
python
LIST
COMPREHENSIONS

BENEFITS:

Create a new list based on the values of an existing list..

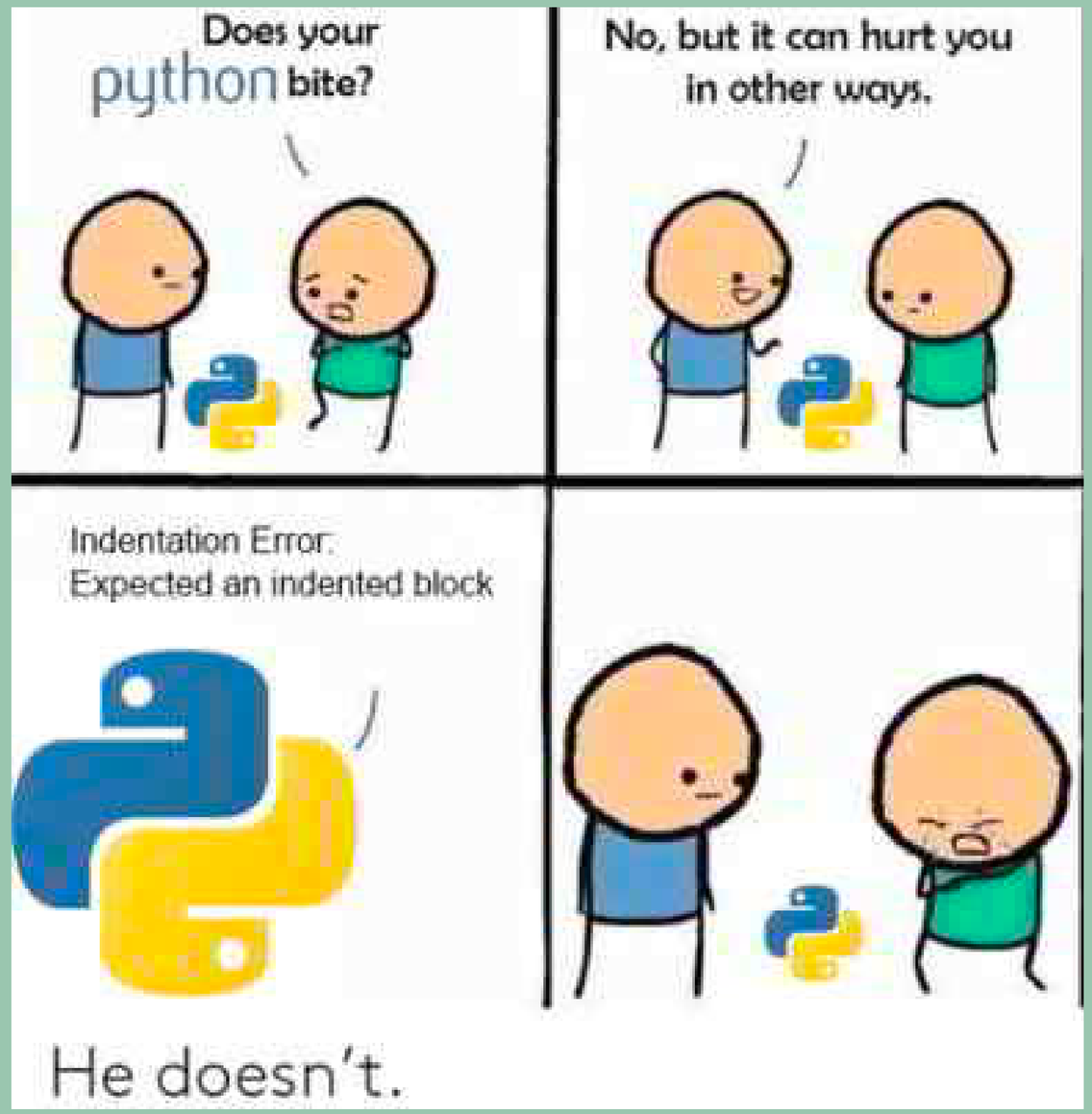
- Clean, elegant way of writing a for loop.
- **Declarative** – tell it what to do, instead of how to do it! (*Imperative)*
- Single tool that can be used in many situations: i.e. mapping, filtering.
- One-liner!

SYNTAX:

```
def multiply_by_two(numbers):
    results = []
    for num in numbers:
        results.append(num * 2)
    return results
```

```
def multiply_by_two(numbers):
    return [num * 2 for num in numbers]
```

Can transform any data type, and from one data type to another.



WRITE PYTHONIC CODE.

WRITE PYTHONIC CODE.

PEP 8 — the Style Guide for Python Code

This stylized presentation of the well-established PEP 8 was created by Kenneth Reitz (for humans).

Introduction

This document gives coding conventions for the Python code comprising the standard library in the main Python distribution. Please see the companion informational PEP describing style guidelines for the C code in the C implementation of Python [1].

This document and PEP 257 (Docstring Conventions) were adapted from Guido's original Python Style Guide essay, with some additions from Barry's style guide [2].

This style guide evolves over time as additional conventions are identified and past conventions are rendered obsolete by changes in the language itself.

Many projects have their own coding style guidelines. In the event of any conflicts, such project-specific guides take precedence for that project.

A Foolish Consistency is the Hobgoblin of Little Minds

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it


python
BUBBLE SORT
& SWAPPING
Algorithms!

BUBBLE SORT

An algorithm for ordering a list by using a swapping function.

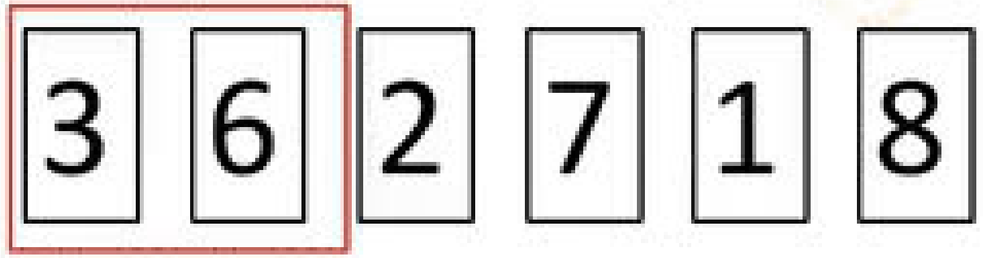
- Time complexity: O(n2)
- Space complexity: O(1)

CONSIDERED THE SIMPLEST SORTING ALGO.

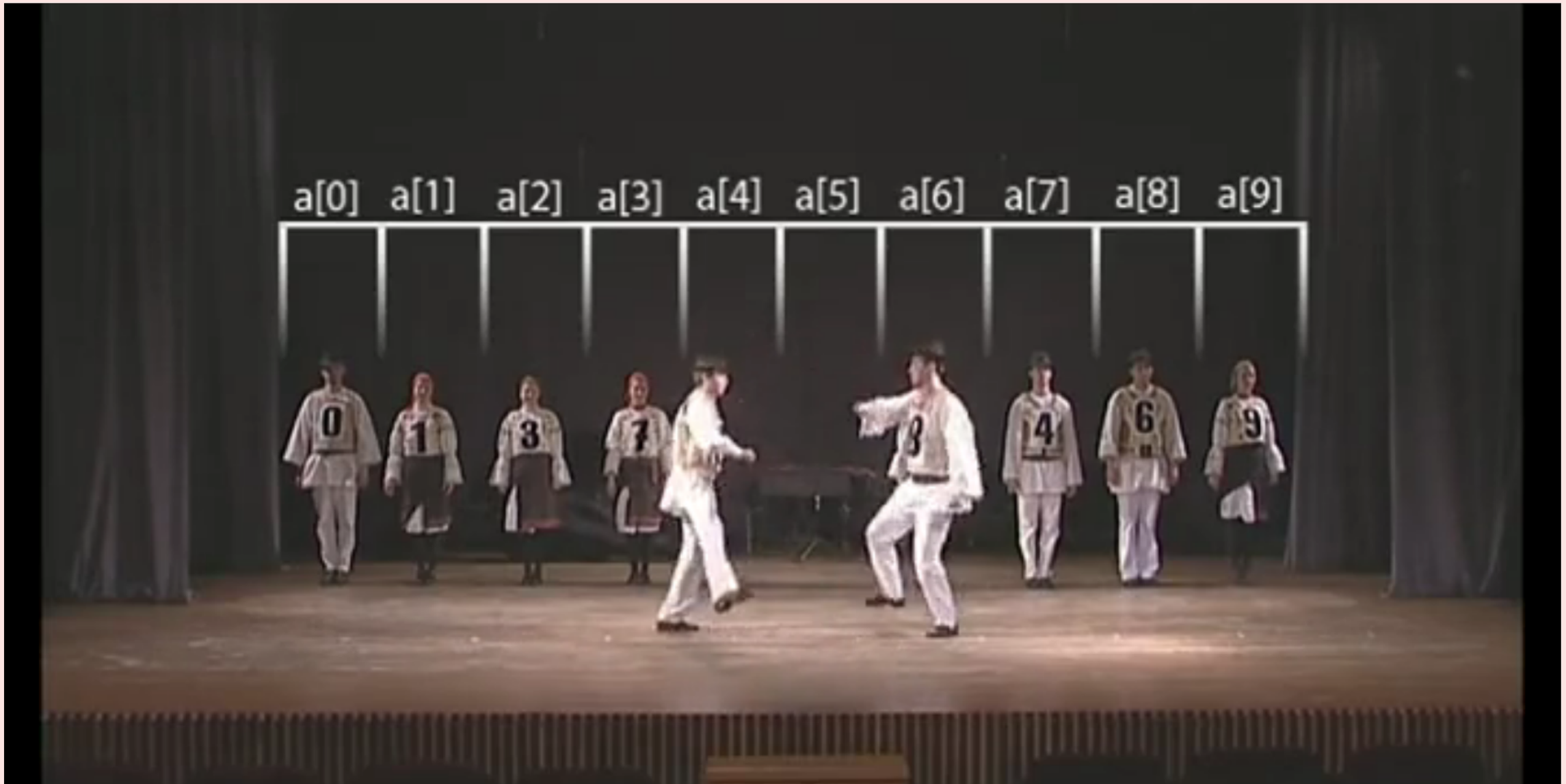
Python swapping makes this cool.

HOW IT WORKS:

Bubble Sort:



BUBBLE SORT:



WRITE PYTHONIC CODE.

Swap two numbers:

```
# Python program to swap two variables

x = 5
y = 10
```

```
# create a temporary variable and swap the values
temp = x
x = y
y = temp
```

WRITE PYTHONIC CODE.

Swap two numbers, pythonically:

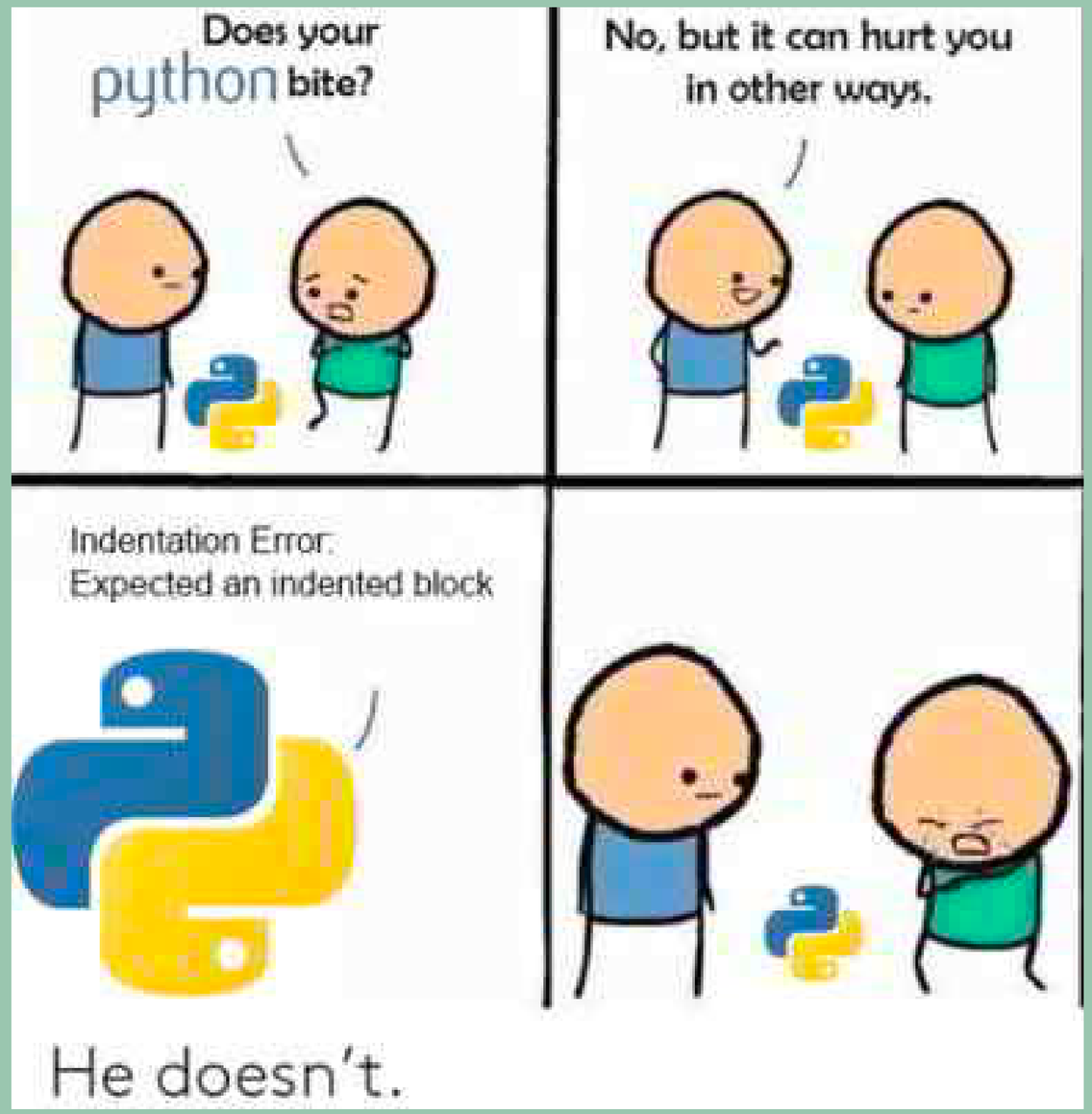
```
x, y = y, x
```

WRITE PYTHONIC CODE.

Swap two numbers, pythonically:

```
x, y = y, x
```

MULTIPLE ASSIGNMENT



WRITE PYTHONIC CODE.

WRITE PYTHONIC CODE.

PEP 8 — the Style Guide for Python Code

This stylized presentation of the well-established PEP 8 was created by Kenneth Reitz (for humans).

Introduction

This document gives coding conventions for the Python code comprising the standard library in the main Python distribution. Please see the companion informational PEP describing style guidelines for the C code in the C implementation of Python [1].

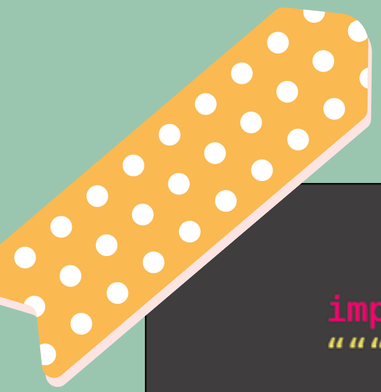
This document and PEP 257 (Docstring Conventions) were adapted from Guido's original Python Style Guide essay, with some additions from Barry's style guide [2].

This style guide evolves over time as additional conventions are identified and past conventions are rendered obsolete by changes in the language itself.

Many projects have their own coding style guidelines. In the event of any conflicts, such project-specific guides take precedence for that project.

A Foolish Consistency is the Hobgoblin of Little Minds

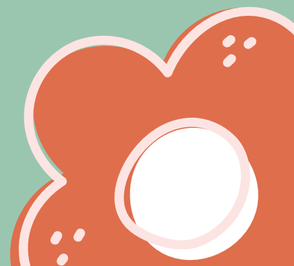
One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it



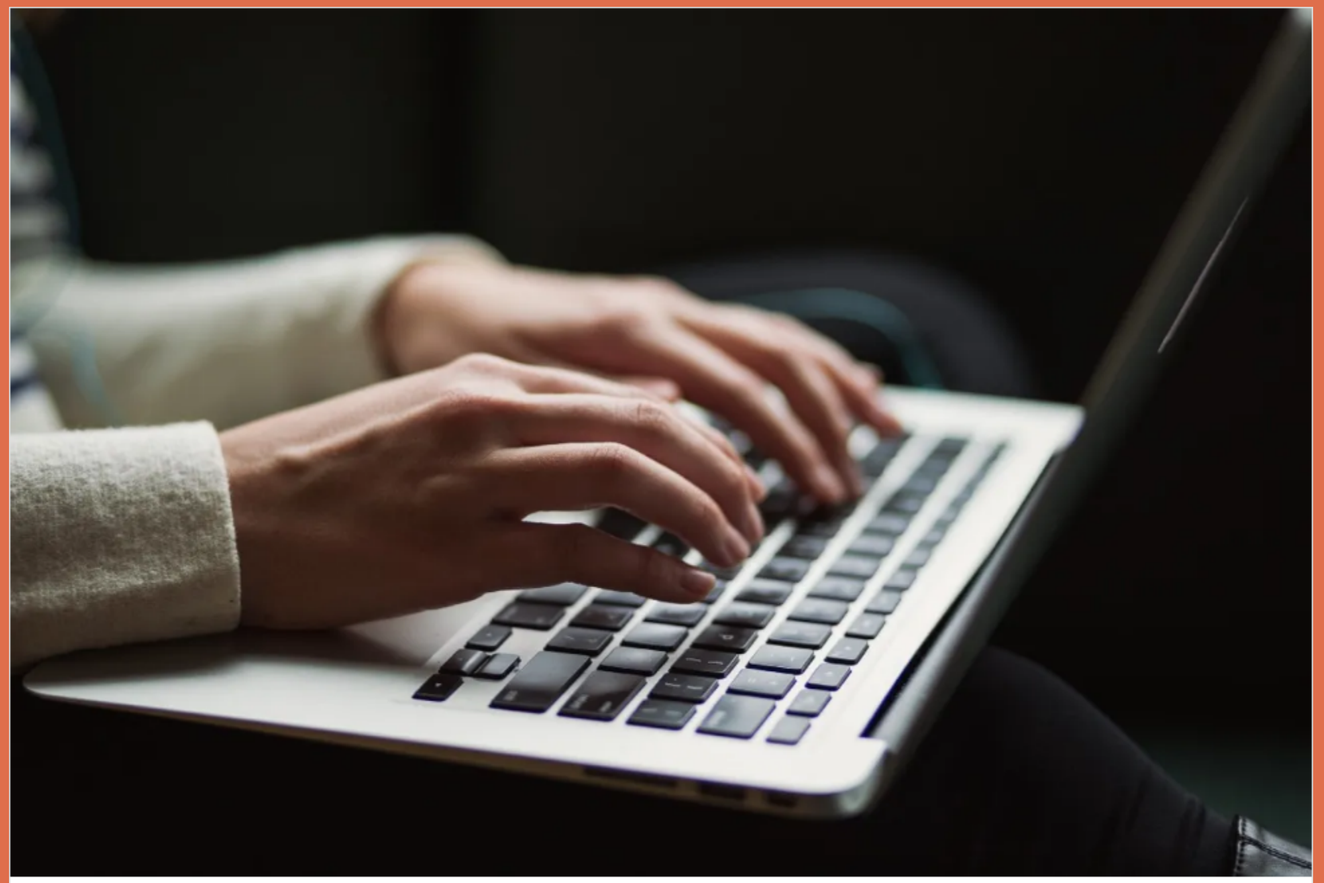
```
import this
"""The Zen of Python, by Tim Peters. (poster by Joachim Jablon)"""

Beautiful is better than ugly.
Explicit is better than impl..
Simple is better than cσmplεx.
Cσmplεx is better than cOmp1|c@ted.
Flat is better than nested.
S p a r s e  is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
raise PythonicError("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 thanrightnow.
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!
```

WRITE PYTHONIC CODE.



TUPLE UNPACKING



5 Features that Make Python Unique

Here are 5 features and syntactical sugars that make Python so different and unique.

Medium / Jun 28, 2021