Advanced Scripting   
Hashtables

Last Updated: 9/23/2021 10:24 AM Version 1  
Document Prepared for: CIT361 Student

# Name Zach Lucas ID 895649438

# 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

Hashtables are used when you need a quick key value pair. Access the value by the key.

# Requirements

PowerShell

# Task 1—Creating Hashtables

## Steps

1. Create an empty hashtable:  
   $h=@{}
   1. What is the type of $h? Hashtable
2. Next create a hashtable $song with several values  
   **Key Value**  
   Title 'Free Will'  
   Band 'Rush'  
   Album 'Permanent Waves'  
   Year 1980  
   Stars 5
   1. Enter the command:  
      $song=@{Title='FreeWill'  
      Band='Rush'  
      Album='Permanent Waves'  
      Year=1980  
      Stars=5}
   2. View the contents of the hashtable:  
      $song
   3. Is the order displayed the same as you entered? Yes
3. Alternatively you can create the hashtable on one line like this:  
   $score=@{home=56;visitor=80}
   1. View the results.

# Task 2—Using Hashtables

## Steps

1. Accessing values. You can access the values several ways.
   1. Access via dot notation using the keyname as a property (tab completion works for the keys). Enter:  
      $song.title
   2. Index notation  
      $song['stars']  
      $song['title','stars']
   3. The name of the key in a variable  
      $key='album'  
      $song.$key  
      $song[$key]  
      $keys='album','band'  
      $song[$keys]
2. Changing values, you can change values all the same ways
   1. Dot notation  
      $song.title='Natural Science'
   2. Index notation  
      $song['stars']=4
   3. View the results.
3. Adding new values.
   1. To add a new key just assign a value to a key that does not currently exist
      1. Using dot notation  
         $score.final=$true
      2. Using Index notation  
         $score['overtime']=$false
   2. Or use the add method  
      $score.add('hadicap',20)
   3. View the results.
4. Removing items
   1. To remove an item from a hashtable use the Remove() method  
      $score.Remove('hadicap')
   2. View the results
5. View all keys with the keys collection  
   $song.keys
6. View all the values with the values collection  
   $song.Values
7. The count property contains the number of items in the hashtable.  
   $song.count
   1. How many Items are in the hashtable? 5
8. Use the Clear() method to remove all items from the hashtable.  
   $song.Clear()
   1. What is the value of the count property now? 0

# Task 3—Ordered Hashtables

Sometimes you would like the hashtable to preserve the keys in the order they were added.

## Steps

1. Create an ordered hashtable:  
   $song=[ordered]@{Title='FreeWill'  
   Band='Rush'  
   Album='Permanent Waves'  
   Year=1980  
   Stars=5}
2. View the hashtable.  
   $song
   1. Are the hashtable keys in the order they were created? Yes

# Task 4—Value or Reference type

## Steps

1. Are hashtables value or reference types? Reference types
2. Prove it. Show the code you used to prove it  
   $song -is [ValueType] returned “False”

# Deliverable

Upload this document with completed answers to i-learn.