LAC: Language and the Computer
Python for non-Programmers

Variables and Simple Data Types

Francis Bond
Division of Asian Studies
Palacký University Olomouc
bond@ieee.org

September 26, 2024

Variables

Naming and Using Variables

- Variable names can contain only letters, numbers, and underscores
  They cannot start with a number
- Spaces are not allowed in variable names
- Avoid using Python keywords and function names as variable names
- Variable names should be short but descriptive.
  - `name` is better than `n`
  - `student_name` is better than `s_n`
  - `name_length` is better `length_of_persons_name`
- Be careful when using the lowercase letter l and the uppercase letter O because they could be confused with the numbers 1 and 0.

Avoiding Name Errors When Using Variables

Everyone mistypes names all the time! Get used to it.

Variables Are Labels

They refer to a space in the computers memory.

Strings

Changing Case in a String with Methods

```
>>> name = "ada lovelace"
>>> print(name.title())
Ada Lovelace
>>> print(name.upper())
ADA LOVELACE
>>> print(name.lower())
ada lovelace
```

Using Variables in Strings

```
>>> name = "ada lovelace"
>>> print(f"I respect {name.title()}.")

I respect Ada Lovelace.
```

Adding Whitespace to Strings with Tabs or Newlines

```
tab = '\t'
newline = '\n'
```

Stripping Whitespace

```
>>> favorite_language = 'python  '
>>> favorite_language = favorite_language.rstrip()
>>> favorite_language
'python'
```

Removing Prefixes

```
>>> nostarch_url = 'https://nostarch.com'
>>> nostarch_url.removeprefix('https://')
'nostarch.com'
```

Avoiding Syntax Errors with Strings

If you need to have a single or double quote in your string, use the other one to construct it:

```
str1 = "python's strings"
str2 = 'I said "Hi!"'
```

If you must use both, then you can either escape the string with a 'ór use three quotes:

```
str3 = """Trust fund, 6'5", blue eyes"""
str4 = '''Trust fund, 6'5", blue eyes'''
str5 = "Trust fund, 6'5\", blue eyes"
str6 = 'Trust fund, 6\'5", blue eyes'
```

Using three quotes also allows multi-line strings:

```
"""
I'm looking for a man in finance
Trust fund, 6'5", blue eyes
Finance, trust fund, 6'5", blue eyes
Finance, trust fund, 6'5", blue eyes
Finance, trust fund, 6'5"
"""
```

Numbers

Integers

Floats

For real numbers — stored as binary, so sometimes weird rounding erros

Integers and Floats

Integers are more precise, floats are more flexible

Underscores in Numbers

These two are the same

```
10\_000\_000 = 10000000
```

But the first is easier for people read!

Comments

How Do You Write Comments?

```
# this is a comment
```

What Kinds of Comments Should You Write?

- It is easier to write code than to read it
- So when you write code, you should make it easier to understand
- Future you will thank you!

  When you're deciding whether to write a comment, ask yourself if you had to consider several approaches before coming up with a reasonable way to make something work; if so, write a comment about your solution. It's much easier to delete extra comments later than to go back and write com- ments for a sparsely commented program. (p30)

The Zen of Python

Multiple Assignment

This can help shorten your programs and make them easier to read

```
x, y = 1, -2
```

Constants

The tradition is to write these as in all caps

```
THRESHOLD = .05
```

Philosophy

```
>>> import this

The Zen of Python, by Tim Peters

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!
```

Acknowledgements

➢ This is basically a summary of Python Crash Course 3rd Edition, Chapter 2 by Eric Matthes (2023)
ISBN-13: 978-1-7185-0270-3 (print)
ISBN-13: 978-1-7185-0271-0 (ebook)