# PowerShell Commands Assignment Template

## 1. Creating and Renaming Files/Directories

**Commands:**

mkdir test\_dir

cd .\test\_dir\

New-Item -ItemType File -Name "example.txt"

Rename-Item -Path "example.txt" -NewName "renamed.txt"

cd ..

**Explanation:**

* mkdir test\_dir creates a directory named test\_dir
* cd .\test\_dir\ moves into the directory
* New-Item -ItemType File -Name "example.txt" creates an empty file named example.txt
* Rename-Item -Path "example.txt" -NewName "renamed.txt" renames example.txt to renamed.txt
* cd .. moves back to the parent directory

**Screenshot:**



## 2. Viewing File Contents

**Commands:**

cd .\test\_dir\

Get-Content -Path "renamed.txt"

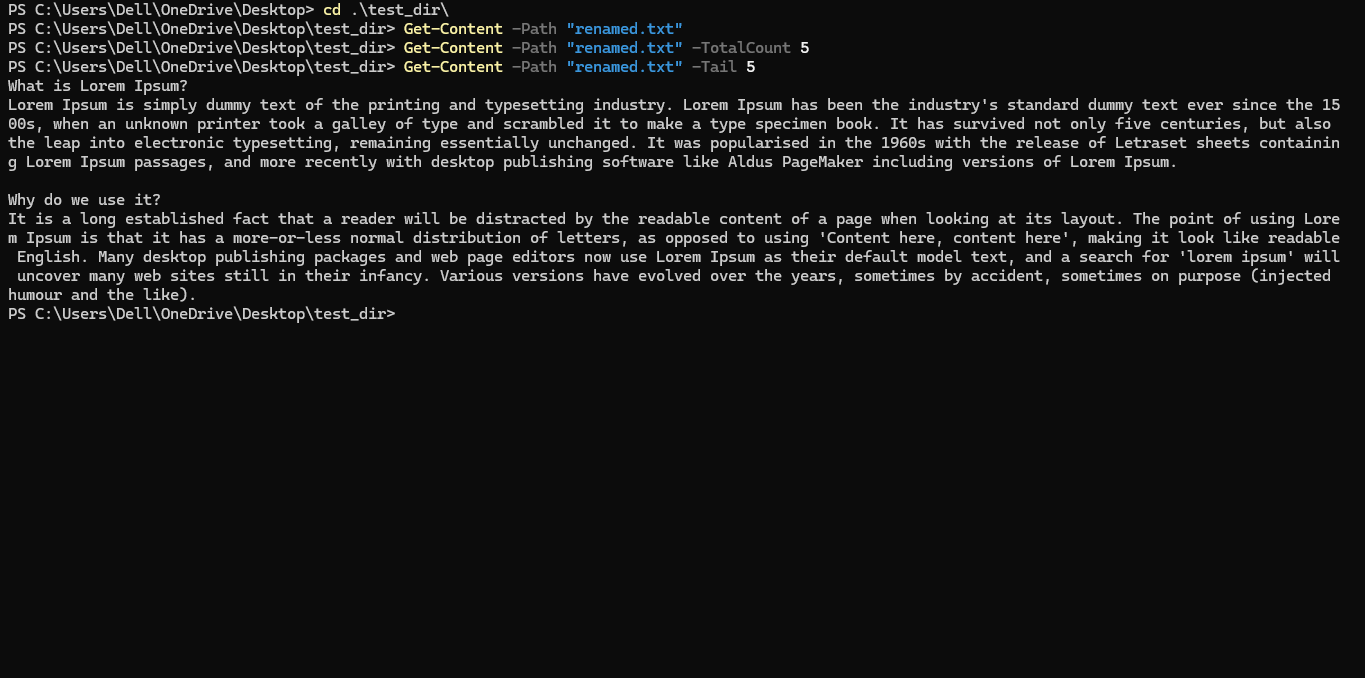
Get-Content -Path "renamed.txt" -TotalCount 5

Get-Content -Path "renamed.txt" -Tail 5

**Explanation:**

* Get-Content -Path "renamed.txt" displays the entire file content
* Get-Content -Path "renamed.txt" -TotalCount 5 shows the first 5 lines
* Get-Content -Path "renamed.txt" -Tail 5 shows the last 5 lines
* The file contains Lorem Ipsum text as shown in the output

**Screenshot:**



## 3. Searching for Patterns

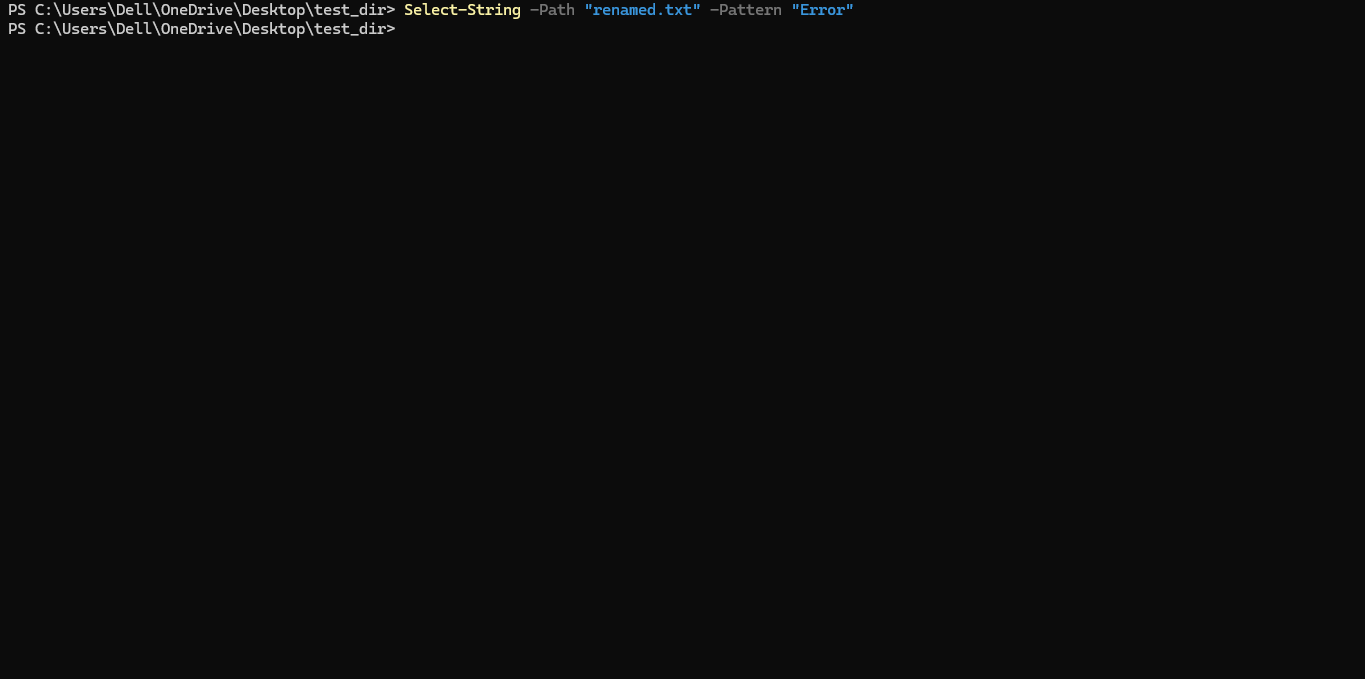
**Commands:**

Select-String -Path "renamed.txt" -Pattern "Error"

**Explanation:**

* Select-String -Path "renamed.txt" -Pattern "Error" searches for the word 'Error' in renamed.txt file
* The command returns no results as "Error" pattern was not found in the file content

**Screenshot:**



## 4. Compressing and Extracting Archives

**Commands:**

Compress-Archive -Path "test\_dir" -DestinationPath "archive.zip"

Expand-Archive -Path "archive.zip" -DestinationPath "Extracted"

**Explanation:**

* Compress-Archive -Path "test\_dir" -DestinationPath "archive.zip" compresses the test\_dir folder into archive.zip
* Expand-Archive -Path "archive.zip" -DestinationPath "Extracted" extracts the contents of archive.zip to the "Extracted" folder

**Screenshot:** 

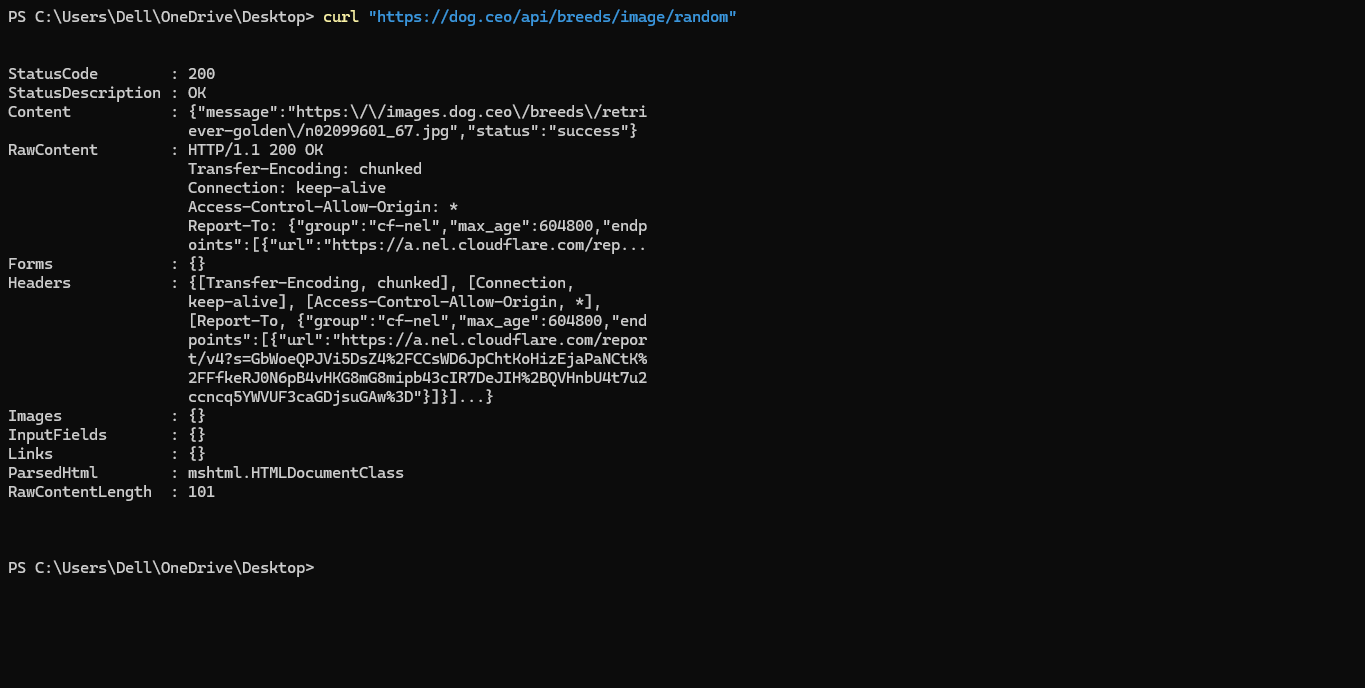
## 5. Downloading Files/Making Web Requests

**Commands:**

curl "https://dog.ceo/api/breeds/image/random"

**Explanation:**

* curl "https://dog.ceo/api/breeds/image/random" makes an HTTP GET request to the Dog API
* The command returns JSON response with dog breed information including image URL and status
* Shows various response details like StatusCode (200), Content, Headers, etc.

**Screenshot:** 

## 6. Changing File Permissions

**Commands:**

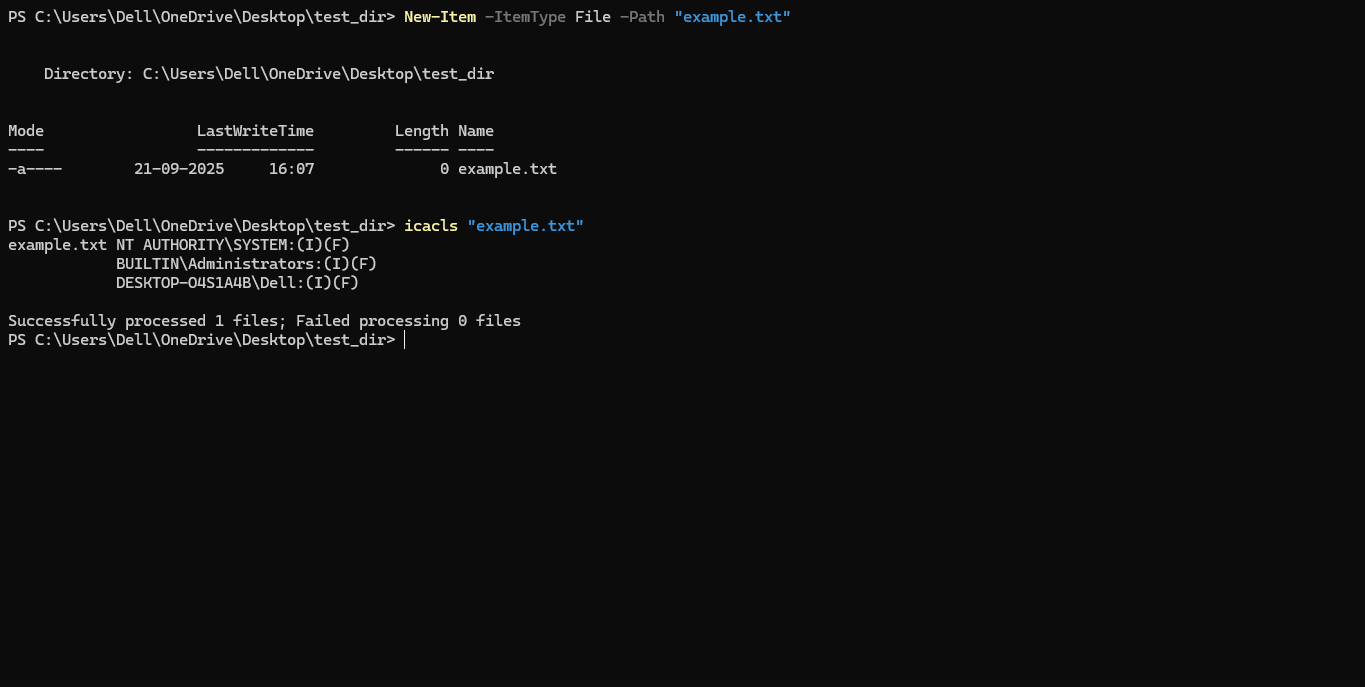
New-Item -ItemType File -Path "example.txt"

icacls "example.txt"

**Explanation:**

* New-Item -ItemType File -Path "example.txt" creates a new file named example.txt
* icacls "example.txt" displays the current access control list (permissions) for the file
* Shows permissions for NT AUTHORITY\SYSTEM, BUILTIN\Administrators, and the current user (DESKTOP-OHS1AUB\Dell)
* The output shows various permission levels like (I)(F) indicating inherited full control

**Screenshot:**



## 7. Working with Environment Variables

**Commands:**

$VAR = "yogesh"

echo $VAR

**Explanation:**

* $VAR = "yogesh" creates a PowerShell variable named VAR with the value "yogesh"
* echo $VAR displays the value of the variable
* The output shows "yogesh" confirming the variable was set and retrieved successfully

**Screenshot:**



**Note:** These commands demonstrate PowerShell equivalents to common Linux/Unix operations, showing how similar tasks can be accomplished in a Windows PowerShell environment.