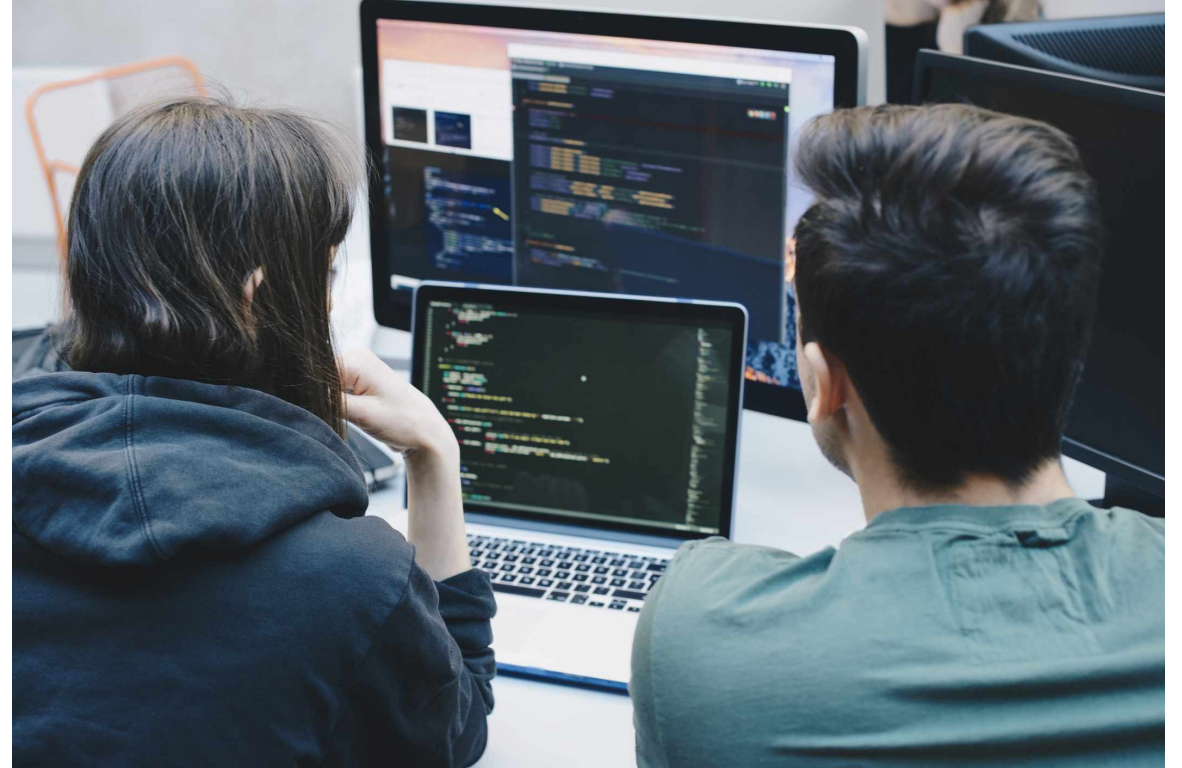


Intro to Programming I: Intro to Python

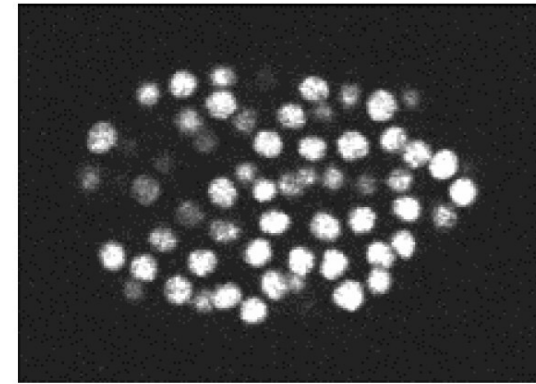
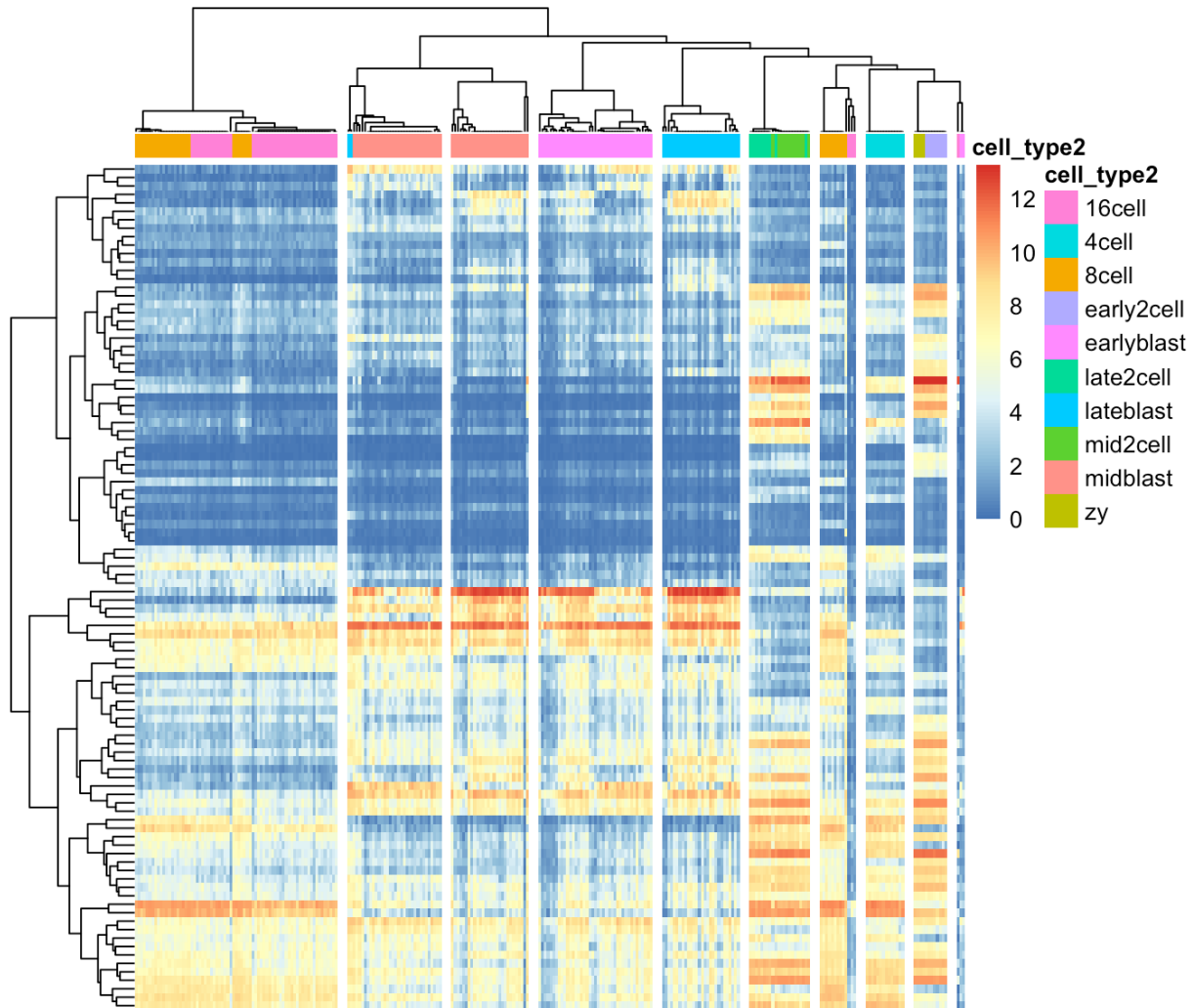
June 20th, 2024

Motivation

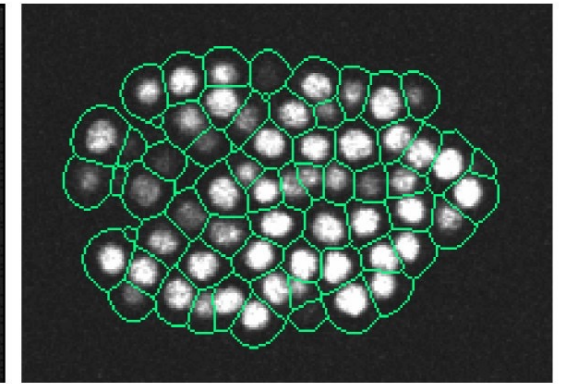
- **Computer Programming:**
Creating a schedule of events for a computer to follow to complete a task.
- Why learn programming?
 - To become a programmer
 - To think computationally
 - To apply computational tools to problems



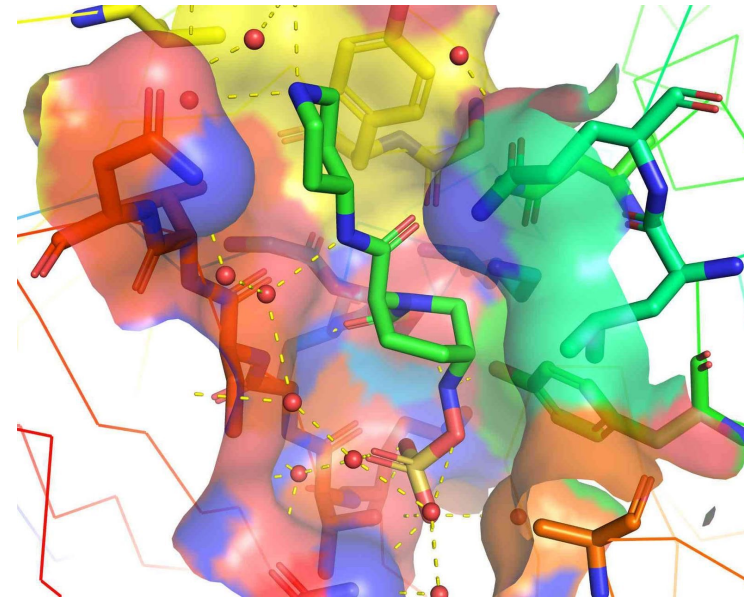
Motivation



(a)



(b)



What do **you** use programming for?



Python is a language

“Calculate the energy
of a protein in this
conformation.”



“01011110010101001010010010101
010101001110110010101111001010
101010100110010011011010010101
101010100101001010100101001001”

Python is a language

“Calculate the energy of a protein in this conformation.”



Human Language

>



(You are here)

Programmer

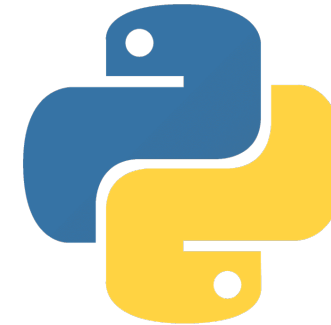
>

```
num1 = int(num1str)
num2 = int(num2str)

valid_operator = True
if operator == "+":
    result = num1 + num2
elif operator == "*":
    result = num1 * num2
elif operator == "-":
    result = num1 - num2
elif operator == "/":
    result = num1 / num2
else:
    valid_operator = False
```

Python

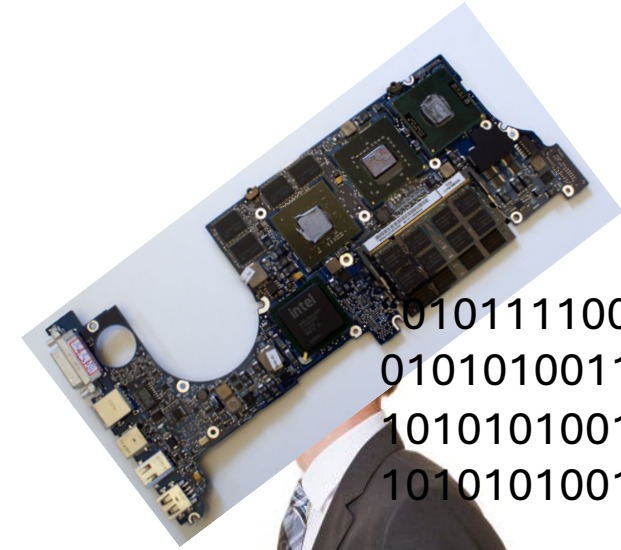
>



Python Interpreter

>

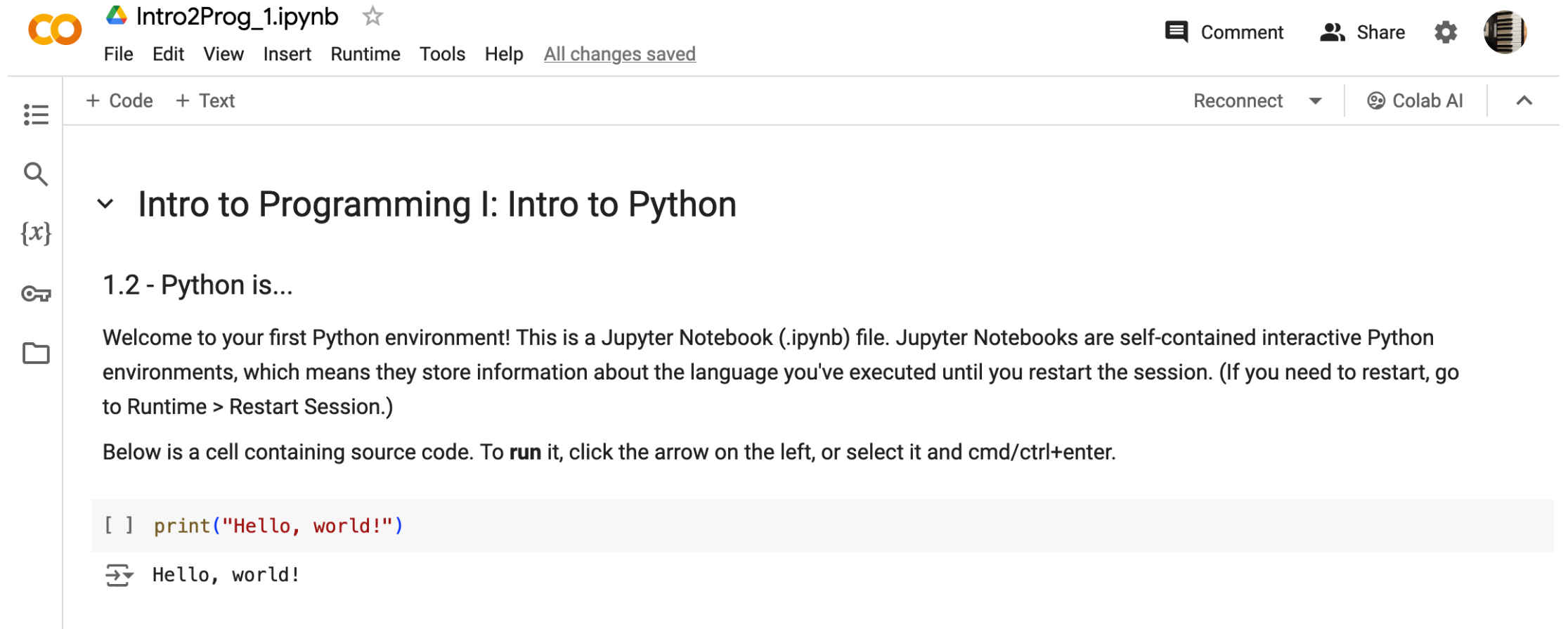
Machine Language



- + easy to pick up
- slower than other languages

Let's try Python!

- [Colab link](#)



The screenshot shows the Google Colab interface for a notebook named "Intro2Prog_1.ipynb". The top bar includes the Colab logo, the notebook name with a star icon, and a menu with options: File, Edit, View, Insert, Runtime, Tools, Help, and a status message "All changes saved". On the right side of the top bar are icons for Comment, Share, Settings, and a user profile. Below the top bar, there's a sidebar on the left with icons for a menu, search, variables, keys, and files. The main area of the notebook shows a section titled "Intro to Programming I: Intro to Python" with a sub-section "1.2 - Python is...". The text in this section reads: "Welcome to your first Python environment! This is a Jupyter Notebook (.ipynb) file. Jupyter Notebooks are self-contained interactive Python environments, which means they store information about the language you've executed until you restart the session. (If you need to restart, go to Runtime > Restart Session.) Below is a cell containing source code. To **run** it, click the arrow on the left, or select it and cmd/ctrl+enter." Below this text is a code cell with the following Python code:

```
[ ] print("Hello, world!")
```

 The output of the code is displayed below the cell:

```
➞ Hello, world!
```

Variables

>>> message = "What's up, Doc?"	<i>Strings</i>
>>> reply = 'Not much'	
>>> n = 17	<i>Integers</i>
>>> pi = 3.14159	<i>Floating point</i>
>>> decide = True	<i>Boolean</i>
>>> empty = None	<i>NoneType</i>

- Now we can fix our bug!

Operators

>>> print((2*6+3)/5-2)	<i>Arithmetic</i>
>>> print(2**2,2**3,2**4)	<i>Exponentiation</i>
>>> print(5 % 2)	<i>Modulus</i>
>>> print("hello" + "sir")	<i>String concatenation</i>
>>> print(not (3 >= 0))	
>>> print((3 > 0) and (3 < 0))	<i>Comparison and Logical</i>
>>> print((3 == 0) or (3 < 0))	

Functions

Function Definition

```
>>> def info(name, color="blue", age=34):  
...     print(name, color, age)
```

```
>>> def distance(x1, y1, x2, y2):
```

```
...     dx = x2 - x1
```

```
...     dy = y2 - y1
```

```
...     return (dx**2 + dy**2)**0.5
```

Whitespace is significant!

Function Invocation

```
>>> info("Spunky")
```

Can you predict what this line of code will do?

In-Built Functions

Built-in Functions			
A abs() aiter() all() any() anext() ascii()	E enumerate() eval() exec()	L len() list() locals()	R range() repr() reversed() round()
B bin() bool() breakpoint() bytearray() bytes()	F filter() float() format() frozenset()	M map() max() memoryview() min()	S set() setattr() slice() sorted() staticmethod() str() sum() super()
C callable() chr() classmethod() compile() complex()	G getattr() globals()	N next()	T tuple() type()
D delattr() dict() dir() divmod()	H hasattr() hash() help() hex()	O object() oct() open() ord()	V vars()
	I id() input() int() isinstance() issubclass() iter()	P pow() print() property()	Z zip() _import__()

Modules

```
import math

print (math.pi)
print (math.factorial(5))
print (math.cos(0))
print (math.log10(100))
```

*Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a **module**; definitions from a module can be imported into other modules or into the main module (the collection of variables that you have access to in a script executed at the top level and in calculator mode).*

Turtles!!!

- Back to colab!

```
import turtle
turtle.forward(100)
turtle.right(90)
turtle.backward(100)
```

```
from turtle import *
forward(100)
right(90)
backward(100)
```

```
help(turtle.forward)
```