

# **Week 1**

## **Python Basics: How to Translate Procedures into Codes**

**Applied Data Science**

**Columbia University - Columbia Engineering**

- ❖ **Week 1: Python Basics: How to Translate Procedures into Codes**
- ❖ Week 2: Intermediate Python — Data structures for Your Analysis
- ❖ Week 3: Relational Databases — Where Big Data is Typically Stored
- ❖ Week 4: SQL — Ubiquitous Database Format/Language
- ❖ Week 5: Statistical Distributions — The Shape of Data
- ❖ Week 6: Sampling — When You Can't or Won't Have ALL the Data
- ❖ Week 7: Hypothesis Testing — Answering Questions about Your Data
- ❖ Week 8: Data Analysis and Visualization — Using Python's NumPy for Analysis
- ❖ Week 9: Data analysis and visualization — Using Python's Pandas for Data Wrangling
- ❖ Week 10: Text Mining — Automatic Understanding of Text
- ❖ Week 11: Machine learning — Basic Regression and Classification
- ❖ Week 12: Machine learning — Decision Trees and Clustering

**Numbers** is one of the data types in Python. This data type is of two types: **integers** and **floating point**.

## Numbers: integers and floating point

### int

```
number_of_students = 47  
print(type(number_of_students))
```

47 is data of  
type int

The value  
associated with  
the variable  
number\_of\_students  
is of type int

### float

```
purchase_price = 93.74  
print(type(purchase_price))
```

93.74 is data of  
type float

Strings are an **ordered collection** of characters.

## string operations

```
x="Always take a banana to a party!"
```

We can extract substrings from a string

```
y=x[7:11] #The value of y is 'take' (locations 7, 8, 9, 10)
```

```
y=x[7:] #The value of y is 'take a banana to a party!'
```

```
y=x[0::2] #The value of y is 'Awy aeabnn oapry' (every 2nd character)
```

```
y=x[::-1] #The value of y is '???' (what does the negative sign mean?)
```

## Boolean Data Types

The boolean (bool) data type typically evaluates to either 'True' or 'False'. Note the casing of the values.

bool

Syntax note:  
uppercase T followed  
by lowercase rue -  
nothing else is True!  
(likewise for False)

bool

```
x=4  
y=2  
z=(x==y) #False  
z=(x==x) #True  
a=True  
b=False
```

*z takes the value False*

the value of z  
changes to True

Variables store values. Values do not change, but the value associated with a variable might change.

Declare variables before you use them. Use the syntax below to declare variables with values.

`<variable name> = <value>`

### Examples

```
price_now = float(input("What is the price now?"))  
pct_return = (price_now - initial_price)/initial_price * 100  
print("The return on the stock is: ", pct_return)
```



# The “If” Statements and Logical Expressions

The 'if' statement is used to control program flow and in Python, if something is true, then you do something, and if it's not true, you do something else

```
purchase_price = float(input("Enter the purchase price of the stock: "))
price_now = float(input("Enter the current price of the stock: "))
if price_now < purchase_price * 0.9:
    print("STOP LOSS: Sell the stock! ")
    print("You've lost", purchase_price - price_now, "Dollars per share")
elif price_now > purchase_price * 1.2:
    print("PROFIT TAKING: Sell the stock!")
    print("You've gained", price_now - purchase_price, "Dollars per share")
else:
    print("HOLD: Don't do anything!")
    print("Your unrealized profit is", price_now - purchase_price, "Dollars per share")
print("Hope you enjoyed this program!")
```

**This is a block. note the indenting!**

**A colon indicates that a block will follow**

**The end of indenting indicates that the block has ended**

A function is a black box and the way to call a function is to use the name of the function and then give it a list of arguments.





