Script Documentation

create_definitions_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing various definitions.

Tutorial:

```powershell# Description: This PowerShell script creates a directory structure for storing various definitions. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_definitions_directory_structure.ps1 $folders = @( "README.md" "Architecture_And_Organization\Cooling_ThermalManagement" "Architecture_And_Organization\Hardware_Specs" "Applied_CS\Business" "Applied_CS\Business\Intraday_Data.md" "Applied_CS\Healthcare" "Applied_CS\Science" "Artificial_Intelligence" "Data_Management" "Foundations_Of_Programming_Languages\Intro_Programming" "Foundations_Of_Programming_Languages\Python" "Foundations_Of_Programming_Languages\Python\Definitions_Python.md" "Foundations_Of_Programming_Languages\PowerShell" "Foundations_Of_Programming_Languages\PowerShell\Get_ChildItem_Cmdlet.md" "Graphics_Interactive_Techniques" "Human_Computer_Interaction\Physical_Interactions" "Human_Computer_Interaction\Physical_Interactions\Safe_Handling_IT_Equipment.md "Human_Computer_Interaction\Physical_Interactions\LED_Indicators.md" "Human_Computer_Interaction\Physical_Interactions\Keyboard_Usage.md" "Math\README.md" "Math\Algebra" "Math\Algebra\Linear_Equation.md" "Math\Business" "Math\Business\Present_Value.md" "Math\Calculus" "Math\Statistics" "Math\Statistics\Standard_Deviation.md" "Math\Glossary" "Math\Glossary\Abbreviations" "Math\Glossary\Mean.md" "Math\Geometry" "Math\Healthcare" "Math\Science" "Network_Communications\Physical_Layer" "Network_Communications\Physical_Layer\Cabling" "Operating_Systems" "Parallel_Distributed_Computing" "Security" "Soc_Ethics_Professionalism" "Software_Development_Fundamentals" "Software_Engineering" "Specialized_Platform_Development" "Systems_Fundamentals\File_Formats" ) # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or
```

```
$item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```

## create_equipment_notes_directory_structure.ps1

**Description:** This PowerShell script creates a directory structure for storing equipment notes.

**Tutorial:**

```powershell# Description: This PowerShell script creates a directory structure for storing equipment notes. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_equipment_notes_directory_structure.ps1 $folders = @( "README.md", "Racks", "Desks", "Building_Properties\Building_A.md", "Building_Properties\Data_Center_A.md", "Building_Properties\Building_A", "Building_Properties\Data_Center_A", "Hallways", "Inventory" ) # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```

## create_glossary_directory_structure.ps1

**Description:** This PowerShell script creates a directory structure for storing glossary terms.

**Tutorial:**

```powershell# Description: This PowerShell script creates a directory structure for storing glossary terms. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_glossary_directory_structure.ps1 $folders = @( "Acronyms", "Abbreviations", "README.md" ) # Create the folders (and files) foreach
```

```
($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```
```

create_hybrid_remote_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing hybrid and remote

Tutorial:

```powershell# Description: This PowerShell script creates a directory structure for storing hybrid and remote # work resources. It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_hybrid_remote_directory_structure.ps1 $folders = @("README.md", "Hybrid_Remote_Job1\README.md", "Hybrid_Remote_Job1\Notes") # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```
```

create_insights_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing insights.

Tutorial:

```powershell# Description: This PowerShell script creates a directory structure for storing insights. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_insights_directory_structure.ps1 $folders = @("README.md", "Recent_Notes\Recent_Notes.md", "News\Business\Notes_Business.md", "News\Healthcare\Notes_Healthcare.md", "News\Science\Notes_Science.md", "News\Technology\Notes_Technology.md", "Opinions\Notes_Opinions.md", "Opinions\Business", "Opinions\Healthcare", "Opinions\Science", "Opinions\Software", "Opinions\Hardware", "Opinions\Devices", "Opinions\Cables", "Opinions\Networks", "Physical_Data_Extraction\Optical_Character_Recognition.md", "Physical_Data_Extraction\Manual_Transcription.md") # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```

create_interactions_notes_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing interaction notes.

Tutorial:

```powershell# Description: This PowerShell script creates a directory structure for storing interaction notes. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_interactions_notes_directory_structure.ps1 $folders = @( "README.md", "OS", "IT_Devices", "Servers", "Programs", "Languages\Python", "Locations\Building_A\Entrance_1.md" ) # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist

```
if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item
-ItemType Directory -Path $parentDir | Out-Null Write-Host "Created
directory: $parentDir" } } # Now create the file or folder if (-not
(Test-Path -Path $item)) { if ($item.EndsWith(".md") -or
$item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType
File -Path $item | Out-Null Write-Host "Created file: $item" } else {
New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created
directory: $item" } } else { Write-Host "Item '$item' already exists." } }
Write-Host "Script completed."```
```

create_my_project_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing project files.

Tutorial:

```
```powershell# Description: This PowerShell script creates a directory
structure for storing project files. # It handles path validation and can
automate the creation of multiple README.md files. # Script Name:
create_my_project_directory_structure.ps1 $folders = @( "README.md",
"Business", "Healthcare", "Science", "Python\Project_A.md\Notes",
"PowerShell", "Project_Gemini\Notes\Building_A_Measurements.md",
"Project_Gemini\Notes\Environment_Conditions.md",
"Project_Gemini\Ideas\Documentation_Project_Structure.md",
"Project_Gemini\Ideas\Metadata_Templates.md",
"Project_Gemini\Ideas\Future_Scripts\Python_Server_Script.py",
"Project_Gemini\Implemented_Scripts\Current_Structure.md",
"Project_Gemini\Implemented_Structures\PowerShell\Rename_Files.ps1",
"Project_Gemini\Implemented_Structures\PowerShell\CamelCase_To_Underscore.ps1",
"Project_Gemini\Implemented_Structures\Python\Update_Metadata_Links.py",
"Project_Gemini\Metadata_Templates",
"Project_Gemini\Programming_Languages\Python\utility.md",
"Project_Gemini\Programming_Languages\PowerShell\utility.md",
"Project_Gemini\Command_Log.md",
"Project_Gemini\CS_Knowledge_Base_Structure.md",
"Project_Gemini\README.md" ) # Create the folders (and files) foreach
($item in $folders) { # Get the parent directory (handle empty string)
$parentDir = Split-Path -Path $item -Parent if (-not
[string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty
# Create the parent directory if it doesn't exist if (-not (Test-Path
-Path $parentDir -PathType Container)) { New-Item -ItemType Directory
-Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } }
# Now create the file or folder if (-not (Test-Path -Path $item)) { if
```
```

```
($item.EndsWith(".md") -or $item.EndsWith(".py") -or
$item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null
Write-Host "Created file: $item" } else { New-Item -ItemType Directory
-Path $item | Out-Null Write-Host "Created directory: $item" } } else {
Write-Host "Item '$item' already exists." } } Write-Host "Script
completed."```
```

create_online_courses_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing online course materials.

Tutorial:

```powershell# Description: This PowerShell script creates a directory
structure for storing online course materials. # It handles path
validation and can automate the creation of multiple README.md files. #
Script Name: create_online_courses_directory_structure.ps1 $folders = @(
"README.md", "Course1\Lecture1.md", "Course1\Notes" ) # Create the folders
(and files) foreach ($item in $folders) { # Get the parent directory
(handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not
[string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty
# Create the parent directory if it doesn't exist if (-not (Test-Path
-Path $parentDir -PathType Container)) { New-Item -ItemType Directory
-Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } }
# Now create the file or folder if (-not (Test-Path -Path $item)) { if
($item.EndsWith(".md") -or $item.EndsWith(".py") -or
$item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null
Write-Host "Created file: $item" } else { New-Item -ItemType Directory
-Path $item | Out-Null Write-Host "Created directory: $item" } } else {
Write-Host "Item '$item' already exists." } } Write-Host "Script
completed."```
```

create_on_location_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing on-location resources.

Tutorial:

```powershell# Description: This PowerShell script creates a directory
structure for storing on-location resources. # It handles path validation
and can automate the creation of multiple README.md files. # Script Name:
create_on_location_directory_structure.ps1 $folders = @( "README.md",
"Location1_Job\Notes", "Location1_Job\README.md" ) # Create the folders
(and files) foreach ($item in $folders) { # Get the parent directory
```

```
(handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```
```

create_professional_development_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing professional development materials.

Tutorial:

```
```powershell# Description: This PowerShell script creates a directory structure for storing professional development materials. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_professional_development_directory_structure.ps1 $folders = @( "Certifications\CompTIA_A+\Study_Guides", "Certifications\CompTIA_A+\Practice_Exams", "Certifications\CompTIA_A+\Resources", "Certifications\CompTIA_A+\Objectives.md", "Career_Roadmaps\IT_Administrator\Software_Developer_Career_Path.md", "Career_Roadmaps\Software_Developer\IT_Administrator_Career_Path.md", "README.md" ) # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```
```
```

create_public_portfolio_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing public portfolio projects.

Tutorial:

```powershell# Description: This PowerShell script creates a directory structure for storing public portfolio projects. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_public_portfolio_directory_structure.ps1 $folders = @("Portfolio\Certifications", "Portfolio\Projects", "Portfolio\Skills", "About.md") # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or $item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null Write-Host "Created file: $item" } else { New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created directory: $item" } } else { Write-Host "Item '$item' already exists." } } Write-Host "Script completed."```

create_tools_and_applications_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing tools and applications.

Tutorial:

```powershell# Description: This PowerShell script creates a directory structure for storing tools and applications. # It handles path validation and can automate the creation of multiple README.md files. # Script Name: create_tools_and_applications_directory_structure.ps1 $folders = @( "Applications\AI_Software_Development", "Applications\Audiobooks", "Tutorials", "Developers", "README.md" ) # Create the folders (and files) foreach ($item in $folders) { # Get the parent directory (handle empty string) $parentDir = Split-Path -Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { # Check if $parentDir is not empty # Create the parent directory if it doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) { New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host "Created directory: $parentDir" } } # Now create the file or folder if (-not (Test-Path -Path $item)) { if
```
```

```
($item.EndsWith(".md") -or $item.EndsWith(".py") -or
$item.EndsWith(".ps1")) { New-Item -ItemType File -Path $item | Out-Null
Write-Host "Created file: $item" } else { New-Item -ItemType Directory
-Path $item | Out-Null Write-Host "Created directory: $item" } } else {
Write-Host "Item '$item' already exists." } } Write-Host "Script
completed."```
```

create_tutorials_directory_structure.ps1

Description: This PowerShell script creates a directory structure for storing tutorials.

Tutorial:

```powershell# Description: This PowerShell script creates a directory
structure for storing tutorials. # It handles path validation and can
automate the creation of multiple README.md files. # Script Name:
create_tutorials_directory_structure.ps1 $folders = @( "Coding\Python",
"Coding\PowerShell", "Coding\YAML", "Physical_Data_Extraction",
"README.md" ) # Create the folders (and files) foreach ($item in $folders)
{ # Get the parent directory (handle empty string) $parentDir = Split-Path
-Path $item -Parent if (-not [string]::IsNullOrEmpty($parentDir)) { #
Check if $parentDir is not empty # Create the parent directory if it
doesn't exist if (-not (Test-Path -Path $parentDir -PathType Container)) {
New-Item -ItemType Directory -Path $parentDir | Out-Null Write-Host
"Created directory: $parentDir" } } # Now create the file or folder if
(-not (Test-Path -Path $item)) { if ($item.EndsWith(".md") -or
$item.EndsWith(".py") -or $item.EndsWith(".ps1")) { New-Item -ItemType
File -Path $item | Out-Null Write-Host "Created file: $item" } else {
New-Item -ItemType Directory -Path $item | Out-Null Write-Host "Created
directory: $item" } } else { Write-Host "Item '$item' already exists." } }
Write-Host "Script completed."```
```