

CSIT110 / CSIT810

Python

Lecture 7

Dr. Joseph Tonien

School of Computing and Information Technology
University of Wollongong

Objectives

Understanding of:

- Rounding off function
- Max and min functions
- Random function
- Write your own functions

Function

The `print` function

```
print("Hello World!")
```

```
a = 10
```

```
print(a)
```

print with a new line



```
print("one", end=", ")
```

```
print("two", end=", ")
```

```
print("three", end=", ")
```

```
print("four", end=", ")
```

```
print("five", end=".")
```

```
print()
```

print with no line break




The `input` function

```
user_input = input("Enter something: ")  
print("You have entered: " + user_input)
```

Note that `input` is a reserved keyword so we can't use it to name a variable

error



```
input = input("Enter something: ")  
print("You have entered: " + input)
```

The round function

```
a = 28.30188679245283
```

```
b = round(a)
```

28

```
b = round(a, 1)
```

28.3

```
b = round(a, 2)
```

28.30

```
b = round(a, 3)
```

28.302

```
b = round(a, 4)
```

28.3019

```
b = round(a, 5)
```

28.30189

```
b = round(a, 6)
```

28.301887

The `min` and `max` functions

```
a = 1.5
```

```
b = 5
```

```
c = 3
```

```
d = min(a, b, c) → 1.5
```

```
e = max(a, b, c) → 5
```

The `random.randint` function

A dice rolling program

```
import random

for i in range(0, 10):
    r = random.randint(1, 6)
    print("Dice result: {0}".format(r))
```

```
Dice result: 3
Dice result: 2
Dice result: 4
Dice result: 1
Dice result: 3
Dice result: 1
Dice result: 3
Dice result: 1
Dice result: 6
Dice result: 5
```


The `random.randint` function

import a python module called `random`

```
import random
```

```
for i in range(0, 10):  
    r = random.randint(1, 6)  
    print("Dice result: {0}".format(r))
```

generate a
random integer
between 1 and 6

```
Dice result: 3  
Dice result: 2  
Dice result: 4  
Dice result: 1  
Dice result: 3  
Dice result: 1  
Dice result: 3  
Dice result: 1  
Dice result: 6  
Dice result: 5
```

The `random.randint` function

import a python module called `random`

```
import random
```

```
r = random.randint(a, b)
```

generate a
random integer
between **a** and **b**

Function

```
r = random.randint(1, 6)
```



these are function's **arguments**

a function may have 0, one, two, or more arguments

Function

```
r = random.randint(1, 6)
```

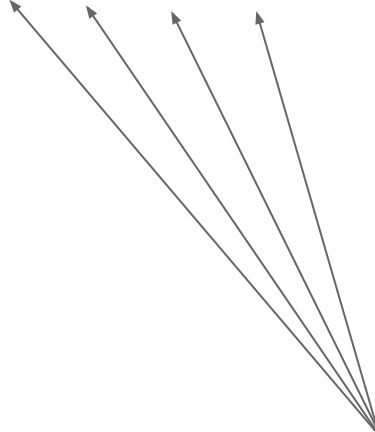
a function can **return value**



a function may return 0, one, two, or more values

Function

```
m = min(4, 6, 3, 1)
```



in this example, the `min` function takes 4 **arguments**

Function

```
m = min(4, 6, 3, 1)
```

and returns one value



Function

```
m = max(-1, 7)
```



in this example, the `max` function takes 2 **arguments**

Function

`m = max(-1, 7)`

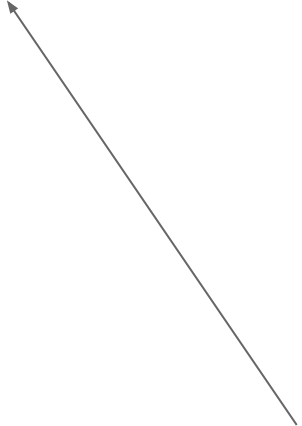
and returns one value



Function

```
user_input = input("Enter something: ")
```

in this example, the `input` function takes 1 **argument**



Function

```
user_input = input("Enter something: ")
```

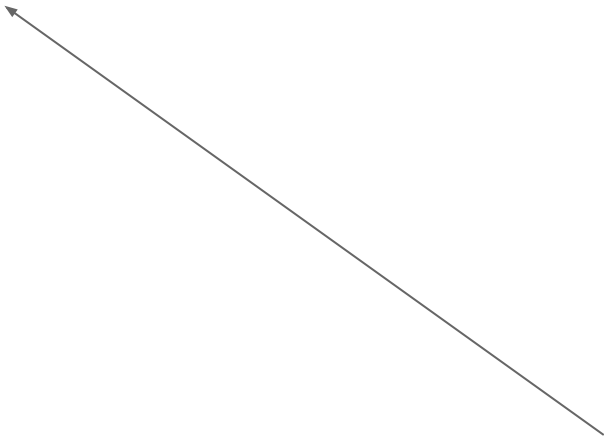
and returns one value



Function

```
print("Hello World!")
```

in this example, the `print` function takes 1 **argument**



Function

```
print("Hello World!")
```

and returns zero value



writing your own functions

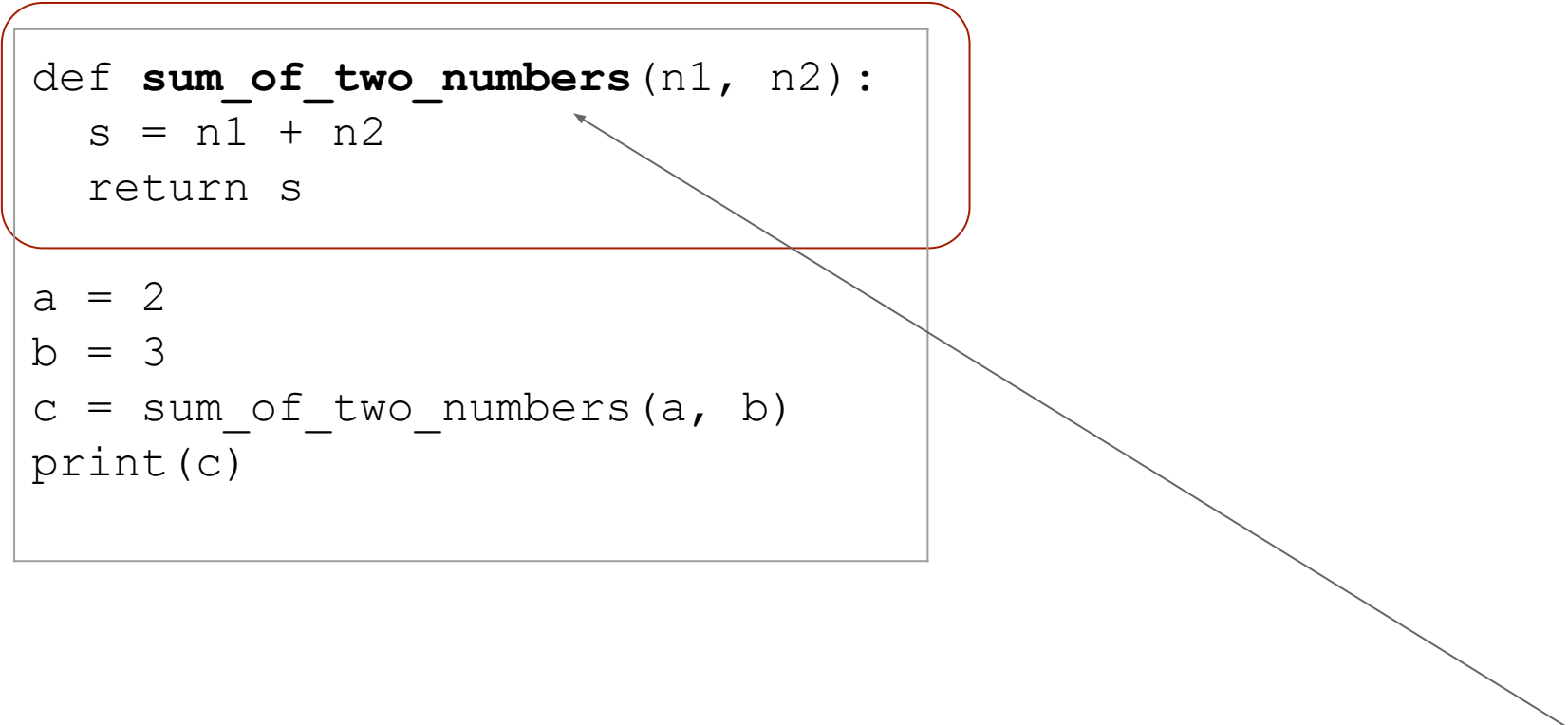
Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s  
  
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```

What do you think this program will do?

Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

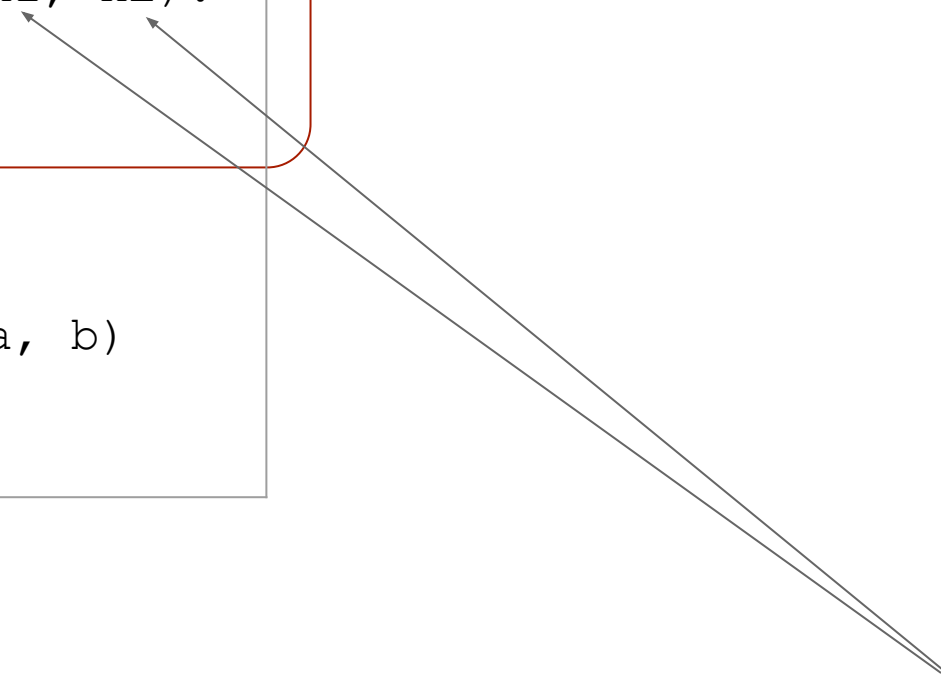


```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```

we are writing a new function
called **sum_of_two_numbers**

Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

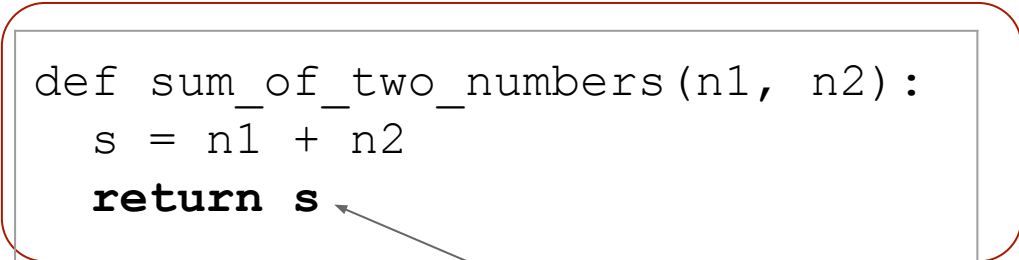
A red rounded rectangle highlights the function definition. Two arrows originate from the text 'this new function has 2 arguments' at the bottom right. One arrow points to the parameter 'n1' and the other points to the parameter 'n2' in the function signature.

```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```

this new function has 2 **arguments**

Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```



```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```

and it returns 1 value



Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```



we have a new function
and we can use it!!!

Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

n1

2

n2

3

s

5

Sum of two numbers

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)  
print(c)
```

```
def sum_of_two_numbers(n1, n2):  
    s = n1 + n2  
    return s
```

c

5

n1

2

n2

3

s

5

writing your own functions

- functions that do not return value**

Say hi

```
def say_hi(name):  
    print("Hi {0}!".format(name))  
  
j = "John"  
say_hi(j)  
  
say_hi("Bob")  
say_hi("Alicia")
```

What do you think this program will do?

Say hi

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

```
j = "John"  
say_hi(j)
```

```
say_hi("Bob")  
say_hi("Alicia")
```

we are writing a new function
called **say_hi**

Say hi

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

```
j = "John"  
say_hi(j)
```

```
say_hi("Bob")  
say_hi("Alicia")
```

this new function has 1 **argument**

Say hi

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

```
j = "John"  
say_hi(j)  
  
say_hi("Bob")  
say_hi("Alicia")
```



this function does NOT return values

Say hi

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

```
j = "John"
```

```
say_hi(j)
```

```
say_hi("Bob")
```

```
say_hi("Alicia")
```

we have a new function
and we can use it!!!

Say hi

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

```
j = "John"  
say_hi(j)
```

```
say_hi("Bob")  
say_hi("Alicia")
```

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

name

John

```
def say_hi(name):  
    print("Hi {0}!".format(name))
```

name

Alicia

Function's return

when function does NOT return values
we just call it

```
say_hi("Bob")  
say_hi("Alicia")
```

```
print("Hello world")  
print("Python")
```

when function returns a value
we can **assign** that
returned value to a **variable**

```
a = 2  
b = 3  
c = sum_of_two_numbers(a, b)
```

```
a = 2  
b = 3  
c = max(a, b)
```

```
number_input = input("Enter an integer: ")  
n = int(number_input)
```

**a function can return
more than one value**

Ask user details

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

```
Enter your first name: John  
Enter your last name: Smith  
User 1: John Smith  
Enter your first name: Bob  
Enter your last name: Lee  
User 2: Bob Lee
```

Ask user details

```
def ask_info() :  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

we are writing a new function
called **ask_info**

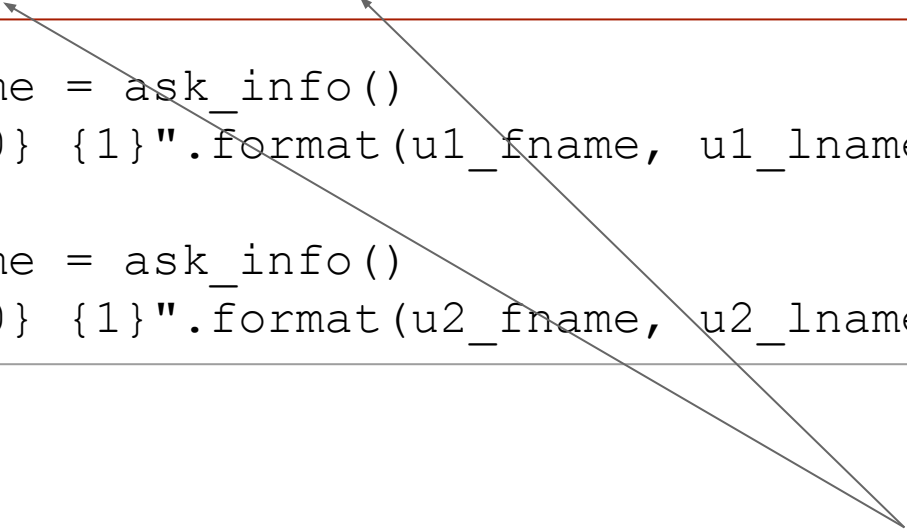
Ask user details

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

this new function has 0 arguments

Ask user details

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

A diagram consisting of two thin grey arrows. One arrow originates from the `first_name` variable in the `return` statement of the `ask_info` function and points to the `u1_fname` variable in the assignment `u1_fname, u1_lname = ask_info()`. The second arrow originates from the `last_name` variable in the `return` statement and points to the `u1_lname` variable in the same assignment. A similar pair of arrows connects the `ask_info` function call to the `u2_fname` and `u2_lname` variables in the second assignment.

this function returns 2 values

Ask user details

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

Since this function returns 2 values,
we can assign the 2 returned values to **2 variables**

Ask user details

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name
```

first_name

John

last_name

Smith

Ask user details

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name  
  
u1_fname, u1_lname = ask_info()  
print("User 1: {0} {1}".format(u1_fname, u1_lname))  
  
u2_fname, u2_lname = ask_info()  
print("User 2: {0} {1}".format(u2_fname, u2_lname))
```

```
def ask_info():  
    first_name = input("Enter your first name: ")  
    last_name = input("Enter your last name: ")  
    return first_name, last_name
```

first_name

Bob

last_name

Lee