# Cmdlets:

|  |  |  |  |
| --- | --- | --- | --- |
| Cmdlet | Parameter | Beschreib | Beispiel |
| Get-Childitem | -Exclude, –Include, –File, -Recurse | Gibt alle Dateien im Momentanem Folder aus | Get-Childitem \* –Exclude «\*.jpg» |
| Get-Command | -Parametername | Gibt (gesuchte) Cmdlets zurück | Get-Command Select-\* |
| Get-Date | -Format | Gibt die jetztige Zeit und Datum zurück | Get-Date –Format yyyy.MM.dd |
| Get-Help | -Detailed, -Full, -Examples, -Parameter | Gibt eine Hilfeseite für ein selektiertes Cmdlet zurück | Get-Help Select-Object |
| Get-History |  | Gibt die vorherig ausgeführten Programme zurück | Get-History |
| Get-Member |  | Gibt alle Members eines Cmdlets zurück (Properties, etc.) | Get-Service | Get-Member |
| Get-Process |  | Gibt alle auf dem Computer laufende Prozesse zurück | Get-Process |
| Get-PSDrive |  | Gibt alle Laufwerke (Auch Powershell spezifische) zurück | Get-PSDrive |
| Get-Service |  | Gibt alle Services auf einem Computer zurück | Get-Service |
| Group-Object |  | Gruppiert die Objekte nach einer spezifizierten Eigenschaft | Get-Childitem | Group-Object Extension | Sort-Object count |
| Measure-Object | -Sum | Misst eine spezifizierte Eigenschaft | (Get-Childitem | Measure-Object Length –sum ).sum |
| Move-Item | -Destination | Verschiebt das angegebene Item zur angegebenen Filelocation | Move-Item PSOrdner –Destination C:\ük\ |
| New-Item | -ItemType | Erstellt ein neues Item | New-Item PSOrdner –ItemType directory |
| New-PSDrive | -Name, –PSProvider, -Root | Erstellt ein neuen Powershelldrive für die momentane Sitzung | New-PSDrive –Name Drive –PSProvider FileSystem –Root C:\Berufsschule |
| Read-Host |  | Schreibt angegebenes in die Konsole und holt einen Input vom User | $a = Read-Host «Please enter something» |
| Remove-Item |  | Löscht spezifizierte Items (Dateien) | Get-Childitem | Remove-Item |
| Remove-PSDrive |  | Löscht den angegebenen PSDrive | Remove-PSDrive Name |
| Select-Object | -First, –Last, –Property, -Index, -Unique | Gibt die spezifizierten Objekte zurück | Get-Process | Select-Object –First 5 –Property Name |
| Set-Location |  | Ändert den Dateipfad/Location | Set-Location Name |
| Sort-Object | -Descending, -Property | Sortiert Objekte nach spezifizierter Eigenschaft | Get-Process | Sort-Object Name |
| Start-Process |  | Startet den spezifizierten Prozess | Start-Process Chrome.exe |
| Update-Help |  | Aktualisieren die Informationen des Get-Help Cmdlets | Update-Help |
| Write-Host | -BackgroundColor, -ForegroundColor | Schreibt angegebenes in die Konsole | Write-Host «Hello World» |

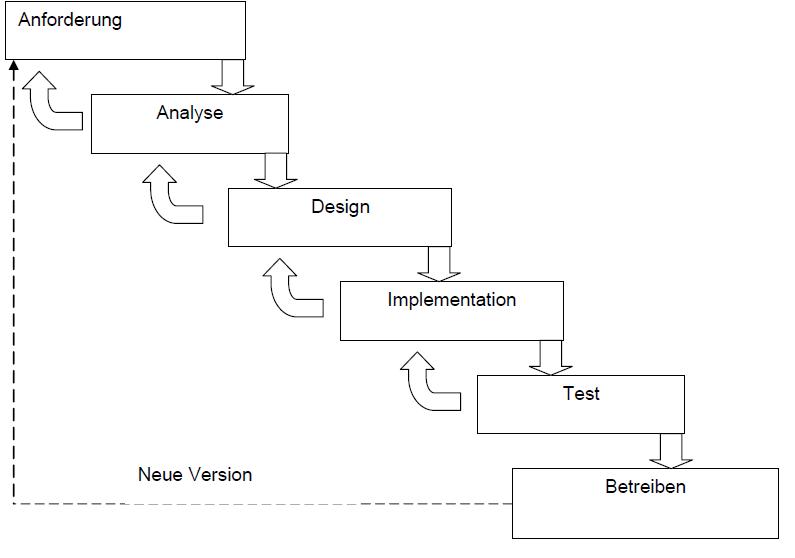
# Wichtige Schleifen und mehr:

|  |
| --- |
| … | Where-Object{Kondition} | … |
| ... | Foreach-Object{ … } |
| If(Kondition) { … } else { … } |
| Switch(kondition) {Case { … } Case{ … } Case{ … } default{ … }} |
| While (Kondition) { … } |
| Do { … } until(Kondition) |
| Do { … } While(Kondition) |
| For(Kondition) { … } |
| Param {$a = 0, $b = 0} |
| [int] $zahl / [string] ${So geht das} |

# Operanden

|  |  |  |  |
| --- | --- | --- | --- |
| -eq | Gleich | -ne | Ungleich |
| -gt | Grösser als | -ge | Grösser gleich |
| -lt | Kleiner als | -le | Kleiner gleich |
| -match | Entspricht | -notmatch | Entspricht nicht |
| -like | Wie | -notlike | Nicht wie |

# Projektablauf Wasserfallmodell:



In unserem Fall werden Analyse und Design zu einem Einzigen Schritt gemacht

**Anforderungen**

Spezifikation der Anforderungen. Meistens eine textuelle und grafische Beschreibung der Anforderungen.

**Analyse/Design**

Die Anforderungen werden analysiert und in ein Design überführt. Bei kleinen Programmen kann die Dokumentation dieser Phase entfallen und die Ergebnisse von Analyse/Design können direkt implementiert werden.

In der Analyse/Design Phase werden die folgenden Hilfsmittel angewandt:

* Textverarbeitungsprogramme zum Dokumentieren von Analyse/Design
* Strukturierte Analyse (SA) für grosse prozedurale Scripts/Makros (selten)
* Klassen- und Objektdiagramme für grosse objektorientierte Scripts/Makros (selten)

**Implementation**

Erstellung des Programms mit einer Script/Makro Sprache. Vor allem bei kleinen Programmen wird erwartet, dass die Analyse/Design Entscheide, sowie die Informationen für den Benutzer als Textinformationen mitimplementiert und in geeigneter Form angezeigt werden können (Help).

In der Implementationsphase werden die folgenden Hilfsmittel angewandt:

* Struktogramme (oder Programmablaufpläne (PAP)) für prozedurale Scripts. Vor allem wenn die Scripts über keine vollständige Implementierung von Kontrollstrukturen verfügen und dadurch mit GOTO Befehlen gearbeitet werden muss.
* Texteditoren für das Erstellen der Scripts

**Test**

Wie jedes andere Programm müssen auch Scripts/Makros systematisch getestet werden.

Dazu werden Testfälle formuliert (z.B. mittels Office Werkzeugen) und in Testdurchläufen abgearbeitet.

**Betreiben**

Betrieben werden diese Programme meistens zur Automatisation von Systemen (Betriebssysteme, Office, Datenbanken usw.) und nehmen dort spezifische Hilfs- und Servicefunktionen wahr.

# Flussdiagramm Symbole:

