Lecture 1 Examples

UCLA, Econ 10P: Introduction to Python for Economists
Dr. Randall R. Rojas

I. Data Types

1. Create 5 variables with names that are NOT allowed

```
# For example, the one below is allowed
x = 1
# but x@ = 1, is not

# Examples:
x#1 = 3
@3y = 4
z.4 = 10
a!! = 100
b_. = 2
```

2. Identify the `type` of each variable

```
t = 1 + 2j
u = letter
w = "True"
x = 3.14
y = 4
z = "Hello"
result1 = 10/3
result2 = 10.0//3
result3 = 5%2
```

In [1]:
```
t = 1 + 2j
print(type(t))

u = "letter"
print(type(u))

w = "True"
print(type(w))

x = 3.14
print(type(x))

y = 4
print(type(y))

z = "Hello"
print(type(z))

result1 = 10/3
print(type(result1))

result2 = 10.0//3
print(type(result2))

result3 = 5%2
print(type(result3))
```

```
<class 'complex'>
<class 'str'>
<class 'str'>
<class 'float'>
<class 'int'>
<class 'str'>
<class 'float'>
<class 'float'>
<class 'int'>
```

3. Identify the order of operations and rewrite eah expression using parenthesis to make them more legible.

```
# 1
5-3*2

# 2
20%5*3-7

#3
40/10*2+1

#4
25//4-21%3-4+5*2
```

In [4]:
```
# 1
5-(3*2)

# 2
#((20%5)*3)-7

#3
#((40/10)*2)+1

#4
(25//4) - (21%3) - 4 + (5*2)
```

Out[4]: 12

4. For the string, x ='Department of Economics', extract

> a. the first 5 charcaters.
> b. the last 5 characters.
> c. the word 'Economics'.

In [3]:
```
x = "Department of Economics"
```

In [4]:
```
# a
x[:5]
```

Out[4]: 'Depar'

In [5]:
```
# b
x[-5:]
```

Out[5]: 'omics'

In [6]:
```
# c
x[-9:]
```

Out[6]:
```
'Economics'
```

5. For the string below,

$x =$'Department of Economics'

> a. find its length.
> b. find the location of the letter 'E'.
> c. convert all letters upper case.
> d. convert all letters to lower case.

In [6]:
```
x = "Department of Economics"
```

In [7]:
```
# a
len(x)
```

Out[7]:
```
23
```

In [10]:
```
# b
x.find("t")
```

Out[10]:
```
5
```

In [8]:
```
# c
x.upper()
```

Out[8]:
```
'DEPARTMENT OF ECONOMICS'
```

In [11]:
```
# d
x.lower()
```

Out[11]:
```
'department of economics'
```

6. Compute the future value of a deposit. Prompt the user to enter the monthly interest rate (in decimal form), the initial deposit amount, and the duration of the deposit (in months). You can use the formula

$$FV = PV(1 + r)^n,$$

where PV = present value of the initial deposit, r = monthly rate, and n = number of months. After computing the future value, output it to the screen formatted with 2 decimal places.

In [5]:
```
rate = 0
deposit = 0
months = 0
FV = 0
print("Enter monthly interest rate in decimal form:")
rate=input()
print("Enter initial deposit amount:")
deposit=input()
print("Enter the number of months your initial deposit will be held:")
months=input()

FV = float(deposit)*(1+float(rate))**float(months)
print("The future value of your deposit is: {:.3f}".format(FV))
```

```
Enter monthly interest rate in decimal form:
.05
Enter initial deposit amount:
100
Enter the number of months your initial deposit will be held:
12
The future value of your deposit is: 179.586
```

7. Repeat Exercise 7 but instead compute the PV assuming you know the FV.

In this case, the formula to use is

$$PV = \frac{FV}{(1+r)^n}$$

.

In [10]:
```
rate = 0
FV = 0
months = 0
PV = 0
print("Enter monthly interest rate in decimal form:")
rate=input()
print("Enter the expected future value of the deposit amount:")
FV=input()
print("Enter the number of months of your your initial deposit will be held
months=input()

PV = float(FV)/((1+float(rate))**float(months))
print("The present value of your deposit is: {:.2f}".format(PV))
```

```
Enter monthly interest rate in decimal form:
.05
Enter the expected future value of the deposit amount:
179
Enter the number of months of your your initial deposit will be held:
12
The present value of your deposit is: 99.67
```

8. Repeat Exercise 6 but instead compute the number of months needed for an initial deposit (PV) to grow to a futuve value (FV). Note: You will need to include the command `import`

`math` and to compute a `log` value, use the command `math.log()`.

In [11]:
```
import math
rate = 0
FV = 0
months = 0
PV = 0
print("Enter monthly interest rate in decimal form:")
rate=input()
print("Enter initial deposit amount:")
PV=input()
print("Enter the expected future value of the deposit amount:")
FV=input()

months = (math.log(float(FV)/float(PV)))/(math.log(1+float(rate)))
print("The number of months needed is: {:.2f}".format(months))
```

```
Enter monthly interest rate in decimal form:
.05
Enter initial deposit amount:
100
Enter the expected future value of the deposit amount:
179
The number of months needed is: 11.93
```

II. Flow Control & Loops

1. Write a `control flow` script that (i) asks a user to input a number, and (ii) prints the sign of the number (e.g., positive or negative). If the number is zero, print "The number is zero".

In [14]:
```
number = input("Please enter a number: ")
if float(number) < 0:
    print("The number is negative")
elif float(number) > 0:
    print("The number is positive")
else:
    print("The number is zero")
```

```
Please enter a number: 6
The number is positive
```

2. What is the ouput of the code below for $X = 8, 9, 10$?

In [14]:
```
x = 9
if x > 9:
    x -= 2
elif x < 9:
    x += 2
else:
    x =x% 9
print(x)
```

```
0
```

3. What is the ouput of the code below?

```
x = 1; y = 2; z = 3
if z < x or y < x:
    print("True")
else:
    print("False")
```

In [17]:
```
x = 4; y = 5; z = 3
if z < x and y < x:
    print("True")
else:
    print("False")
```

```
False
```

4. Write a nested `for` loop (where i and j range from 1 to 10) that prints only the odd (i, j) pairs (i.e., both numbers are odd). Hint: use the `range()` function.

In [21]:
```
for i in range(11):
    for j in range(11):
        if i%2 == 0 or j%2 == 0:
            continue
        else:
            print(i,j)
```

```
1 1
1 3
1 5
1 7
1 9
3 1
3 3
3 5
3 7
3 9
5 1
5 3
5 5
5 7
5 9
7 1
7 3
7 5
7 7
7 9
9 1
9 3
9 5
9 7
9 9
```

5. Using a `while` control flow, write a script that adds the numbers from 1 to 100.

In [18]:
```
y = range(101)
i = 0
total = 0
while i  < 89:
    total = total + y[i]
    i += 1
print(total)
```

```
3916
```

6. What is the ouput of the code below? What is the last value of x?

```
    x = 100
```

In [19]:
```
x = 1000
tot = 0
while x > 0:
    if tot > 200:
        break
    tot += x
    x = x //2
    print(x)
```

```
500
```

In [ ]:
```
print(type(5//2))
```