# לימוד עצמי אובייקטים בPowershell

PowerShell היא שפה של אובייקטים. כלומר, כל דבר אליו אנו פונים ובו אנו משתמשים, נחשב אובייקט. פלט של כל פקודה ב-PS הוא אובייקט, קלט שנדרש להכניס לפקודות הוא אובייקט. כל אחד מהאובייקטים מכיל מאפיינים מסוימים.

אובייקט הוא עצם שמכיל תכונות ומשמש לייצוג נתונים. בסופו של דבר, לאובייקט יהיה **ערך** ספציפי (הוא מכיל מידע על משהו מסוים) ויש **פעולות** שניתן לבצע עליו.

**מרכיבי האובייקט:**

**Property** - מאפיין של אובייקט (תכונה).

לכל סוג אובייקט יש מאפיינים מוגדרים. אפשר לחשוב על property כתכונות שנשתמש בהם כדי לאפיין סוג אובייקט מסוים. לאובייקטים מאותו הסוג יהיו את אותם מאפיינים (לדוגמה לכל החתולים יש שם, מין, צבע פרווה וגיל כלשהו), ומה שמייחד בין המופעים השונים של אותו אובייקט הוא הערכים של הproperties שלהם (חתול א' מאופיין בשם "משה", במין "זכר" גיל 3 ובצבע פרווה שחור, חתול ב' מאפיין בשם "צילה", במין "נקבה, גיל 8 וצבע פרווה חום- לשניהם אותם מאפיינים כללים שיש לכל חתול, אך אנחנו מבדילים בין שני מופעי החתול השונים בגלל שערכי המאפיינים שלהם שונים אחד מהשני).



אובייקט כללי מסוג "חתול":

מופע חתול א':

שם: "משה"

צבע פרווה: "שחור"

מין: זכר

גיל: 3

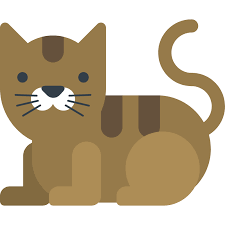
מופע חתול ב':

שם: "צילה"

צבע פרווה: "חום"

מין: נקבה

גיל: 8



**Method** - פעולה שניתן לבצע על אובייקט מסוים.

בPS נשתמש בנקודה (.) כדי למצוא property ולהפעיל method על אובייקט. לדוגמה:

$my\_file = Get-Item -Path 'C:\file.txt'

$my\_file.directory

$my\_file.delete()

שמרנו מידע על הקובץ ‘file.txt’ בתוך משתנה בשם my\_file. לאחר מכן אנחנו פונים למשתנה (זכרו שמשתנה זה הוא אובייקט שכן כל דבר בPS הוא אובייקט) ופולטים למסך את התיקייה בה הקובץ נמצא באמצעות הproperty של הdirectory. לבסוף, אנחנו משתמשים בMethod ה-delete כדי למחוק את הקובץ. נשים לב שלעומת property, שימוש בmethod מסתיים בסוגרים.

השתמשו בפקודה Get-Service למצוא את שירות ה"dhcp", הכניסו את הפלט למשתנה בשם dhcp\_service

|  |
| --- |
| $dhcp\_service = Get-Service -Name "dhcp" |

השתמשו במשתנה שיצרתם כדי לבדוק האם שירות ה”dhcp” רץ (מצאו את הproperty המתאים).

|  |
| --- |
| $dhcp\_service.Status |

**תווים מיוחדים**

ניתן להשתמש בתווים מיוחדים בעלי משמעות שונה (תלמדו בהמשך הקורס בפירוט יותר בנושא "ביטויים רגולריים"):

* \* - כולל את כל התווים מ-0 עד אין סוף פעמים.  
  משמע, הביטוי dev\* יתאים ל-devops ול-develop team אך לא יתאים ל- cyber או ל-dveops.
* ? - כולל את כל התווים פעם אחת בלבד.   
  לדוגמה, de? יתאים ל-dev או der אך לא יתאים ל-devops או derops.

לדוגמה, אפשר להשתמש בסימן האלו לחיפוש נוח של אובייקטים:

Get-Command –Name '\*Service\*'

Get-Command '\*ACL\*'

Get-Process –Name '?????'

חפשו Process ששמו מתחיל ב-S ונגמר ב-s.

|  |
| --- |
| Get-Process -Name S\*s |

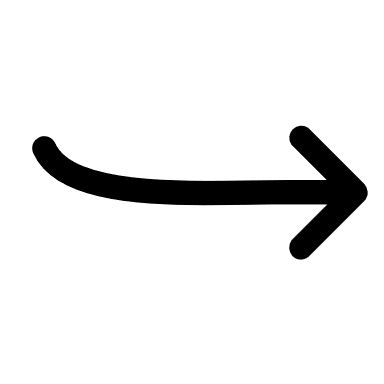
חפשו את כל הפקודות שמתחילות ב-Set.

|  |
| --- |
| Get-Command -Name set\* |

**שרשור פקודות**

**Pipe** – מאפשר לפלט של פקודה א' להיות **קלט** של פקודה ב'. השימוש ב-Pipe יעשה באמצעות התו Pipe-Line – |.

שימוש ב-Pipe-Line יהיה בצורה הבאה:

Command A | Command B

**סינון וסידור אובייקטים**נוכל לסנן ולסדר את הפלט שמוצג לנו ולהפוך אותו לנוח יותר לעבודה באמצעות פקודות שונות.   
  
Select-Object: פקודה השולפת אובייקטים או מאפיינים (properties) מסוימים שלהם.   
נניח נשבר לי הטלפון, וארצה להיכנס לחנות טלפונים לבחור טלפון חדש. כשאגש אל המוכר אוכל לבקש שהוא יציג לי טלפונים לפי תכונות מסוימות, select-object יהיה כמו לבקש מהמוכר שיציג לי ספציפית את איכות המצלמה ושטח האחסון של כל הטלפונים.  
  
לדוגמה: כשנשתמש בפקודה get-service ונרצה להשיג מידע על הServiceים הקיימים במערכת ולדעת האם הם במצב running או לא, נכתוב את הפקודה  
Get-Service.  
יוצג לנו הפלט הבא:  
Text

Description automatically generated  
העניין הוא שלGet-Service יש עוד properties מלבד הסטטוס, ואם לא מעניין אותי לראות אותן, אוכל לשלוף רק את הproperties שארצה לקבל (במקרה זה את הסטטוס) באמצעות שימוש בselect-object  
Graphical user interface

Description automatically generated with medium confidence

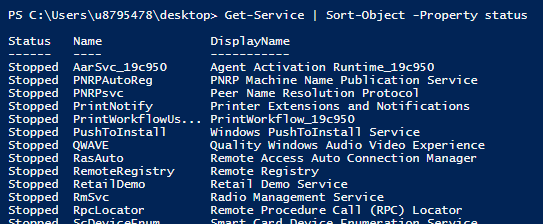
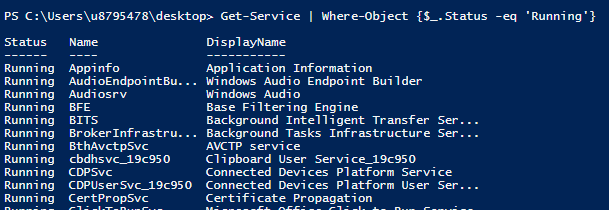
ניתן להשתמש בפקודה:

Select-Object -Property \*

כדי למצוא את כל התכונות של אובייקט מסוים, לרוב לא כל התכונות של האובייקט מוצגות באופן דיפולטי.

השתמשו בפקודה Get-ChildItem, הראו את השם ואת זמן היצירה של כל הקבצים בתיקיה הנוכחית.

|  |
| --- |
| Get-Childitem -File | Select-Object -Property name, lastwritetime |

נוכל גם להשתמש בפקודה על מנת שתציג לנו למשל את האובייקטים הראשונים או האחרונים בפלט, למשל על מנת לשלוף את 5 האובייקטים האחרונים בפלט (כלומר חמשת הserviceים האחרונים המוצגים לנו) נוכל להשתמש בפקודה הבאה:  
  
  
Sort-Object: פקודה המסדרת את הפלט בצורה נוחה.  
נוכל גם לבקש מהמוכר בחנות הטלפונים להסתכל על כל הטלפונים בחנות, אבל שיהיו מסודרים מהזול ליקר.  
לדוגמה: נניח שעכשיו נרצה לראות לא רק סטטוס של service אחד, אלא לראות בדיוק איזה serviceים במצב running ואיזה לא. נוכל להשתמש בפקודה sort-object ולסדר את הפלט:  
  
כמו שרואים הפלט מציג קודם כל את הserviceים שהם stopped, ולאחר מכן יופיעו כל הserviceים שהם running, בצורה מסודרת.   
הפקודה ממש אוספת את כל הפלט, ואז מסדרת אותו לפי מה שביקשנו.   
  
where-object: פקודה המסננת אובייקטים לפי **הערך** של מאפיין מסויים.   
נוכל לבקש גם מהמוכר בחנות הטלפונים שיציג לי את כל הטלפונים של חברת אפל.  
למשל אם אני ארצה לראות את כל הserviceים **שרצים** אוכל להשתמש בפקודה where-object  
  
  
קצת על הסינטקס של Where-Object:

הסימן $\_ - מציין משתנה שמכיל את **הפלט של הפקודה האחרונה** שהועברה ב- Pipeline. בדוגמה מעל, \_$ יחזיק את הפלט Get-Service, כלומר האובייקט שנוצר עקב הפקודה, רשימת השירותים.

נקודה – כידוע, כדי לגשת לproperty של אובייקט נשתמש בנקודה ונציין תכונה ספציפית שלו (במקרה הזה ניגשנו לתכונת ה’status’.

את התנאי נציב בעזרת eq – נלמד בהמשך על תנאים בPS, שימוש בסימן -eq (מלשון equal) מאפשר במקרה זה למצוא את כל השירותים הרצים (כל השירותים שproperty ה-‘status’ הוא ‘running’). כמובן ניתן להשתמש בתנאים אחרים.

פקודה שימושית נוספת היא Get-Member. בעזרתה נוכל לבחון אילו מאפיינים (Properties) נוכל להציג עבור האובייקטים, ואילו פעולות (Methods) ניתן לבצע עליהם.   
נסו לבצע בעצמכם את הפקודה הבאה:

****

כיצד נוכל לקבל רק את הproperties של האובייקט הנ"ל?

|  |
| --- |
| Get-Childitem C:\Course | Get-Member | Where-Object {$\_.MemberType -eq "property" } |

כיצד נוכל לקבל רק את הmethods שלו?

|  |
| --- |
| Get-Childitem C:\Course | Get-Member | Where-Object {$\_.MemberType -eq "method" } |

**Alias**

ב-PowerShell יש לנו הרבה קיצורים לפקודות, וכל קיצור כזה נקרא **Alias**.

Alias מהווה "קיצור דרך" עבור פקודות נפוצות (למשל ps בשביל Get-Process) או על מנת להתאים את פקודות PowerShell לפקודות CMD (למשל dir בשביל Get-ChildItem).

* **שימו לב! בתרגולים סגורים לא נקבל תשובות המכילות פקודות מקוצרות לכן אין להשתמש בaliasים במסגרת המבחנים בקורס!**

\*לרוב שימוש בalias יהיה בShell ולא בכתיבת סקריפטים שכן שימוש בפקודה המלאה יותר קריא בקטעי קוד.

כיצד ניתן לראות את כל ה-Alias הקיימים? (היעזר במבנה ה-Cmdlet ובפעלים הנפוצים).

מצאו Alias עבור Set-Location.

צרו Alias משלכם עבור הפקודה Get-Help. ה-Alias ייקרא geh. רשמו כיצד יצרתם את ה-Alias וכיצד תשתמש בו.

|  |
| --- |
| Get-Alias  Alias cd -> Set-Location  Alias sl -> Set-Location |
| Set-Alias -Name "geh" -Value Get-Help |
| Geh <Command> |

**תרגול**

שלפו מתוך תיקיית C:\Windows\System32 את 10 הקבצים הכי גדולים בתיקייה. הצג את הקבצים בסדר יורד, כלומר מהגדול לקטן. בנוסף לכך, הצג עבור כל אובייקט את שם הקובץ, גודלו ונתיבו המלא בלבד.

|  |
| --- |
| Get-Childitem C:\Windows\System32 -File | Sort-Object -Property length -Descending | Select-Object -Property name, length -First 10 |

תמצאו את שלושת הקבצים הקטנים ביותר בגודלם בתיקיית   
C:\Windows\System32

|  |
| --- |
| Get-Childitem C:\Windows\System32 -File | Sort-Object -Property length -Descending | Select-Object -Property name, length -Last 3 |

באותה התיקייה, תמצאו את כל הקבצים שגודלם גדול מ100 KB.

|  |
| --- |
| Get-Childitem C:\Windows\System32 -File | Sort-Object -Property length -Descending | Where-Object {$\_.Length -ge 100000} |

תכתבו סקריפט המציג את תאריך היצירה לכל הקבצים והתיקיות הנמצאים בC:\. הסקריפט יציג את התאריך בלבד.   
**אקסטרה:** עשו זאת ב2 דרכים שונות.

|  |
| --- |
|  |

**אקסטרה:**  
היעזרו בפקודות group-object ו- measure-object  
1. תחלקו את כל הקבצים בכונן C: ל2 קבוצות: תיקיות וקבצים.

2. תבדקו מה הממוצע והסכום של 1-1000. תבדקו גם מה המינימום והמקסימום של כל המספרים.

3. צור Script המבקש מהמשתמש להזין 2 מספרים.  
הפלט ימיין את המספרים בטווח בין המספר הראשון לשני לפי:

1. המספרים שמתחלקים ב2
2. כל השאר

עליך להציג אך ורק את מספר האלמנטים בכל קבוצה.  
4. שלוף מתוך תיקיית C:\Windows\System32 את הקבצים בעלי הסיומת DLL וEXE.  
עליך להציג את שם הסיומת ואת מספר הקבצים בלבד.