4th FEB
Powershell Activity

```
Loading personal and system profiles took 1833ms.
PS C:\Users\jatin.saini3> Write-Host 'Hello, World!'
Hello, World!
```

```
Hello, World!
PS C:\Users\jatin.saini3> $a = 29
PS C:\Users\jatin.saini3> $b = "Kendrick"
```

```
PS C:\Users\jatin.saini3> New-Variable -Name myVar -Value 1337 -Option ReadOnly
PS C:\Users\jatin.saini3> $myVar
1337
```

```
PS C:\Users\jatin.saini3> New-Variable -Name myConst -Value "This CANNOT be changed" -Option Constant
PS C:\Users\jatin.saini3> $myConst
This CANNOT be changed
```

```
PS C:\Users\jatin.saini3> for ($i = 1; $i -le 5; $i++){
>>     Write-Host $i
>> }
1
2
3
4
5
```

```
PS C:\Users\jatin.saini3> $list = @('a', 'b', 'c', 'd');
PS C:\Users\jatin.saini3> foreach($item in $list){
>>     Write-Host $item
>> }
a
b
c
d
```

```
PS C:\Users\jatin.saini3> $value = 5
PS C:\Users\jatin.saini3> if ($value -gt 1) {
>>     Write-Host "value is greater than 1"
>> }
value is greater than 1
```

```
PS C:\Users\jatin.saini3> $month = 4
PS C:\Users\jatin.saini3> switch ($month) {
>>     1 { Write-Host "January" }
>>     2 { Write-Host "February" }
>>     3 { Write-Host "March" }
>>     4 { Write-Host "April" }
>>     5 { Write-Host "May" }
>>     6 { Write-Host "June" }
>>     7 { Write-Host "July" }
>>     8 { Write-Host "August" }
>>     9 { Write-Host "September" }
>>     10 { Write-Host "October" }
>>     11 { Write-Host "November" }
>>     12 { Write-Host "December" }
>>     default{ Write-Host "Invalid month" }
>> }
April
```

```
PS C:\Users\jatin.saini3> $values = @("One", "Two", "Three", "Four", "Five")
PS C:\Users\jatin.saini3> $values
One
Two
Three
Four
Five
PS C:\Users\jatin.saini3> $values.GetType()

IsPublic IsSerial Name                                     BaseType
-------- -------- ----                                     --------
True     True     Object[]                                 System.Array
```

```
PS C:\Users\jatin.saini3> $array = New-Object 'object[,]' 5,8
PS C:\Users\jatin.saini3> $array[2,5] = 'Hello'
PS C:\Users\jatin.saini3> $array[3,7] = 'World!'
PS C:\Users\jatin.saini3> $array
Hello
World!
```

```
PS C:\Users\jatin.saini3> $employees = @{}
PS C:\Users\jatin.saini3> $employees.Add(1, "John")
>> $employees.Add(2, "Mary")
>> $employees.Add(3, "Bob")
>> $employees.Add(4, "Sam")
PS C:\Users\jatin.saini3> $employee[2]
Cannot index into a null array.
At line:1 char:1
+ $employee[2]
+ ~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [], RuntimeException
    + FullyQualifiedErrorId : NullArray

PS C:\Users\jatin.saini3> $employees[2]
Mary
```

```
PS C:\Users\jatin.saini3> function writeHelloWorld() {
>>     Write-Host "Hello World!"
>> }
PS C:\Users\jatin.saini3> writeHelloWorld
Hello World!
```

```
PS C:\Users\jatin.saini3> "Lorem ipsum dolor sit amet..."
Lorem ipsum dolor sit amet...
PS C:\Users\jatin.saini3> $value1 = "Ut enim ad minim veniam... $var"
```

```
PS C:\Users\jatin.saini3> try {
>>     # This will generate an error
>>     1/0
>>     Write-Host "This is executed after the error"
>> } catch {
>>     # Catch all errors
>>     Write-Host "Oh oh! Error occurred.`n$_"
>> }
Oh oh! Error occurred.
Attempted to divide by zero.
```

```
PS C:\Users\jatin.saini3> $object1 = New-Object PSObject
PS C:\Users\jatin.saini3>
>> Add-Member -InputObject $object1 -MemberType NoteProperty -Name prop1 -Value "value1"
PS C:\Users\jatin.saini3> Add-Member -InputObject $object1 -MemberType NoteProperty -Name prop2 -Value "value2"
PS C:\Users\jatin.saini3> $object1

prop1  prop2
-----  -----
value1 value2


PS C:\Users\jatin.saini3> $object1.GetType()

IsPublic IsSerial Name                                     BaseType
-------- -------- ----                                     --------
True     False    PSCustomObject                           System.Object
```

```
PS C:\Users\jatin.saini3> Get-Date

05 February 2024 11:29:38
```

```
PS C:\Users\jatin.saini3> $names = @("Muffin","Romeo","Noodle","Zoe","Jack","Luna","Gracie","mittens","Phoebe","Peanut","Harley","Jake")
PS C:\Users\jatin.saini3> $names | Sort-Object
Gracie
Harley
Jack
Jake
Luna
mittens
Muffin
Noodle
Peanut
Phoebe
Romeo
Zoe
```

```
PS C:\Users\jatin.saini3> $object2 = [PSCustomObject]@{
>>     prop1 = "value1"
>>     prop2 = "value2"
>> }
PS C:\Users\jatin.saini3>
>> $object2

prop1  prop2
-----  -----
value1 value2
```

```
PS C:\Users\jatin.saini3> class Tree {
>>     [int]$Height
>>     [int]$Age
>>     [string]$Color
>> }
PS C:\Users\jatin.saini3> $tree1 = new-object Tree
>> $tree2 = [Tree]::new()
PS C:\Users\jatin.saini3> $tree1.Height = 10
>> $tree1.Age = 5
>> $tree1.Color = "Red"
PS C:\Users\jatin.saini3> $tree2.Height = 20
>> $tree2.Age = 10
>> $tree2.Color = "Green"
PS C:\Users\jatin.saini3> $tree1
>> $tree2

Height Age Color
------ --- -----
    10   5 Red
    20  10 Green
```

```
PS C:\Users\jatin.saini3> $url = "https://gist.githubusercontent.com/sanders
>> $json = (New-Object System.Net.WebClient).DownloadString($url)
>>
>> $data = $json | ConvertFrom-Json
>>
>> $data.colors

color  category type      code
-----  -------- ----      ----
black  hue      primary   @{rgba=System.Object[]; hex=#000}
white  value              @{rgba=System.Object[]; hex=#FFF}
red    hue      primary   @{rgba=System.Object[]; hex=#FF0}
blue   hue      primary   @{rgba=System.Object[]; hex=#00F}
yellow hue      primary   @{rgba=System.Object[]; hex=#FF0}
green  hue      secondary @{rgba=System.Object[]; hex=#0F0}
```