

CS 1340:Fall 2020:Lecture 02

Intro to Python for CS and Data Science

Mark Fontenot, PhD

Southern Methodist University

Reminders

- Slack
- Zybooks (hopefully you did the assignment due before class today?)
- Anaconda?

Getting Started

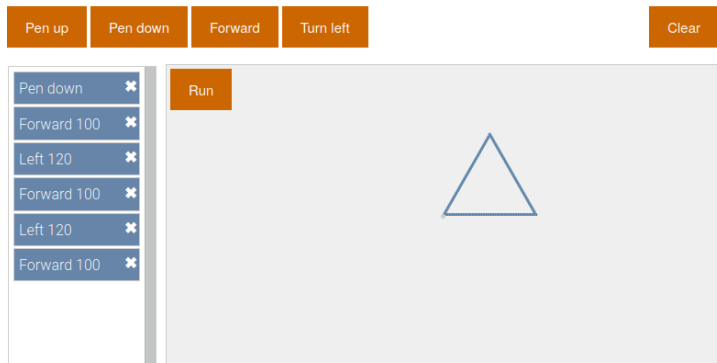
All computer programs have ...

1. Input -

2. Processing -

3. Output -

Drawing with the Turtle



1. What's the input?
2. What's the processing?
3. What's the output?

Challenge on your own time ..

Can you draw 'SMU' (block letters of course) with the Turtle?

Humans vs. Computers

- Humans don't understand long strings of 1's and 0's
- Computers don't understand

```
print('Hello 1340')
```

So what do we do???

General *Algo* for Programming

1. You write source code in a code editor or IDE (Integrated Development Environment)
2. Save it with a file extension of `.py`. Example: `project01.py`
3. Use the Python interpreter to execute the source code.
 - `python project01.py`
 - OR could be `python3 project01.py`
4. Back to Step 1 to add more code.

Converting Python Code to Machine Code

- Python is an **interpreted** language
 - as your program is running, the Python Interpreter / Runtime is converting source code to machine code one line by line.
 - The alternative is a **compiled** language which converts all the source code to machine code before you run your program.

```
def load_data(filename):
    rows = []
    max_skill_num = 0
    max_num_problems = 0
    with open(filename, "r") as csvfile:
        reader = csv.reader(csvfile, delimiter=',')
        for row in reader:
            rows.append(row)
    index = 0
    print("the number of rows is " + str(len(rows)))
    tuple_rows = []
    while(index < len(rows)-1):
        problems_num = int(rows[index][0])
        tmp_max_skill = max(map(int, rows[index+1]))
        if(tmp_max_skill > max_skill_num):
            max_skill_num = tmp_max_skill
            if(problems_num == 2):
                index += 3
```

Python Code



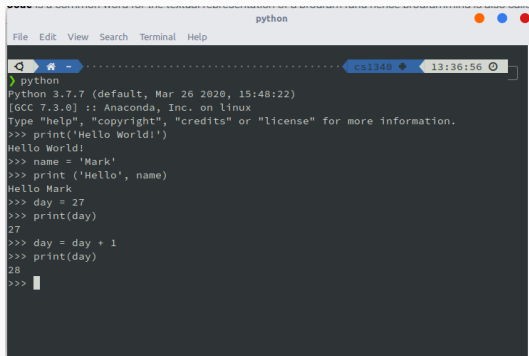
Interpreter

```
00111111000000111111001001100011011100101
11101111100001101011111101100011110111
1101111100001110001101010010111010011111
110001110111011111110111111111010001111
111110011011101110001111011101011101000
11001101111101100010010000001011011101
111110001111100011111111111110011110011
011111011001110111101111110101111101111
111111110001111111001010010100011111011
000111111111011000101000011110010000000
10111100111111101010111111000101110111
011100001111011011111001111110011111111
101110110100001001100110001101110000110
110110011000101011101101000111110100011
111111110011000111100111101010000110100
11101000010110011111110000011110111111
111001001101011111111110011111111111111
000000000111111100001100111101100100011
01000100111111001111111111111001111000
01000111001001111111000010111011100110
```

Binary Code

Interactive Python

- `python somescript.py` - runs the code inside `somescript.py`
- `python` - Starts the **interactive interpreter**
 - each line of code is interpreted right after you type it.



```
python
File Edit View Search Terminal Help
cs1348 13:36:56
>>> python
Python 3.7.7 (default, Mar 26 2020, 15:48:22)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World!')
Hello World!
>>> name = 'Mark'
>>> print('Hello', name)
Hello Mark
>>> day = 27
>>> print(day)
27
>>> day = day + 1
>>> print(day)
28
>>> 
```

Elements of Python Code

- **statement** -
- **variable** -
- **expression** -
- **assignment** -

Some Python Code

```
def advance_cars():  
    """Calculate new positions of the cars"""  
    global car1_speed, car1_location  
    global car2_speed, car2_location  
    car1_speed += car1_acceleration  
    car1_speed = car1_top_speed if car1_speed > car1_top_speed else car1_speed  
    car1_location += car1_speed  
  
    car2_speed += car2_acceleration  
    car2_speed = car2_top_speed if car2_speed > car2_top_speed else car2_speed  
    car2_location += car2_speed
```

```
print(...)
```

- Notice:
 - printed in mono-spaced font
 - ... means other stuff will be put there
 - () indicate a method or function call

```
print('Hello')
```

```
print('World')
```

```
print('Hello World')
```

'Hello', 'World' are called **string literals**.

```
print(...)
```

```
name = 'Mark'  
print(name)
```

- You can print **string literals** OR values contained in variables.
- What is the variable in this example?

```
print('Hello', end='')  
print('World', end='')
```

- What will this print?

Can you do it?

Use print statements to draw a diamond shape.

Use print statements to draw a heart shape.