CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

Outline Sept 24:

- Functions
  - add_mult.py
  - class_functions.py
  - factorial.py
  - lettercount.py

Notes

- **Lab 3** due **Saturday** night

- **Quiz 1** returned on **Wednesday**

- When we come back together as a class, please end side conversations – distracting for others around you

Recap if/elif/else, and/or

movie.py

```
Assume that movie tickets have the following prices based on age:
 0 - 12 years: $8
13 - 64 years: $12
65 and up: $8
```

```python
# 1) if/elif/else-----------------------------------------------
if age <= 12:
    print("Your ticket price is", 8)
elif age < 64:
    print("Your ticket price is", 12)
else:
    print("Your ticket price is", 8)
```

movie.py

```
Assume that movie tickets have the following prices based on age:
 0 - 12 years: $8
13 - 64 years: $12
65 and up: $8
```

```python
# 2) if/else + or-----------------------------------------------
if age <= 12 or age >= 65:
    print("Your ticket price is", 8)
else:
    print("Your ticket price is", 12)
```

movie.py

```
Assume that movie tickets have the following prices based on age:
 0 - 12 years: $8
13 - 64 years: $12
65 and up: $8
```

```python
# 3) if/else + and-----------------------------------------
if age > 12 and age < 65:
    print("Your ticket price is", 12)
else:
    print("Your ticket price is", 8)
```

movie.py

```
Assume that movie tickets have the following prices based on age:
 0 - 12 years: $8
13 - 64 years: $12
65 and up: $8
```

```python
# 4) if/else-----------------------------------------------------
if 12 < age < 65: # implicit "and"
    print("Your ticket price is", 12)
else:
    print("Your ticket price is", 8)
```

Functions

```
def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Calling a function, just like input(..), print(..), int(..)

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s
```

```python
def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Calling a function, just like input(..), print(..), int(..)

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Keyword “def” defines a function

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Arguments or *parameters* (input)

Keyword “def” defines a function

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Description of function in triple quotes

Arguments or *parameters* (input)

Keyword “def” defines a function

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Description of function in triple quotes

Arguments or *parameters* (input)

Indented: *body* of function

Keyword “def” defines a function

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Description of function in triple quotes

Arguments or *parameters* (input)

Indented: *body* of function

Keyword “def” defines a function

main() is also a function (no arguments!)

```
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Description of function in triple quotes

Arguments or *parameters* (input)

Indented: *body* of function

Keyword “def” defines a function

main() is also a function (no arguments!)

Calling or *invoking* a function

```python
def add(x, y):
    """
    This function takes two numerical arguments and returns their sum.
    """
    s = x + y
    return s

def multiply(x, y):
    """
    This function takes two numerical arguments (x,y) and returns x*y
    """
    m = x * y
    return m

def main():
    """
    In the main function, test our functions
    """

    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)

main()
```

Description of function in triple quotes

Arguments or *parameters* (input)

Indented: *body* of function

Keyword “def” defines a function

main() is also a function (no arguments!)

Calling or *invoking* a function

Passing in arguments

class_functions.py

Section 2

```
Yasmin
Eva
Abby
Angelina
Chris
Josh
Ricky
Chrisbet
Colin
Gwen
Julian
Luke
Dean
Rachel
Kat
Matt
Eudy
Juan
Karin
Juliette
Karinna
Ghazi
Jordan
Sam
Ella
Mirabai
Bilal
Christina
Iris
Tiffany
Tom
Owen
Larkin
Robert
```

Section 3

```
Faith
Santi
Sarah
Egor
Elizabeth
Carina
Jade
Ilana
Charles
Ollie
Maggie
Kyle
Maeve
Jason
Aron
Ellie
Charlie
Lauren
Bess
Francesco
Beluchi
Otis
Fefa
Grace
Camryn
Greta
Vivian
Helen
Avery
Theo
Tyler
Gene
Dominic
Jefferson
```

Problems to try now with a partner

- cs21/inclass/w04/factorial.py
- cs21/inclass/w04/lettercount.py

If complete: return to class_functions.py

- in_class(name, section_lst)
- random_student(section_lst)

cs21/inclass/w04/factorial.py

$n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1$

$7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$

i	n! accumulate	multiply by
0	1	1
1	2	2
2	6	3
3	24	4
4	120	5
5	720	6
⋮	5040	7

i+1