Python Tutorial

EECS 498: Intro to Algorithmic Robotics

Tianyi Liu

09/10/2021

With content adapted from CS231n

Outline

- Python
- Numpy
- Matplotlib
- Best practices

Outline

- Python
- Numpy
- Matplotlib
- Best practices

What is Python

- High-level
- Dynamically typed
- Powerful and succinct

- We use Python 3 in this course

Hello World!

```python
# This is a comment
if True:
    print('Hello World!')
    print('Indentation matters!')
else:
    print('Nothing here')
```

Basic data types - Number

```
x = 3
print(type(x)) # Prints "<class 'int'>"
print(x)       # Prints "3"
print(x + 1)   # Addition; prints "4"
print(x * 2)   # Multiplication; prints "6"
print(x ** 2)  # Exponentiation; prints "9"
x += 1
print(x)  # Prints "4"
x *= 2
print(x)  # Prints "8"
y = 2.5
print(type(y)) # Prints "<class 'float'>"
print(y, y + 1, y * 2, y ** 2) # Prints "2.5 3.5 5.0 6.25"
```

Basic data types - Boolean

```
t = True
f = False
print(type(t)) # Prints "<class 'bool'>"
print(t and f) # Logical AND; prints "False"
print(t or f)  # Logical OR; prints "True"
print(not t)   # Logical NOT; prints "False"
print(t != f)  # Logical XOR; prints "True"
```

Basic data types - string

```
hello = 'hello'    # single or double quote
world = "world"    # doesn't matter
print(hello)       # Prints "hello"
print(len(hello))  # prints "5"
hw = hello + ' ' + world  # concatenation
print(hw)  # prints "hello world"
hw12 = '%s %s %d' % (hello, world, 12)
print(hw12)  # prints "hello world 12"
```

Containers - List

```
xs = [3, 1, 2]     # Create a list
print(xs[0])       # Access index 0
print(xs[-1])      # Access last idx
xs.append(4)       # Add an element to the end
print(xs)          # Prints "[3, 1, 2, 4]"
x = xs.pop()       # Remove the last element
print(x)           # Prints "4"
```

Iterating over lists

```python
animals = ['cat', 'dog', 'monkey']
for animal in animals:
    print(animal)
```

Functions

```
def hello(name, loud=False):
    if loud:
        print('HELLO, %s!' % name.upper())
    else:
        print('Hello, %s' % name)
hello('Bob') # Prints "Hello, Bob"
hello('Fred', loud=True)  # Prints "HELLO, FRED!"
```

File I/O

```
• f = open('sensorReading.txt', 'r') # Open a file as read
• list = []
• for line in f.readlines():
•     line = line[:-1] # get rid of newline character
•     list.append(float(line))
• f.close()
• print(list)
```

Outline

- Python
- Numpy
- Matplotlib
- Best practices

Arrays

```
import numpy as np
a = np.array([1, 2, 3])   # Create a rank 1 array
print(a.shape)            # Prints "(3,)"
print(a[1])               # Prints "2"
b = np.array([[1,2,3],
              [4,5,6]])   # Create a rank 2 array
print(b.shape)            # Prints "(2, 3)"
print(b[0, 1])            # Prints "2"
```

Array Slicing

```
import numpy as np
a = np.array([[1,2,3],
              [4,5,6],
              [7,8,9]])
# slice first 2 rows and col 1 & 2
b = a[:2, 1:3]
# A slice is a shallow copy
print(a[0, 1])   # Prints "2"
b[0, 0] = 77
print(a[0, 1])   # Prints "77"
```

Array Math

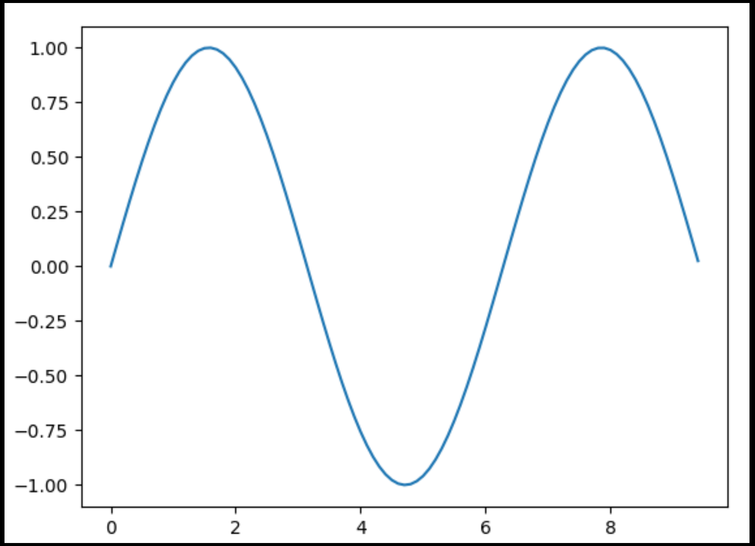
```
import numpy as np

x = np.array([[1,2],[3,4]], dtype=np.float64)
y = np.array([[5,6],[7,8]], dtype=np.float64)
# Elementwise
print(x + y)
print(x * y)
# Matrix multiplication
print(np.dot(x, y))
```

Outline

- Python
- Numpy
- Matplotlib
- Best practices

Plotting

```
import numpy as np
import matplotlib.pyplot as plt
# Sine data
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
# plot and show
plt.plot(x, y)
plt.show()
```



Outline

- Python
- Numpy
- Matplotlib
- Best practices

Best practices

- Pick your favorite text editor
  - Visual Studio Code
- Follow a style guide
  - https://google.github.io/styleguide/pyguide.html
- Use Google
- Start early on the homework