

## Objectives

- Review algorithms
- Programming in Python
  - Data types
  - Expressions
  - Variables
  - Arithmetic

Jan 13, 2017

Sprenkle - CSCI111

1

## Review

- What is an algorithm?
- What did we learn from the PB&J demonstration?

Jan 13, 2017

Sprenkle - CSCI111

2

## Review: Parts of an Algorithm

- Input, Output
- Primitive operations
  - What data you have, what you can do to the data
- Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

An overview for the semester!

Jan 13, 2017

Sprenkle - CSCI111

3

## Review: Computational Problem Solving 101

- **Computational Problem:**  
A problem that can be solved by logic
- To solve the problem:
  - Create a **model** of the problem
  - Design an **algorithm** for solving the problem using the model
  - Write a **program** that *implements* the algorithm

Jan 13, 2017

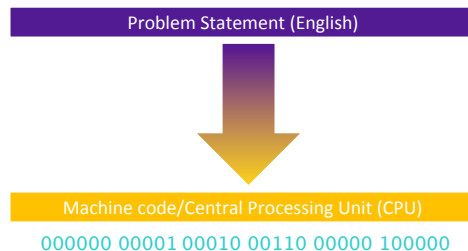
Sprenkle - CSCI111

4

## Why Do We Need Programming Languages?

- Computers can't understand English
  - Too ambiguous
- Humans can't easily write machine code

Live Jazz!



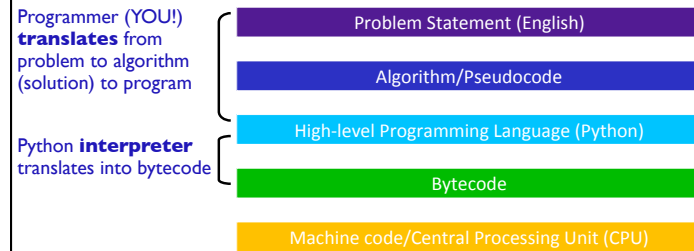
Jan 13, 2017

Sprenkle - CSCI111

5

## Why Do We Need Programming Languages?

- Computers can't understand English
  - Too ambiguous
- Humans can't easily write machine code



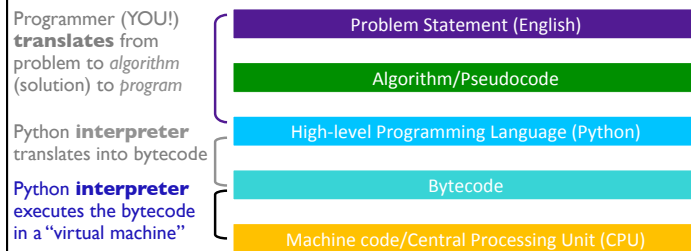
Jan 13, 2017

Sprenkle - CSCI111

6

## Why Do We Need Programming Languages?

- Computers can't understand English
  - Too ambiguous
- Humans can't easily write machine code



Jan 13, 2017

Sprenkle - CSCI111

7

## Programming Languages

- Programming language:
  - Specific rules for what is and isn't allowed
  - Must be exact
  - Computer carries out commands as they are given
- **Syntax:** the symbols given
- **Semantics:** what it means
- Example: III \* IV means  $3 \times 4$  which evaluates to 12
- Programming languages are **unambiguous**

Jan 13, 2017

Sprenkle - CSCI111

8

## Another Syntax and Semantics Example

What is the syntax? What does the syntax mean (semantics)?



Jan 13, 2017

Sprenkle - CSCI111

9

## Python Is ...

- A programming language
  - 5<sup>th</sup> most popular programming language  
<http://www.tiobe.com/tiobe-index/>
- An interpreter (which is a program) that executes Python code

Jan 13, 2017

Sprenkle - CSCI111

10

## Python

- A common *interpreted* programming language
  - Runs on many operating systems
- First released by Guido van Rossum in 1991
- Named after *Monty Python's Flying Circus*
- Minimalist syntax, emphasizes readability
- Flexible, fast, useful language
- Used by scientists, engineers, systems programmers

Jan 13, 2017

Sprenkle - CSCI111

11

## Python Interpreter:

Interactive Mode Run by typing "python3" in terminal

```
sprenkle@perلمان:~/courses/private$ python3
Python 3.4.1 (default, Sep 24 2015, 20:41:10)
[GCC 4.9.2 20150212 (Red Hat 4.9.2-6)] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3
3
>>> 4+5
9
>>> 1-7
-6
>>> "word"
'word'
>>> word
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'word' is not defined
>>> print 4+5
File "<stdin>", line 1
  print 4+5
      ^
SyntaxError: invalid syntax
>>> print(4+5)
9
><<<
```

Python displays the result

Type in the expression

Error Message: We'll talk more later about why this is an error

print: Special function to display output

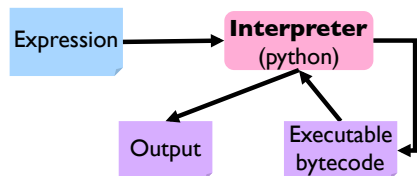
Jan 13, 2017

Sprenkle - CSCI111

12

## Python Interpreter

1. Validates Python programming language expression(s)
  - Enforces Python **syntax**
  - Reports **syntax** errors
2. Executes expression(s)
  - Runtime errors (e.g., divide by 0)
  - **Semantic** errors (not what you *meant*)



Jan 13, 2017

Sprenkle - CSCI111

13

## Two Modes to Execute Python Code

- **Interactive:** using the interpreter
  - Try out Python expressions
- **Batch:** execute *scripts* (i.e., files containing Python code)
  - What we'll usually write

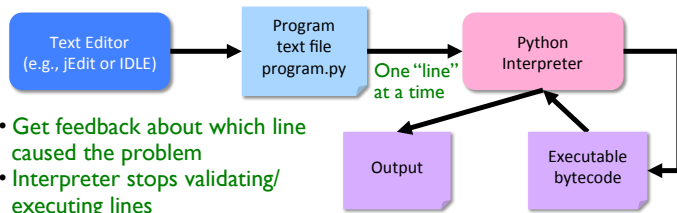
Jan 13, 2017

Sprenkle - CSCI111

14

## Batch Mode

1. Programmer types a **program/script** into a **text editor** (jEdit or IDLE).
2. An **interpreter** turns each expression into **bytecode** and then executes each expression



Jan 13, 2017

Sprenkle - CSCI111

15

## Parts of an Algorithm

- ➔ **Input, Output**
  - Primitive operations
    - What data you have, what you can do to the data
  - Naming
    - Identify things we're using
  - Sequence of operations
  - Conditionals
    - Handle special cases
  - Repetition/Loops
  - Subroutines
    - Call, reuse similar techniques

Jan 13, 2017

Sprenkle - CSCI111

16

## Printing Output

- **print** is a special command or a *function*
  - Displays the result of expression(s) to the terminal
  - Automatically adds a '\n' (carriage return) after it's printed
    - Relevant when have multiple print statements

● `print("Hello, class")`

string literal

**Syntax:** a set of double quotes  
**Semantics:** represents text

Jan 13, 2017

Sprenkle - CSCI111

17

## Printing Output

- **print** is a special command
  - Displays the result of expression(s) to the terminal
- `print("Hello, class")`

string literal

**print** automatically adds a '\n' (carriage return) after it's printed

● `print("Your answer is", 4*4)`

**Syntax:** comma  
**Semantics:** print multiple "things" in one line

Jan 13, 2017

Sprenkle - CSCI111

18

## Parts of an Algorithm

- Input, Output
- ➔ Primitive operations
  - What data you have, what you can do to the data
- Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

Jan 13, 2017

Sprenkle - CSCI111

19

## Primitive Data Types

- Primitive data types represent *data*
  - In PB&J example, our data had **types** slice of bread, PB jar, jelly jar, etc.
- Python provides some basic or *primitive data types*
- Broadly, the categories of primitive types are
  - Numeric
  - Boolean
  - Strings

Jan 13, 2017

Sprenkle - CSCI111

20

## Numeric Primitive Types

| Python Data Type     | Description                                      | Examples                         |
|----------------------|--|----------------------------------|
| <code>int</code>     | Plain integers (32-bit precision)                | -214, -2, 0, 2, 100              |
| <code>float</code>   | Real numbers                                     | .001, -1.234, 1000.1, 0.00, 2.45 |
| <code>complex</code> | Imaginary numbers (have real and imaginary part) | $1j * 1j \rightarrow (-1+0j)$    |

Jan 13, 2017

Sprenkle - CSCI111

21

## How big (or small or precise) can we get?

- Computer cannot represent all values
- Problem: Computer has a **finite** capacity
  - The computer only has so much memory that it can devote to one value.
  - Eventually, reach a cutoff
    - Limits size of value
    - Limits precision of value

PI has more decimals, but we're out of space!

0 0 0 0 0 3 . 1 4 1 5 9 2 6 5

Example: in Python interpreter, `.1 + .1 + .1` yields `0.30000000000000004`.  
\* In reality, computers represent data in binary.

## Strings: `str`

- Indicated by double quotes "" or single quotes ''
- Treat what is in the "" or '' literally
  - Known as **string literals**
- Examples:
  - "Hello, world!"
  - 'c'
  - "That is Buddy's dog."

Single quote must be inside double quotes\*  
\* Exception later

Jan 13, 2017

Sprenkle - CSCI111

23

## Booleans: `bool`

- 2 values
  - True
  - False
- More on these later...

Jan 13, 2017

Sprenkle - CSCI111

24

## What is the value's type?

| Value      | Type |
|------------|------|
| 52         |      |
| -0.01      |      |
| 4+6j       |      |
| "3.7"      |      |
| 4047583648 |      |
| True       |      |
| 'false'    |      |

Jan 13, 2017

Sprenkle - CSCI111

25

## What is the value's type?

| Value      | Type    |
|------------|---------|
| 52         | int     |
| -0.01      | float   |
| 4+6j       | complex |
| "3.7"      | str     |
| 4047583648 | int     |
| True       | boolean |
| 'false'    | str     |

Jan 13, 2017

Sprenkle - CSCI111

26

## Parts of an Algorithm

- Input, Output
- Primitive operations
  - What data you have, what you can do to the data
- ➔ Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

Jan 13, 2017

Sprenkle - CSCI111

27

## Introduction to Variables

- Variables save data/information
  - Example: first slice of bread or knife A
  - Type of data the variable holds can be any of primitive data types as well as other data types we'll learn about later
- Variables have names, called *identifiers*

Jan 13, 2017

Sprenkle - CSCI111

28

## Variable Names/Identifiers

- A variable name (identifier) can be any one word that:
  - Consists of letters, numbers, or \_
  - Does *not* start with a number
  - Is not a Python reserved word
    - Examples: **for while def**
- Python is case-sensitive:
  - **change** isn't the same as **Change**

Jan 13, 2017

Sprenkle - CSCI111

29

## Variable Name Conventions

- **Variables** start with lowercase letter
- Convention: **Constants** (values that won't change) are all capitals
  - More on Monday
- Example: Variable for the current year
  - **currentYear**
  - **current\_year**
  - **CURRENT\_YEAR**
  - ~~currentyear~~ Harder to read
  - ~~current year~~ No spaces allowed

Naming doesn't matter to computer.  
Matters to humans

Jan 13, 2017

Sprenkle - CSCI111

30

## Importance of Variable Naming

- Helps you *remember* what the variable represents
- Easier for others to *understand* your program
- Examples:

| Info Represented              | Good Variable Name    |
|-------------------------------|-----------------------|
| A person's first name         | firstName, first_name |
| Radius of a circle            | radius                |
| If someone is employed or not | isEmployed            |

Jan 13, 2017

Sprenkle - CSCI111

31

## Review: Computational Problem Solving

- **Computational Problem:**  
A problem that can be solved by logic
- To solve the problem:
  - Create a **model** of the problem
  - Design an **algorithm** for solving the problem using the model
  - Write a **program** that *implements* the algorithm

Jan 13, 2017

Sprenkle - CSCI111

32



## Modeling Information

- How would you **model** this information?
- What data type best represents the info?

| Info Represented   | Data Type | Variable Name |
|--------------------|-----------|---------------|
| A person's salary  |           |               |
| Sales tax          |           |               |
| If item is taxable |           |               |
| Course name        |           |               |
| Graduation Year    |           |               |

Jan 13, 2017

Sprenkle - CSCI111

33

## Modeling Information

- How would you **model** this information?
- What data type best represents the info?

| Info Represented   | Data Type    | Variable Name |
|--------------------|--------------|---------------|
| A person's salary  | int or float | salary        |
| Sales tax          | float        | salesTax      |
| If item is taxable | boolean      | isTaxable     |
| Course name        | str          | course_name   |
| Graduation Year    | int          | gradYear      |

Jan 13, 2017

Sprenkle Variable names are just suggestions,  
Many other possible variable names

## Assignment Statements

- Variables can be given any value using =
  - **Syntax:** <variable> = <expression>
  - **Semantics:** <variable> is set to value of <expression>
- After a variable is set to a value, the variable is said to be **initialized**
- Examples:

```
month = 1
impt_num = 4.5
monthName = 'January'
```

These are **not** equations!  
Read "=" as "is set to"

Jan 13, 2017

Sprenkle - CSCI111

35

## Assignment Statements

```
x = 5
y = x
```



- Statements execute in order, from top to bottom
- Value of **x** does not change because of second assignment statement

Jan 13, 2017

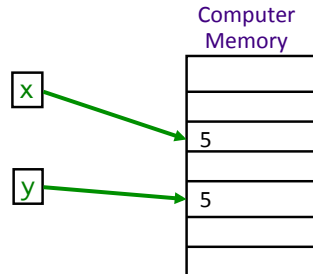
Sprenkle - CSCI111

36

## Assignment Statements

```
x = 5  
y = x
```

Does a “lookup”  
in memory to find  
value of X



- Statements execute in order, from top to bottom
- Value of **x** does not change because of second assignment statement

Jan 13, 2017

Sprenkle - CSCI111

37

## What are the values?

- After executing the following statements, what are the values of each variable?

```
➤ a = 5  
➤ y = a + -1 * a  
➤ z = a + y / 2  
➤ a = a + 3  
➤ y = (7+x)*z  
➤ x = z*2
```

### Runtime error:

- x doesn't have a value yet!
- We say “x was not initialized”
- Can't use a variable on RHS until seen on LHS!\*

Jan 13, 2017

Sprenkle - CSCI111

38

## Groups for Broader Issues

Ashley  
Collin  
George  
Molly  
Robert

Buddy  
Lexi  
Mira  
Win  
Zander

Alex  
Anna Kate  
Mike  
Sarah  
Victor

Austin  
Charlotte  
Jae  
Tony

Burke  
John  
Josette  
Leslie

Jan 13, 2017

Sprenkle - CSCI111

39

## Broader CS Issues

- Good summaries!
  - Good English, complete sentences
- Good, thoughtful questions
- Mechanics details
  - Follow instructions on “CS Issues” about what summary should contain
  - Should be able to edit your own posts
  - Characters from Word
    - Click button “Paste from Word”
  - Don't attach Word documents

Jan 13, 2017

Sprenkle - CSCI111

40

## AI Everywhere

- What is AI (which is based on algorithms) useful for?
  - What *aren't* algorithms useful for?
- What would be some useful algorithms, specific to W&L students?
  - What are problems that are difficult—but useful—to solve?
- “Stephen Hawking warned that artificial intelligence could be the end of mankind. The ability to learn and evolve is an incredibly powerful ability, especially for robots which do not feel empathy or other human emotions. While the benefits of AI are undeniable, do the risks outweigh the benefits? Is there a line we have to draw in the development of AI?”

Jan 13, 2017

Sprenkle - CSCI111

41

## Algorithmic Transparency and Accountability

- Potential algorithmic bias:
  - **Job hunting web sites:** Do these sites send more listings of high paying jobs to men than to women
  - **Credit reporting bureaus:** Does the data set that algorithms weigh in determining credit scores contain prejudicial information?
  - **Social media sites:** What factors go into determining the news items that are served up to users?
  - **The criminal justice system:** Are computer generated reports that influence sentencing and parole decisions biased against African Americans?

Jan 13, 2017

Sprenkle - CSCI111

42

## Extra Credit Opportunities

- Read an article that relates to CS
- Summarize it on the forum under “Extra Credit”
  - 5 pts extra credit on lab grade

Jan 13, 2017

Sprenkle - CSCI111

43

## Looking Ahead

- Read Chapter 1
- Read Chapter 2, skipping section 8

Jan 13, 2017

Sprenkle - CSCI111

44