Scripting & Automation Cheatsheet

A comprehensive cheat sheet covering essential scripting and automation concepts, tools, and techniques.



Scripting Fundamentals

Basic Concepts

Scripting: Writing a sequence of commands to automate tasks. Automation: Using scripts and tools to perform tasks automatically, reducing manual intervention. Key Benefits: Increased efficiency, reduced errors, and improved consistency. Shebang: #! - Specifies the interpreter for the script (e.g., #!/bin/bash for Bash, #!/usr/bin/env python3 for Python). Variables: Used to store and manipulate data. Control Flow: Statements like if , else , for , and while to control the execution flow. Functions: Reusable blocks of code to perform specific Comments: # (Bash, Python, PowerShell) - Used to add explanatory notes to the code.

Scripting Languages

Bash	Primarily used for Unix-like operating systems. Great for system administration tasks.
Python	Versatile language suitable for web development, data analysis, and general-purpose scripting.
PowerShell	Designed for Windows system administration and automation. Includes powerful cmdlets.

Input/Output

Standard Input (stdin): Input from the keyboard or redirected from a file.

Standard Output (stdout): Output displayed on the screen or redirected to a file.

Standard Error (stderr): Error messages displayed on the screen or redirected to a file.

Redirection:

- > Redirect stdout to a file (overwrites).
- >> Redirect stdout to a file (appends).
- 2> Redirect stderr to a file.
- &> or 2>&1 Redirect both stdout and stderr to a

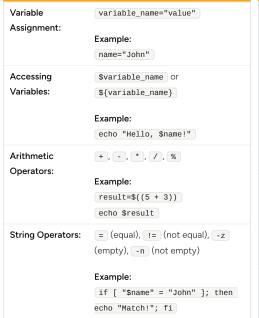
Piping:

Connect the stdout of one command to the stdin of another.

Example: ls -1 | grep 'myfile.txt'

Bash Scripting

Variables and Operators



Control Structures

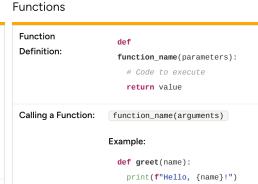
```
if [ condition ]; then
Statement:
                 # Code to execute if condition
               elif [ condition2 ]; then
                 # Code to execute if
               condition2 is true
                 # Code to execute if all
               conditions are false
For Loop:
                for variable in list;
                 # Code to execute for each
                item in the list
While Loop:
               while [ condition ]; do
                 # Code to execute while the
               condition is true
```

Functions

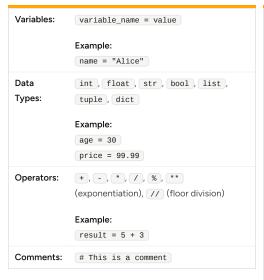
```
Function Definition:
                          function_name() {
                            # Code to execute
                            return value
Calling a Function:
                        function name
                        Example:
                          greet() {
                            echo "Hello, $1!"
                          greet "World"
```

Python Scripting

Page 1 of 2 https://cheatsheetshero.com



greet("World")



If if condition: Statement: # Code to execute if condition is true elif condition2: # Code to execute if condition2 is true else: # Code to execute if all conditions are false For Loop: for variable in iterable: # Code to execute for each item in the iterable While Loop: while condition:

Code to execute while the

condition is true

Modules

```
Importing Modules:
  import module_name
  from module_name import specific_item
  import module_name as alias

Example:
  import math
  print(math.sqrt(16))

  from datetime import datetime
  print(datetime.now())
```

PowerShell Scripting

Basic Concepts

```
Cmdlets: Commands in PowerShell (e.g., Get-Process , Write-Host ).

Variables: Start with a $ (e.g., $name = "John" ).

Piping: Use | to pass objects between cmdlets (e.g., Get-Process | Where-Object {$_.CPU -gt 10} ).
```

Variables and Data Types

```
Variable
                 $variable_name = value
Assignment:
                 Example:
                 $name = "John"
Data Types:
                 [int], [string], [bool],
                 [array] , [hashtable]
                 Example:
                  [int]$age = 30
                 $myArray = @("item1", "item2",
Arrays:
                 "item3")
Hashtables:
                  $myHash = @{Name="John";
                 Age=30}
```

Control Structures

```
lf
               if (condition) {
Statement:
                 # Code to execute if condition
               is true
               } elseif (condition2) {
                 # Code to execute if condition2
               is true
               } else {
                 # Code to execute if all
               conditions are false
For Loop:
               foreach ($item in $collection) {
                 # Code to execute for each item
               in the collection
               }
While Loop:
               while (condition) {
                 # Code to execute while the
               condition is true
               }
```

Functions

```
Function
                   function function_name {
Definition:
                     param (
                       $parameter1,
                       $parameter2
                     # Code to execute
                     return value
                   }
Calling a
                  function_name -parameter1 value1
Function:
                 -parameter2 value2
                 Example:
                   function Greet {
                     param (
                       $Name
                     Write-Host "Hello, $Name!"
                   }
                   Greet -Name "World"
```