



 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:

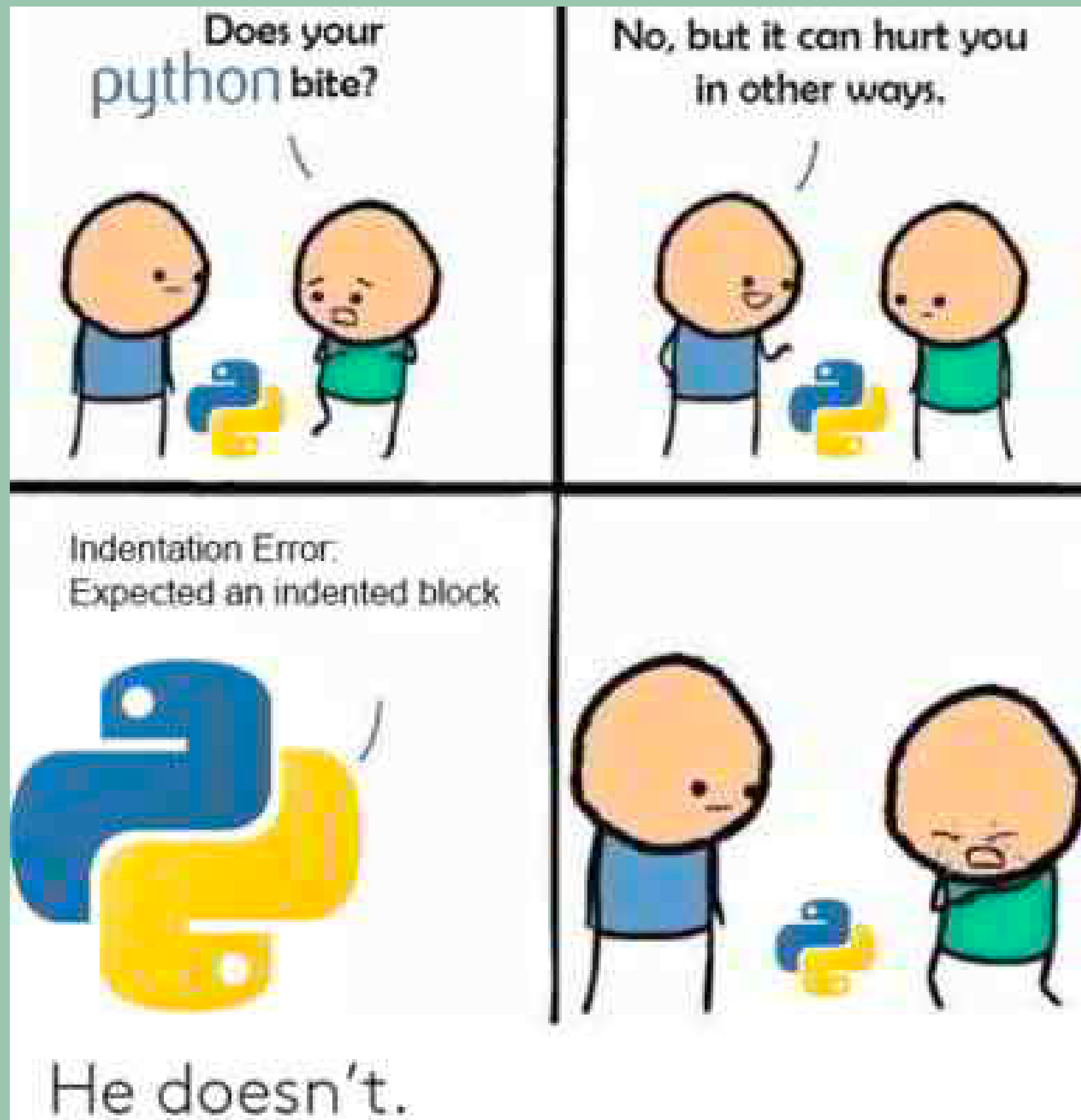
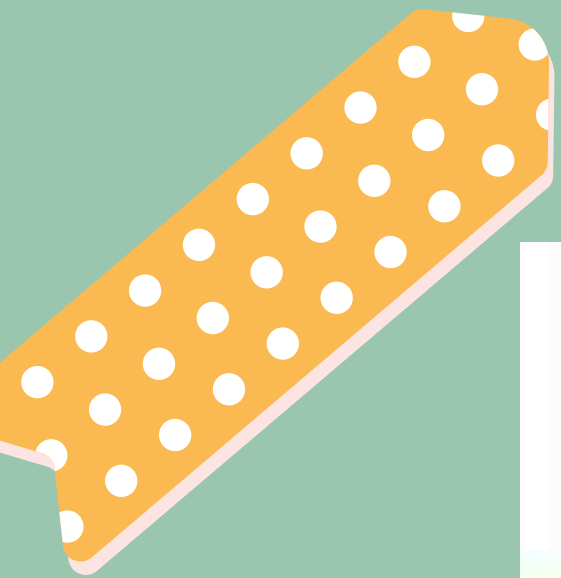
```
def multiply_by_two(numbers):  
    results = []  
    for num in numbers:  
        results.append(num * 2)  
    return results
```

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

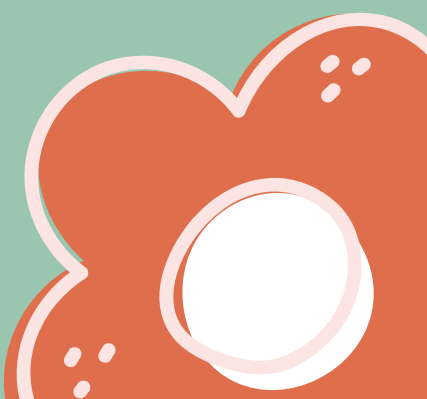
The diagram illustrates the transformation of a traditional loop-based function into a more concise list comprehension. Arrows show the following mappings:

- A grey arrow from the `results = []` line to the opening square bracket of the list comprehension `[num * 2 for num in numbers]`.
- A yellow arrow from the `for num in numbers:` line to the `for num in numbers` part of the list comprehension.
- A light blue arrow from the `results.append(num * 2)` line to the `num * 2` part of the list comprehension.
- A grey arrow from the `return results` line to the closing square bracket of the list comprehension.

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

A Foolish Consistency is the
Hobgoblin of Little Minds

Code lay-out

- *Indentation*
- *Tabs or Spaces?*
- *Maximum Line Length*
- *Should a line break before or after a binary operator?*
- *Blank Lines*
- *Source File Encoding*
- *Imports*
- *Module level dunder names*

String Quotes

Whitespace in Expressions and
Statements

- *Pet Peeves*
- *Other Recommendations*

When to use trailing commas

Comments

- *Block Comments*
- *Inline Comments*
- *Documentation Strings*

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