(Programming) Languages, Python(3), PythonAnywhere, Canopy

CORE-UA 109.01, Joanna Klukowska
adapted from slides for CSCI-UA.002 by D. Engle, C. Kapp and J. Versoza

Natural Languages

What are some characteristics of natural languages?

Natural Languages

What are some characteristics of natural languages?

- meant for communication between people (usually!)
- these are the languages that people speak (English, Urdu, Catalan, etc.)
- not intentionally designed - evolved naturally / organically
- usually verbose
- room for ambiguity, interpretation, etc.

A Programming Language...

on the other hand ...

A Programming Language...

on the other hand ...

- is artificial / synthetic
- is created specifically to communicate **instructions** to a machine
- usually has a rigid structure or grammar
- is usually very strict about **syntax**
- some words / symbols have special meanings
- is usually exact / explicit in its meaning
- information dense; each character counts!
- provides a layer of **abstraction** between programmer and machine

A Programming Language...

on the other hand ...

- is artificial / synthetic
- is created specifically to communicate **instructions** to a machine
- usually has a rigid structure or grammar
- is usually very strict about **syntax**
- some words / symbols have special meanings
- is usually exact / explicit in its meaning
- information dense; each character counts!
- provides a layer of **abstraction** between programmer and machine

This Makes Reading and Writing Programs Much Different From Natural Languages

Reading a Programming Language

- reading a program requires analysis of a program's **structure**
- information density can make comprehension a slow process
- again, meaning is usually unambiguous
  - behavior / meaning usually backed by formal specifications
  - though language is unambiguous, programmer intent may not be!

Reading a Programming Language

- reading a program requires analysis of a program's **structure**
- information density can make comprehension a slow process
- again, meaning is usually unambiguous
  - behavior / meaning usually backed by formal specifications
  - though language is unambiguous, programmer intent may not be!

Writing a Program

- easy to introduce minor syntactic errors (you must be careful with syntax and grammar)
- may have multiple implementation possibilities
- may offer constructs for abstraction and preventing repetition
  - potentially further improves on human readability
  - important for refactoring, improving your program, and future maintenance
- don't necessarily have to know entire language to write a program!

Sooo Many Programming Languages...

(You could even make one yourself!)

These Are 20 of the Most Used Languages Today (At least, according to TIOBE, August 2017)

1. Java
2. C
3. C++
4. C#
5. **Python**
6. Visual Basic .NET
7. PHP
7. JavaScript
8. Perl
9. Ruby
10. Swift
11. Delphi
12. VisualBasic
14. Assembly
15. R
16. Go
17. MATLAB
18. Objective-C
19. Scratch
20. Dart

Sooo Many Programming Languages...

(You could even make one yourself!)

These Are 20 of the Most Used Languages Today (At least, according to TIOBE, August 2017)

1. Java
2. C
3. C++
4. C#
5. **Python**
6. Visual Basic .NET
7. PHP
7. JavaScript
8. Perl
9. Ruby
10. Swift
11. Delphi
12. VisualBasic
14. Assembly
15. R
16. Go
17. MATLAB
18. Objective-C
19. Scratch
20. Dart

Python is a friendly high-level programming language that we will be using in this class.

Which Python

- Python currently has two active versions 2.x and 3.x
- we will be using the version 3.x in this class

Warning: Many online resources do not mention the version of Python that they are refering to. The two versions are not always compatible and you need to be very careful to always use 3.x in this class.

How will we write Python?

Python programs are just text

- we will use text editors to edit the programs
- the files end in a `.py`, for example, `hello.py` or `problem1.py`

Note: we cannot use word processors like Microsoft Word or Pages. Why? Because they are not just text editors; they save a lot of information about text formatting, structure of documents, etc.

How will we write Python?

Python programs are just text

- we will use text editors to edit the programs
- the files end in a `.py`, for example, `hello.py` or `problem1.py`

Note: we cannot use word processors like Microsoft Word or Pages. Why? Because they are not just text editors; they save a lot of information about text formatting, structure of documents, etc.

Which text editors will we use to write programs?

- well ...
- we will use one of the two **IDEs** (= Integrated Development Environment) instead of just a plain text editor
- why?
- because an **IDE** comes with many features, in addition to a text editor, that help programmers

Which IDEs will we use?

PythonAnywhere

pythonanywhere

- https://www.pythonanywhere.com/
- cloud-based / web-based, not on your own computer
- 👍 does not require any installation
- 👍 you can use it from any computer (lab, friend, library)
- 👎 requires that you are connected to the Internet
- 👎 tends to be slow and there are limits on usage with a free version
- 👎 does not support graphics that just open up (like a plot) - you need to save those to files

Which IDEs will we use?

PythonAnywhere

pythonanywhere

- https://www.pythonanywhere.com/
- cloud-based / web-based, not on your own computer
- 👍 does not require any installation
- 👍 you can use it from any computer (lab, friend, library)
- 👎 requires that you are connected to the Internet
- 👎 tends to be slow and there are limits on usage with a free version
- 👎 does not support graphics that just open up (like a plot) - you need to save those to files

Conopy

ENTHOUGHT SCIENTIFIC COMPUTING SOLUTIONS

- https://www.enthought.com/products/ca
- on your own computer
- requires installation on your computer
- 👎 if anything happens to your own computer, you cannot access it
- 👍 no need for the Internet connection (except for the installation time)

Which IDEs will we use?

PythonAnywhere

pythonanywhere

- https://www.pythonanywhere.com/
- cloud-based / web-based, not on your own computer
- 👍 does not require any installation
- 👍 you can use it from any computer (lab, friend, library)
- 👎 requires that you are connected to the Internet
- 👎 tends to be slow and there are limits on usage with a free version
- 👎 does not support graphics that just open up (like a plot) - you need to save those to files

Conopy

ENTHOUGHT SCIENTIFIC COMPUTING SOLUTIONS

- https://www.enthought.com/products/ca
- on your own computer
- requires installation on your computer
- 👎 if anything happens to your own computer, you cannot access it
- 👍 no need for the Internet connection (except for the installation time)

We will use both of them to take advantage of all the 👍 and minimize problems resulting from 👎.