**Tell:** Use the text provided explain to the group.

**Do:** Follow the instructions for the activity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Session/Title/Topic** | **LO** | **Content / Description** | **Assets/Media** | **Duration** |
| Introduction |  | **SAY:**  Hello and welcome to this course on PowerShell!  **DO:**  Facilitator to spend a brief 30 seconds introduction themselves. If they have relevant experience of PowerShell it would be useful to mention it here, but only in brief.  Create a “parking space” flipchart and stick it to the wall. Explain that any questions which the facilitator asks to park, due to them not being relevant at the time, attendees should write on a post-it note and stick to this flipchart.  Draw a “P” in blue whiteboard marker with a box round it, then a big box around the whole flipchart. | 3Variables.ps1 |  |
| Learning Objective Overview |  | **SAY:**  Welcome to the second workshop on this course! This workshop will cover control structures, functions and standards. But before we go on, lets do a quick recap of what we covered in the 2nd workshop.  **DO:**  Recap | 5Control Structures.ps1 |  |
| Control Structures | 1, 2 | **SAY:**  Now that we’ve recapped over the second workshop, lets crack on with Control Structures, Functions and Standards.  **SAY:**  Control Structures are mechanisms for us to introduce logic into our scripts. We can perform actions based on the result of our comparison operators, while something is true, or for each item in an array.  The first control structure we’re going to cover is an if/else statement.  The syntax is:  If (comparison operator) {  thing  } else {  Other thing  }  Where the thing happens if the comparison operator returns true if singular, or a true object if an array. If this is false, or nothing is returned, then the other thing happens. | 5Control Structures.ps1 |  |
| If statement | 1, 2 | **DO:**  Go through examples on 5Control Structures.ps1 sheet. | 5Control Structures.ps1 |  |
| While statement | 1,2 | **SAY:**  A while statement will do something whilst a condition is true.  The syntax is:  While (comparison operator) {  thing  }  Where thing is done provided that the comparison operator returns true. If the statement is true it is executed, and then the statement is checked again. This continues until the statement is false.  **DO:**  Go through examples on 5Control Structures.ps1 sheet | 5Control Structures.ps1 |  |
| Foreach statement | 1, 2 | **SAY:**  Finally, the last statement we’ll cover is a foreach loop.  A foreach loop will do something for each item in an array.  The syntax is:  Foreach (tempvar in array) {  thing  }  This tempvar is used to reference the currently indexed item in the array. The code block is executed as many times as there are items in the array, with the tempvar automatically updated on each run to reference the next object in the array, and finishes when it has been executed as many times as the array count.  **DO:**  Go through examples on 5Control Structures.ps1 sheet | 5Control Structures.ps1 |  |
| Activity 1 | 1,2 | **SAY:**  Now, quite a lot of people struggle with trying to understand the concept of control structures and how they work. We do have some difficult task sheets about 2/3rds of the way through the workshops, but this activity is easier and intended to help you understand control structures in more depth.  **DO:**  Hand out activity 1 sheet and give attendees 15 minutes to complete. | 5Control Structures.ps1 / Control Structures, Functions and Standards Activity 1 |  |
| Activity 1 recap | 1,2 | Go through the activity sheet together, answer any questions the group may have by encourage answers to come from within the group rather than the facilitator. Once questions have been cleared then move on to the second part of the workshop. | 5Control Structures.ps1 / Control Structures, Functions and Standards Activity 1 - Facilitator |  |
| Break | NA | **DO:**  Ask attendees if they would like 10 minute break | NA | NA |
| Function Syntax | 1,3 | **SAY:**  Now on to the 2nd 3rd of this workshop, functions.  Just like we can use variables to store datatypes for use later on in our scripts, we can use functions to store code for use later on in our scripts.  The syntax is fairly simple for a basic function:  Function name (params) {  stuff  }  There are more advanced ways to write functions, but we do not cover these in this course. | 6Functions.ps1 |  |
| Function examples | 1,3 | **DO:**  Go through examples on 6Functions.ps1 and answer questions. | 6Functions.ps1 |  |
| Standards | 1,3 | **DO:**  Go through examples on 7Standard-Good and 7Standard-Bad. Ensure you cover: whitespace, intention and camel casing. |  |  |
| Task Sheet | 1,3 | **SAY:**  Now, if there are no-more questions, you have an activity sheet to keep you occupied until the ned of the workshop.  **DO:**  Hand out the learner sheet for this workshop and give attendees till the end of the workshop to complete. | Task Sheet |  |
| End | 1,3 | **DO:** Wrap up workshop |  |  |