CS 115, Autumn 2021

Lecture 1: Introduction; Python

Thank you to Marty Stepp, Stuart Reges and Larry Snyder for parts of these slides

Important Information

- Course website:
  - http://allisonobourn.com/edmonds/115
- Instructor email:
  - allison.obourn@edmonds.edu

Course goals

- be able to write a useful computer program
- a better understanding of what computer science is
- a better understanding of what is hard and what is easy for a computer to do

How to be successful

- Read the **syllabus**!
- **Attend** classes and labs
- Start assignments **early**
  - Taking breaks will help you figure out the answer, if you start too late you won't have time to take breaks
- **Be persistent**! Many things will seem confusing at first, but you CAN figure them out stay with it
- **Ask questions** when you get stuck. I want to help you but I won't know that you are stuck unless you tell me.

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

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

- **Pascal** (1970): designed for education
- **C** (1972): low-level operating systems and device drivers
- **Lisp** (1958) / **Scheme** (1975), **ML** (1973), **Haskell** (1990)
- **Smalltalk** (1980): first major object-oriented language
- **C++** (1985): "object-oriented" improvements to C
- **Python** (1991): The language taught in this course
- **Java** (1995): designed for embedded systems, web apps/servers
- **Visual Basic .NET** (2001)
  - Based on Visual Basic (1991) which was based on Basic (1963)

Why Python?

- Relatively simple syntax – it looks a lot like English
- Pre-written software
- Widely used
  - Currently the second most commonly used language
  - Used by
    - Google
    - Netflix
    - PayPal
    - Goldman Sachs
    - Uber
    - ... and many, many more

A Python program

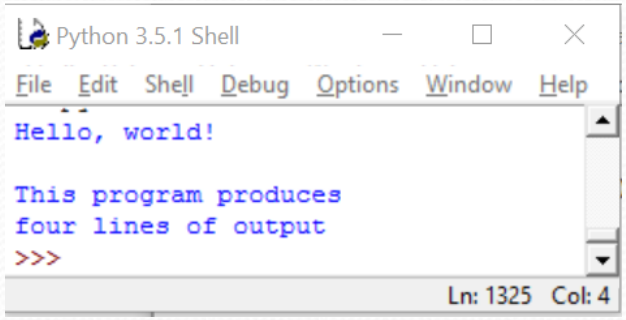
```
print("Hello, world!")
print()
print("This program produces")
print("four lines of output")
```

- Its output:

```
Hello, world!

This program produces
four lines of output
```

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



`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:

```
print("\\hello\nhow\tare \"you\"?\\\\")
```

- Output:

```
\hello
how     are "you"?\\
```

Questions

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

```
print("\ta\tb\tc")
print("\\\\")
print("'")
print("\"\"\"")
print("C:\nin\the 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:

```
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 \"")
```