PowerShell Conference Europe

# Unlocking the Power of Regular Expressions with PowerShell

https://github.com/jdhitsolutions/PSConfEU2023

PRAGUE23

PSCONF.EU

PowerShell Conference Europe

# Jeff Hicks

Author | Teacher | Guide

@jeffhicks

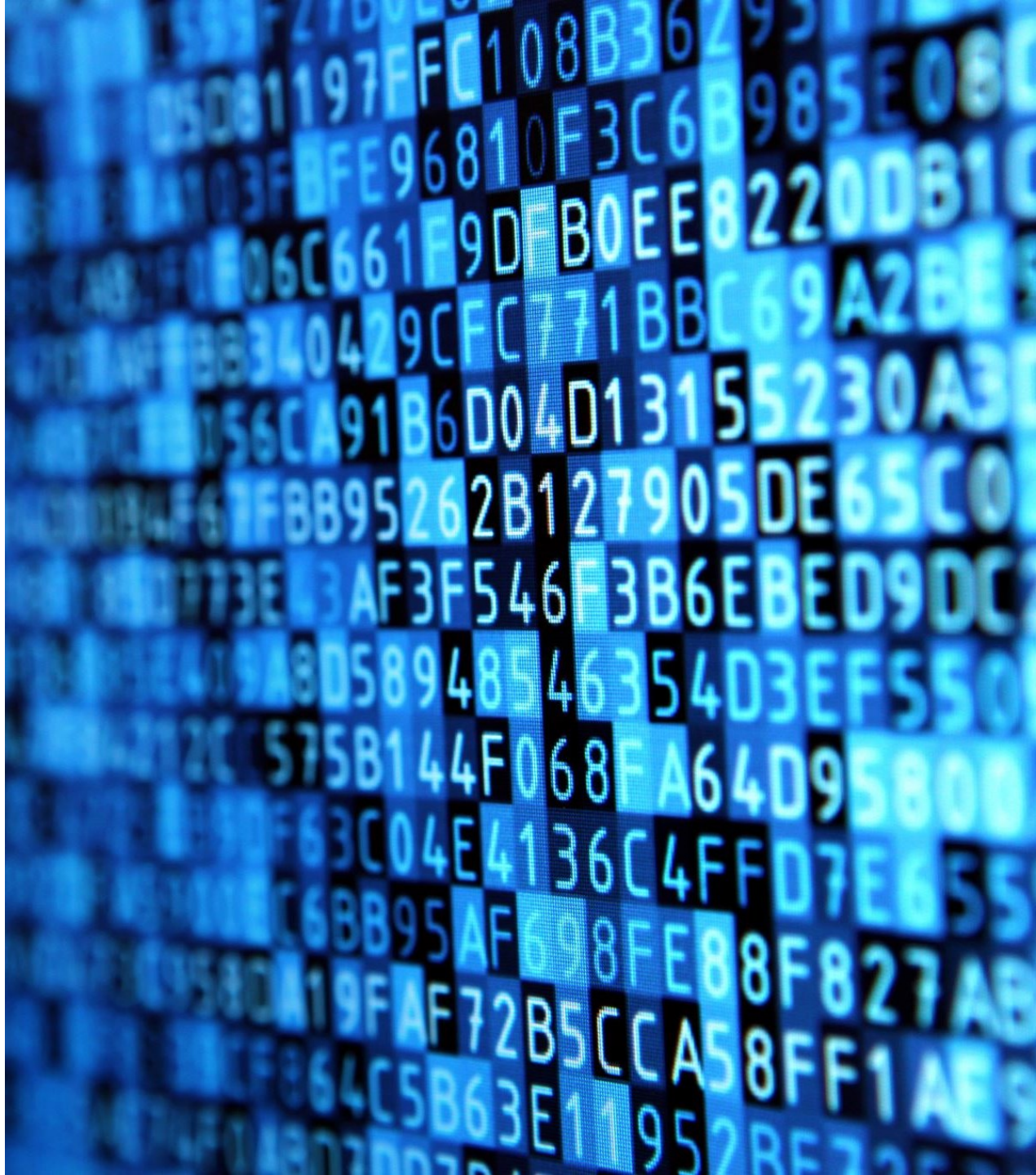
https://jdhitsolutions.github.io

Microsoft® Most Valuable Professional

PRAGUE23

# What is a Regular Expression?

- A way of describing data (text)
- Based on a set of accepted and standardized patterns
- Introduced in 1951
  - Incorporated into Unix
  - Led to *grep*
  - *“Global search for Regular Expressions and Print lines”*
- PowerShell supports the full regular expression library from .NET



# Why Regular Expressions?

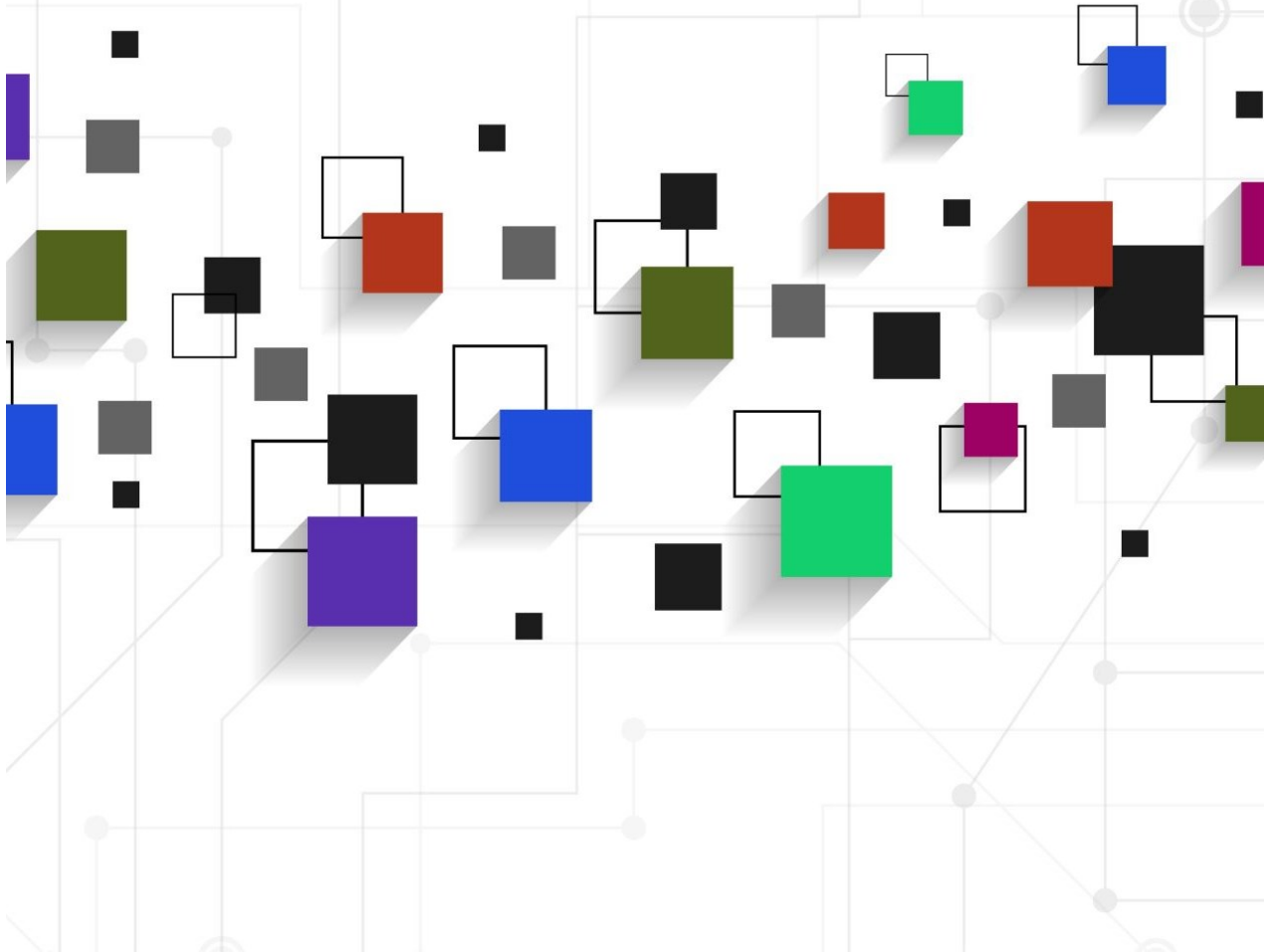
- Validating data
- Finding data
- Fixing data
  - Splitting
  - Replacing

# Patterns

- (202) 867-5309
- 192.168.2.3
- \\server01\public

# Patterns

- \(\d{3}\)\s\d{3}-\d{4}
- (\d{1,3}\.){3}\d{1,3}
- \\\\\w+\\\w+



# Regular Expressions are Classy

| Pattern | Match |
| --- | --- |
| \w | Any word character, including numbers and the underscore ( _ ) |
| \d | Any digit |
| \D | Any non-digit |
| \s | Any whitespace |
| \W | Any non-word character |
| \S | Any non-whitespace |

Case-Sensitive

Help about_regular_expressions

# With Qualifiers

| Qualifier | Description |
|---|---|
| * | Match 0 or more of the preceding pattern |
| + | Match 1 or more of the preceding pattern |
| ? | Match 0 or 1 instance of the preceding pattern |
| {n} | Match *exactly* N number of the preceding pattern |
| {n,m} | Match *at least* N number of the preceding pattern and *no more* than M number |
| {n,} | Match *at least* N number of the preceding pattern |

# Special Characters

| Character | Description |
|---|---|
| . | Any single character |
| [xyz] | Match at least one of the characters in the brackets |
| [a-zA-Z] | Match at least one of the characters in the range. Case sensitive. |
| [^xyz] | Match any character *except* what is in brackets |
| ^ | Match the beginning characters |
| $ | Match the ending characters |
| \ | The regular expression escape character |

# Escape Clause

- You'll need to escape regular expression characters if you need to match them
  - []<>?^$
- Escape with \
- `[2022-12-22] Some data goes here`
  - \[[\d-]+\]
    - Match on the literal [
      - \[
    - Match on a digit or the dash
      - [\d-]
    - Match one or more times
      - +
    - Match on the literal ]
      - \]
    - Match: `[2022-12-22]`

PowerShell Conference Europe

# Show Me

https://github.com/jdhitsolutions/PSConfEU2023

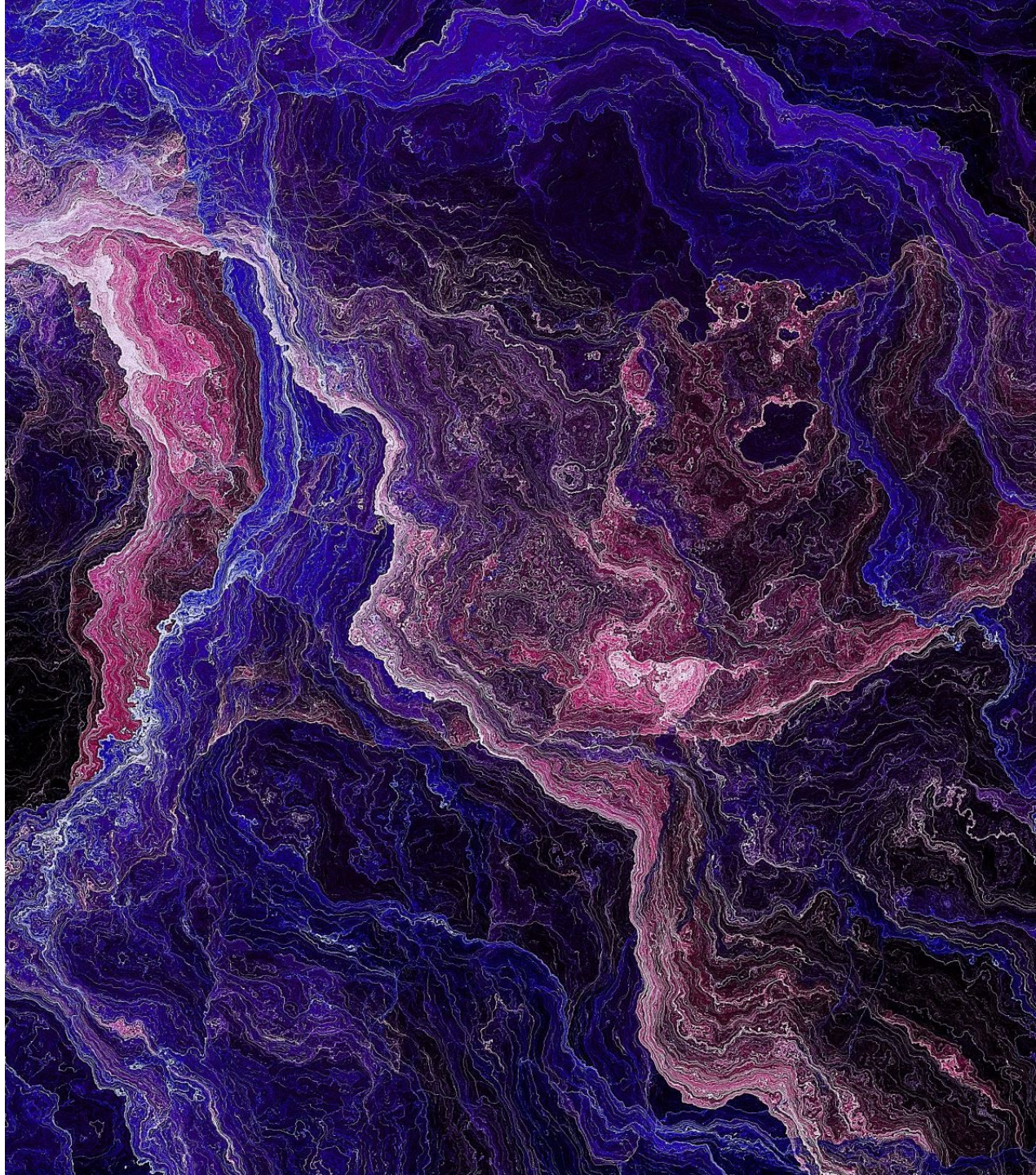
PRAGUE23

# Regular Expression “Gotchas”

- Comparisons are True/False
- Watch the float
  - How exact do you need to be?
- Test for known failures
- You still might need to validate
- You have to know your data
  - It should be predictable
  - It should be consistent

# Advanced Stuff

- Look ahead and look behind
  - (?<=\d\s)\w+\.\w+
  - 35 foo.bar
  - foo.bar
- Optional matches
  - ^\d{3}(\-[a-z]+)?$
  - 432
  - 678-xyj
- Regex Options
  - Configuring case
  - Multiline search
  - Ignore pattern whitespace



# Advanced Stuff

- More with the [regex] type accelerator
  - [regex]$rx = "^[a-z]{1}\.\w{2,7}@company.com$"
  - $rx.ismatch("f.bar@company.com")
    - True
  - $rx.match("f.bar@company.com")

```
Groups     : {0}
Success    : True
Name       : 0
Captures   : {0}
Index      : 0
Length     : 17
Value      : f.bar@company.com
ValueSpan :
```

# Resources

Help about_regular_expressions

Help about_comparison_operators

Help about_wildcards

# Resources

https://www.regular-expressions.info

https://regexlib.com

https://rubular.com

https://regex101.com

https://pluralsight.pxf.io/psregex

# Find Me Online

@jeffhicks

https://jdhitsolutions.github.io

SCAN ME


PowerShell Conference Europe

# Thank you.

PRAGUE23