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

# 2) if/else + or

```python
if age <= 12 or age >= 65:
    print("Your ticket price is", 8)
else:
    print("Your ticket price is", 12)
```
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)
```
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
    
    Calling a function, just like input(..), print(..), int(..)

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

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

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()
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()
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()
def add(x, y):
    s = x + y
    return s

def multiply(x, y):
    m = x * y
    return m

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

main()
def add(x, y):
    
    # Description of function in triple quotes
    
    """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()
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()
def add(x, y):
    
    s = x + y
    return s

def multiply(x, y):
    
    m = x * y
    return m

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

main()
Arguments or parameters (input)

```
def add(x, y):
    s = x + y
    return s
```

Keyword “def” defines a function

```
def multiply(x, y):
    m = x * y
    return m
```

Indented: body of function

```
def main():
    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(sum_result, mult_result)
```

Description of function in triple quotes

```````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````
def add(x, y):
    result = x + y
    return result

def multiply(x, y):
    result = x * y
    return result

def main():
    num1 = 3
    num2 = 5
    sum_result = add(num1, num2)
    mult_result = multiply(num1, num2)
    print(multiply(sum_result, mult_result)
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)`
$n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1$

<table>
<thead>
<tr>
<th>i</th>
<th>accumulate</th>
<th>multiply by</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>720</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>5040</td>
<td>7</td>
</tr>
</tbody>
</table>

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