Advanced Scripting   
JSON and YAML

Last Updated: 2/4/2022 8:06 AM Version 1  
Document Prepared for: CIT361 Student

# Name Daniel Harris ID 235868292

# Instructions

Save a copy of this document. Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

You will explore JSON as a serialization format for data.

# Requirements

* PowerShell
* Internet Connectivity

# Setup

Make sure you have access to the psfiles example files. You can get a copy here <http://cf.esage.com/psfiles.zip>

# Task 1—Convert the Metal Data to JSON

## Steps

1. Import the metals.csv data  
   $m=Import-Csv .\Metals.csv
2. Since the numeric values will become strings due to the way csv files are processes we are going to fix the data up.  
   $m|%{$\_.SpecificGravity=+$\_.SpecificGravity;$\_.MeltingPoint=+$\_.MeltingPoint}
3. View the results
   1. What type is the SpecificGravity Property? int
   2. What type is the MeltingPoint Property? int
4. Convert this data to JSON and save to the file Metals.json  
   ConvertTo-Json -InputObject $m|Set-Content Metals.json
5. View the resulting file  
   Get-Content Metals.json

How are the Symbol and Name properties different than the MeltingPoint and SpecificGravity Properties? Not presented as numbers

# Task 2—Reading JSON Data

## Steps

1. Read the JSON data you just created into PowerShell and convert to PowerShell Objects. Since the ConvertFrom-JSON takes a string you will need to provide a string. Remember that the normal behavior of Get-Content is to read the file into an array of strings, one element for each line of text in the file. To make Get-Content import as a single string use the -raw switch  
   $jm=ConvertFrom-Json (Get-Content .\Metals.json -raw)
2. View the data  
   $jm
3. For each property list the datatype
   1. Symbol: System.Object[]
   2. Name: System.Object[]
   3. MeltingPoint: System.Object[]
   4. SpecificGravity: System.Object[]

# Task 3—YAML

PowerShell does not provide native support for YAML. However there are PowerShell Libraries that will. Try one out.

## Steps

1. Install a YAML library  
   Install-Module powershell-yaml -Scope CurrentUser
2. The library provides 2 cmdlets. ConvertFrom-Yaml, and ConvertTo-Yaml, they work pretty much the same way the JSON cmdlets work
3. Load a yaml file. Yaml files are used by Kubernetes to configure pods. View the sample file  
   Get-Content .\kube.yml
4. Load the sample file  
   $kube=ConvertFrom-Yaml (Get-Content .\kube.yml -Raw)
5. Explore the object  
   $kube  
   $kube.metadata.name
6. Write a command that accesses the containerPort
   1. Enter your command here. $kube.spec.template.spec.containers.ports

# Deliverable

Upload this document with completed answers to i-learn.