# REGEX

Regular Expressions

# WHAT ARE REGULAR EXPRESSIONS?

A way of describing patterns within search strings

# VALIDATING EMAILS

colt@gmail.com carly.simon@yahoo.com rosa-98@meals.org

shoe_queen91@hello.net david+lee+roth@hotmail.com

Starts with 1 or more letter,number,+,_,-,. then

A single @ sign then

1 or more letter, number, or - then

A single dot then

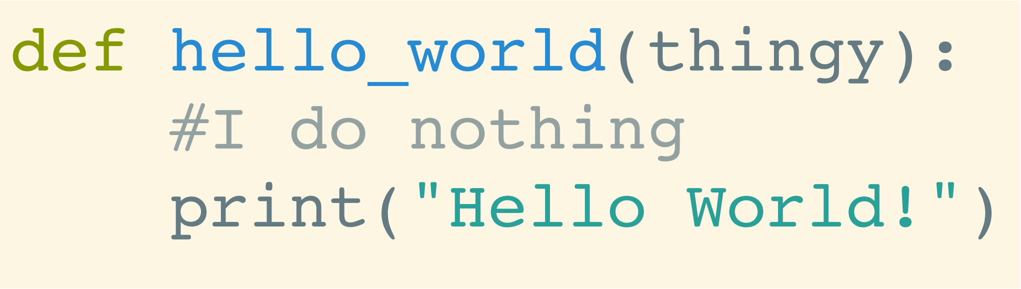
ends with 1 or more letter, number,-, or .

Let's use a regular expression!

```
(^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$)
```

# POTENTIAL USE CASES

- Credit Card number validating
- Phone number validating
- Advanced find/replace in text
- Formatting text/output
- Syntax highlighting



```
def hello_world(thingy):
    #I do nothing
    print("Hello World!")
```

# FIRST,

# LET'S START WITH REGEX OUTSIDE OF PYTHON

# THERE ARE A TON OF REGEX SYMBOLS.

I'm just going to cover some important ones

# SOME REGEX SYNTAX

## Some Characters

| | |
|---|---|
| \d | digit 0-9 |
| \w | letter, digit, or underscore |
| \s | whitespace character |
| \D | not a digit |
| \W | not a word character |
| \S | not a whitespace character |
| . | any character except line break |

# SOME REGEX SYNTAX

Quantifers

| | |
|---|---|
| + | One or more |
| {3} | Exactly x times. {3} - 3 times |
| {3,5} | Three to five times |
| {4,} | Four or more times |
| * | Zero or more times |
| ? | Once or none (optional) |

# SOME REGEX SYNTAX

## Anchors and Boundaries

| | |
|---|---|
| ^ | Start of string or line |
| $ | End of string or line |
| \b | Word boundary |

# LOGICAL OR

| (pipe character)

"Mr|Mrs|Ms"

I COULD MAKE 20+ SLIDES ON REHEX SYNTAX.

IT'S EASIER TO LOOK AT A CHEAT SHEET

# NOW,
# LET'S MOVE ON TO
# USING REGEX WITH PYTHON

# USING REGEX IN PYTHON

```python
#import regex module
import re

#define our phone number regex
pattern = re.compile(r'\d{3} \d{3}-\d{4}')

#search a string with our regex
res = pattern.search('Call me at 415 555-4242!')
```