Week 1: Basics of a Program

csci_128

THE TWO STATES OF EVERY PROGRAMMER
I AM A GOD.
I HAVE NO IDEA WHAT I'M DOING.
VIA 9GAG.COM

Topics for the Week

- Read about: General Introduction (ch 1), Variables (ch 2)
- Lecture about: Program basics, good coding style, code reading
- Apply new knowledge via studios and worksheets!

Week 1 Learning Goals

- Correctly recognize the major components of any program: variables, expressions, and statements
- Implement and run a program that utilizes common operators to take in simple text input and produces simple text output
- Recognize and produce well-formatted and well-styled Python code
- Write descriptive variable names that aid in code readability

Announcements

Upcoming Assignments

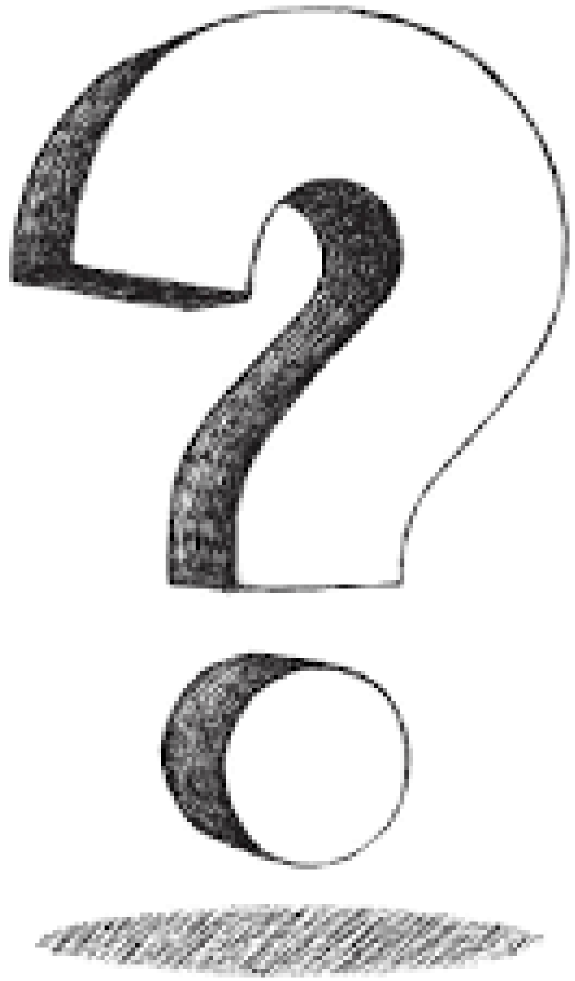
- ✓ **Worksheet 1**
  Due **Today, August 21** in-class
- ✓ **Intro Survey and Syllabus Quiz**
  Due **Friday, August 23** by **11:59pm**

Also be sure your Python Environment is set up for Friday!

From the reading you learned about...

- Common programming vocabulary
- 3 kinds of errors
  - Syntactic, runtime, and semantic
- Variables, statements, and expressions
- Data types
  - int, float, string, etc.
- Operators
  - `+, -, *, /, %, //`
- Functions
  - `print(), input(), int()`, etc.

Questions on the reading?



Source: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.facebook.com%2FTeam-Question-Mark-102976737809512%2F&psig=AOvVaw0eabUCOH9zbelJiflciCrg&ust=1676352812623000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCIiep4Hjkf0CFQAAAAAdAAAAABAK

What is a program?

Definition from the textbook (1.5):

"A program is a sequence of instructions that specifies how to perform a computation."

You can also think of a program as a series of "statements" that tell the computer what to do

Statements

Statement: piece of code that carries out a task or an action. Usually made up of expressions.

- *Ex. value = 5*
- *Ex. print('ABC')*
- *Ex. for value in lst:*
- *Ex. if value == 5:*

A statement is a complete, logical "thought" in a program. Usually a single line of code.

Expressions

Expression: *a value, or anything that executes and ends up being a value.*

- *Ex. 5*
- *Ex. 5 + 3*
- *Ex. Value + 6*
- *Ex. "Carrot" + "Cake"*
- *Ex. len("Carrot Cake")*

Expressions vs. Statements

There is a lot of overlap between expressions and statements, so what's the difference?

- **You can make all the expressions you want, but without any statements your program isn't doing anything.**
- Most statements are made up of many nested/combined expressions
- Expressions allow us to insert values into statements, but they don't do anything on their own
- Statements don't always do something immediately

Variables

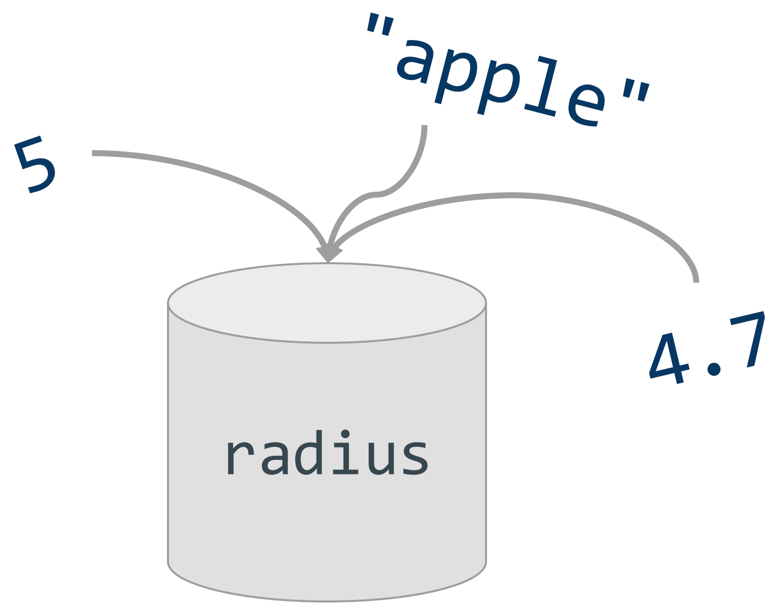
<u>Variable</u>: *a data item that may take on more than one value during the execution of a program*

Ex. x = 5

Ex. name = "Kelly"

Ex. value = 0.66

Ex. lst = [1, 2, 3, 4, 5]



A variable is a name you give to a place you can put information.

Variables

Each Python variable (or object) has three defining properties:

Value: A value such as “CSCI101", “13”, or 191.

Type: The type of the variable, such as integer or string.

Identity: A unique identifier for the variable (or object). Typically where the variable is stored in memory

Ex. X = 5

Value: 5

Type: integer

Identity: Depends...

Types

Types let us predict how a value will behave, by limiting it

Our first types: **int, float, string**

Check your understanding:

- What is a string? What values can go into a string?
- What is the difference between an int and float?

Operators

We are all familiar with mathematical operators for addition, subtraction, division, etc. Python allows us to use these, plus more, to operate on values and variables in different ways.

x + y	Addition
x – y	Subtraction
x * y	Multiplication
x ** y	Exponentiation
x / y	Division
x // y	Integer division (Round down)
x % y	Modulus (remainder)

Functions

Chunks of code stored elsewhere (we'll come back to this).

We can make that code run by "calling" it with "function_name()".

Some functions need to be given data to run, these values are called **arguments**.

E.g. `print(“test”)`, `“test”` is the string argument given to the print function.

Every non-trivial function can:

- Cause something to happen (some immediate change) **and/or**
- "Return" a value (the function call acts as an expression)

Example Functions

- **print()** is the first function we saw – it outputs a variable to the terminal.
- **input()** takes input from the terminal. Optionally, provide a string inside of the parenthesis to print a prompt.
- **Type conversion**: int(), str(), float()

Which functions cause something to happen?

Which functions take arguments?

Which function calls are also expressions? What values do they turn into?

Code Reading

- How do we define readable code?
- Why do we want to write readable code?
- What makes our code better? What makes our code worse?

Code Reading

- How do we define readable code?
  - Readable code is easy to understand by someone who didn't write it.
- Why do we want to write readable code?
  - ..so that others can understand it (**readability**)
  - ..so that we can make changes easily (**maintainability**)
- What makes our code better? What makes our code worse?
  - ***Better***: *Following the* ***Python Style Guide*** *(next slide)*
  - ***Worse***: *Not thinking about how others might view our code*

The Python Style Guide
https://peps.python.org/pep-0008/

- Indentation
- Maximum Line Length
- Imports
- Variable Names
- Whitespace

Our First Worksheet

- Blank copy is available on Canvas
- Submit proof you did the worksheet by the end of the day
  - OK to work in groups, but everyone submits their own copy
  - Submitted on Gradescope (linked from Canvas)
  - Submission can be any type (typed pdf, hand-written, etc.)

Our First Worksheet

- Graded mostly on *attempt*
    - Make a real attempt at all problems for full credit
- One problem selected at random graded on correctness
    - To encourage you to try all problems earnestly

- We will usually do worksheets on Monday
    - Future sheets will be available on Canvas before class
- We can discuss answers at the end of class
    - Or they can be seen on Canvas the day after

Our First Worksheet

Emphasis on:

- Reading code
- Developing readable code
- Python conventions

Worksheet 1 Answers

What problems do you have questions on?

Upcoming Assignments

- ✓ **Worksheet 1**

  Due **Today, August 21** by **11:59pm**
- ✓ **Intro Survey and Syllabus Quiz**

  Due **Friday, August 23** by **11:59pm**

Be sure your Python Environment is set up for Friday!

HAVE A GREAT DAY!