Python Day 1

Variable
&
Operation

Data
types

List
&
Numpy

Function

Variable

- Don’t really need to declare the type of a variable

```
temp_var = 10  # an int
temp_var = "Hello, World!"  # now become a str
```

Python Variable Name Rules:

- Must start with a letter or the underscore character
- Cannot start with a number
- Can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Case sensitive
- Cannot be any of the Python keywords

Operation

Arithmetic Operators

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Power
%	Remainder

Comparson Operators

Operator	Name
==	Equal
!=	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Data types

Category	Types
Text Type	`str`
Numeric Types	`int`, `float`, `complex`
Sequence Types	`list`, `tuple`, `range`
Mapping Type	`dict`
Set Types	`set`, `frozenset`
Boolean Type	`bool`
Binary Types	`bytes`, `bytearray`, `memoryview`

List

- Python List functions like a **dynamic array**, capable of **holding multiple types** of variables.
- Lists are created using square brackets or the **list()** constructor.

```
my_list = ["abc", 34, True, 40, "male"]
this_list = list((1, "one", 2, "two", 3, "three"))
```

- Lists are ***ordered***, ***changeable,*** and ***allow duplicate values***.
- List items are **indexed**, the first item has index 0.

List methods:

- len()
- index()
- copy()
- append()
- insert()
- extend()
- remove()
- del()
- pop()
- clear()
- sort()
- reverse()

Numpy

- NumPy is a Python **library** used for working with arrays.
- It has functions for working in domain of linear algebra, fourier transform, and matrices.
- Numpy array object is 50x **faster** than traditional Python list.

To install numpy, type this command in your terminal:

```
pip install numpy
```

- The array object in NumPy is called **ndarray**.
- We can create a NumPy ndarray object by using the **array()** function.

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])
```

Numpy array methods:

- ndim
- shape
- reshape()
- copy()
- concatenate()
- sort()

Function

- In Python function is defined using the **def** keyword.
- To call a function, use the function name followed by parentheses.

```
def my_function():
  print("Hello from a function")

my_function()
```

- Information can be passed into functions as arguments.
- To let a function return a value, use the **return** statement

```
def my_function(name1, name2):
  print("Hello " + name1 + " and " + name2)

my_function("Bob", "Fred")  # Hello Bob and Fred
my_function("Harry")  # This will cause an error
```

```
def my_function(x):
  return 5 * x

print(my_function(3))  # 15
print(my_function(5))  # 25
```

Function

Parameter vs Argument

- A **parameter** is the variable listed inside the parentheses in the function definition.
- An **argument** is the value that is sent to the function when it is called.

Scope

- **Local:** Accessible in a code block, function.
- **Global:** Accessible in the program, script, and module.

Docstring

- A special kind of string used to describe what a function, class, or module does.