

Python

beginner to CRUD Maven

About me

- CTO of RESTORE-Skills, gamification platform for physical and occupational therapy in older adults.
- 20+ years of dev experience
- Developing in Python since 2016
- 1 year of software bootcamp instruction
- 20+ years of dad and husband experience



Curriculum

Curriculum

- Intro and getting started
- Python basics
- Functions, modules and decorators
- Testing
- Pydantic
- CRUD magic with FastAPI



Intro and getting started

Intro and getting started

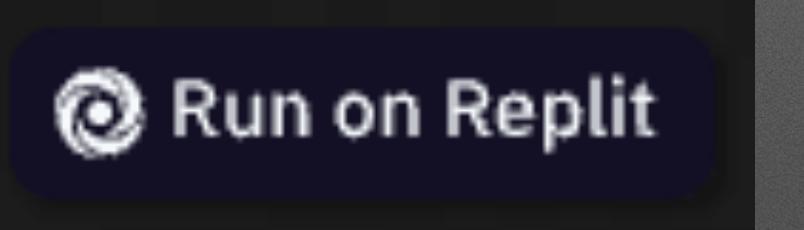
- About Python
- Launch learning materials from GitHub repo
- IDE environment overview
- “Hello, our robot overlords!” - getting environment to sing

About Python

- Conceived by Guido van Possum in the 80's
- It's implementation began in Dec 1989
- Multi-paradigm language supporting: OOP, structured programming, many functional programming features and AOP
- [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))



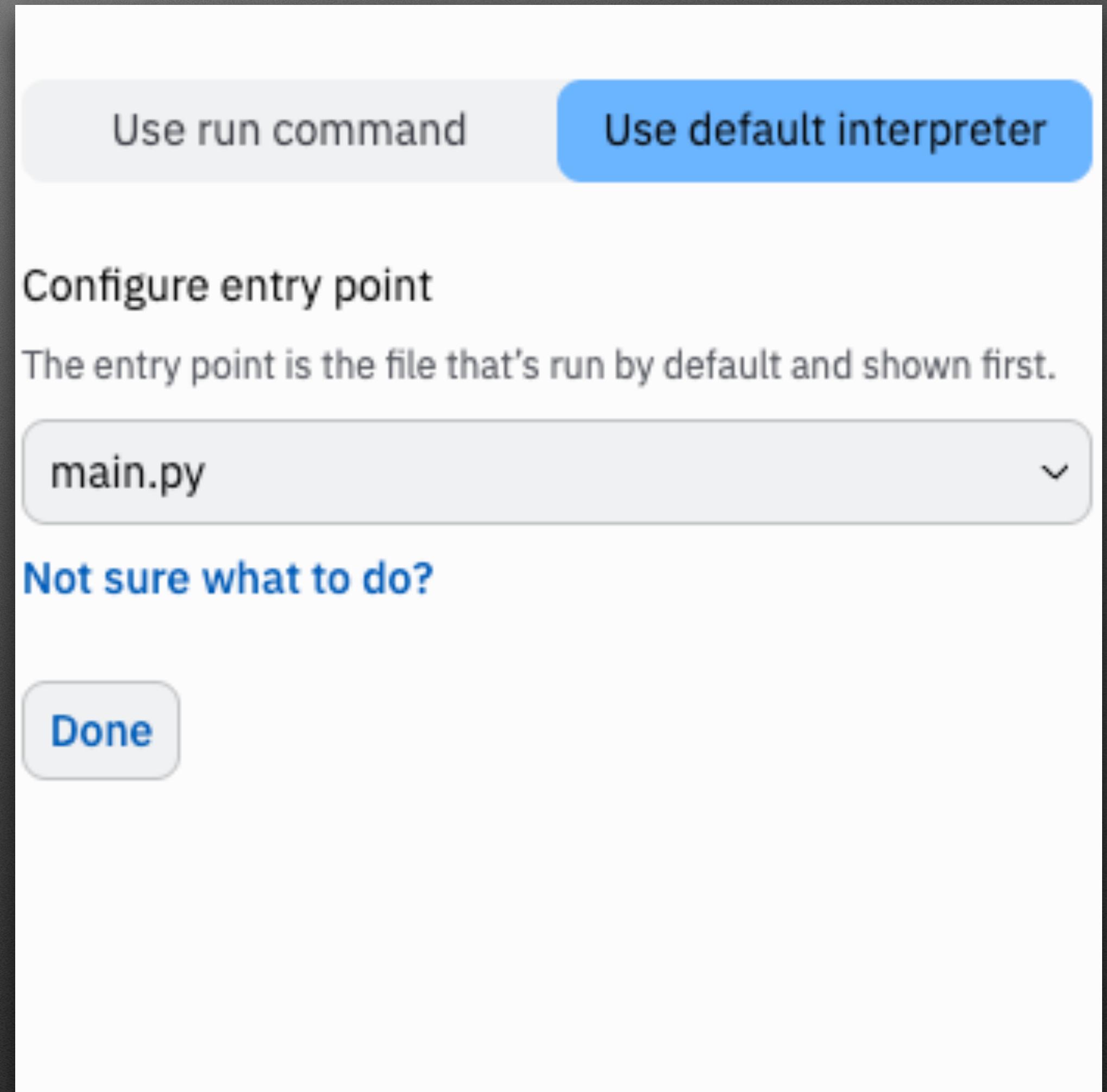
Launch learning materials from GitHub

- <http://bit.ly/3Z3EIUX>
- Click on ==>  Run on Replit
- Let the environment pull and load GitHub repo
- Let's check out our surroundings

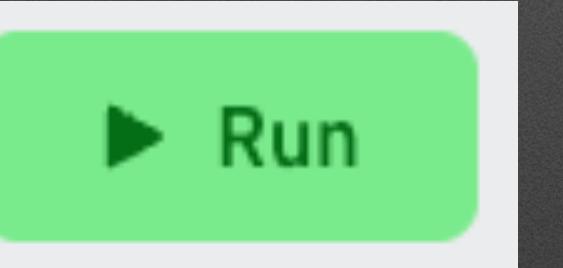


IDE environment overview

- Sign up for free replit.com account
- Select **main.py** from dropdown to get us started
- Take a look at **Console** and **Shell** tabs on the right. We are going to be observing Python execution and running some of the code from there.



Getting Environment to sing

- Click on **main.py** in the file list on the left
- Enter `print("Hello, our AI overlords!")`
- Hit this button on top ==> 



Python Basics

Python Basics

- Data
- Flow control
- Collections

Data

- Variables
- Strings
 - Multiline strings
 - fstrings
- date
 - date math
- dynamic strong typing
 - install and run **mypy**
 - type hint example



Flow control

- IF statements
 - turneries
- loops
 - for range (0, n)
 - “for each”
 - enumerate



Collections

- tuple
 - unrolling tuples
- list
 - comprehension
 - list splicing `test[start:end:step]`
 - filtering
- dictionary
 - comprehension
- set
 - set math



functions, modules and decorators

functions

- structure and type hints
- import standard library into modules
- build a password generator function from what we have learned so far

decorators

- basic AOP (Aspect Oriented Programming) concepts
- import performance decorator
- apply performance decorator
- pass in different params into function
- pass different params into decorator

Testing

Testing

- Setup pytest
- basic test conventions
 - `__init__.py`
 - test suite structure
 - running a test



Pydantic

Pydantic

Data validation and settings management using Python type annotations.

pydantic enforces type hints at runtime, and provides user friendly errors when data is invalid.

Define how data should be in pure, canonical Python; validate it with pydantic.

<https://docs.pydantic.dev/>

Pydantic

- build User model
- test User model
- basics
- serialization/deserialization
- custom validation



CRUD with FastAPI

CRUD with FastAPI



Created by : Sebastián Ramírez

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.

<https://fastapi.tiangolo.com/>

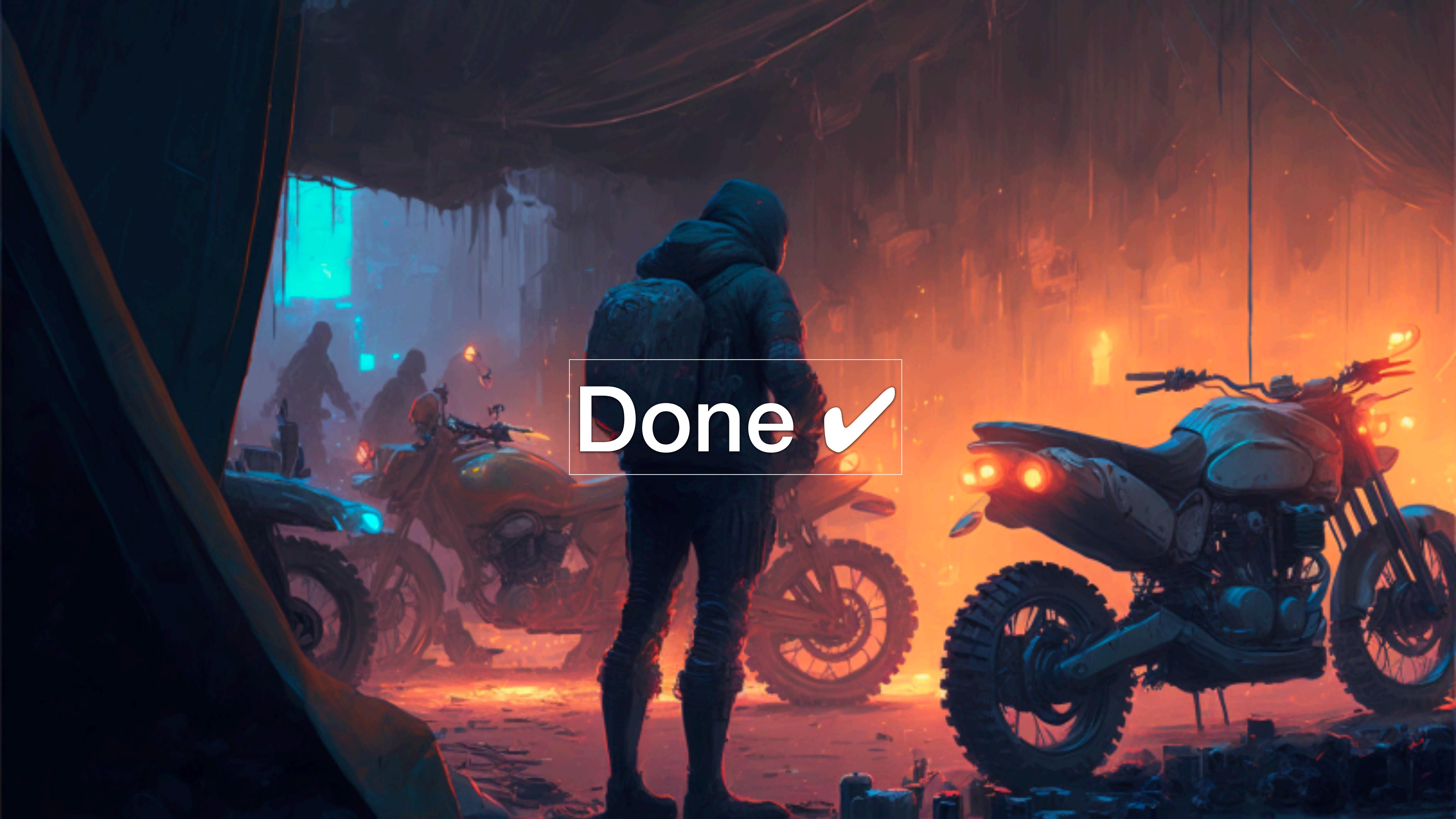


FastAPI

- Install **FastAPI**
- Install **unicorn**
- Create a “**Hello World**” **FastAPI** style
- Launch and access Swagger Docs

FastAPI

- Build a **GET** endpoint to pull funny quotes by ID
- Build **GET** endpoint to create username and password
- Build **POST** login endpoint taking in **User** model
- Secure our precious “quotes” endpoint with **JWT** based auth



Done ✓