1.

# Path to the folder containing pictures

$pictureFolder = "C:\BackgroundChange\pics"

- This line defines a variable named 'pictureFolder' and assigns it the path to the folder containing the pictures.

2.

# Function to set the desktop background

function Set-Wallpaper {

- The function 'Set-Wallpaper' is declared here with one parameter 'path', which is the path to the image file.

3.

param([string]$path)

Add-Type -TypeDefinition @"

using System;

using System.Runtime.InteropServices;

public class Wallpaper {

[DllImport("user32.dll", CharSet = CharSet.Auto)]

public static extern int SystemParametersInfo(int uAction, int uParam, string lpvParam, int fuWinIni);

}

"@

- These lines add a new type definition to the current PowerShell session. They define a class 'Wallpaper' with a method to change the desktop wallpaper using system-level functions via DLL imports.

4.

[Wallpaper]::SystemParametersInfo(20, 0, $path, 3)

- This line calls the 'SystemParametersInfo' function from the 'Wallpaper' class to change the desktop background to the specified path.

5.

}

- This marks the end of the 'Set-Wallpaper' function definition.

6.

# Main loop to cycle through each picture

while ($true) {

- This line starts an infinite loop. The script will continuously execute the code inside this loop.

7.

# Get a random picture from the folder

$randomPicture = Get-ChildItem -Path $pictureFolder -Filter \*.jpg | Get-Random

- This line selects a random picture from the folder specified in the 'pictureFolder' variable. The filter '\*.jpg' means it will only consider JPG files.

8.

# Set the random picture as the desktop background

Set-Wallpaper -path $randomPicture.FullName

- This line calls the 'Set-Wallpaper' function with the path of the randomly selected picture to set it as the desktop background.

9.

# Wait for 10 seconds

Start-Sleep -Seconds 10

- This line makes the script wait for 10 seconds before the loop restarts, thus changing the wallpaper every 10 seconds.

}