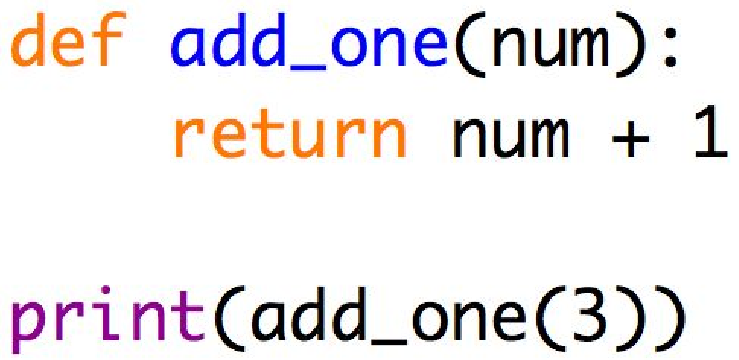
#CodeYork

Session 4

Recap on Functions

- Functions are defined using python’s “def” keyword
- Functions may or may not return values
- Functions may call themselves (recursion)



```python
def add_one(num):
    return num + 1

print(add_one(3))
```

Functions Calling Functions

- Functions can call other functions as well as themselves if needed

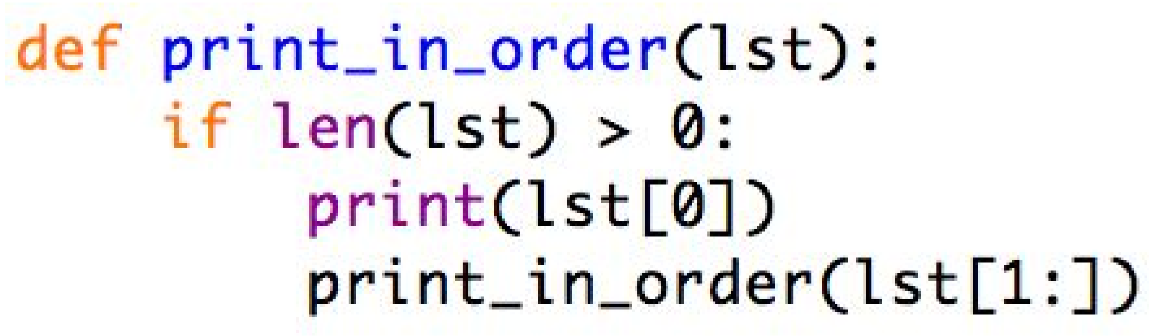
```
def add_one(num):
    return num + 1

def add_two(num):
    return add_one(add_one(num))


print(add_two(3))
```

Printing in Functions

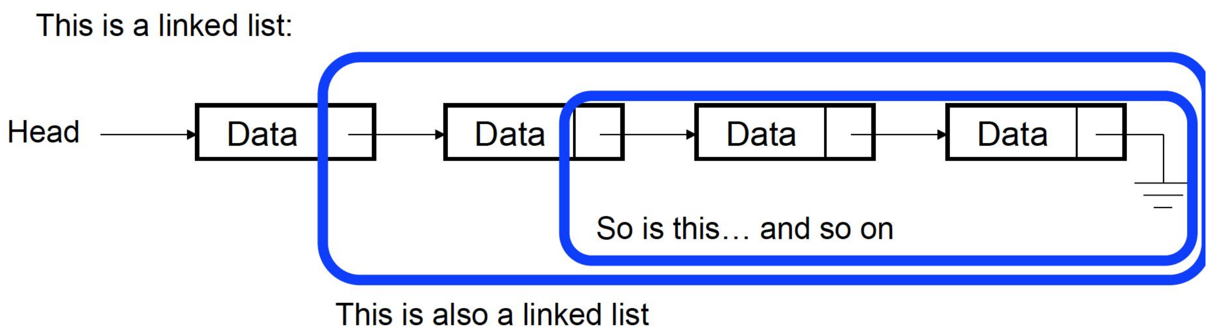
- Note in particular, that “return” is **not the same** as “print”.
- Recall from our last session:



```python
def print_in_order(lst):
    if len(lst) > 0:
        print(lst[0])
        print_in_order(lst[1:])
```

Linked Lists

- Just like algorithms, we can define data structures recursively.
- A linked list is an example of such a data structure.
    - Base case: The linked list is nothing, eg. None in Python.
    - Recursive case: The linked list has two items: the first element and the rest of the list.



Today's Practical Work

Please do these first:

- Exercises 1
  - Questions 1, 2, 4
- Exercises 3
  - Questions 1, 2, 3, 4, 5
- Exercises 4
  - Questions 1, 2, 3

If you finish all those:

- Exercises 1
  - Questions 3, 5
- Exercises 3
  - Questions 6, 7
- Exercises 4
  - Question 4

Finished everything? Take a look at the extra extension tasks.

“Now is better than never.
Although never is often better than *right* now.”

- Tim Peters

Thanks!

Contact us:

gjc510@york.ac.uk
jr1161@york.ac.uk

https://york.gjcampbell.co.uk/