**# PowerShell script to print "Hello World"**

Write-Output "Hello World"

**# Declare variables**

$greeting = "Hello"

$name = "World"

**# Concatenate strings**

$message = "$greeting, $name!"

# Output the message

Write-Output $message

**# Basic arithmetic**

$a = 5

$b = 10

$sum = $a + $b

Write-Output "Sum: $sum"

**# If-else statement**

$number = 10

if ($number -gt 5) {

Write-Output "$number is greater than 5"

} else {

Write-Output "$number is not greater than 5"

}

**# For loop**

for ($i = 1; $i -le 5; $i++) {

Write-Output "Iteration $i"

}

# While loop

$j = 1

while ($j -le 5) {

Write-Output "While iteration $j"

$j++

}

**#!/bin/bash # Bash script to print "Hello World"** echo "Hello World"

**# Declare variables**

greeting="Hello"

name="World"

**# Concatenate strings**

message="$greeting, $name!"

# Output the message

echo $message

**# Basic arithmetic**

a=5

b=10

sum=$((a + b))

echo "Sum: $sum"

**# If-else statement**

number=10

if [ $number -gt 5 ]; then

echo "$number is greater than 5"

else

echo "$number is not greater than 5"

fi

**# For loop**

for i in {1..5}; do

echo "Iteration $i"

done

# While loop

j=1

while [ $j -le 5 ]; do

echo "While iteration $j"

((j++))

done

**# Define a function**

function Get-Sum {

param (

[int]$a,

[int]$b

)

return $a + $b

}

**# Call the function**

$result = Get-Sum -a 5 -b 10

Write-Output "Sum: $result"

**# Using splatting to pass parameters**

$params = @{

Name = "John"

Age = 30

}

function Get-UserInfo {

param (

[string]$Name,

[int]$Age

)

Write-Output "Name: $Name, Age: $Age"

}

Get-UserInfo @params

**# Read from a file**

$content = Get-Content -Path "example.txt"

Write-Output $content

**# Write to a file**

"Hello, World!" | Out-File -FilePath "output.txt"

# Try-Catch block

try {

$result = 1 / 0

} catch {

Write-Output "An error occurred: $\_"

}

**# Define a function**

get\_sum() {

local a=$1

local b=$2

echo $((a + b))

}

**# Call the function**

result=$(get\_sum 5 10)

echo "Sum: $result"

**# Function using arrays**

process\_array() {

local arr=("$@")

for i in "${arr[@]}"; do

echo "Processing $i"

done

}

# Call the function

data=("apple" "banana" "cherry")

process\_array "${data[@]}"

**# Read from a file**

content=$(cat example.txt)

echo "$content"

**# Write to a file**

echo "Hello, World!" > output.txt

# Try-Catch equivalent using if statements

result=$(some\_command 2>&1)

if [ $? -ne 0 ]; then

echo "An error occurred: $result"

fi