EXPLORING PYTHON BYTECODE

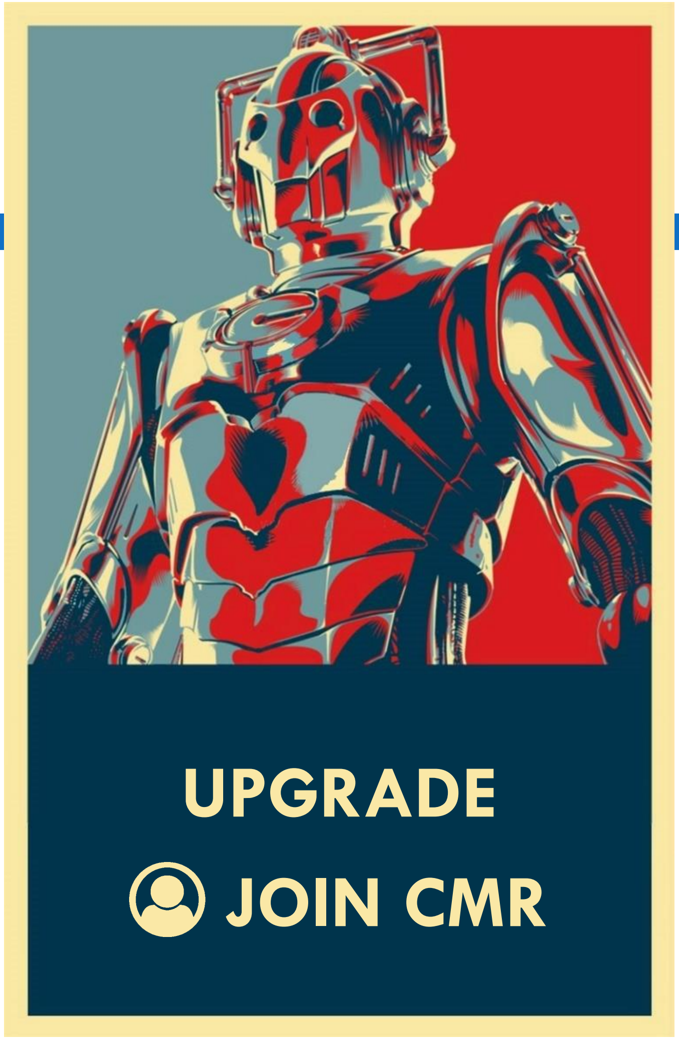
Jan Chwiejczak

www.github.com/janhak/bytecode

iamjanhak

whoami

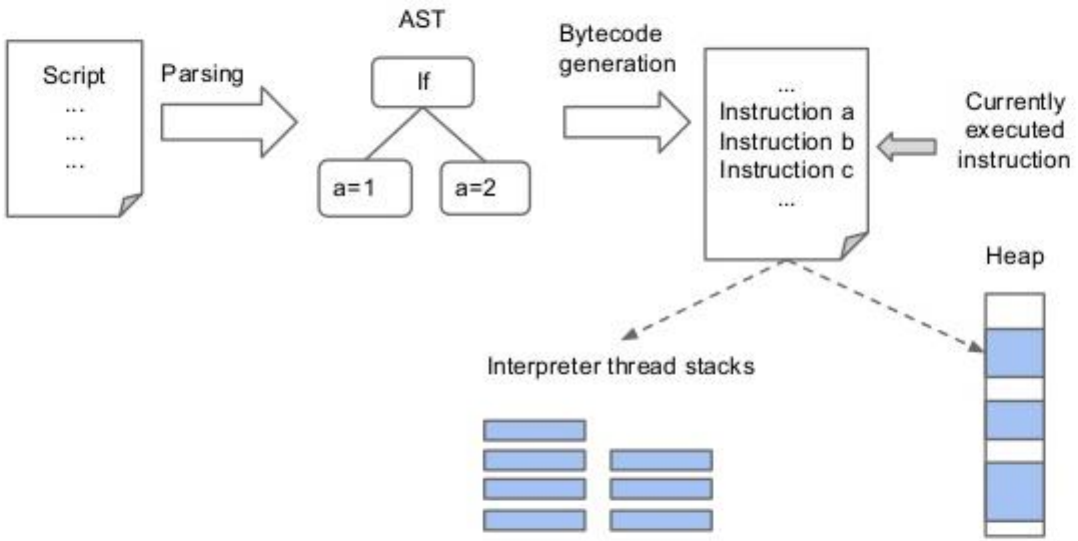
- Python Dev
- I work with robots
- Cambridge Medical Robotics
- iamjanhak
- janhak/bytecode



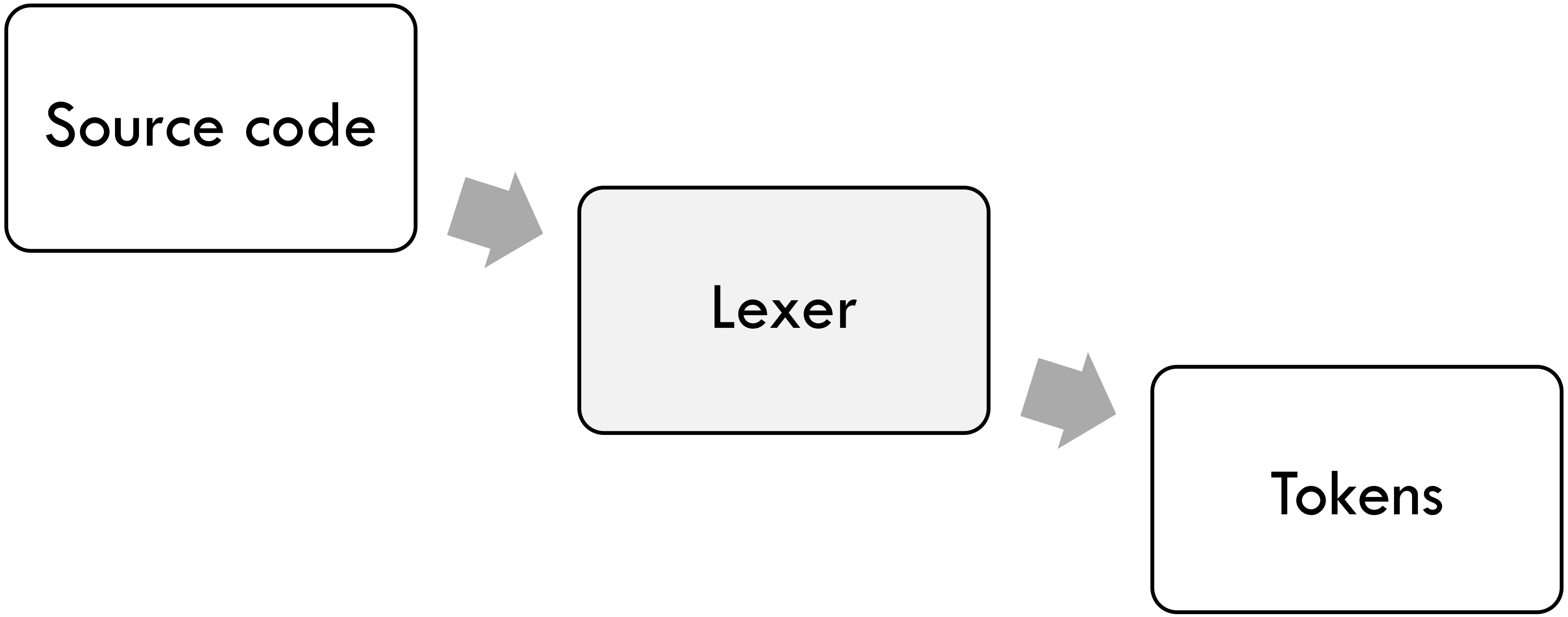
What I would like to explore

- Gain insight into how Python executes code
- Get hands on practice using the dis module to look at Python Bytecode
- For this talk by Python I mean *CPython*

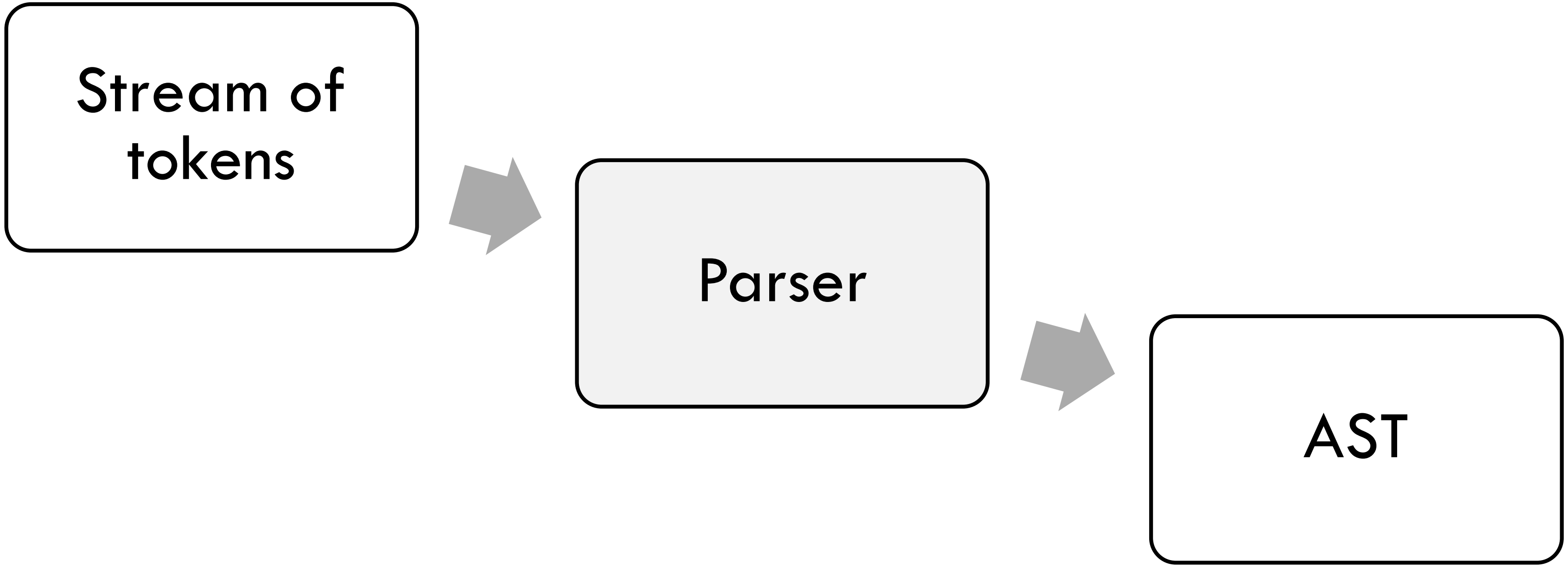
Python execution model



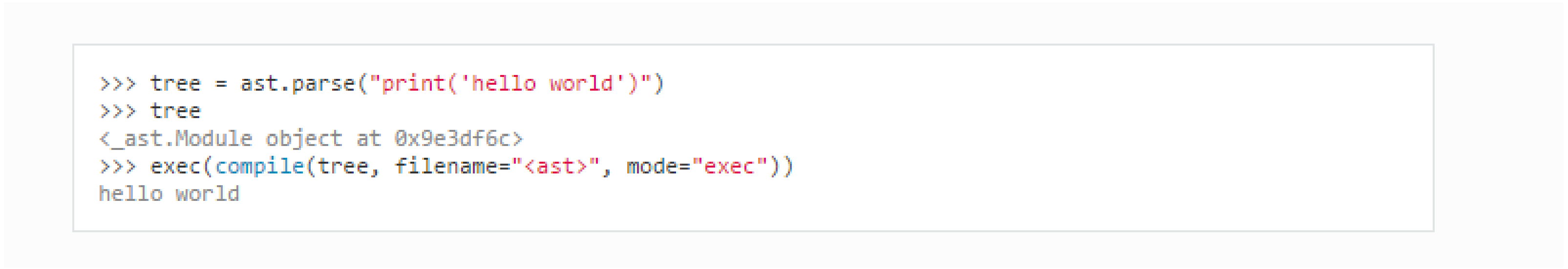
Lexing



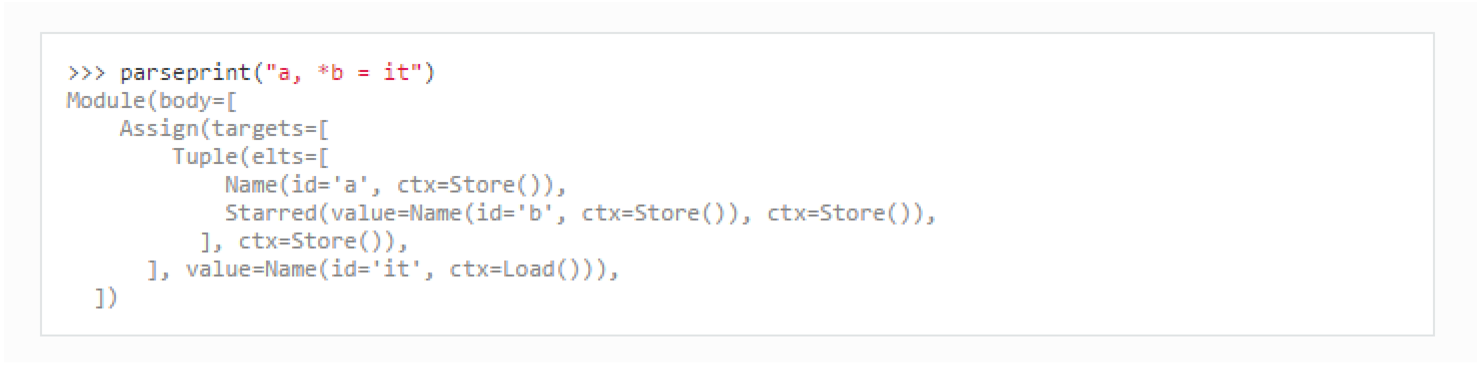
Parsing



Abstract Syntax Tree



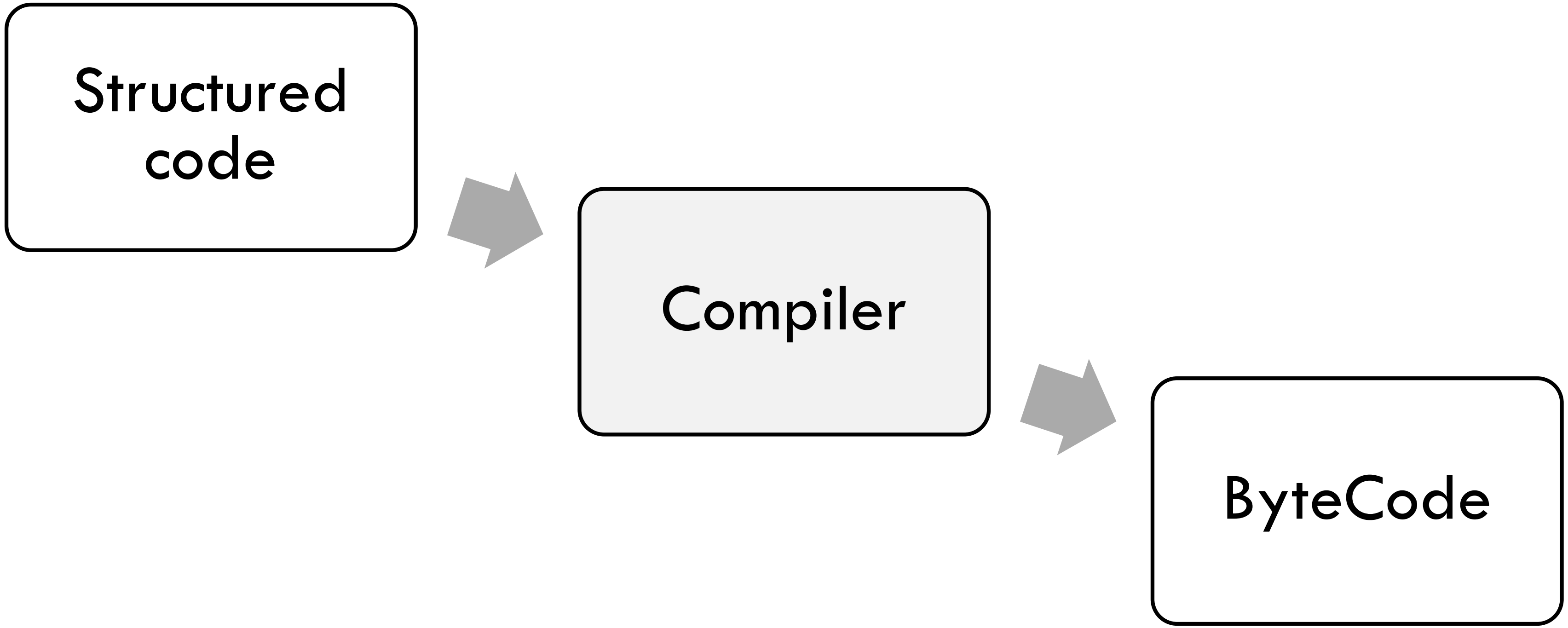
```
>>> tree = ast.parse("print('hello world')")
>>> tree
<_ast.Module object at 0x9e3df6c>
>>> exec(compile(tree, filename="<ast>", mode="exec"))
hello world
```



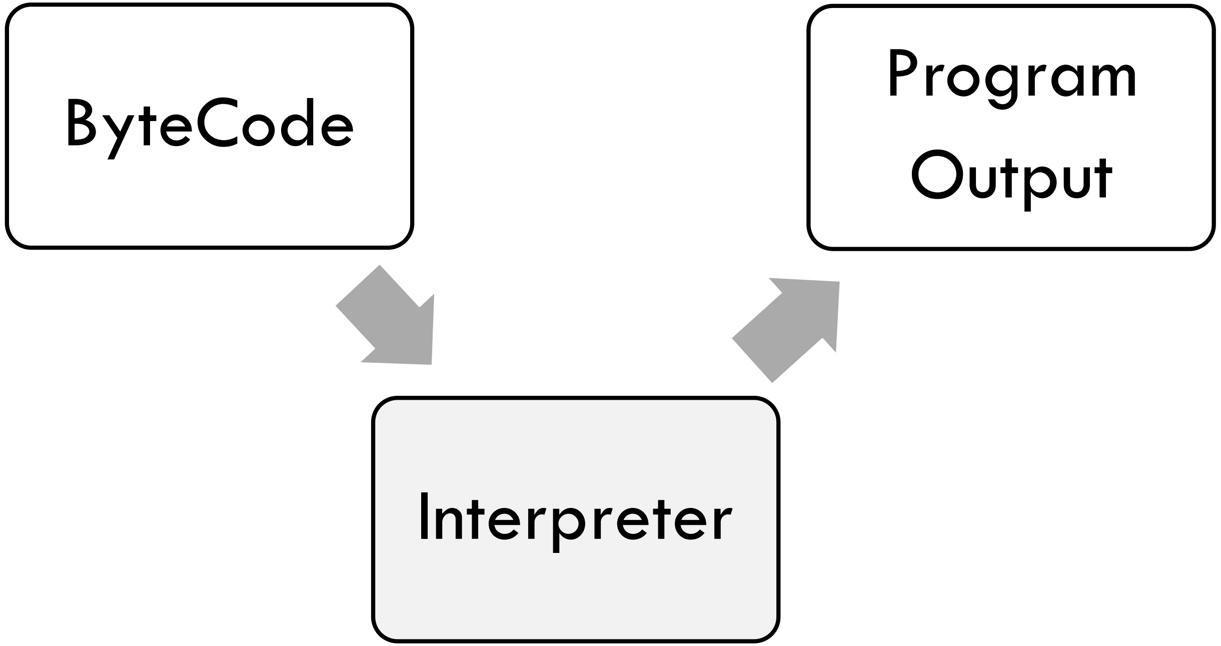
```
>>> parseprint("a, *b = it")
Module(body=[
    Assign(targets=[
        Tuple(elts=[
            Name(id='a', ctx=Store()),
            Starred(value=Name(id='b', ctx=Store()), ctx=Store()),
          ], ctx=Store()),
      ], value=Name(id='it', ctx=Load())),
  ])
```

missing docs https://greentreesnakes.readthedocs.io/

Compiling



Interpreter



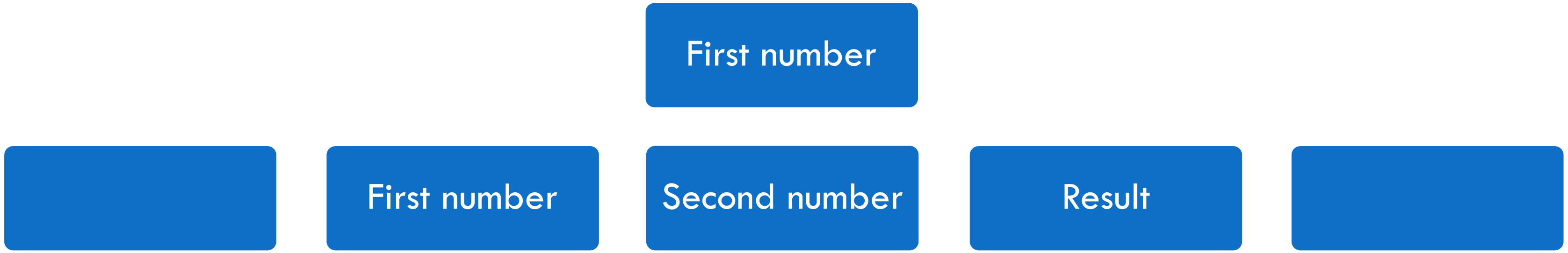
www.github.com/janhak/bytecode

Simple stack based interpreter

- Let's start with minimal interpreter that understands three instructions:
  - *LOAD_VALUE*
  - *ADD_TWO_VALUES*
  - *PRINT_ANSWER*

Interpreter Code Execution

- Suppose we want to execute “7 + 5”
  - *LOAD_VALUE - 0   # the first number*
  - *LOAD_VALUE - 1   # the second number*
  - *ADD_TWO_VALUES - None*
  - *PRINT_ANSWER - None*



What is bytecode?

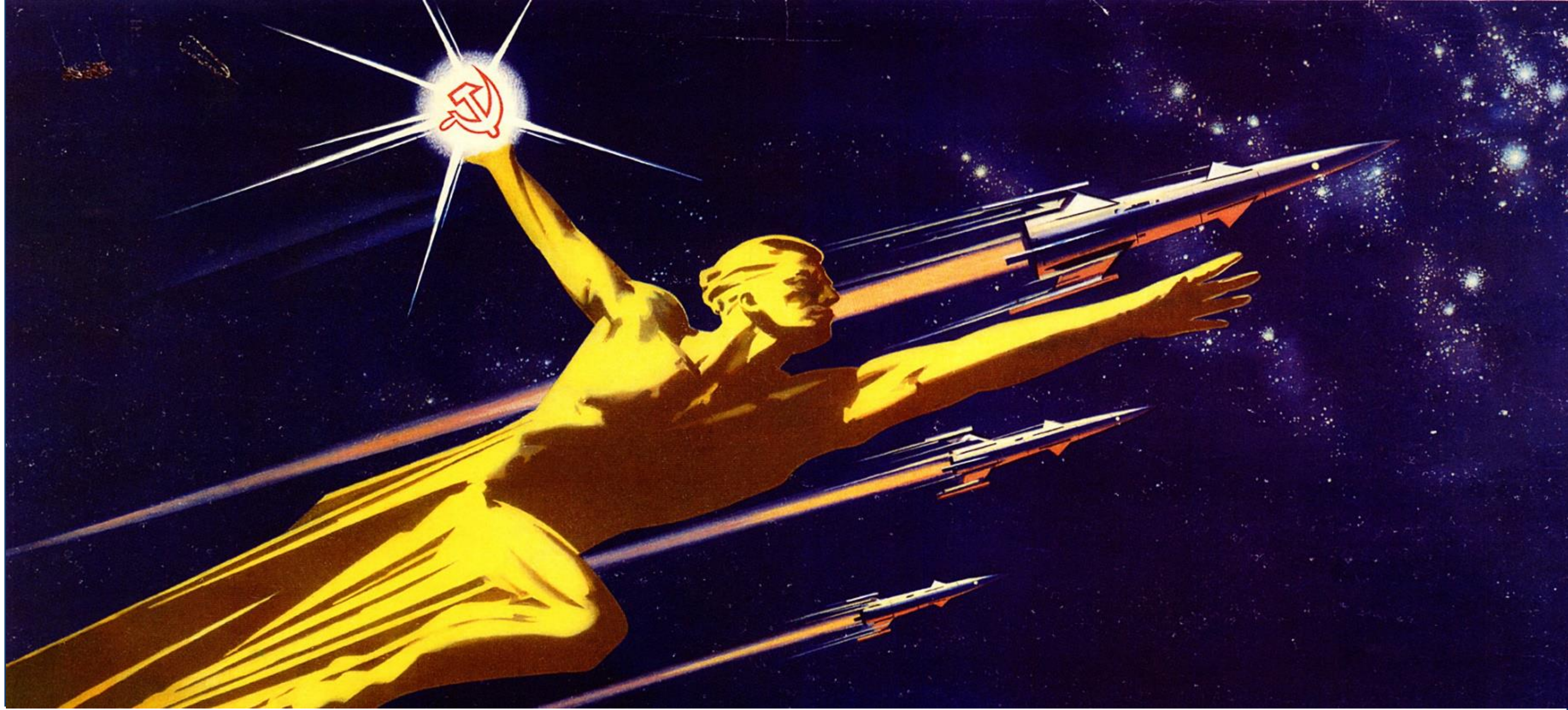
AN **INTERMEDIATE REPRESENTATION** OF YOUR PROGRAM

WHAT THE **INTERPRETER** WORKS WITH WHEN IT RUNS YOUR PROGRAM

MACHINE CODE FOR A VIRTUAL MACHINE

A SERIES OF **INSTRUCTIONS** FOR STACK OPERATIONS

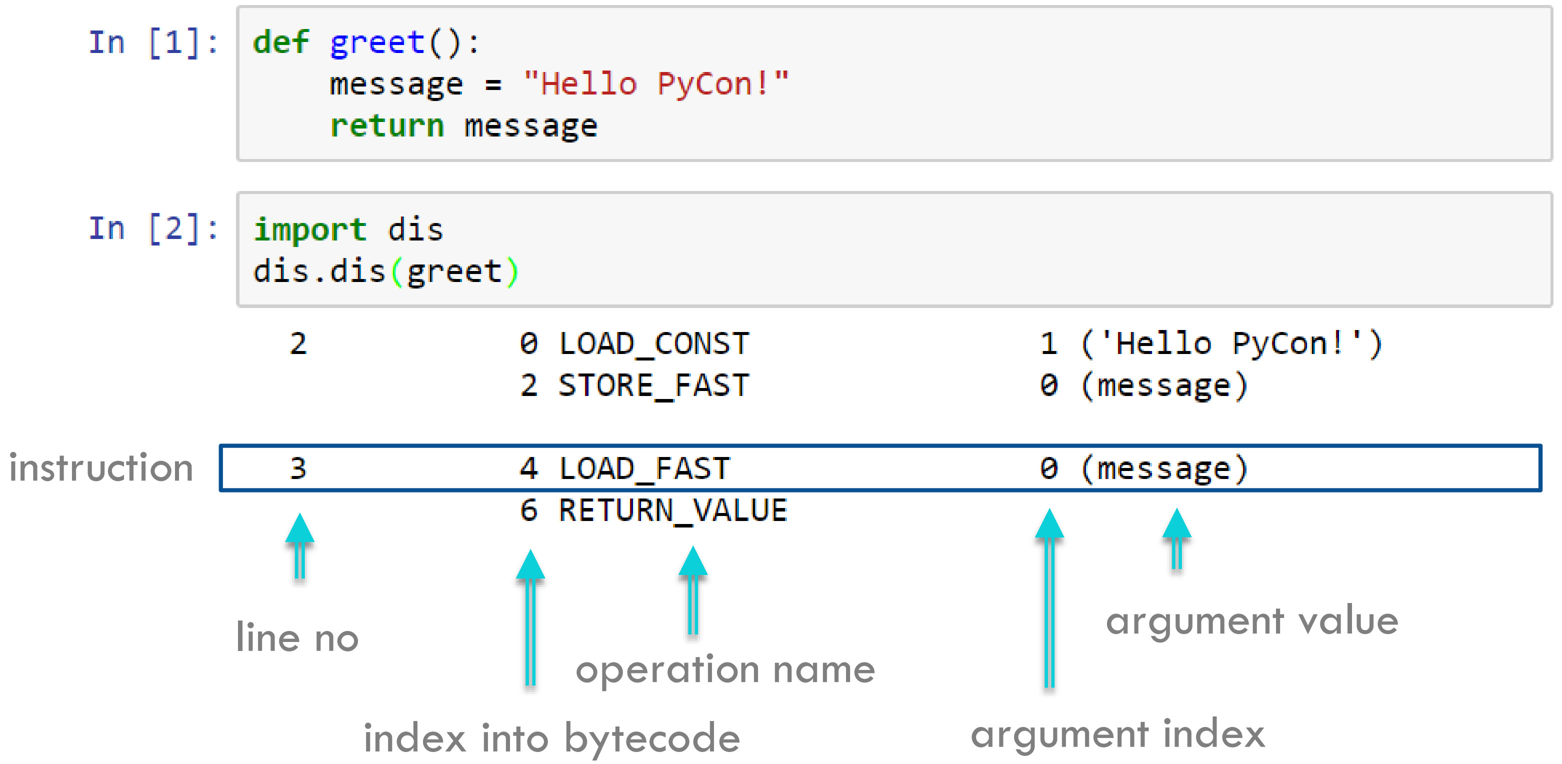
BUNCH OF **.PYC** FILES



Let’s switch to the real deal

Finally!

Dis module in action!



```
In [1]: def greet():
            message = "Hello PyCon!"
            return message
```

```
In [2]: import dis
        dis.dis(greet)
```

```
  2           0 LOAD_CONST               1 ('Hello PyCon!')
              2 STORE_FAST               0 (message)

  3           4 LOAD_FAST                0 (message)
              6 RETURN_VALUE
```

Thanks

Thank you for coming and contributing to the learning of others

Further Resources:

- A Python Interpreter written in Python:
  - http://www.aosabook.org/en/500L/a-python-interpreter-written-in-python.html
  - https://github.com/nedbat/byterun
- Hand crafted ByteCode:
  - http://multigrad.blogspot.co.uk/2014/06/fun-with-python-bytecode.html
- Anjana Vakil presentation:
  - https://speakerdeck.com/vakila/exploring-python-bytecode
- Docs for the dis module:
  - https://docs.python.org/3/library/dis.html
- Exploring ceval.c at the heart of interpreter:
  - https://tech.blog.aknin.name/category/my-projects/pythons-innards/