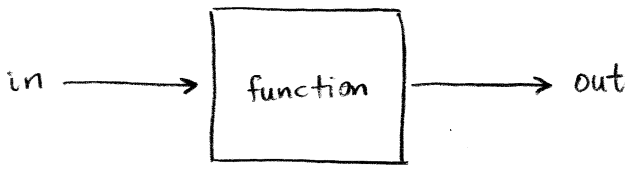
Lecture 4 - Functions

* Prayer / AMA

I. Functions

* A function is a map between an input (or multiple) and an output.



↖ can also think of this as a "black box."

example:

math $x \mapsto f(x)$, e.g. $x \mapsto x^2$

(in) (out)

other 'c b a' ⟶ 'a b c' (sort function)

* A function can sometimes be written as a formula.

$$f(x) = x^2$$

* However, there are a lot of functions that are better described as a procedure.

- step through each letters in a list and compare pairs of letters.
- Swap pairs to correct order
- Go through list again & again until sorted

} bubble sort

* If we think of a function as a procedure, we can also think of it as an abstraction or a way to name, organize & re-use code.

II. Functions in Excel

* Excel has many built-in functions
  - mathematics (e.g. abs, cos, log10)
  - statistics (e.g max, average)
  - logical (e.g. if)
  - date and time (e.g. today)

* The Excel documentation has a comprehensive list of functions and what the input & outputs are.

* In Excel, functions are written with the following syntax:

= Function(Arg1, Arg2, ...)

↑ cell is output ↑ ↑ inputs.

* You cannot define your own function in Excel (unless you use VBA - a programming language)

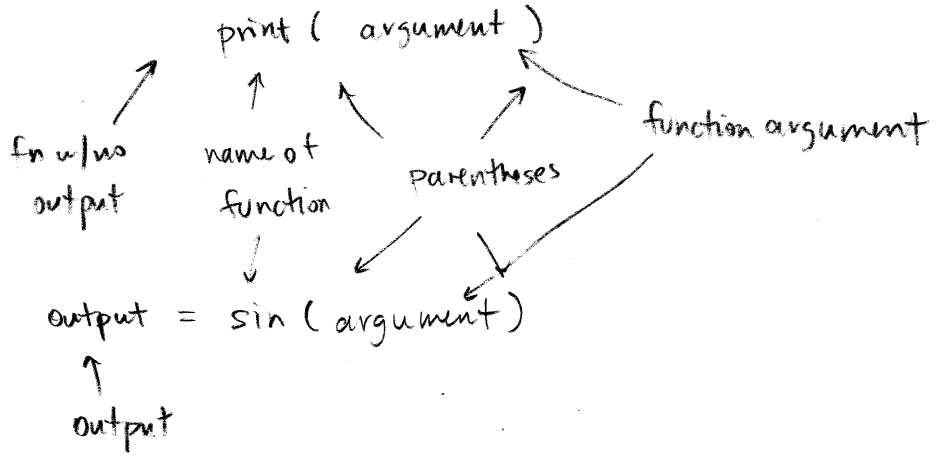
Activity Functions in Excel

- Read through the list of functions
- Practice using these functions.

II. Functions in Python

* Python also has many built in functions.
  - some are part of the language,
    e.g. print( )
    float ()
  - Python also has many libraries (modules) which contain many more functions.
    e.g. • math (standard library for math
    • csv (standard library for files Excel can read)
    • numpy (numerical python)
    • scipy (scientific python)
    • matplotlib (plotting)
  - modules can be imported using the import command. (see example)

* A function is used or called by the following Syntax:



* You can define a custom function in Python. This is a powerful feature of the language.

- Function syntax:

```
def my_function(arg1, arg2, ...):
    # code or formula goes here
    return output
```

There must be an indent here.

- you call this function the same way you call a built-in function.

```
out = my_function(arg)
```

* Variables you define inside a function are local to that function. They don't exist outside of that function. Languages do this so you can isolate code. Variables outside of a function are called global variables

e.g.
```
def my_f():
    x = 1

my_f()
print(x)   # Error!
```

vs.

```
x = 1
def my_f():
    x = 2
my_f()
print(x)   # 2
```

Activity

- Import modules
- Call functions
- Define functions

} Python