Week 4 Tutorial

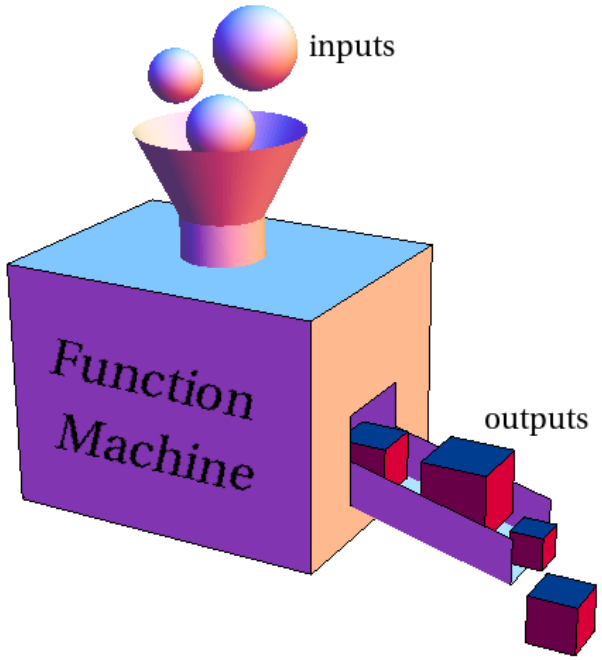
COMP10001 – Foundations of Computing

Semester 2, 2025

Clement Chau

- Introduction to Functions
- More on Strings
- Methods, Tuples, Sequences

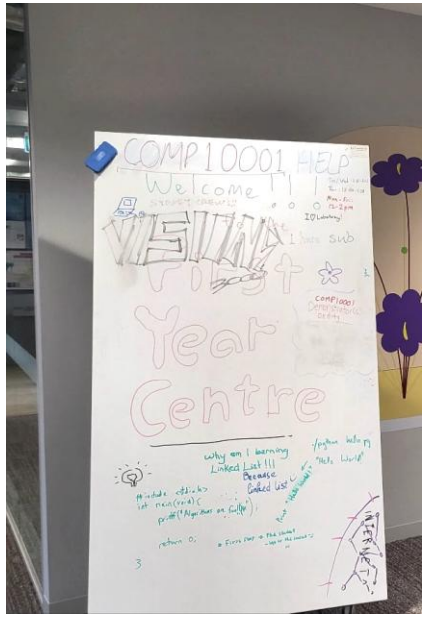
Revision: Functions!

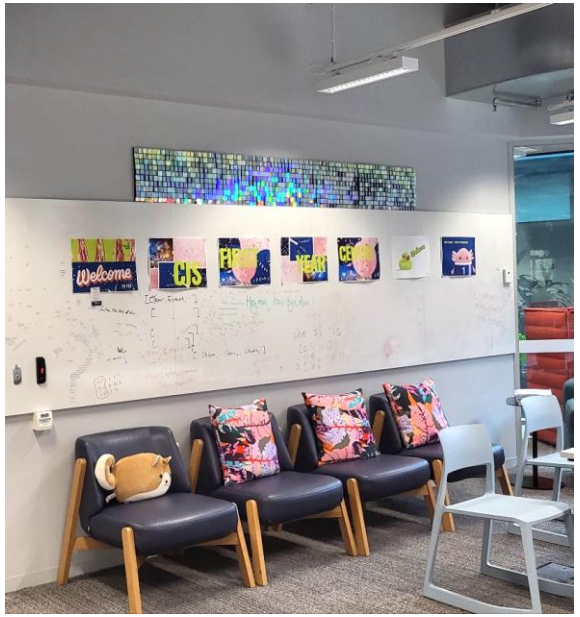



```
sem1-2025 > week-4 > functions.py > ...
1   # This is an example of a function in Python
2   def add(a, b):
3       return a + b
4
5   result = add(1, 2)
6   print(result) # result = 3
```

How do I Get Help?

- *During the **workshop** hours*
- *For **subject content** enquiries, please use **Ed Discussion** (Please check previous threads)*
- *For email enquiries (e.g. **special consideration**), please use comp10001-semester2@unimelb.edu.au*
- *CIS **First Year Centre** (FYC): Level 3 @ Melbourne Connect*
  - ***Mon-Fri 12pm-2pm***





TutSheet Week 4 – Question 1

1. What is shown on screen after you execute the program below? What is the value of `res`, `res_p`, `res_r`, and `res_pr`?

```
def ave(a, b):
    result = (a + b) / 2
    return ?
def ave_p(a, b):
    result = (a + b) / 2
    print("p", result)

def ave_r(a, b):
    result = (a + b) / 2
    return result

def ave_pr(a, b):
    result = (a + b) / 2
    print("pr", result)
    return result

res = ave(1, 2)
res_p = ave_p(1, 2)
res_r = ave_r(1, 2)
res_pr = ave_pr(1, 2)
```

	Shown on screen	Value of variables
res		None
res_p	p 1.5	None
res_r		1.5
res_pr	pr 1.5	1.5

return:

*how a function stops execution and **"gives back a value"**, so that the value can be, for example, assigned to a variable.*

print:

show the value on your screen**, but does **NOT** "gives back a value" as return does. As such, the value it "give back" is **None

Ed lessons > Worksheet 5 > Defining functions > Viewing Output [Link]

TutSheet Week 4 – Question 2

2. What's wrong with this code? How can you fix it?

```
def calc(n1, n2):
    answer = n1 + (n1 * n2)
    print(answer)
    return answer

num = int(input("Enter the second number: "))
result = calc(2, num)
print("The result is:", result)
```

"The result is: None"

*This function **prints the answer** to the calculation it's performed, **rather than returning it**.*

This means that the value of `result` will be `None` and the last line will not work as intended.

TutSheet Week 4 – Question 3

3. Evaluate the following method calls given the assignment `s = "Computing is FUN!"` Think about the input and output of each method. You're not expected to know all methods for all types: if you haven't seen some of these before, your best guess based on the name will probably be right!

(a) `s.isupper()` *False*

(b) `s.upper()` *"COMPUTING IS FUN!"*

(c) `s.endswith("FUN!")` *True*

(d) `s.count('i')` *2*

(e) `s.strip('!')` *"Computing is FUN"*

(f) `s.replace('i', '!')` *"Comput!ng !s FUN!"*

(g) `s.split()` *["Computing", "is", "FUN!"]*

(h) `s.isdigit()` *False*

https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str
https://www.geeksforgeeks.org/python-string-strip/

TutSheet Week 4 – Question 4

4. Evaluate the following given the assignment `lst = [2, ("green", "eggs", "ham"), False]`. Assume the list is reset after each part.

(a) `lst[2]` ***False***

(b) `lst[1][-2]` ***"eggs"***

(c) `lst[1][-2][:3]` ***"egg"***

(d) `lst.append(5); print(lst)`

[2,("green", "eggs", "ham"), False, 5]

(e) `lst.pop(2); print(lst)`

False, [2,("green", "eggs", "ham")]

(f) `lst.reverse(); print(lst)`

[False,("green", "eggs", "ham"), 2]

list	[2,	("green","eggs","ham"),			False]
index	0	1			2
	-3	-2			-1
		0	1	2	
		-3	-2	-1	

.pop()

- *remove an element from a list at a specified index*
- *If no index : remove the last element from list*

https://www.geeksforgeeks.org/python-list-pop-method/

TutSheet Week 4 – Past Exam Q1

1. What does the following code produce as output?

(a) `22 % 4 * 4 + 1.5 - 22 // 3`

2 *7*

*2 * 4 + 1.5 – 7 = 2.5*

(b) `"hello"[:3] + "p"`

"help"

(c) `(1,2) + (3) + (4,5)`

TypeError

(3) *: Not tuple, Integer*
because a single value inside parentheses without a trailing comma is interpreted as a regular integer

(d) `tuple('abc') + (4,)`

('a','b','c',4)

https://docs.python.org/3/tutorial/datastructures.html#tuples-and-sequences

TutSheet Week 4 – Past Exam Q2 (a)

2. One liners!

(a) Suppose that `str1` and `str2` are two strings, and that `k` is a positive integer. Give a single Python assignment statement that assigns `True` to `match` if `str1` and `str2` have the same first `k` characters, and assigns False to `match` if not.

```
str1[:k] == str2[:k]

match = str1[:k] == str2[:k]
```

```
k = 2
str1 = "apple"
str2 = "ale"

match = str1[:k] == str2[:k]
print(match)
```

```
False
```

TutSheet Week 4 – Past Exam Q2 (b), (c)

(b) Suppose that `lst` is a non-empty list of numbers. Give a single Python assignment statement that assigns the difference between the largest and smallest numbers in `lst` to the variable `diff`.

```
diff = max(lst) - min(lst)
```

(c) Suppose that `text` is a Python string. Give a single Python assignment statement that assigns the number of words in text to `wrds`, where a "word" is any non-blank sequence of characters. (Hint: A method covered in previous exercises may be useful).

```
wrds = len(text.split())
```

```
text = "Computing is FUN!"

wrds = len(text.split())

print(text.split())
print(wrds)
```

```
['Computing', 'is', 'FUN!']
3
```

Independent Work

- **Do worksheets** 6, 7, 8 on Ed (due **next Monday at 6pm**)
  - Remember that **Ed worksheets contributes to 10% of your total score!**
- **Raise your hand** if you have any questions!

Scan here for annotated slides