CS 152: Functions

CS 152: Python for STEM

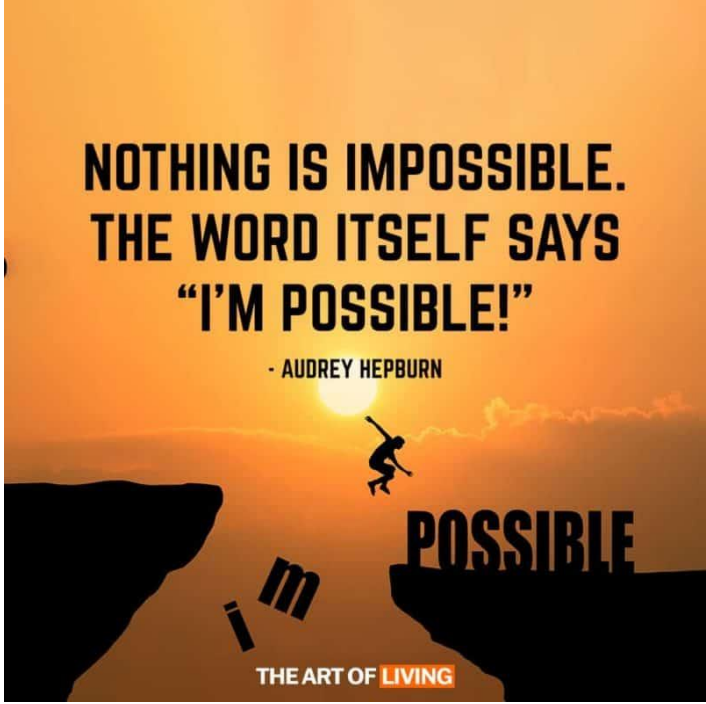
Colorado State University
Computer Science Department
Slides Originally Created by Albert Lionelle and Updated by Marcia Moraes

Colorado State University

Weekly Announcements!

TODO Reminders:

- Reading 3 (zyBooks) – you already should have done that for today's class ☺
- Lab 01 – Warm Up
- Reading 4 (zyBooks)
- Lab 02 - Application
- Reading 5 (zyBooks)



Remembering - Coding from Last Class

- Dr. Green is looking for a bank that will give the most return on her money over the next 5 years. She has P100,000.00 into a savings account. The standard equation to calculate principal plus interest at the end of a period is:
  - amount = P * (1 + I/M) ^ (N * M)
- Where:
  - P – principal (amount of money to invest)
  - I – interest (percentage rate the bank pays to the investor)
  - N – number of years (time for which the principal is invested)
  - M – compound interval (the number of times per year the interest is calculated and added to the principal)
- Think about what problem do you need to solve, how you are doing to solve it (write in English the steps to do that), write a Python code to solve that.

One Possible Solution

```
p = 10000000
n = 5
print("Enter the interest rate: ", end="")
i = float(input())
print("Enter the compound interval: ", end="")
m = float(input())
amount = p * (1 + i / m) ** (n * m)
print(f'Principal plus interest after {n:d} is {amount:.2f}')
```

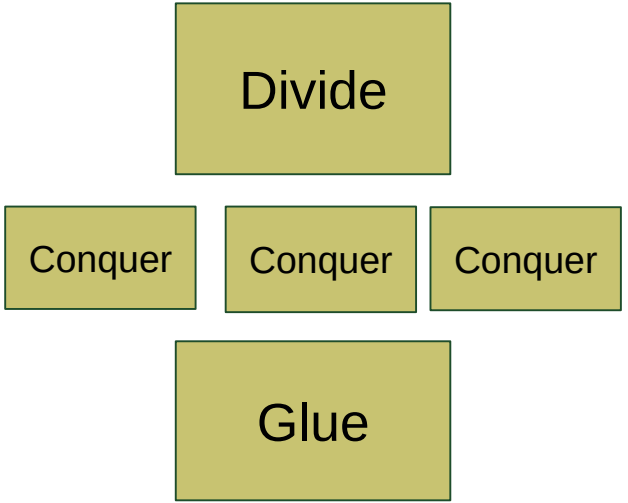
- What happen if we want to repeat this code to calculate the principal plus interest for another bank? Or for 10, 100 banks?

Code should be **Reusable**
Code should be **DRY**
- **D**on't **R**epeat **Y**ourself

Functions

Reusable Code

- Programming == Problem Solving
  - You look at the problem to solve
    - Clarify the problem and constraints
  - Break it up into *smaller* parts (Divide)
  - Outline the steps needed
    - Solve each step (Conquer)
  - Reassemble the pieces (Glue)
  - Completed program



Function – Part 1

```
p = 10000000
n = 5
print("Enter the interest rate: ", end="")
i = float(input())
print("Enter the compound interval: ", end="")
m = float(input())
amount = p * (1 + i / m) ** (n * m)
print(f'Principal plus interest after {n:d} is {amount:.2f}')
```

- def
  - defines the start of a function
  - indents keep the 'code' with the function
  - spacing matters!
- parameters
  - allows for variables to the functions
  - print(value)
    - function name print
    - value is a parameter!

What should be the parameters for our code? Why?

Which commands are part of the function? Why?

Function – Part 2

```
p = 10000000
n = 5
print("Enter the interest rate: ", end="")
i = float(input())
print("Enter the compound interval: ", end="")
m = float(input())
amount = p * (1 + i / m) ** (n * m)
print(f'Principal plus interest after {n:d} is {amount:.2f}')
```

```
def principal_plus_interest(i, m):
    amount = p * (1 + i / m) ** (n * m)
    print(f'Principal plus interest after {n:d} is {amount:.2f}')
```

```
p = 10000000
n = 5
print("Enter the interest rate: ", end="")
i = float(input())
print("Enter the compound interval: ", end="")
m = float(input())
principal_plus_interest(i, m)
```

How are we going to call the function we just created?

Could we improve this code a little bit more? How about p and n variables?

Function – Part 3

```
def principal_plus_interest(p, i, m, n):
    amount = p * (1 + i/m) ** (n * m)
    print(f'Principal plus interes after {n:d} is {amount:.2f}')

p = float(input("Enter the principal amount: "))
n = int(input("Enter the number of years to invest: "))
print("Enter the interest rate: ", end="")
i = float(input())
print("Enter the compound interval: ", end="")
m = float(input())
principal_plus_interest(p,i,m,n)
```

Return Values

- Functions do some work
  - and then return the answer of that work
- Other programs can then use those answers
- Always the best paradigm to follow

```
def principal_plus_interest(p, i, m, n):
    amount = p * (1 + i/m) ** (n * m)
    return amount

p = float(input("Enter the principal amount: "))
n = int(input("Enter the number of years to invest: "))
print("Enter the interest rate: ", end="")
i = float(input())
print("Enter the compound interval: ", end="")
m = float(input())
amount = principal_plus_interest(p,i,m,n)
print(f'Principal plus interes after {n:d} is {amount:.2f}')
```

Function principal_plus_interest is returning the value that was calculated

We need to store the variable that is returned when we call the function, in order to be able to use it

Return Values – More Examples

- Lets analyze the program below:

```
def arithmeticExpression(val1, val2, val3):
    return (val1+val2+val3)/3;

def powerOfTwo(val):
    return val ** 2;

def powerOfSomething(val, something):
    return val ** something;

def main():
    val1 = float(input("Enter a first number: "))
    val2 = float(input("Enter a second number: "))
    val3 = float(input("Enter a third number: "))
    average = arithmeticExpression(val1, val2, val3)
    print(f"Average of {val1:.1f}, {val2:.1f} and {val3:.1f} is {average:.1f}")
    print(powerOfTwo(val1));
    print(powerOfSomething(powerOfTwo(2), val1))

main()
```

Incremental Development and Function Stubs

- What is incremental development?
- What is function stubs?
- What does the pass keyword mean? When is it used?

```
def steps_to_feet(num_steps):
    feet_per_step = 3
    feet = num_steps * feet_per_step
    return feet

def steps_to_calories(num_steps):
    pass

steps = int(input('Enter number of steps walked: '))

feet = steps_to_feet(steps)
print('Feet:', feet)

calories = steps_to_calories(steps)
print('Calories:', calories)
```

Docstring and help function

- What is a docstring?
- What is a help function? How would you use it in the example below?

```
def num_seats(airliner_type):
    """Determines number of seats on a plane"""
    #Function body statements ...

def ticket_price(origin, destination, coach=True, first_class=False):
    """Calculates the price of a ticket between two airports.
    Only one of coach or first_class must be True.

    Arguments:
    origin -- string representing code of origin airport
    destination -- string representing code of destination airport

    Optional keyword arguments:
    coach -- Boolean. True if ticket cost priced for a coach class ticket (default True)
    first_class -- Boolean. True if ticket cost priced for a first class ticket (default False)

    """
    #Function body statements ...
```

Student Challenge

- Write two functions
  - The first function takes in two parameters – first, last
    - It prints the ”welcome to the class (last), (first)”
  - The second function
    - Calls input to ask the client their first name
    - It calls input a second time to ask them their last name
    - it calls your first function to print out the result

- Use the Python Tutor editor (https://pythontutor.com/visualize.html#mode=edit) to see a step-by-step execution of your program.