



# PowerShell Fundamentals

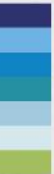
Jaap Brasser

AZURE  
OFFICE 365  
ENTERPRISE MOBILITY SUITE  
OPERATIONS MANAGEMENT SUITE  
AZURE STACK  
HYPER-V  
WINDOWS



# Module 2 – Variables and objects

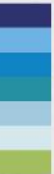
- Create variables
- Object types
- Accessing methods and properties of an object
- Create objects as a specific type
- Work with strings in PowerShell
- Create arrays and hash tables
- Create custom objects





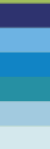
# Define a variable in PowerShell

- Using the '\$Variable = ...' notation
- Using the -Variable cmdlets
- Using Tee-Object





# Demo Define a variable





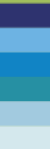
# Object types in PowerShell

- Each value displayed in the console or stored in a variable has an object type
- Get-Member can be used to reveal the object
- Examples of object types:
  - String / Integer / Boolean / Array
  - File / Folder / Registry entry / ACL



WINDOWS

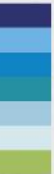
# Demo Create a specific type





# Objects in PowerShell

- Each object has set of properties & methods
- These can be discovered by using Get-Member







# Properties and methods

- It is possible for a property to have multiple sub properties
- Methods are called using () notation
- To discover the possible arguments for a method try to access with brackets





# Demo Access properties&methods



# Set an object as a specific type

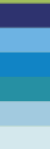


- PowerShell will determine which type
- Use special notation `@()` " `$()` to indicate type
- Cast the value or variable as a certain type
- Use the `-as` operator to cast a type



WINDOWS

# Demo PowerShell Object Type





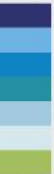
# Strings in PowerShell

- Four different types:
- Expanding strings
  - `"""`
  - `@" .... "@`
- Literal strings
  - `''`
  - `@' .... '@`



# Expanding strings

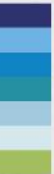
Displays contents of variables / statement  
Useful if variables are included in a string





# Literal strings

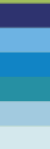
- Special characters do not have to be escaped





WINDOWS

# Demo Working with Strings

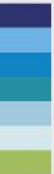






# Arrays in PowerShell

- Arrays are a collection of objects
- An array can contain any type of object
- PowerShell arrays can be appended to





# Hash Tables

- Key-Value pairs
- Value can contain any type of object
- Unstructured, a hash table is not ordered
- Useful for fast lookup
- Can be used to create PSCustomObject



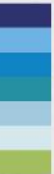
# Demo Create an array&hash table





# PowerShell Custom Object

- Structured object
- Can be easily created with properties of choice
- Can be converted to any type of output





WINDOWS

# Demo Create Custom Object

