Introduction to Python

Fondren Library
Research Data Services

ASSIGNING VARIABLES

In [28]:

```
a = 3
b = 4

c = a + b
d = a*b + c
e = a**b/c

print (c)
print (d)
print (e)

```

```
7
19
11.571428571428571
```

What is a Function?

A **function** in programming is a reusable block of code that performs a specific task. You can pass data to a function, and it can return data back. Functions help organize your code, make it more readable, and reduce repetition.

#Define a function called greet that takes a name as a parameter and prints a greeting.
def greet(name):
    print("Hello, " + name + "!")

#Call the function greet with "Alice" as the argument.

greet("Alice")
This function, when called, will output: `Hello, Alice!`

What is a Function?

A **function** in programming is a reusable block of code that performs a specific task. You can pass data to a function, and it can return data back. Functions help organize your code, make it more readable, and reduce repetition.

```
# Define a function called greet that takes a name as a parameter and prints a greeting.
def greet(name):
    print("Hello, " + name + "!")

# Prompt the user for their name
user_name = input("Enter your name: ")

# Call the function greet with the user-provided name as the argument.
greet(user_name)
```

Built-In Function: INPUT

```
How old are you?
```

```
How old are you? 6
```

```
How old are you?6
Your age is  6
```

```
In [20]:  1 Age = input("How old are you?")
          2 print ("Your age is ",Age)
```

Exercise 2:
Create a variable affiliation, prompt a question, "Are you a student or a staff member?"

print "You are a " + input

USER-DEFINED FUNCTION

In [26]:
```
def C_F(C):
    F = 1.8*C+32
    return F

temp = C_F(20)
print(temp)

```

```
68.0
```

Exercise 3: Create a BMI function and calculate BMI for person1 and person2. BMI = weight/height²

person1: height:1.65m, weight:60kg
person2: height:1.75m, weight:75kg

```
def BMI(H,W):
    bmi=

person1 =
person2 =
print(person1)
print(person2)
```

BASIC DATA STRUCTURES IN PYTHON

- Lists [1,2,3] ordered and changeable
- Tuples (1,2,3) ordered and unchangeable
- Dictionary {'a': 1, 'b':2, 'c':3} changeable, key-value pairs

LIST

Create a list:

```
1 mylist = ['apple', 'orange', 'banana']
2 print (mylist)
```

```
['apple', 'orange', 'banana']
```

Access item:

```
1 mylist = ['apple', 'orange', 'banana']
2 print (mylist[1])
```

```
orange
```

Change Item Value:

```
1 mylist = ['apple', 'orange', 'banana']
2 mylist[1] = 'cherry'
3 print (mylist)
```

```
['apple', 'cherry', 'banana']
```

Add Items:

```
1 mylist = ['apple', 'orange', 'banana']
2 mylist.append('pear')
3 print(mylist)
```

```
['apple', 'orange', 'banana', 'pear']
```

Remove Items:

```
1 mylist = ['apple', 'orange', 'banana']
2 mylist.remove('apple')
3 print(mylist)
```

```
['orange', 'banana']
```

Exercise 4:

1) Create a list of your favorite songs, print the list
2) Print the 3rd item in the list
3) Change the 3rd item into another song
4) Add one more song
5) Remove one song

CONTROL FLOW – IF/ELSE

In [31]:

```
1 GPA = 4.0
2 if GPA >3.8 and GPA <= 4.0:
3     print ("Welcome to Rice!")
4 elif GPA <= 3.8:
5     print ("Sorry")
6 else:
7     print ("Ooops, type a GPA in range")
```

```
Welcome to Rice!
```

Exercise 5: Create a variable called "behavior", assign a value "good" to it

if "good" print "candy"
elif "bad" print "no candy"
else print "ask your mom"

Control Flow - dictate how a program runs under different conditions or inputs

IF - Allows for us to execute on code, only if a set of conditional statements are true.

X is True IF

    6 = 6

    7 < 5

ELSE

T = “Apple”

Return T

ELIF, ELSE - Used in addition to IF statements, to execute code when a set of conditional statements aren’t true.

FOR and WHILE Loops - Repeat code based on a set of conditionals, iterating on a sequence of code, conditionals, or outputs

```
behavior = "good"
if behavior == "good":
    print("candy")
elif behavior == "bad":
    print("no candy")
else:
    print("ask your mom")
```

CONTROL FLOW – FOR LOOP

```
1 for x in range(1,6):
2     print (x)
```

```
1
2
3
4
5
```

Exercise 6:
Create a list called “animals" and put "cat","dog","pig"...in it
Use for loop to print each one out

```
5 animals=['cat','dog','pig']
6 for x in animals:
7     print (x)
```

```
cat
dog
pig
```

While Loop - Execute on a set of statements as long as a set of conditional statements are true.

```
i = 1

while i < 6:

    print(i)

    i += 1
```

objectives

- to introduce how to perform an analysis in python
- to understand the typical workflow of an analysis
- to understand some basic functions in python and associated libraries

packages (or libraries)

- review: what is a function?
- review: what is a program?
- packages expand our list of offerings of pre-constructed functions
- pre-constructed vs. customized functions
- packages are typically organized according to some specific task (data cleaning, data visualization, accessing a database, machine learning, etc.)
- whatever you're trying to do with data and whatever field/area/domain you're trying to do it in, chances are there is already an established library for that (and probably even more than one)

some important packages for data analytics

- pandas - adds spreadsheet functionality to python (dataframes)
- numpy - adds support for multi-dimensional arrays and matrices, and several relevant mathematical functions (linear algebra)
- matplotlib - adds data visualization functionality and several options for customization
- seaborn - adds a layer on top of matplotlib for additional out-of-the-box visualization options and several options for statistical visualizations

Next Class: Data Visualizations with Pandas and MatPlotLib Packages

Break Time

Office Hours - 8PM to 9PM