



# WELCOME!





python



# Hi, I'm Colt

I've lead in-person programming bootcamps in the Bay Area for a decade.

I've been a lead-instructor and curriculum director for multiple bootcamps including Galvanize and Rithm School where we guarantee students jobs.

I've taught millions of students to code online and was selected as the Best New Instructor on Udemy!



NEVER KNOW WHAT TO PUT UP HERE

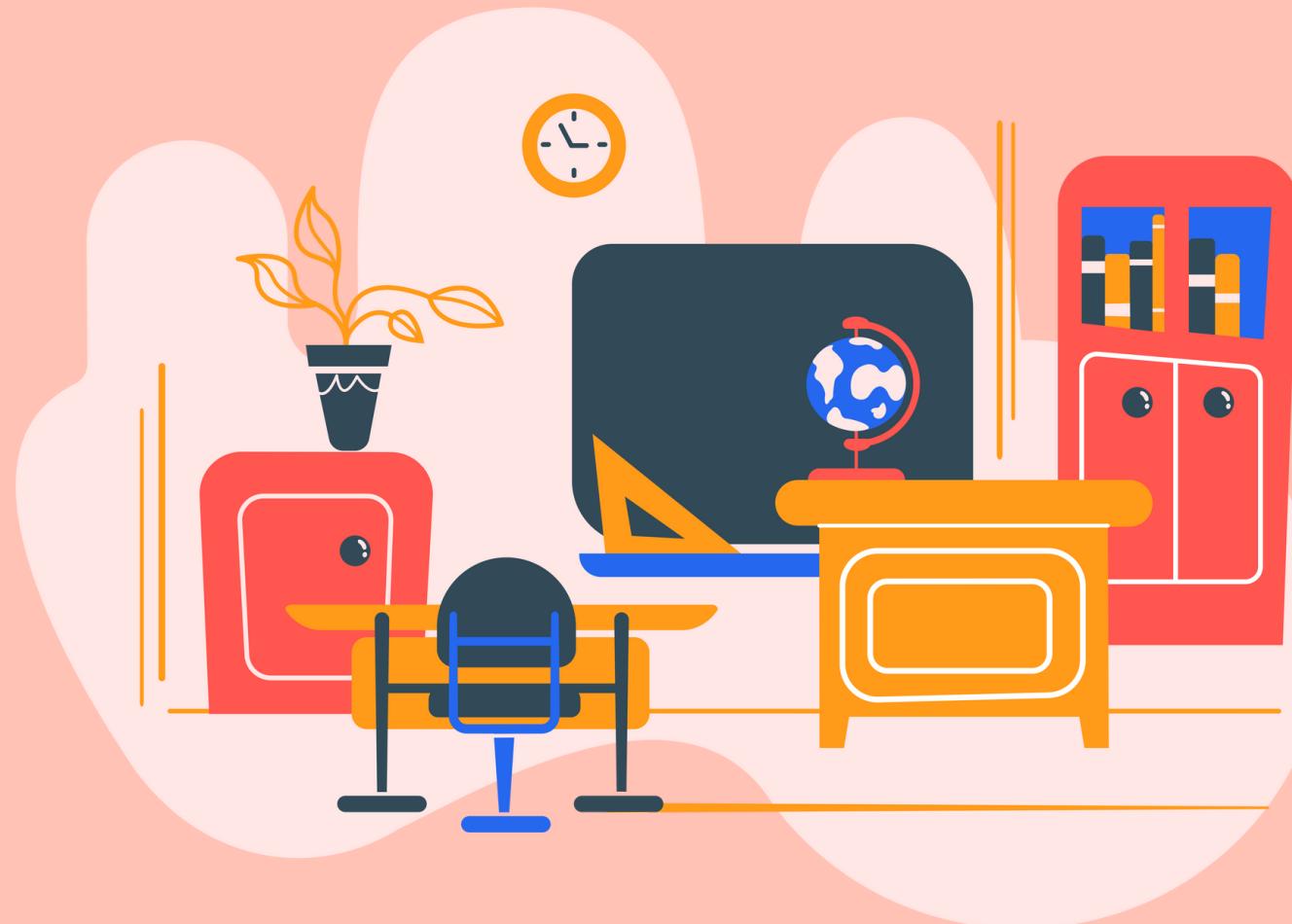
My students work at  
companies including...

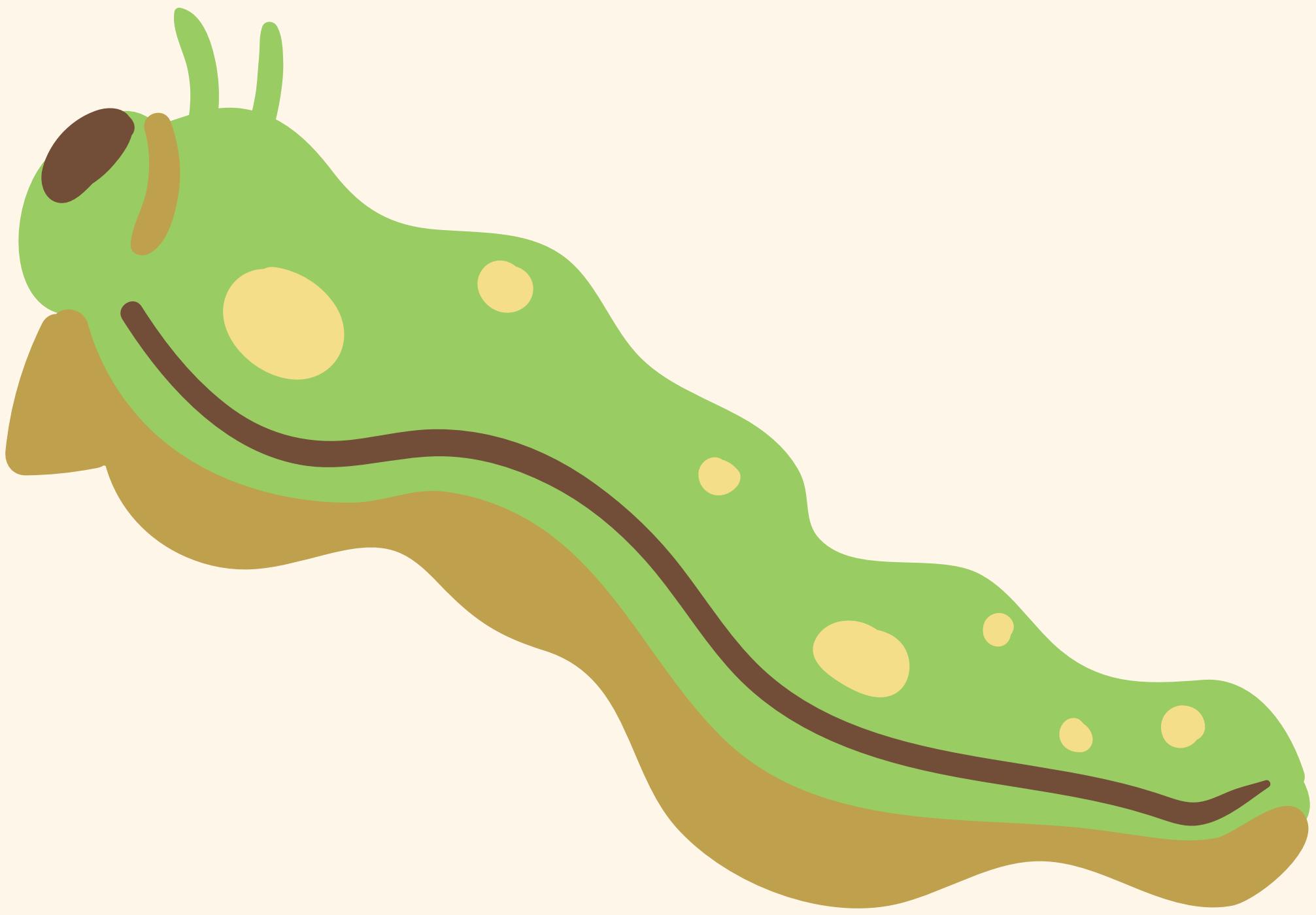


# What really matters...

Is that I've spent years trying to perfect teaching  
programming to beginners

I've learned how to keep students engaged from my  
years of in-person teaching experience.





# This Course Is...

- A fast, effective pathway to learn Python
- Full of exercises/challenges/quizzes
- Made for normal people who can't stand 20 minute videos. Average video is 4 minutes.
- An ideal on-ramp to data science, ML, or web development courses.



# WHAT THIS COURSE IS NOT



# This course isn't

- A 60 hour Python encyclopedia course that covers every possible feature of Python
- An advanced level Python course for experts
- A web development, data science, or machine learning oriented course



# High Demand

There are more Python jobs today than ever before, and salaries are at an all-time high. If you could only pick one language to learn, it should be Python.



# Huge Community

Python has been around for 30+ years now, and in that time a massive, supportive online community has formed. You can always get help and find answers (relatively) easily.



# Libraries

There are thousands of Python libraries, frameworks and tools available. From django to matplotlib to scikitlearn, there's usually something for every need.



# Easy To Learn

Python is (in my opinion) the easiest programming language to learn. It has far fewer quirks and oddities than languages like JavaScript.



# Versatility

You can use Python for simple scripting tasks.

You can use it to build huge web applications.

You can use it for data processing and analysis.

And of course, you can use it for machine learning!



# What can we build with python



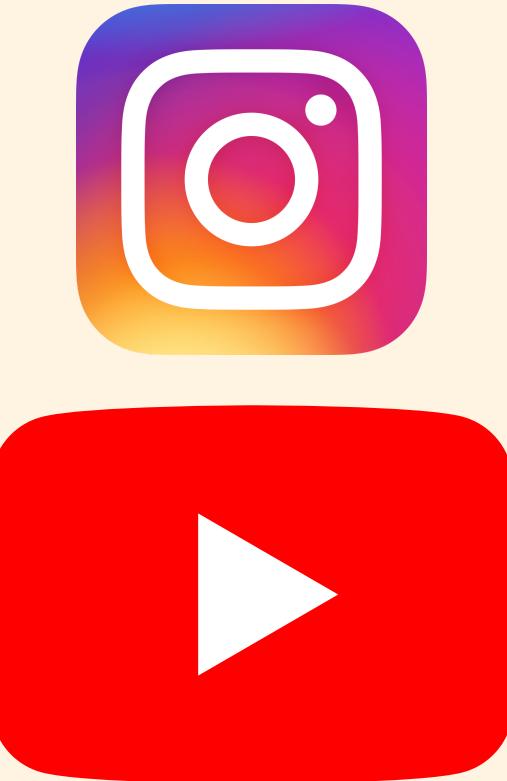
# Game Dev

Hugely popular games including the Sims 4, Civilization, EVE Online, and many others are built using Python!



# Web Dev

Companies ranging from tiny startups to massive unicorns use Python to build web applications.



# Data Stuff

Python is hugely popular in the world of data science and data analysis. Python tools like numpy, pandas, and matplotlib are widely used.



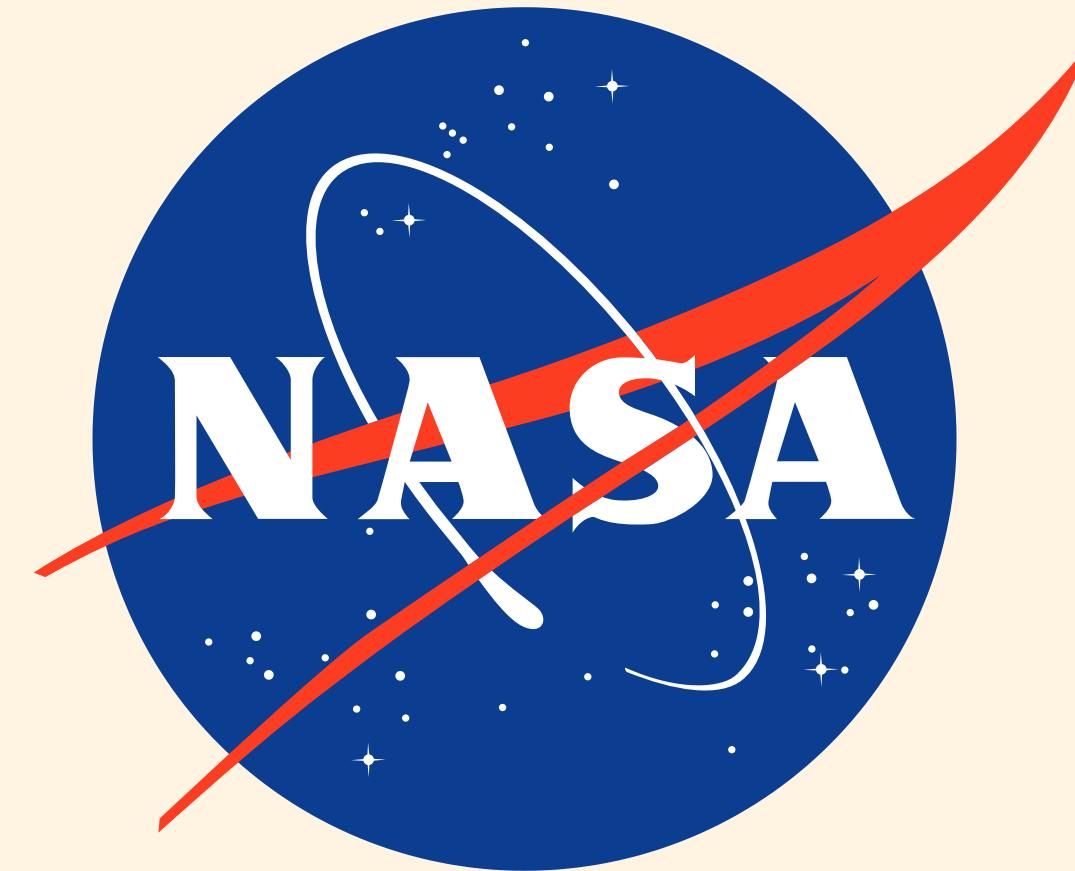
# Machine Learning

Python is "the" language for machine learning. Tools including TensorFlow, Scikitlearn, OpenCV, and NLTK require Python knowledge



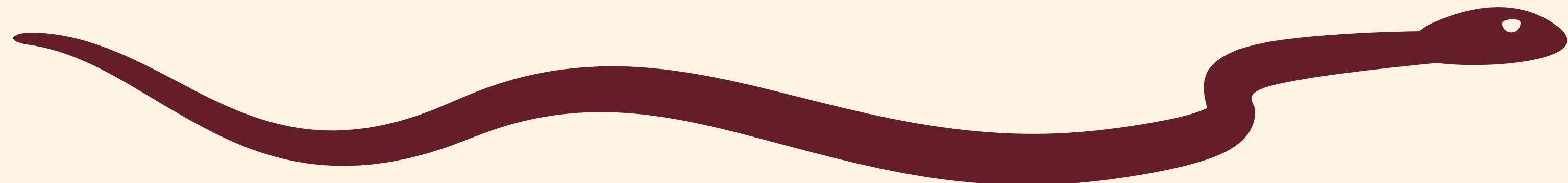
# Science

Python is used across academia and the science world, from biology research labs to NASA engineering teams





# A Tiny Little Bit Of Python History



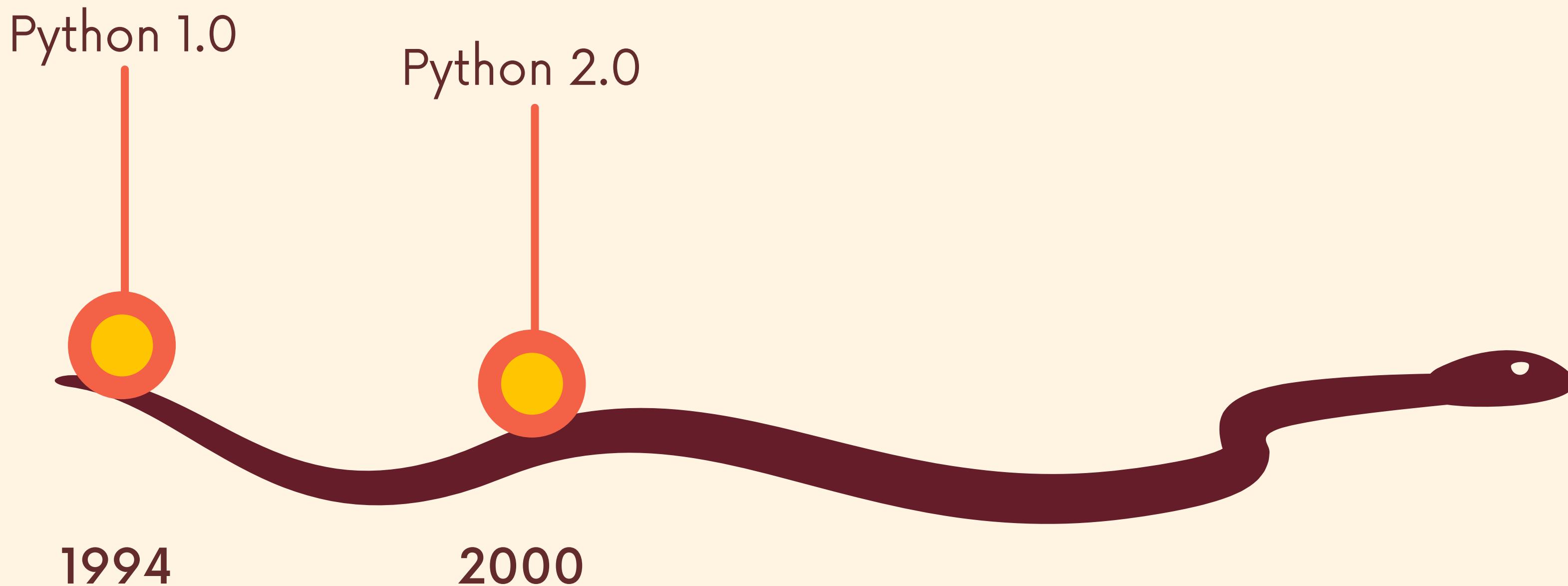
# A Tiny Little Bit Of Python History

Python 1.0

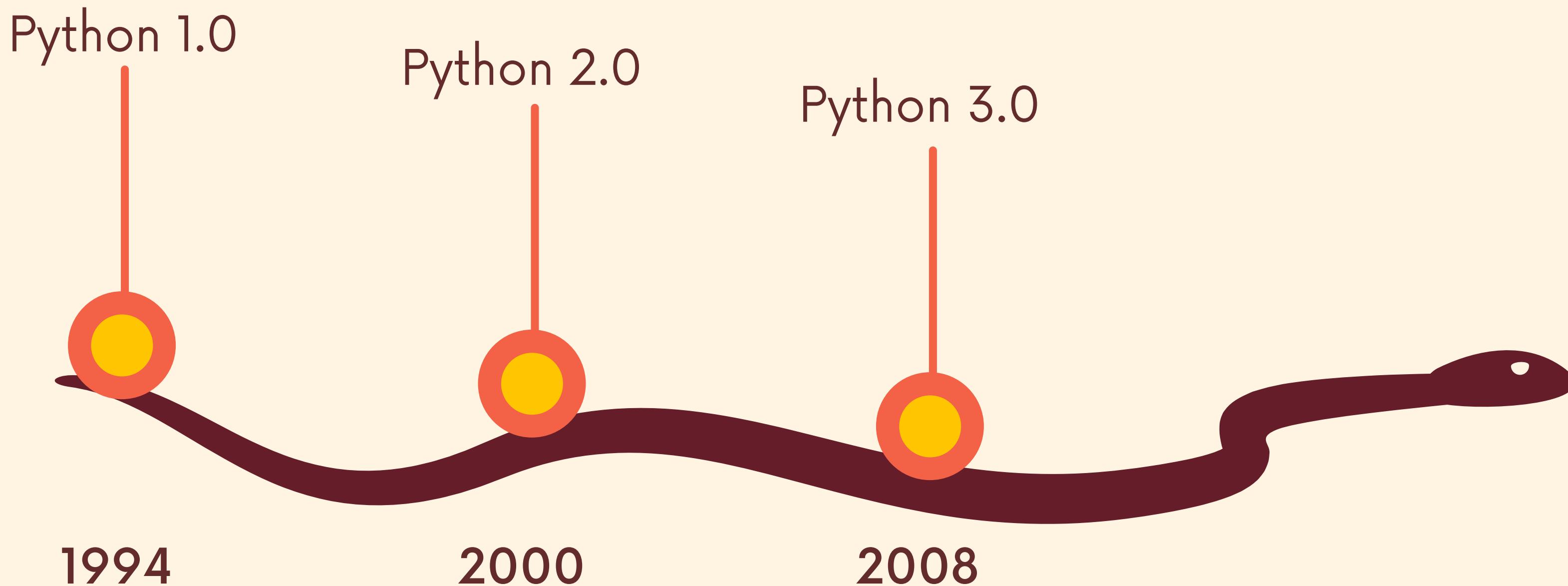


1994

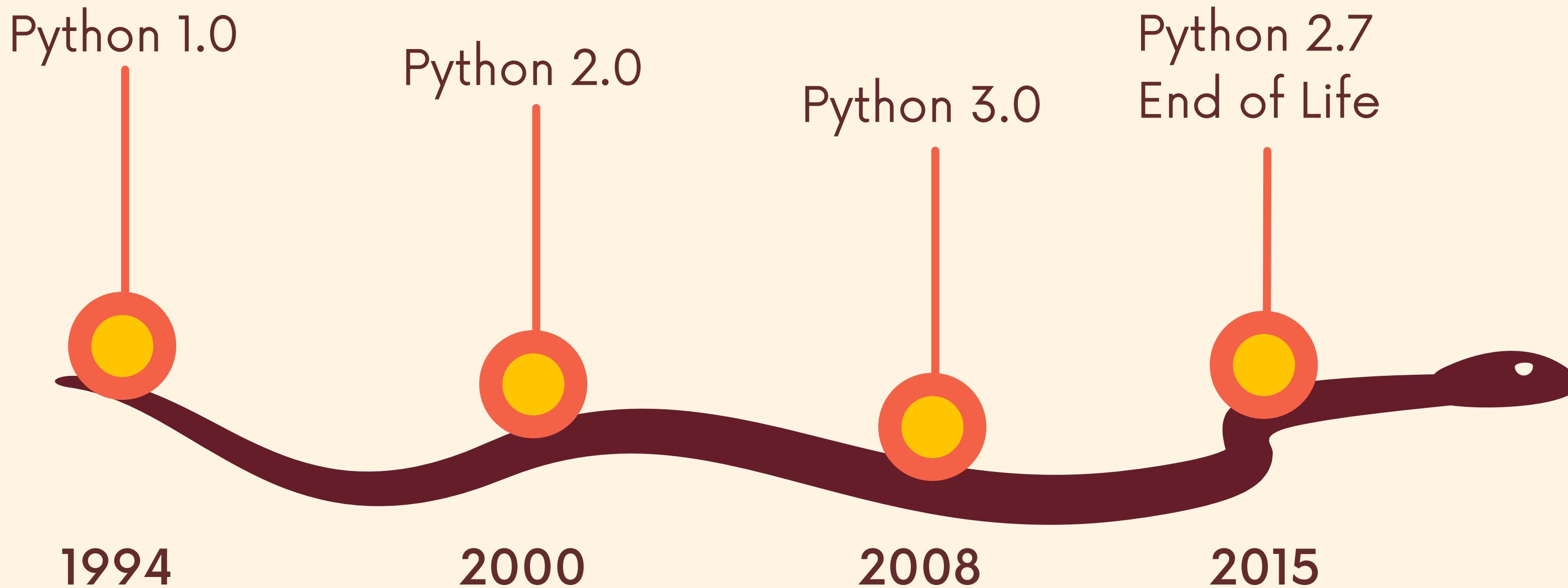
# A Tiny Little Bit Of Python History



# A Tiny Little Bit Of Python History



# A Tiny Little Bit Of Python History



A Tiny Little Bit Of

# Python History

Python 1.0

Python 2.0

Python 3.0

Python 2.7  
End of Life

Python 2  
End of Life

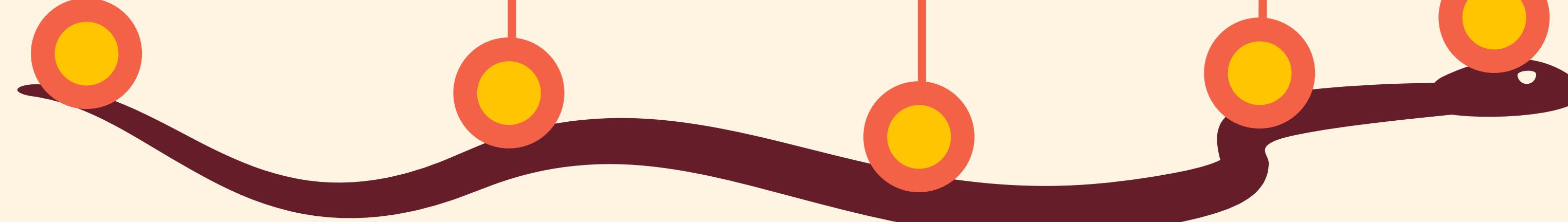
1994

2000

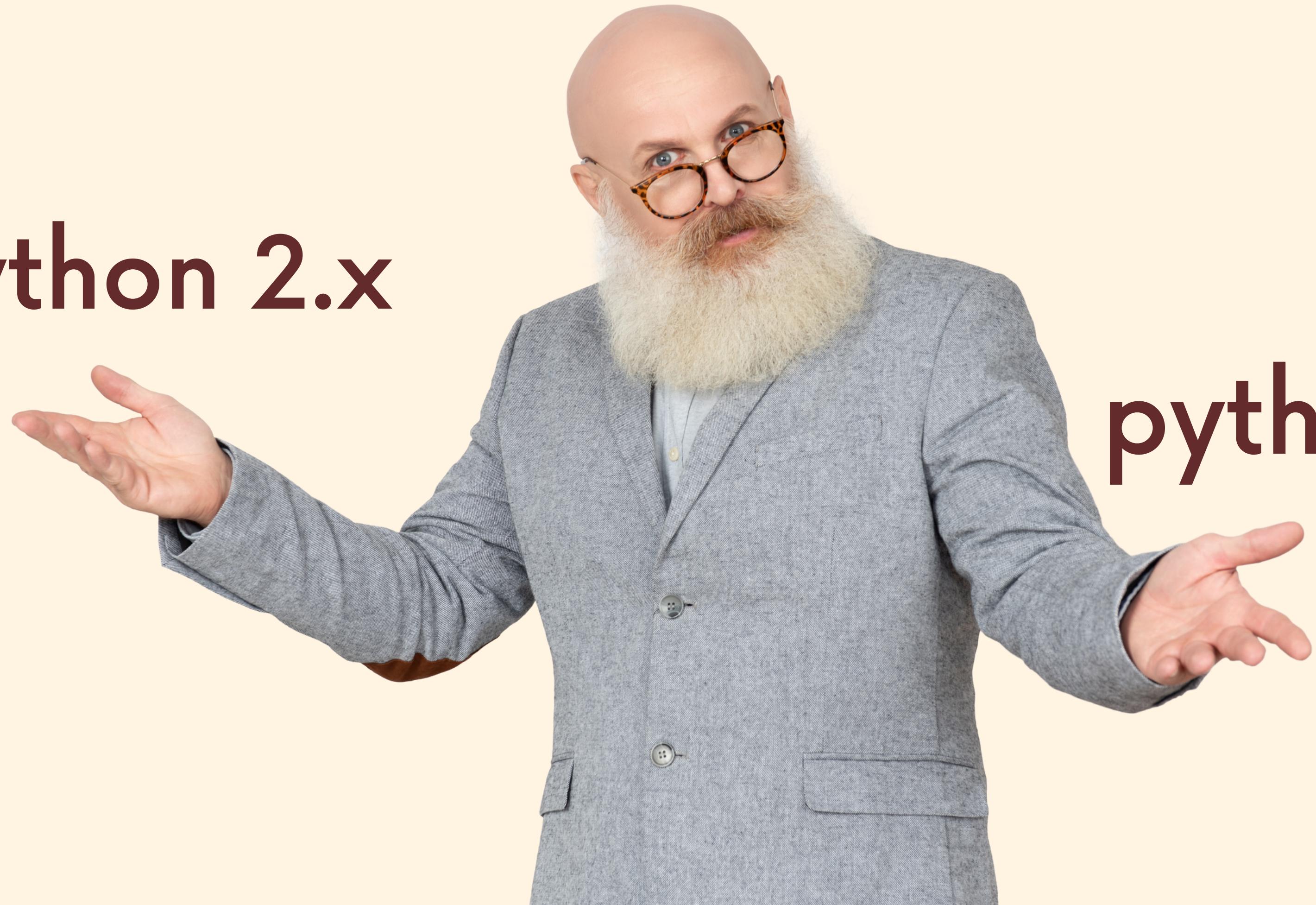
2008

2015

2020



**python 2.x**

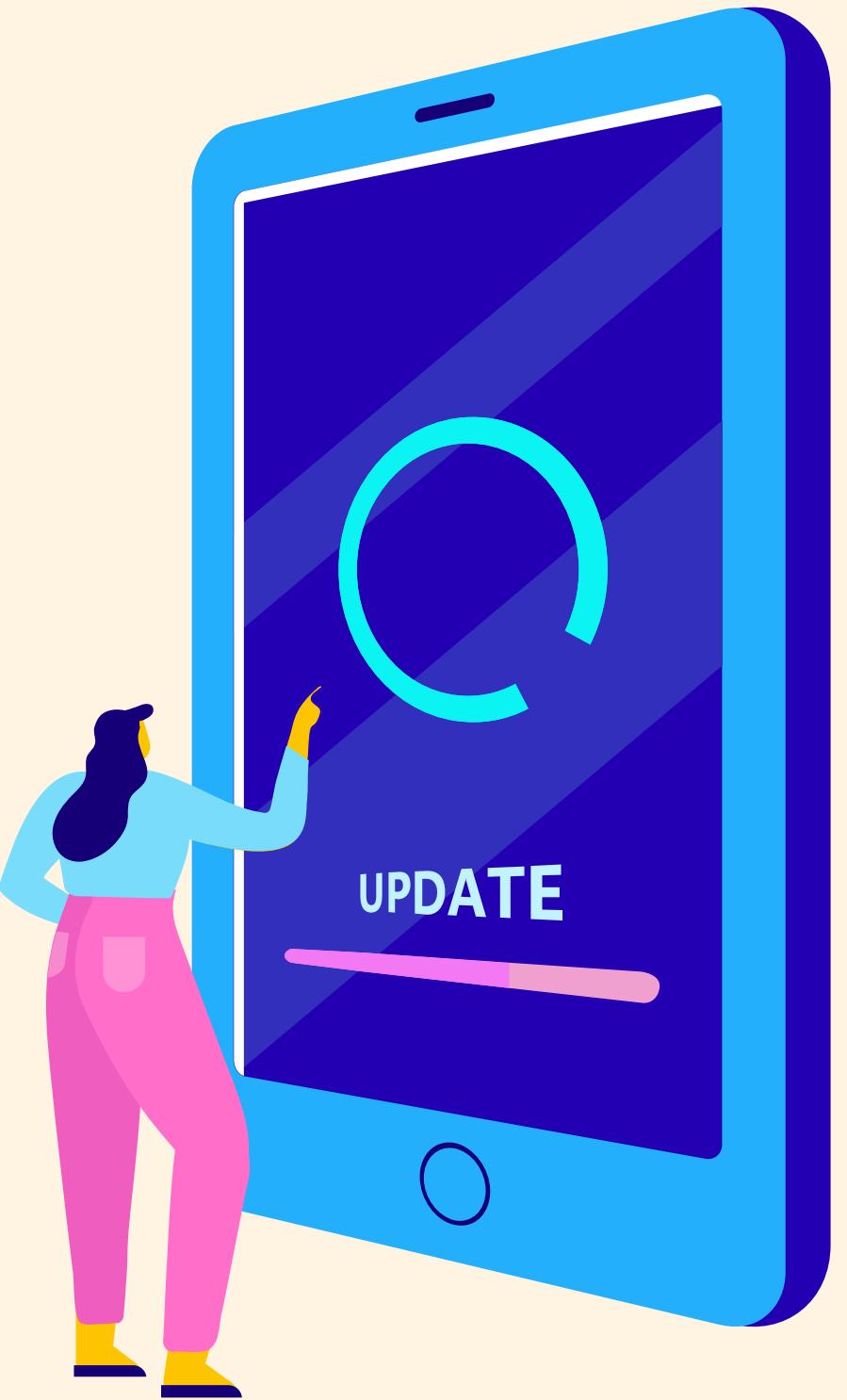


**python 3.x**

# python 2 vs. 3

**Python 3 was a major change**

- Not backwards compatible with python 2
- It included some syntax changes
- Featured major changes under the hood
- It took 10+ years for Python 2 to be officially declared dead



# python 2.x

```
>>> print "Hello"  
Hello
```

```
>>> 5/2  
2
```

# python 3.x

```
>>> print("Hello")  
Hello
```

```
>>> 5/2  
2.5
```

# Who Cares?

At this point, there is no real reason to learn 2.x, but it's worth knowing the history because...

- Lots of the tutorials online are for 2.x
- Your computer may have 2.x pre-installed
- Some dead libraries/tools still haven't been updated to work with 3.x



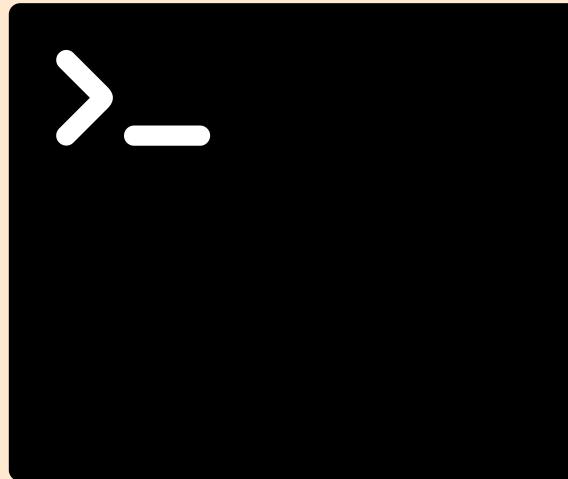


# Installation

(it's actually not bad!)

# Running Python

There are two main ways we can run Python code



Interactively



From A File



# Interactive Python

In interactive mode, any code you enter is immediately run.

Python executes each line as you enter it and then displays the output.

Great for learning and trying things out, but not great for "real" applications



```
>>> 1 + 2
3
>>> "heyyoo" * 3
' heyyooheyyooheyyoo'
>>>
```

# Python Scripts

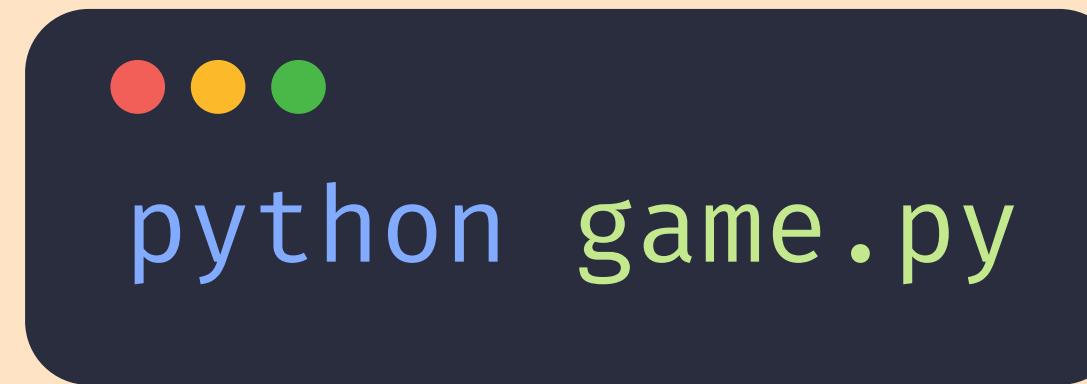
Alternatively, you can write python code in a file.



game.py



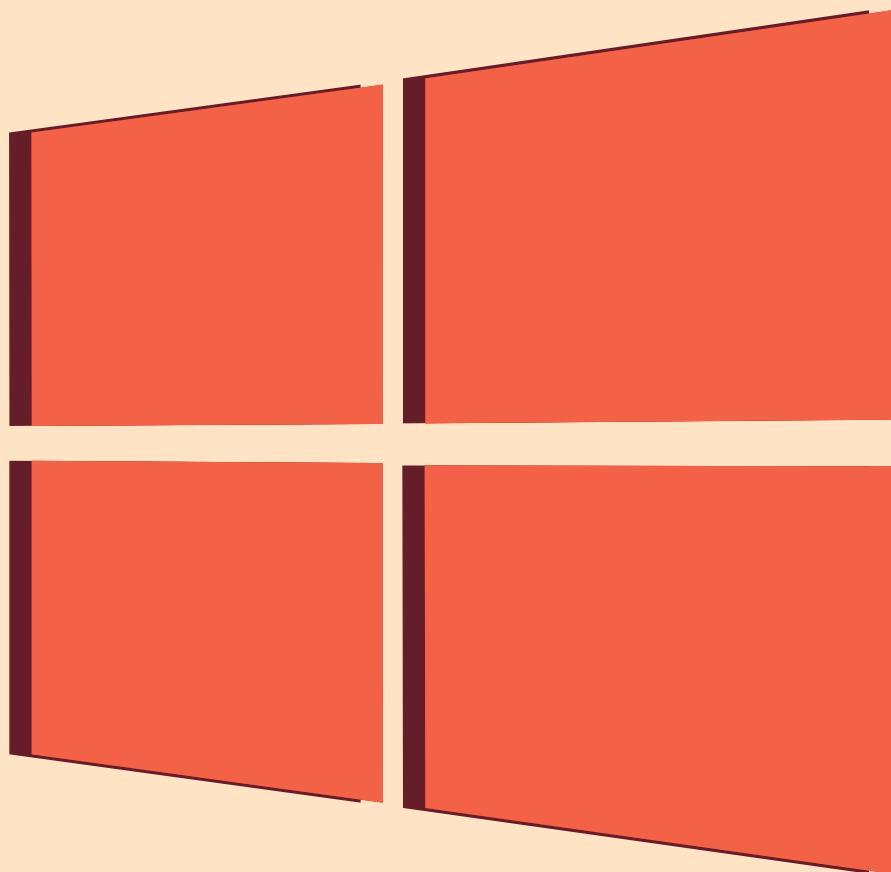
And then you can separately execute that script with Python



The code only runs when you manually tell it to.



Mac  
Install



# Windows Install



Visual Studio Code

The "I want to  
get started right  
now" option



repol.it