



An introduction course for biologists





**Gothenburg Global Biodiversity  
Centre - GGBC**



**Swedish LifeWatch**

# Why python?

- widely used in the scientific community
- consistent syntax, very intuitive (after getting used to it)
- great libraries for doing complex scientific computing
- very fast (much faster than R)
- very handy for 'in terminal coding' as well as writing full programs
- Lot's of great resources:
  - <https://pythonforbiologists.com/>
  - <https://www.python.org/doc/>
  - tons of online forums, you always find something:
    - e.g. <https://stackoverflow.com/>
- it's a lot of FUN!!  

# General comments

- python counting starts with 0

e.g. **list[0]** prints the **first item** of a list

- python is sensitive to white-spaces (' ') and tabs ('\t')

e.g. loops, functions, etc.

# python terminology

- `5` is an integer
- `0.3` is a float
- `'hello world'` is a string
- `['some','example','words']` or `[1,2,4,6]` are lists
- `{Adam : 8, Berta : 9, Christian : 5}` is a dictionary (key : value)
- `example_string = 'hello world'` is defining a variable named `example_string` with the value `'hello world'`

Note: if you don't assign something to a variable, it will not be saved in the python environment (few exceptions)

# jupyter, spyder, or command line

- **jupyter:** notebook, supporting python (also bash and R) *(recommended for this course!!)*
- **spyder:** python studio, similar to Rstudio for R
- **command line:** run python from terminal, no fancy visualisation but good for quick tests
- **text editor:** write code in text editor (textEdit, textWrangler, Note++, atom, visual studio code) and execute with python

**[https://github.com/tobiashofmann88/  
workshops](https://github.com/tobiashofmann88/workshops)**

# first tutorial

- introduction to basic python objects

## **Skills:**

- learn basic python syntax
- basic modification skills for strings, lists and arrays
- loops
- solve small exercises

--> this is what it looks like

**Let's go!**



# second tutorial

- two options, you can choose one (or do both)

# option A

- write small program that changes format of alignment (e.g. from nexus to fasta)

## Skills:

- use of Biopython (library)
- define functions in python
- read, modify, and output alignments
- write python program with command line interaction

--> this is what it looks like

```
dhcp-140-133:personal tobias$ python alignment_formatter.py -h
usage: alignment_formatter.py [-h] --input INPUT --input_format
                             {fasta,nexus,stockholm,phylip,clustal,emboss,phylip-sequential,phylip-relaxed,fasta-m10,ig,maf}
                             --output OUTPUT --output_format
                             {fasta,nexus,stockholm,phylip,clustal,phylip-sequential,phylip-relaxed,maf}
                             [--fix_invalid_characters]

optional arguments:
  -h, --help            show this help message and exit
  --input INPUT          Path to your input alignment file.
  --input_format {fasta,nexus,stockholm,phylip,clustal,emboss,phylip-sequential,phylip-relaxed,fasta-m10,ig,maf}
                        Alignment format of input file.
  --output OUTPUT        Name of output alignment.
  --output_format {fasta,nexus,stockholm,phylip,clustal,phylip-sequential,phylip-relaxed,maf}
                        Desired alignment format of output file.
  --fix_invalid_characters
                        Replace all invalid bases (not A,C,T,G,a,c,t,g,-) with
                        "N"
```

# option B

only for jupyter or spyder users (graphics)

- read, modify and cluster biological data, basic data science

## Skills:

- read tabular data with python
- data filtering and modification
- supervised machine learning (very basic)
- plotting

--> this is what it looks like

