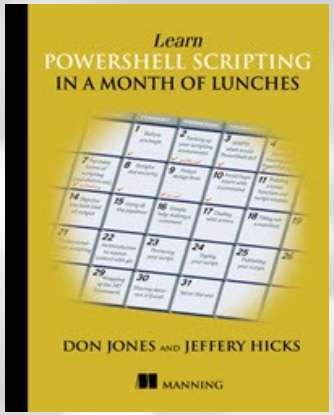
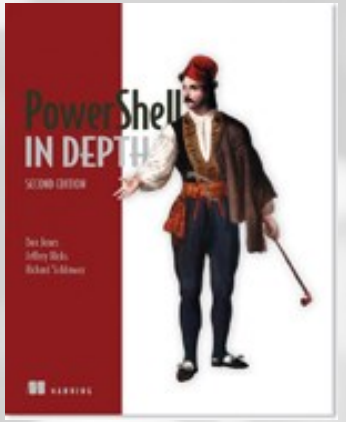
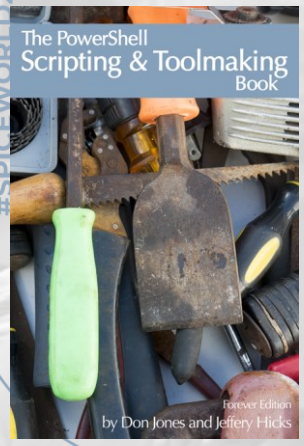
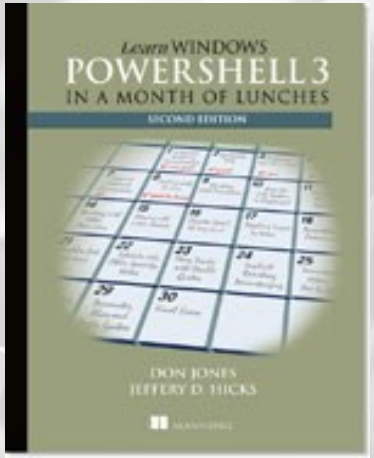
PowerShell Toolmaking Patterns and Practices

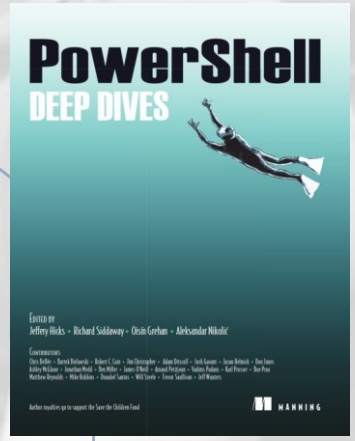




https://jeffhicks.substack.com

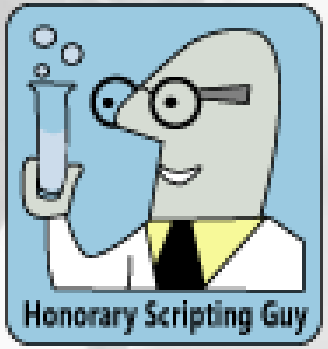






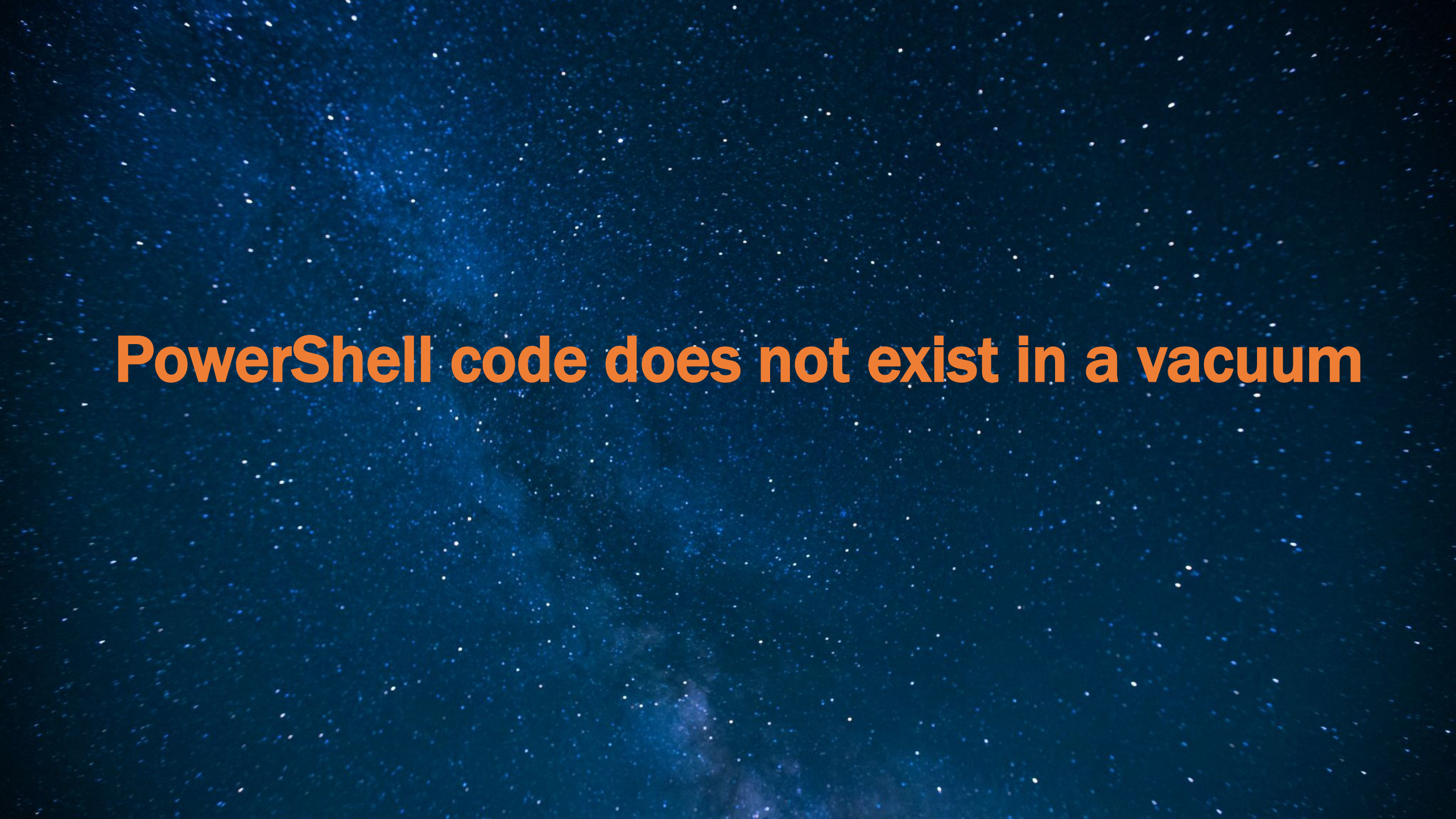
Jeff Hicks

https://jdhitsolutions.github.io



pluralsight

SPICEWORLD
IT CONFERENCE


PowerShell code does not exist in a vacuum

Code as Art

It doesn’t matter what you are automating

Anyone can learn syntax

Generating code is easy

But is it the right code?

Command Input

Who (or what) will be using your code?

What are their expectations?

Don’t assume anything

Anticipated usage scenarios

Can you leverage the pipeline?

Command Output

Functions do one thing

Functions write one type of object to the pipeline

What will the user do with your command?

What type of object do you need?

How can you seamlessly join your command output to the pipeline?

Separate formatting from code

Demonstration

Recommended Practices

Naming standards

Scripting Best Practices

- Full names and parameters
- Documentation
- Code layout

Script for Scale

Separate data from code

Separate formatting from code

Beyond PowerShell

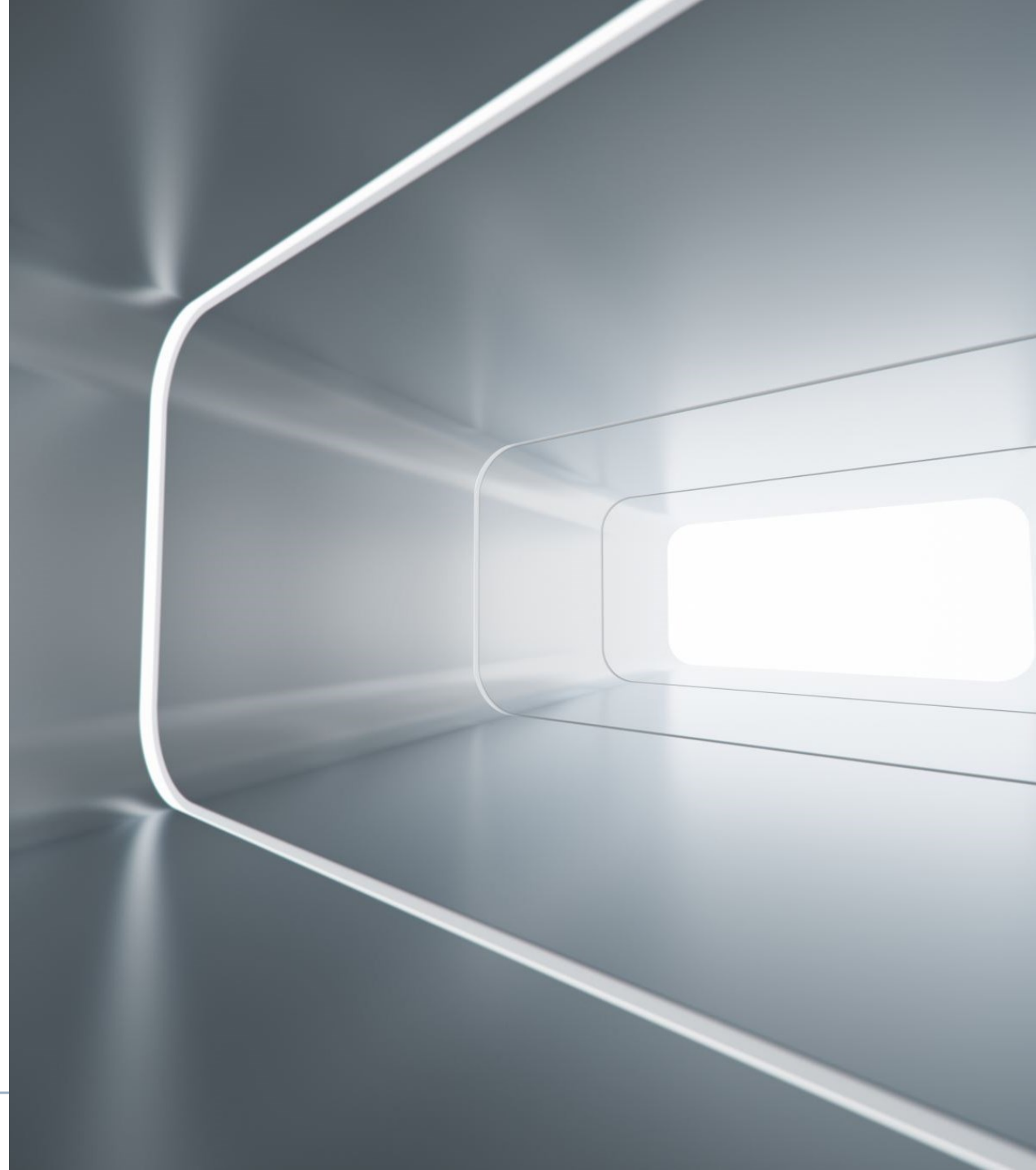
Use source control

Pester Tests

Validate in a clean environment

  PowerShell [pwsh] -noprofile

Consider code signing



Jeff's Suggestions

Understand the technology first

Write for the console before scripting

Write advanced functions

Include error handling (Try/Catch)

Use Verbose output

Learn to use Write-Progress

Script in layers

Write typed objects to the pipeline

Learn to create and use format and type extensions

PS Goals

Reduce friction

Be flexible

Plug into the PowerShell ecosystem

Set yourself up for sucess

...plan for the worst

Thank you.

#SPICEWORLD2023

THE HOTTEST I.T. CONFERENCE IN THE UNIVERSE

SPICEWORLD
— IT CONFERENCE —