name \*notepad\* >>ps.txt** Öppna ps.txt filen och undersök resultatet. Stäng filen och ta bort den.
28. Kör kommandot **Get-Process -Name \*notepad\* | Get-Member**  
Nu visas alla egenskaper, metoder mm. för process-objektet som första kommandot ger.
29. Kör kommandot **Get-Process -Name \*notepad\* | Format-List \***  
Nu ser vi all output om processen.
30. Kör kommandot **Get-Process -Name \*notepad\* | Stop-Process** och kontrollera att det fungerar (processen stängs)

**Detta skall du kunna efter genomförd labb:**

- ✓ Starta Powershell
- ✓ Använda vanligt förekommande kommandon och pipe
- ✓ Hitta kommandon och hjälp om kommandon