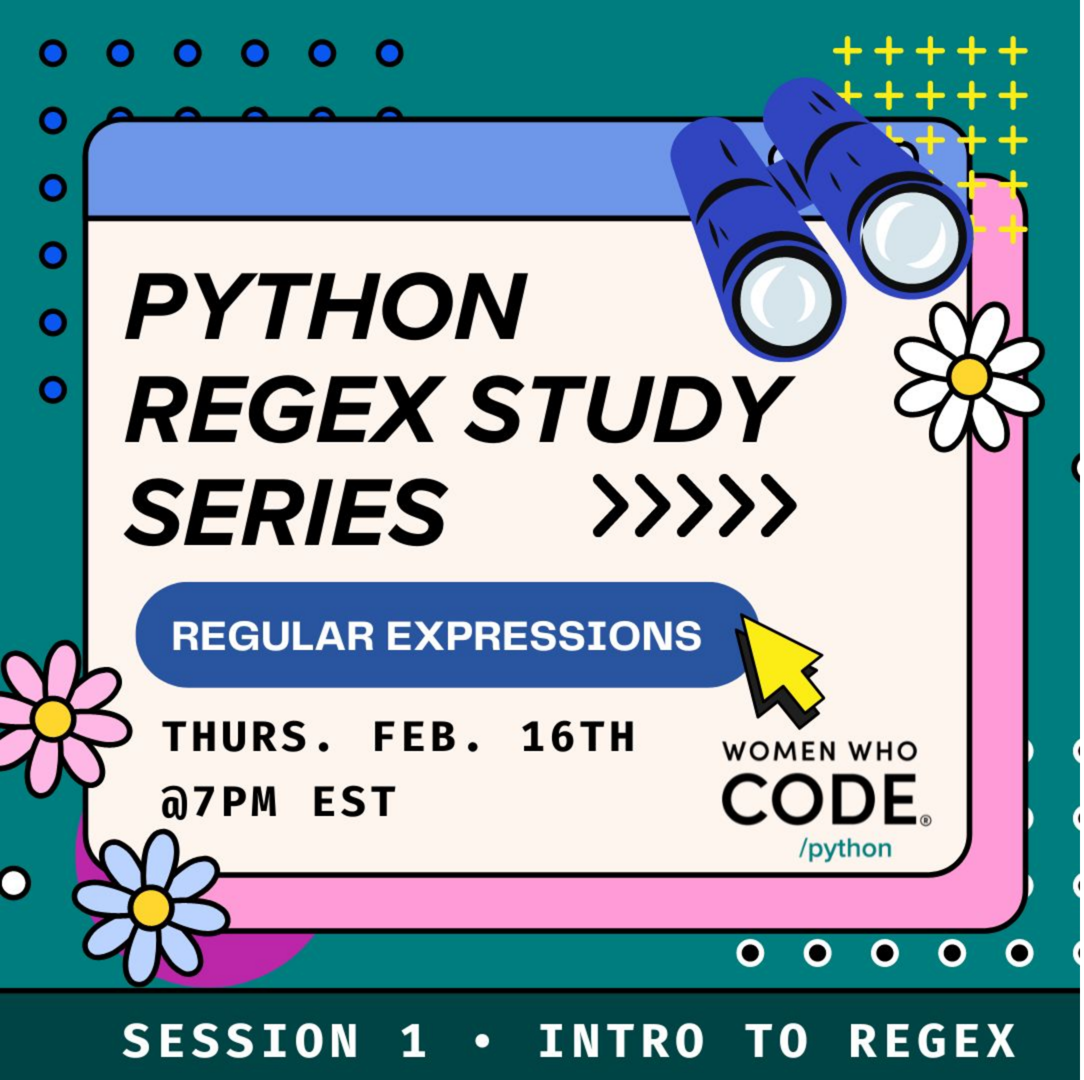


PYTHON REGEX STUDY SERIES

REGULAR EXPRESSIONS

THURS. FEB. 16TH
@7PM EST

WOMEN WHO CODE /python

SESSION 1 • INTRO TO REGEX

# WELCOME!

We will get started shortly.

Please share in the Chat:

- Where you are joining from
- What the weather is like today

***Reminder: Set your Chat to "Everyone".***

WOMEN WHO CODE /python

# Our Mission

Empower diverse women to excel in technology careers

WOMEN WHO CODE /python

# Our Vision

A tech industry where diverse women and historically excluded people thrive at every level.

WOMEN WHO CODE /python

# CODE OF CONDUCT

**WWCode is an inclusive community**, dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste, creed, political affiliation, or preferred programming language(s).

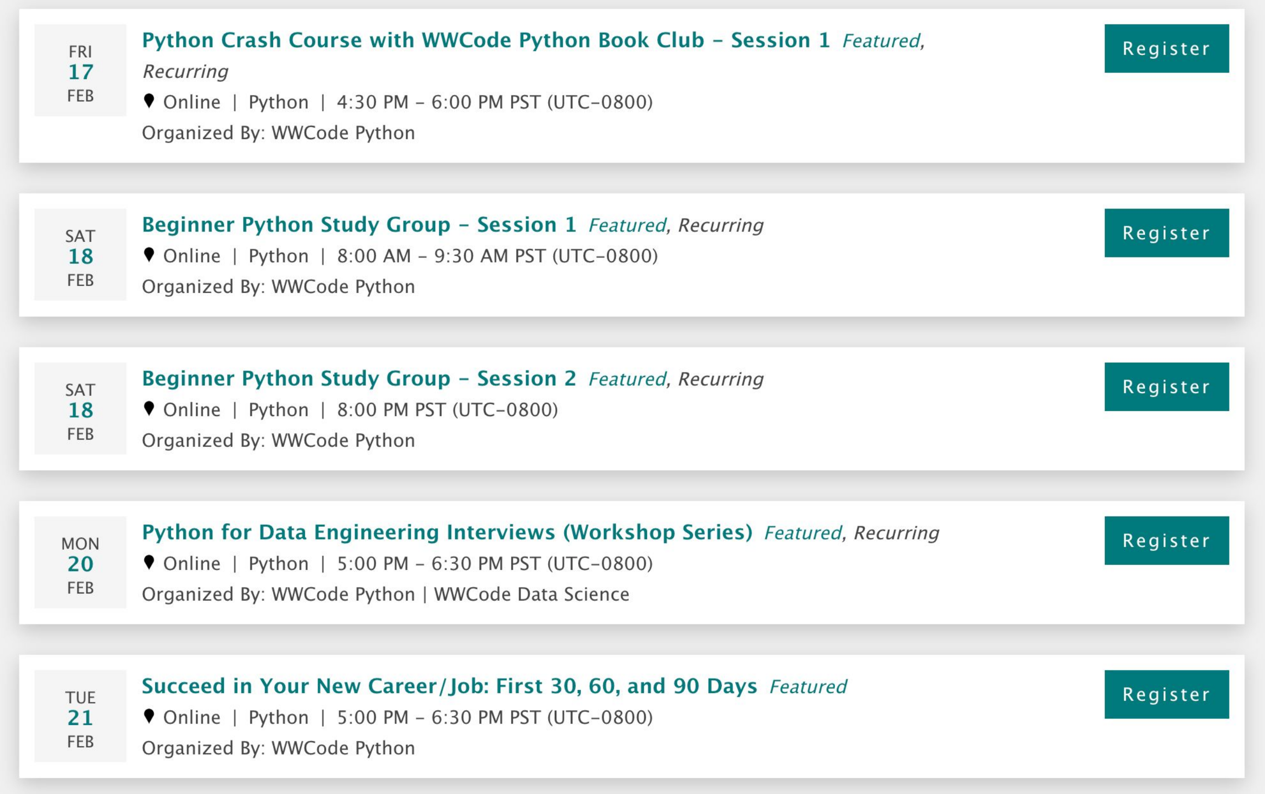
Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. We do not tolerate harassment of members in any form. Our **Code of Conduct** applies to all WWCode events and online communities.

Read the full version and access our incident report form at
**womenwhocode.com/codeofconduct**

WOMEN WHO
CODE®
/python

# /upcoming events

Learn more and register at
**womenwhocode.com/python**



FRI
17
FEB

**Python Crash Course with WWCode Python Book Club – Session 1** *Featured, Recurring*

📍 Online | Python | 4:30 PM – 6:00 PM PST (UTC-0800)

Organized By: WWCode Python

SAT
18
FEB

**Beginner Python Study Group – Session 1** *Featured, Recurring*

📍 Online | Python | 8:00 AM – 9:30 AM PST (UTC-0800)

Organized By: WWCode Python

SAT
18
FEB

**Beginner Python Study Group – Session 2** *Featured, Recurring*

📍 Online | Python | 8:00 PM PST (UTC-0800)

Organized By: WWCode Python

MON
20
FEB

**Python for Data Engineering Interviews (Workshop Series)** *Featured, Recurring*

📍 Online | Python | 5:00 PM – 6:30 PM PST (UTC-0800)

Organized By: WWCode Python | WWCode Data Science

TUE
21
FEB

**Succeed in Your New Career/Job: First 30, 60, and 90 Days** *Featured*

📍 Online | Python | 5:00 PM – 6:30 PM PST (UTC-0800)

Organized By: WWCode Python

WOMEN WHO
CODE®
/python

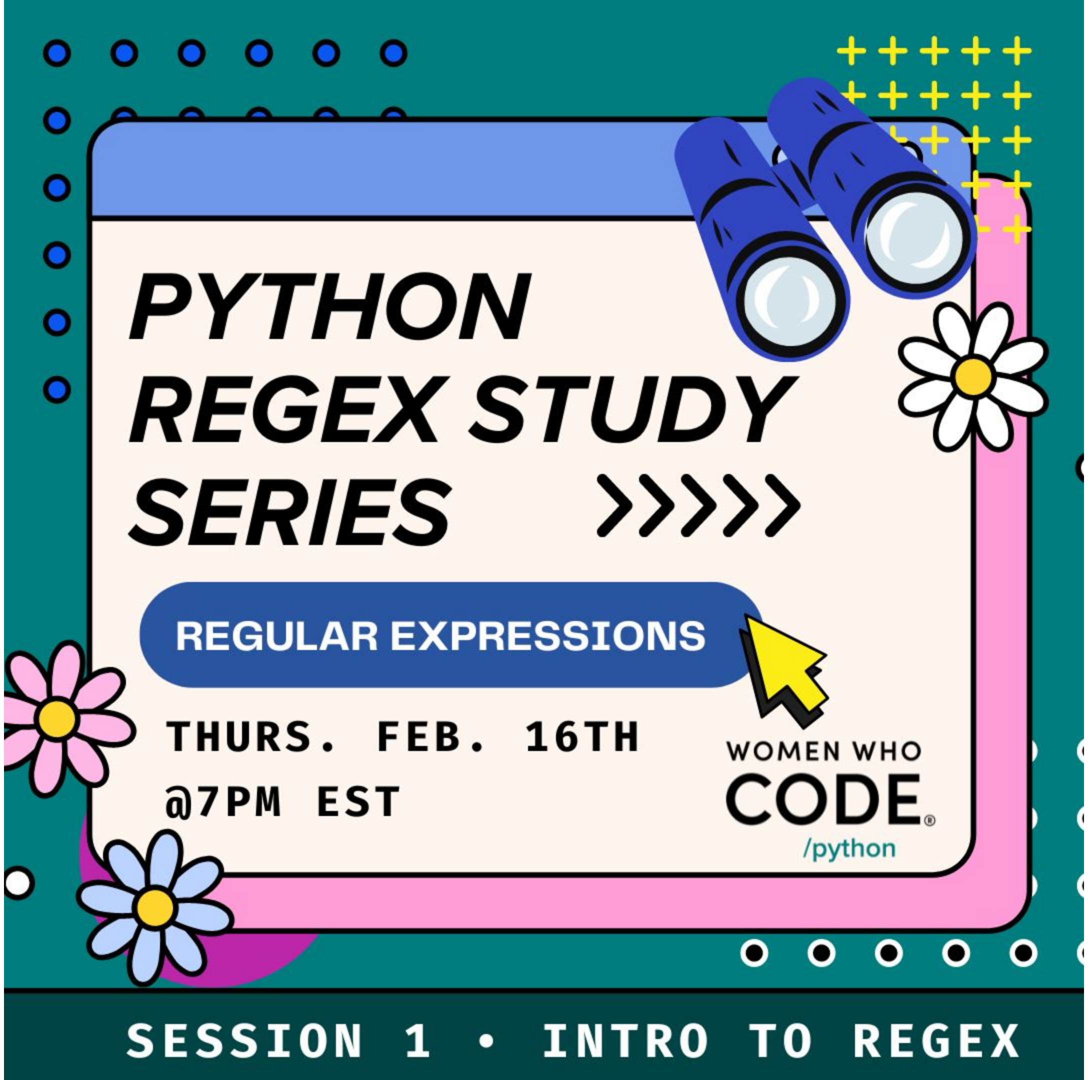
# /stay connected

Join Women Who Code Python on Slack

Find us on LinkedIn, Twitter, Instagram & Facebook:
**@WWCodePython**

Find links to join our community and follow us:
`beacons.ai/WWCodePython`

WOMEN WHO
CODE®
/python



# PYTHON REGEX STUDY SERIES

REGULAR EXPRESSIONS

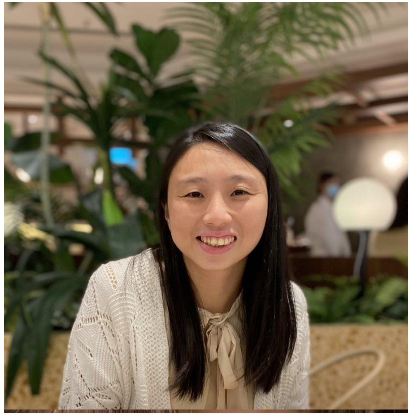
THURS. FEB. 16TH
@7PM EST

WOMEN WHO CODE
/python

SESSION 1 • INTRO TO REGEX

WOMEN WHO CODE
/python

# MEET THE TEAM



**Karen Wong**
Senior Programmer
R&D Company

**Mansi Jain**
MS in DS
Indiana University

**Katerina Hanson**
Eng Manager
Kojo

# Our Speaker

## *Speaker name here*

A short speaker bio goes here.

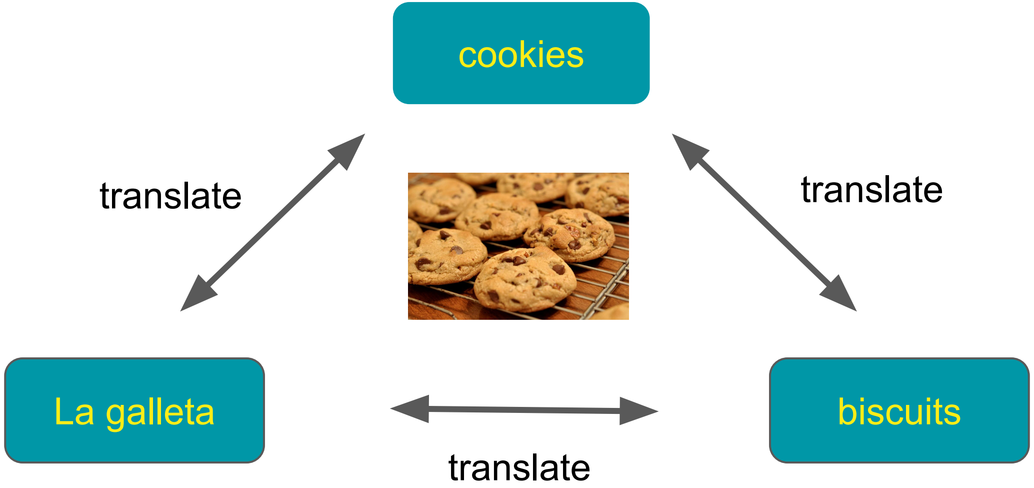
Add a headshot of the speaker on the right.

# Introduction of Regular Expression

WOMEN WHO CODE® /python

# *What is Regular Expression?*

# Language



cookies

translate

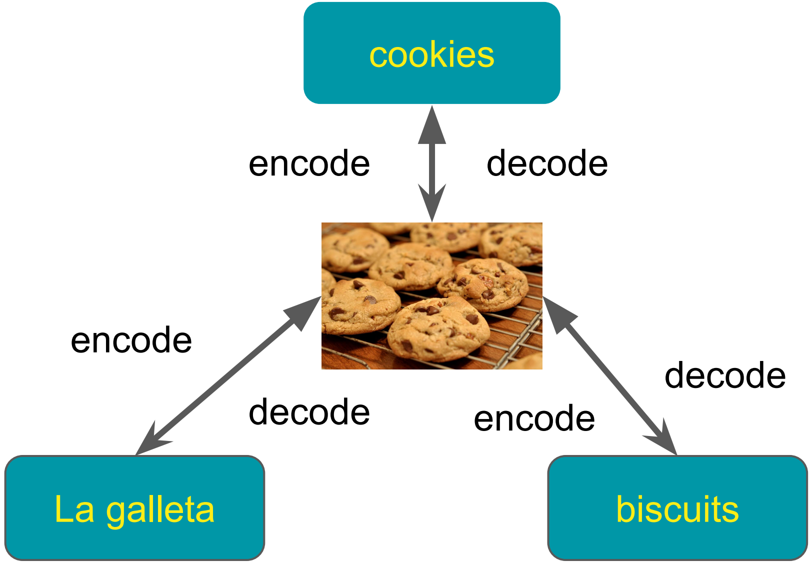
translate

La galleta

biscuits

translate

# Language with Machine



cookies

encode

decode

encode

decode

decode

encode

La galleta

biscuits

# Regular Language

- Set of rules to find specific patterns that can be recognized in a Finite State Machine
- Finite State Machine (FSM)?
  - Computational memory of a FSM is limited
  - Any languages that cannot be recognized by FSM cannot be considered as regular language

**A*B***

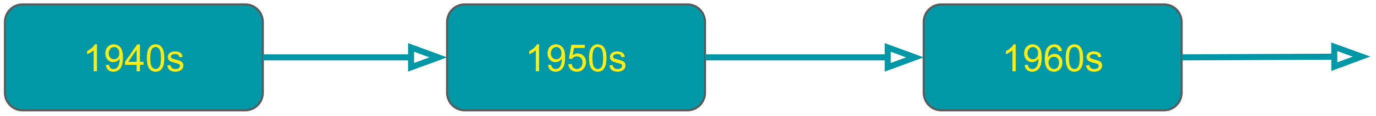
**(a+b-c)***

# So What is Regular Expression?

- Set of specially encoded text strings used as pattern for matching set of strings
- A notation of regular expression

```
[^ ]*?@[^ ]*?\.[^ ]*
```

# A Brief History of Regex



1940s

1950s

1960s

With implementations with different groups dialects of regular expressions are born

Neuroscientist Warren McCulloch and logician Walter Pitts published first mathematical model an artificial neuron

Mathematician Stephen Kleene described the model in algebra -> regular expression

Ken Thompson implemented regular expression on text editors searches

# A Brief History of Regex

Various dialects are transformed from the original regular expression

- grep
- egrep
- sed
- awk

| Feature | Modern grep | Modern egrep | awk |
|---|---|---|---|
| *, ^, $, [...] | same | same | same |
| ? + \| | \? \+ \\| | ? + \| | ? + \| |
| Word boundary | · | \< \> | · |
| grouping | \(...\) | (...) | (...) |

# Regex Engine

- **Just like how cars can be built differently, a regex engine reflects how patterns are exactly matched with the search text and its computational time**

| Engine Type | Example |
|---|---|
| DFA | Awk, egrep, lex |
| NFA | Tel, Perl, Python, GNU Emacs, sed |

# The Power of Regex

- Lots of valuable information are lying within textual data
- Regex will be the fundamental technology for text data exploration and analysis

**With regex, you become more powerful in retrieving any textual data**

# Understanding Regex

| Pattern | What the computer does | Examples |
|---|---|---|
| cookie | Start scanning from the beginning of the line, find the pattern where c follows immediately with o, o, k, i, e until the end of line | ✅chocolate chip cookie<br>❌shortbread ookie |
| ^coo | The pattern of coo literally is found at the beginning of the line | ✅cookie bars<br>❌chocolate chip cookie |
| cookie$ | The pattern of cookie matches literally at the end of line | ✅chocolate chip cookie<br>❌cookie bars |

WOMEN WHO CODE®
/python

# Types of Characters in Regex

**Metacharacter**

- Special characters that represent regular expression patterns
- *, ?, $

**Metasequence**

- Sets of characters that represent regular expression patterns
- \w, \d

**Literal Text**

- Like normal text characters

# Looks Simple Eh?

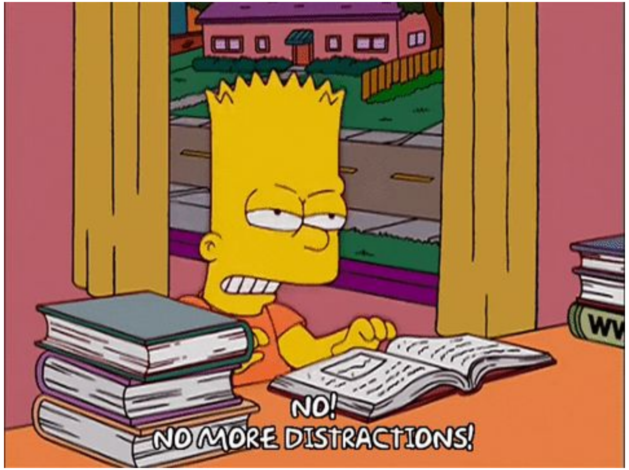
*But...*

- Syntax takes a bit of time to digest
- Takes TONS of practices to fully understand
- Complexity (usually) increases when the pattern is more inclusive

```
^(?:(?:31(\/|-|\.)(?:0?[13578]|1[02]|(?:Jan|Mar|May|Jul|Aug|Oct|Dec)))\1|(?:(?:29|30)(\/|-|\.)(?:0?[1,3-9]|1[0-2]|(?:Jan|Mar|Apr|May|Jun
|Jul|Aug|Sep|Oct|Nov|Dec))\2))(?:(?:1[6-9]|[2-9]\d)?\d{2})$|^(?:29(\/|-|\.)(?:0?2|(?:Feb))\3(?:(?:(?:1[6-9]|[2-9]\d)?(?:0[48]|[2468][048]
|[13579][26])|(?:(?:16|[2468][048]|[3579][26])00))))$|^(?:0?[1-9]|1\d|2[0-8])(\/|-|\.)(?:(?:0?[1-9]|(?:Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|
Sep))|(?:1[0-2]|(?:Oct|Nov|Dec)))\4(?:(?:1[6-9]|[2-9]\d)?\d{2})$
```

# Tips and Tricks

- Same as you learning a new language:
  - Spend some time to understand the rules
  - Practice practice practice!
- Space matters ! (especially sneaky ones at the beginning / end



# Q&A

# SECTION

WOMEN WHO
CODE®
/python

# Heading

Body

# *Heading*

Body

WOMEN WHO CODE®
/python

# Heading

Body