CSc 110, Autumn 2017

Lecture 1: Introduction; Basic Python Programs

```python
import os
CUR_PATH = os.getcwd()
IGNORE_SET = set(['__init__.py',
                  'count_source_lines.py'])
```

I didn't know Harry spoke Python.

Yeah, he's a parser-tongue.
Course Staff

• Allison Obourn (aeobourn@cs.arizona.edu)

• 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
  • "efficiently implementing automated abstractions" (Philip Guo)

• Computers are a tool
  • Currently the best implementation platform
  • What kinds of problems can they solve?
  • How can they be made faster, cheaper, more efficient...?

• Science?
  • More like engineering, art, magic...
  • Hypothesis creation, testing, refinement important

• CS is still a young field finding itself
Why should you take Computer Science?

• ... like solving tricky problems

• ... like building things

• ... (will) work with large data sets

• ... are curious about how Facebook, Google, etc work

• ... are shopping around for a major
  • 110 is a good predictor of who will enjoy and succeed in CSc
Are you in the right class?

CSc 101
Intro to Computer Science

CSc 110
Intro to Computer Programming I

CSc 120
Intro to Computer Programming I

You are here
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 systematic set of rules used to describe computations in a format that is editable by humans.
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?

- Relatively simple
- Pre-written software
- Widely used
A Python program

```python
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.
print

• A statement that prints 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

- **string**: A sequence of characters to be printed.
  - Starts and ends with a " quote " character or a ' quote ' character.
    - The quotes do not appear in the output.
  - Examples:
    "hello"
    "This is a string. It's very long!"
    'Here is "another" with quotes in"
    """"I can span multiple lines
    because I'm surrounded by 3 quotes"

- **Restrictions**:
  - 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.'
Escape sequences

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

\t  tab character
\n  new line character
\"  quotation mark character
\'  quotation mark character
\\  backslash character

• **Example:**
  ```python
  print("\hello\nhow\tare \"you\"?\\\n")
  ```

• **Output:**
  ```python
  \hello
  how  are "you"?\\
  ```
Questions

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

```python
print("\ta\tb\tc")
print("\\\\")
print("\\")
print("\\\\\\")
print("C:\nin\nthe downward spiral")
```

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

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

Answers

• **Output of each print statement:**

  ```
  a   b   c
  //
  \
  ""
  C: in he downward spiral
  ```

• **print statement to produce the line of output:**

  ```python
  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."

  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:

```python
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:

```python
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 \"")
```