

BINF2111 - Introduction to Bioinformatics

Computing

Python - you have the power!



**Richard Allen White III, PhD
RAW Lab**

Lecture 15 - Tuesday Oct 26th, 2021

Learning Objectives

- Python History
- Why Python?
- Installing python/bioconda
- Python Interpreter/Syntax
- Conda
- Quiz 15

Mid-term results

- Lab Write in:

Mean: 70%

Max: 92%

Min: 32%

- Multiple choice:

Mean: 88%

Max: 100%

Min: 76%

Stdev: 3.89

Time: 31:17 mins

Python History

- Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language
- December 1989 (same month the Simpsons appeared on FOX)
 - Monty python flying circus
(<https://docs.python.org/3/library/pprint.html>)
- Benevolent Dictator For Life?

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Readability counts.



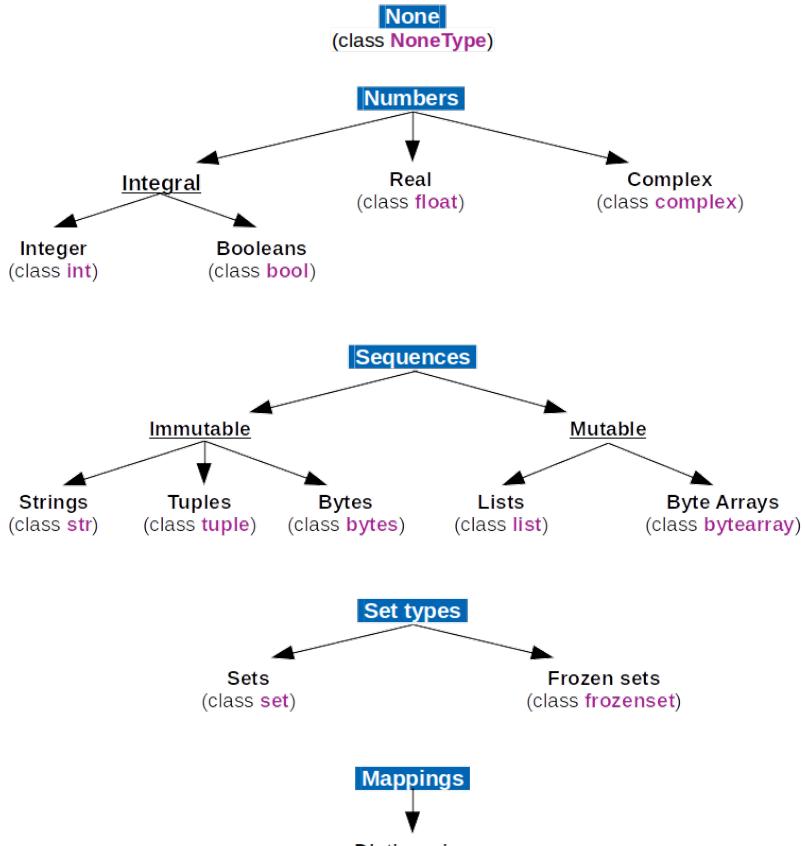
Guido van Rossum
"Benevolent
Dictator For Life"

Now - "permanent
vacation" 2018

Why python?

- Python is a powerful scripting language that can be used at many skill levels
- Beginner users can accomplish a lot with simple scripts
- Advanced users can write fully functioning object-oriented code
- Bioinformatics involves a lot of work with sequences, which are strings – and python is great for manipulating strings

Python 3
The standard type hierarchy



Callable
< Functions, Methods, Classes >

Modules

Type	Mutability	Description	Syntax examples
<code>bool</code>	immutable	Boolean value	<code>True</code> <code>False</code>
<code>bytearray</code>	mutable	Sequence of bytes	<code>bytearray(b'Some ASCII')</code> <code>bytearray(b"Some ASCII")</code> <code>bytearray([119, 105, 107, 105])</code>
<code>bytes</code>	immutable	Sequence of bytes	<code>b'Some ASCII'</code> <code>b"Some ASCII"</code> <code>bytes([119, 105, 107, 105])</code>
<code>complex</code>	immutable	Complex number with real and imaginary parts	<code>3+2.7j</code> <code>3 + 2.7j</code>
<code>dict</code>	mutable	Associative array (or dictionary) of key and value pairs; can contain mixed types (keys and values), keys must be a hashable type	<code>{'key1': 1.0, 3: False}</code> <code>{}</code>
<code>ellipsis^a</code>	immutable	An ellipsis placeholder to be used as an index in NumPy arrays	<code>...</code> <code>Ellipsis</code>
<code>float</code>	immutable	Double-precision floating-point number. The precision is machine-dependent but in practice is generally implemented as a 64-bit IEEE 754 number with 53 bits of precision. ^[101]	<code>1.33333</code>
<code>frozenset</code>	immutable	Unordered set, contains no duplicates; can contain mixed types, if hashable	<code>frozenset([4.0, 'string', True])</code>
<code>int</code>	immutable	Integer of unlimited magnitude ^[102]	<code>42</code>
<code>list</code>	mutable	List, can contain mixed types	<code>[4.0, 'string', True]</code> <code>[]</code>
<code>NoneType^a</code>	immutable	An object representing the absence of a value, often called <code>null</code> in other languages	<code>None</code>
<code>NotImplementedType^a</code>	immutable	A placeholder that can be returned from overloaded operators to indicate unsupported operand types.	<code>NotImplemented</code>
<code>range</code>	immutable	A Sequence of numbers commonly used for looping specific number of times in <code>for</code> loops ^[103]	<code>range(-1, 10)</code> <code>range(10, -5, -2)</code>
<code>set</code>	mutable	Unordered set, contains no duplicates; can contain mixed types, if hashable	<code>{4.0, 'string', True}</code> <code>set()</code>
<code>str</code>	immutable	A character string: sequence of Unicode codepoints	<code>"Wikipedia"</code> <code>"Wikipedia"</code> <code>"""Spanning multiple lines"""</code>
<code>tuple</code>	immutable	Can contain mixed types	<code>(4.0, 'string', True)</code> <code>('single element',)</code> <code>()</code>

	Python	R
General	Python is a general-purpose programming language for data analysis and scientific computing.	R is a functional programming environment and language for statistical computing and graphics.
Objective	Data Science, Web Development, Embedded Systems	Data Science & Statistical Modeling
IDE	iPython, Pycharm, Jupyter Notebook, Spyder	Rstudio, R GUI, R KWARD
Data Collection	Supports CSV files, SQL, JSON , and webscraping with BeautifulSoup .	Can also import csv files with built-in readr library. R's library RCurl provides a simple way to make API requests, similar to Python's requests package.
Data Analysis	Organize dataframes with Pandas filtering, sorting. Python takes a more streamlined approach for data science projects.	Complex data visualization tools make the exploratory data analysis (EDA) process much more complex than Python.
Essential Packages & Libraries	Numpy, Pandas, matplotlib, scipy, scikit-learn, TensorFlow	caret, stringr, ggplot2, knitr, tidyverse, markdown, shiny, forcats, haven
Database Handling Capacity	Can easily handle large data because there are less constraints for memory usage	R computes everything in memory, so its capabilities are limited by RAM size. A major downfall of R is the inability to handle massive amounts of data
Data Visualization	Despite the capabilities of data visualization tools like Matplotlib and Seaborn , Python fails to measure up to data visualization features of R.	Developed by and for statisticians, R has complex data visualization features.
Syntax	The 'zen of python' is that there's a proper way to write code.	R doesn't have this set of rules. Also indexing starts at 1, which can be considered unconventional for general programmers.
Learning Curve	Simple and readable code structure makes it easier for beginners to learn. It also allows for object-oriented programming. It also offers a wide range of data structures that you wouldn't expect from a general-purpose language.	R's functional syntax isn't easy for beginners, but not too challenging for those well versed in programming. It also offers a few data structures, but fails to handle large amounts of data.

Installing Python

Linux: sudo yum install python
sudo apt-get install python

Mac: instructions can be found on
<https://www.python.org/about/>

Python Command line

On your computer, type “python” into a terminal

Your command line prompt should change to

>>>

Now you are in a new environment where what you type is interpreted as python commands and not as bash commands (your linux environment)

Python Interpreter

The interpreter translates one statement at a time.

Easy to debug as it stops at the first error.

Slower than compiled code.

Memory efficient.

Lines of commands (scripts) can be passed through the interpreter.

Python Interpreter

In order to quit or exit back to the command line type exit() or quit().

If you mess up a line, you can always interrupt the command using ctrl-C.

File Edit View Search Terminal Help

(base) docwhite@system76-pc:~\$

which python

File Edit View Search Terminal Help

```
docwhite@system76-PC:~$ which python  
/home/docwhite/miniconda3/bin/python  
docwhite@system76-PC:~$ █
```

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>



Python Syntax

Functions

Comments

Variables

Operators

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

print("Hello World")



Python Functions

Functions - self-contained programs that execute a specific task. Once defined, can call the function.

Example:

print is a built-in python function

In your python window, type:

```
print("Hello world!")
```

functions are always followed by parentheses

e.g. exit() or quit() to quit the interpreter

the text in parentheses is an **argument** that the function can take

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> 

print('Hello World')



#testing

print('Hello World')

print("hello world")

Python Syntax

Functions

Comments

Variables

Operators

Python Comments

Comments - annotation of python code

Example:

print is a built-in python function

In your python window, type:

```
print("Hello world!") #prints hello world
```

Comments are denoted by the # sign and are terminated at the end of the line

This helps you remember what your command does, and to annotate things within your script/program.

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

Var = 1

Var = 2



Python Syntax

Functions

Comments

Variables

Operators

Python Variables

Variables - variables allow you to store values within data types

Example:

```
var1 = 1
```

```
var2 = 10
```

```
x = y = z = 2
```

Try to print out var1, var2, x, y, and z from your interpreter.

Python Variables

Variables - variables allow you to store values within data types

Data types:

Numbers

Strings

List

Tuple

Dictionary

Python Variables

Variables - variables allow you to store values within data types

Data types:

Numbers: (int, long, float, complex, 45, 0xDE, 4.52, 3e-26j)

Strings: ("words go in here")

Array: contiguous elements in memory, homogeneous data

List: dynamic array, scattered in memory, heterogeneous data

Tuple: heterogeneous data, immutable, static

Dictionary: index of words and relationships

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> □

X = int(4)

X = ?

X = int(4.256)

X = ?

X = float(4.256)

X = ?

type(X)



File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> 

X = int(4)

X = 4

X = int(4.256)

X = 4

X = float(4.256)

X = 4.256

type(X) = <type 'float'>



Python Variables

Variable type becomes important when you try to use operators and built in methods

Math operators work on numbers

String methods work on strings

Very few of these overlap

Some important types:

`int()` = an integer value

`float()` = a floating point value

`str()` = a character string

`bool()` = a Boolean value (`True/False`)

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> 

print('Hello\nWorld')



File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
print('Hello\nWorld')
```

Hello
World

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

dna = 'ATGTGC'

print(dna)

len(dna)



Python Syntax

Functions

Comments

Variables

Operators

Python Operators

Math Operators - Mathematical operators to manipulate variables.

Addition: $a + b$

Subtraction: $a - b$

Multiplication: $a * b$

Division: a / b

Exponential: $a ** b$

Modulo: $a \% b$

Order of operations and parenthesis are important in mathematical order.

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

A = 1

B = 2

print(A+B)

int(A) or int(B)



File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

dna = 'ATG' + 'TGC'

print(dna)
len(dna)

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

start = 'ATG'

stop = start + 'TAG'

print(stop)

len(stop)

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
print(start.upper())
print(start.lower())
```

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

print(dna.replace('T', 'U'))

print(dna.replace('ATG','GGC'))

File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
print("Hello" + "" + "world")
```



File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
 dna = "ATGGCGC"
 dna_length = len(dna)
 print("The length of the DNA
 sequence is " + str(dna_length))
```



File Edit View Search Terminal Help

docwhite@system76-**pc**:~\$ python

Python 3.7.4 (default, Aug 13 2019, 20:35:49)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
 dna = "ATGGCGC"
 dna_length = len(dna)
 print("The length of the DNA
 sequence is " + str(dna_length))
```



User guide

Concepts

Conda commands

Conda packages

Conda package specification

Conda channels

Conda environments

Installing with conda

Conda update versus conda install

Installing conda packages offline

Installing conda packages with a specific build number

Conda performance

Conda for data scientists

Getting started with conda

Installation

Configuration

Tasks

Cheat sheet

Troubleshooting

Conda configuration

Conda Python API

Release notes

Command reference

Glossary

Read the Docs

v: latest ▾

Installing with conda

To install conda packages, in the terminal or an Anaconda Prompt, run:

```
conda install [packagename]
```

During the install process, files are extracted into the specified environment, defaulting to the current environment if none is specified. Installing the files of a conda package into an environment can be thought of as changing the directory to an environment, and then downloading and extracting the artifact and its dependencies---all with the single `conda install [packagename]` command.

Read more about [conda environments and directory structure](#).

When you `conda install` a package that exists in a channel and has no dependencies, conda:

Looks at your configured channels (in priority).

Reaches out to the repodata associated with your channels/platform.

Parses repodata to search for the package.

Once the package is found, conda pulls it down and installs.

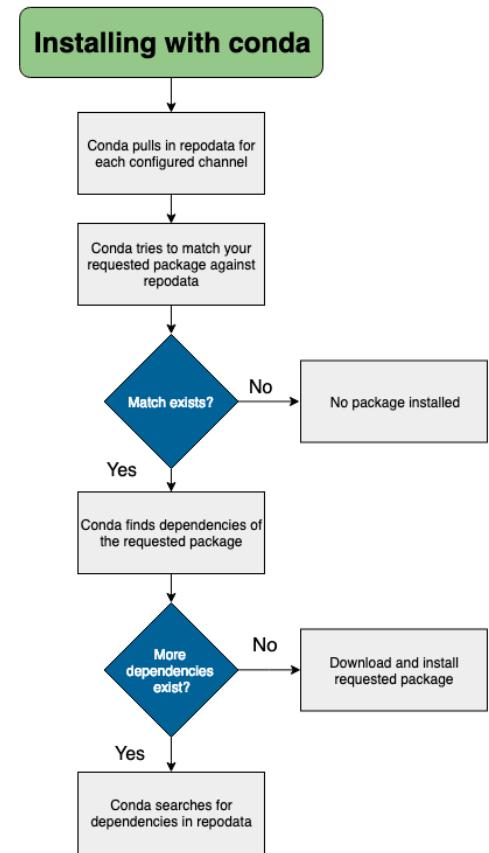
Conda update versus conda install

`conda update` is used to update to the latest compatible version. `conda install` can be used to install any version.

Example:

If Python 2.7.0 is currently installed, and the latest version of Python 2 is 2.7.5, then `conda update python` installs Python 2.7.5. It does not install Python 3.

If Python 2.7.0 is currently installed, and the latest version of Python is 2.9.0, then `conda install python` installs Python 2.9.0



File Edit View Search Terminal Help

(base) docwhite@system76-pc:~\$



File Edit View Search Terminal Help

(base) docwhite@system76-pc:~\$ ┌─[]



(base) ?

When conda is installed this will
show up.

How do you get rid of it?

File Edit View Search Terminal H

docwhite@system76-pc:~\$



(base) ?
conda deactivate

File Edit View Search Terminal H

docwhite@system76-pc:~\$



What if I want base back?

File Edit View Search Terminal Help

```
docwhite@system76-pc:~$ conda activate base
(base) docwhite@system76-pc:~$ which python
/home/docwhite/miniconda3/bin/python
(base) docwhite@system76-pc:~$ █
```



File Edit View Search Terminal Help

```
docwhite@system76-pc:~$ conda activate base
(base) docwhite@system76-pc:~$ which python
/home/docwhite/miniconda3/bin/python
(base) docwhite@system76-pc:~$ █
```

How do I list all the
conda packages?



File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$ conda info --envs
```

```
# conda environments:
```

```
#
```

```
base
```

```
* /home/docwhite/miniconda3
```

```
cerberus_env
```

```
/home/docwhite/miniconda3/envs/cerberus_env
```

```
prokka_env
```

```
/home/docwhite/miniconda3/envs/prokka_env
```

```
py3.6
```

```
/home/docwhite/miniconda3/envs/py3.6
```

conda info --envs

How do I activate a package?

Activities Terminal Mon 23:01 ● docwhite@system76-pc: ~

```
File Edit View Search Terminal Help
(base) docwhite@system76-pc:~$ conda info --envs
# conda environments:
#
base          * /home/docwhite/miniconda3
cerberus_env      /home/docwhite/miniconda3/envs/cerberus_env
prokka_env       /home/docwhite/miniconda3/envs/prokka_env
py3.6           /home/docwhite/miniconda3/envs/py3.6

(base) docwhite@system76-pc:~$ conda activate py3.6
(py3.6) docwhite@system76-pc:~$ 
```

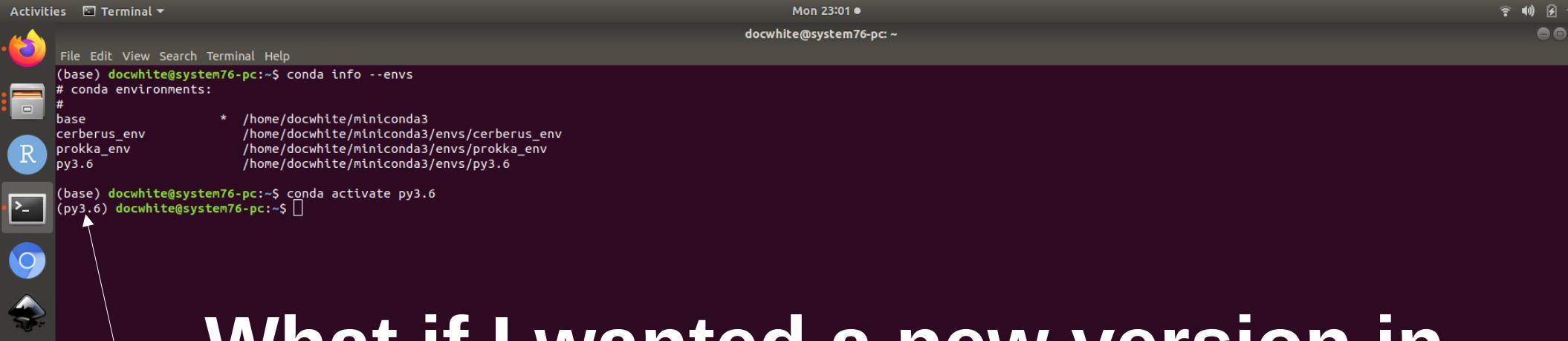
How do I activate a package?

`conda activate py3.6`

which python

`/home/docwhite/miniconda3/envs/py3.6/bin/python`

Activities Terminal Mon 23:01 ● docwhite@system76-pc: ~



```
File Edit View Search Terminal Help
(base) docwhite@system76-pc:~$ conda info --envs
# conda environments:
#
base          * /home/docwhite/miniconda3
cerberus_env      /home/docwhite/miniconda3/envs/cerberus_env
prokka_env       /home/docwhite/miniconda3/envs/prokka_env
py3.6           /home/docwhite/miniconda3/envs/py3.6

(base) docwhite@system76-pc:~$ conda activate py3.6
(py3.6) docwhite@system76-pc:~$ 
```

What if I wanted a new version in
python 3.9?



Mon 23:07 ●
docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$ conda create -n py3.9 python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
--> WARNING: A newer version of conda exists. <=-
  current version: 4.8.2
  latest version: 4.10.3
```

Please update conda by running

```
$ conda update -n base conda
```

Package Plan

```
environment location: /home/docwhite/miniconda3/envs/py3.9
added / updated specs:
- python=3.9
```

The following packages will be downloaded:

package	build				
ca-certificates-2021.10.8	ha878542_0	139 KB	conda-forge		
ld_impl_linux-64-2.36.1	hea4e1c9_2	667 KB	conda-forge		
libffi-3.4.2	h9c3ff4c_4	57 KB	conda-forge		
libgcc-ng-11.2.0	h1d223b6_11	887 KB	conda-forge		
libgomp-11.2.0	h1d223b6_11	427 KB	conda-forge		
libstdcxx-ng-11.2.0	he4da1e4_11	4.2 MB	conda-forge		
libzlib-1.2.11	h36c2ea0_1013	59 KB	conda-forge		
openssl-3.0.0	h7f98852_1	2.9 MB	conda-forge		
pip-21.3.1	pyhd8ed1ab_0	1.2 MB	conda-forge		
python-3.9.7	hf930737_3_cpython	27.5 MB	conda-forge		
python_abi-3.9	_2_cp39	4 KB	conda-forge		
setuptools-58.2.0	py39hf3d152e_0	1011 KB	conda-forge		
sqlite-3.36.0	h9cd32fc_2	1.4 MB	conda-forge		
tk-8.6.11	h27826a3_1	3.3 MB	conda-forge		
tzdata-2021e	he74cb21_0	121 KB	conda-forge		
wheel-0.37.0	pyhd8ed1ab_1	31 KB	conda-forge		
zlib-1.2.11	h36c2ea0_1013	86 KB	conda-forge		
Total:		43.9 MB			

The following NEW packages will be INSTALLED:

```
_libgcc_mutex      conda-forge/linux-64::_libgcc_mutex-0.1-conda_forge
_ompmp_mutex       conda-forge/linux-64::_ompmp_mutex-4.5-1_gnu
ca-certificates    conda-forge/linux-64::ca-certificates-2021.10.8-ha878542_0
ld_impl_linux-64   conda-forge/linux-64::ld_impl_linux-64-2.36.1-hea4e1c9_2
libffi              conda-forge/linux-64::libffi-3.4.2-h9c3ff4c_4
```

What if I wanted a
new version in python
3.9?
**conda create -n py3.9
python=3.9**

Activities Terminal

Mon 23:09



docwhite@system76-pc: ~

```
File Edit View Search Terminal Help
(base) docwhite@system76-pc:~$ conda info --envs
# conda environments:
#
base          * /home/docwhite/miniconda3
cerberus_env   /home/docwhite/miniconda3/envs/cerberus_env
prokka_env     /home/docwhite/miniconda3/envs/prokka_env
py3.6          /home/docwhite/miniconda3/envs/py3.6
py3.9          /home/docwhite/miniconda3/envs/py3.9
```

```
(base) docwhite@system76-pc:~$ conda activate py3.9
(py3.9) docwhite@system76-pc:~$ 
```

conda activate py3.9

Quiz 15

- On canvas now