

# Python coding conversion

# Python's principles

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one -- obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than \*right\* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

# Python coding conventions

- There are many Python coding conventions.
  - PEP8
  - Google Python Style Guide
  - Many more

# Code layout

- Indentation
- Tabs/ `**spaces**`
- Max line length = 79
- Blank lines: 2 lines for top level functions / definitions

# How to write names

- There are many styles for writing names.
  - ifyouaregoodyoumightwritesome thinglikethis
  - CamelCaseIsUsedOften
  - maybeStartWithSmallCase
  - or\_you\_might\_want\_to\_use\_underscore
  - OR\_UPPER\_CASE\_WITH\_UNDERSCORE
  - ORIFYOUIARE REALLYGOOD
- It is a common practice to use different styles for different kinds of names.

# Names

- Class names – **CamelCase**
- Function and variable names – **lower\_case\_with\_underscore**
- Arguments
  - Use: **self** for 1<sup>st</sup> argument of instance methods
- Constants – **UPPER\_CASE\_WITH**

Last words

**Be Consistent**