CSc 110, Spring 2017

Introduction to Programming I

Lecture 1: Introduction; Basic Python Programs

Adapted from slides by Marty Stepp and Stuart Reges

CSc 110: Introduction to Computer Programming I

webs
IMPORT OS
CUR_PATH = OS.GETCWD()
IGNORE_SET = SET(["__INIT__.PY", "COUNT_SOURCELINES.PY"])
I DIDN'T KNOW HARRY SPOKE PYTHON.
YEAH, HE'S A PARSER-TONGUE.

Course Staff

- Allison Obourn (aeobourn@cs.arizona.edu)
  - B.S. M.S. Computer Science and Engineering - University of Washington
  - Lecturer - University of Washington
- Janalee O'Bagy (jobagy@cs.arizona.edu)
  - B.S. Math, Ph.D. Computer Science – University of Arizona
  - Academia – University of Virginia
  - Industry
    - High Availability Systems Architect (clients such as Apple, Inc.)
    - Software Developer (part of a team that implemented a soft real-time version of Java)
    - Independent Futures Trader (S&P mini-Futures)
- Section Leaders
  - Your primary point of contact
  - Ask them about their experiences in CSc

Computer Science

- CS is about PROCESS – describing how to accomplish tasks
  - Algorithm: a step-by-step procedure for solving a problem
  - Computers are “brainless” machines that execute specific instructions; they have perfect memories
  - Our task is to develop those very specific instructions for the problem at hand
- Computers are a tool
  - Currently the best implementation platform
  - What kinds of problems can they solve?
- Science?
  - More like engineering, art, magic…
  - Hypothesis creation, testing, refinement important

Take this course if you…

- … like solving problems
- … like building things
- … (will) work with data sets or very large data sets
- … are curious about how Facebook, Google, etc work
- … have never written a computer program before
- … are shopping around for a major
    - 110 is a good predictor of who will enjoy and succeed in CSc

Programming

- **program**: A set of instructions to be carried out by a computer
- **program execution**: The act of carrying out the instructions contained in a program.
- **programming language**: A set of rules used to describe computations in a format that is readable by humans.

Programming

- A **programming language specification** consists of two parts
  - **syntax:** specifies the sequences of symbols that are valid programs in the language
  - **semantics:** specifies the meaning of a sequence of symbols

  Example of syntax and semantics from math:
  (3,8)
  a point in a coordinate plane

Some modern languages

- *procedural languages*: programs are a series of commands
  - **Pascal** (1970): designed for education
  - **C** (1972): low-level operating systems and device drivers
- *functional programming*: functions map inputs to outputs
  - **Lisp** (1958) / **Scheme** (1975), **ML** (1973), **Haskell** (1990)
- *object-oriented languages*: programs use interacting "objects"
  - **Smalltalk** (1980): first major object-oriented language
  - **C++** (1985): "object-oriented" improvements to C
    - successful in industry; used to build major OSes such as Windows
  - **Python** (1991):
    - The language taught in this course

Why Python?

- Expressive language

  expresses complex ideas in a simple way

  strong philosophy

  well-designed

- Object-oriented

- Pre-written software

- Widely used

A Python program

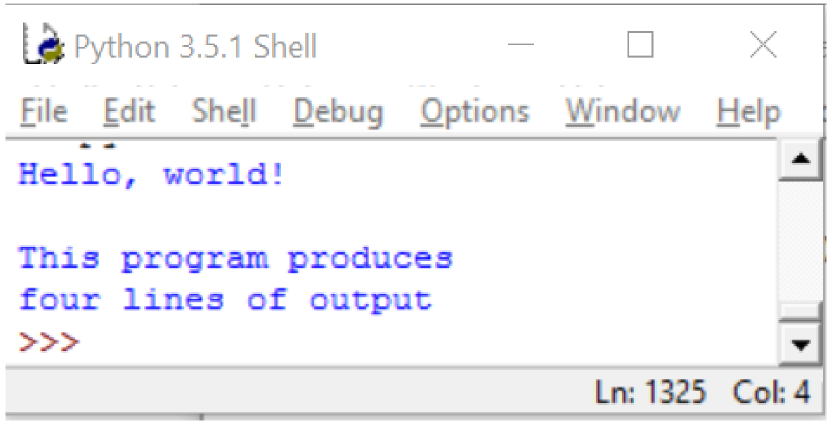
```
print("Hello, world!")
print()
print("This program produces")
print("four lines of output")
```

- Its output:

```
Hello, world!

This program produces
four lines of output
```

- **console**: Text box into which the program's output is printed.



```
Hello, world!

This program produces
four lines of output
>>>
```

`print`

- Used to print a line of output on the console

- Two ways to use `print`:
  - `print("…`*`text`*`…")`

    Prints the given message as output.

  - `print()`

    Prints a blank line of output.

Strings and escape sequences

Strings

- **string**: A sequence of characters
  - Starts and ends with a " quote " character or a ' quote ' character.
    - The quotes do not appear in the output when printed
  - Examples:

```
"hello"
"This is a string.  It's very long!"
'Here is "another" with quotes in'
```

- Syntax Rules:
  - Strings surrounded by " " or ' ' may not span multiple lines

```
"This is not
a legal String."
```

  - Strings surrounded by " " may not contain a " character.

```
"This is not a "legal" String either."
```

  - Strings surrounded by ' ' may not contain a ' character.

```
'This is not a 'legal' String either.`
```

  - Strings surrounded by 3 " may span lines.

```
  """I can span multiple lines
  because I'm surrounded by 3 double quotes"""
```

Problem: What if you want to have both double and single quotes in the output?

- Consider printing the following output:

  She said, “Who’s there?”

- The syntax rules tell us the following is not correct:

```
print("She said, "Who’s there?")
```

  We need a new convention to express this, i.e., more syntax and semantics.

Escape sequences

- **escape sequence**: A sequence of characters used to represent certain special characters in a string.

`\t`	tab character
`\n`	new line character
`\"`	quotation mark character
`\\`	backslash character

- Example:
```
print("\\hello\nhow\tare \"you\"?\\\\")
```
- Output:
```
\hello
how     are "you"?\\
```

Questions

- What is the output of the following `print` statements?

```
print("She said, \"Who’s there?\"")
print("\\\\")
print("'")
print("\"\"\"")
print("C:\nin\the directory")
```

- Write a `print` statement to produce this output:

```
/ \ // \\ /// \\\
```

Answers

- Output of each `print` statement:

```
"She said, Who’s there?"
\\
'
"""
C:
in        he directory
```

- `print` statement to produce the line of output:

```
print("/ \\ // \\\\ /// \\\\\\")
```

Questions

- What `print` statements will generate this output?

```
This quote is from
Irish poet Oscar Wilde:

"Music makes one feel so romantic
- at least it always gets on one's nerves –
which is the same thing nowadays."
```

- What `print` statements will generate this output?

```
A "quoted" String is
'much' better if you learn
the rules of "escape sequences."

Also, "" represents an empty String.
Don't forget: use \" instead of " !
'' is not the same as "
```

Answers

- `print` statements to generate the output:

```
print("This quote is from")
print("Irish poet Oscar Wilde:”)
print()
print("\"Music makes one feel so romantic")
print("- at least it always gets on one's nerves -")
print("which is the same thing nowadays.\"")
```

- `print` statements to generate the output:

```
print("A \"quoted\" String is")
print("'much' better if you learn")
print("the rules of \"escape sequences.\"")
print()
print("Also, \"\" represents an empty String.")
print("Don't forget: use \\\" instead of \" !")
print("'' is not the same as \"")
```