CS 1112: Introduction To Programming

Python Basics; Hello World; Printing; Comments (Confirming Python & PyCharm!)

Dr. Nada Basit // basit[*at*]Virginia[*dot*]edu

Spring 2024

Friendly Reminders

- Your ***safety*** and ***comfort*** is important!
  - If you choose to wear a mask you are welcome to do so
  - *We will interpret wearing a mask as being considerate and caring of others in the classroom (not that you are sick), and realize that some may choose to mask to remain distanced*
- Be an ***active*** participant in your learning!
  You're welcome and ***encouraged*** to ask questions during class!
- If you feel ***unwell***, or think you are, please stay home
  - *We will work with you!*
  - Get some rest ☺
  - View the recorded lectures – *please allow 24-48 hours to post*
  - *Contact us!*

Place-out? Waitlist?

- Place-out Test for CS 11xx
  - Think you are already familiar with the fundamentals of programming? Consider taking the place-out test for CS 11xx!
  - URL: https://sherlock.cs.virginia.edu/?c=exam&e=339
  - Test will be open through Tuesday night (11:59pm on 1/23)
- Waitlist
  - If you need CS 1112 – feel free to stay on the waitlist
  - **Don't forget to sign-up!** *(So that I know you are active and attending!)*
  - Considering ***switching*** to CS 1110 or CS 1111? Please let me know.
- Note: being on a waitlist **doesn't guarantee** enrollment into a course
  - Your instructor cannot force your enrollment into a section that is already full
  - In rare circumstances, a dean or the registrar may be able to help

Syllabus Quiz

Don't forget to take the Syllabuzz Quizz!

- This quiz is *Mandatory!*
- This quiz is located on Canvas (see tab on left-hand side).
- Take this quiz ***individually***. Absolutely no collaboration permitted.
- Must get **100%** to stay in the course! *May take it as many times as needed.*
  - Review the detailed Syllabus
  - This quiz is ***open-book***
  - See score out of 12 points on Canvas Grades to confirm you've completed the quiz
- ***Where?:*** *"Assignments" tab > "Syllabus Quiz (Required)"*; **or** *"Quizzes" tab*
- ***Deadline:*** **January 31, 2024 @ 11:00pm**. *Take it early!*
  - *Most students should aim to finish the Syllabus Quiz by January 26, 2024*

One-Time Office Hours This Week

- **Extended Help Session for Python and PyCharm Installation**
  - Date: Monday, January 22, 2024
  - Time: **3:30 – 5:00pm**
    Come and go as you please
  - Location: Rice Hall 540

CS 1112 Pledge!

Taking this pledge is mandatory for our class to have a community of trust

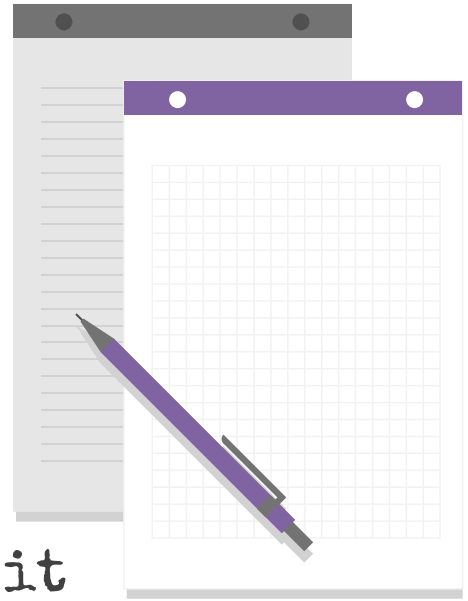
★ Google form: https://forms.gle/eksrS9pcinGLYNKu7 (try this first)

★ Microsoft form: https://forms.office.com/r/wVdPysWRN6

[Please submit your pledge at only ONE of the above links!]

I gratefully acknowledge Prof. J. Cohoon for introducing the course pledge in this course!

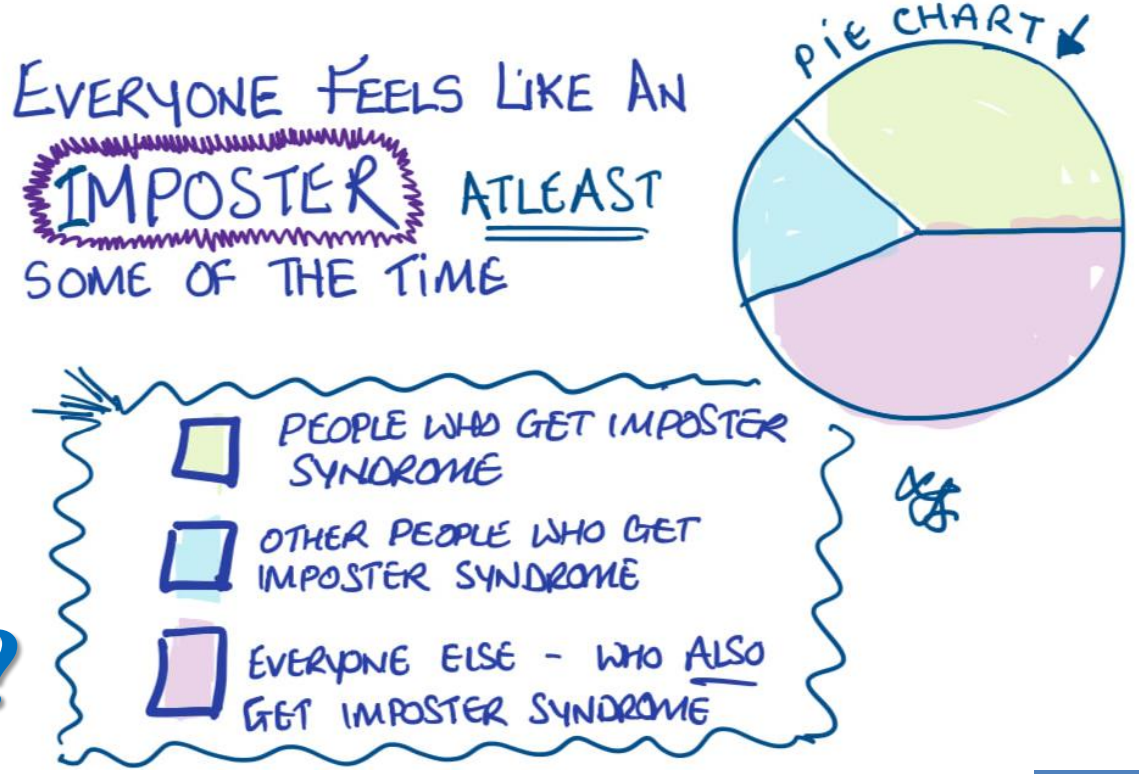
A Little Bit of Housekeeping…



Create a folder somewhere where you can easily find it (Desktop, Documents, …) and name it "CS 1112"

Put everything related to this course inside this folder – you will have many artifacts by the end of the semester!

- Ever Think You Are An Imposter?
- Wonder whether UVA made the right decision?



Well, we think UVA made the right decision! ☺ Watch this video this weekend to be sure: Click HERE!! The video is about 20 minutes long. Watching it will be a great investment in your education, should help your test performance, and improve your job interviewing prowess!

I gratefully acknowledge Prof. J. Cohoon for discovering & sharing this wonderful resource!

Quick & Fun Survey Questions

Get to know your peers! ☺

Cold weather or Hot weather?

Were You Successful In Installing Python and PyCharm?

In-class "lab" activity

Follow the installation guide corresponding to your computer's operating system (Windows or Mac).

On Canvas: Files > Installation Documents > MacOS_Installation.zip

Files > Installation Documents > WindowsOS_Installation.zip

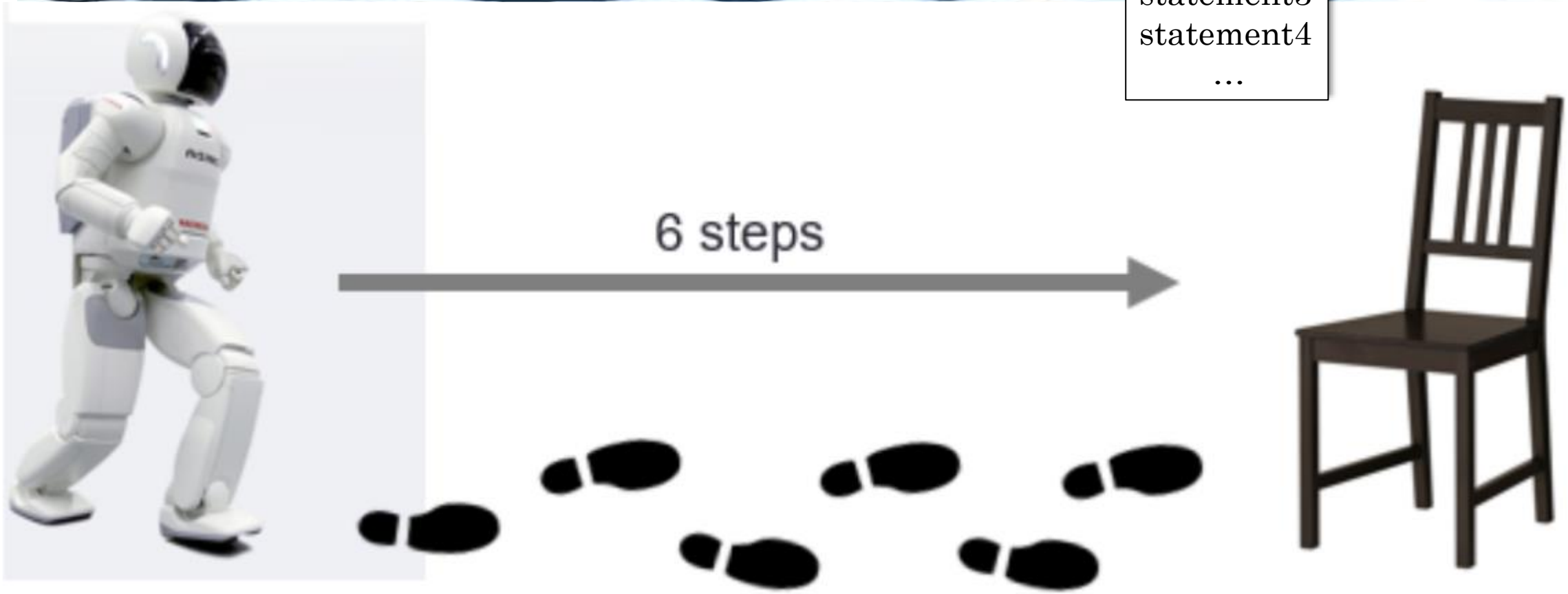
Building Blocks of Programs

Building Blocks of Programs

- **Sequence**
  - We start with the instruction written at the top
  - We go in order, one instruction at a time
  - Each line is "one" thing to do
- **Repetition** = *repeat something*
  - Repeat a fixed number of times (e.g., repeat 3 times)
  - Repeat **UNTIL** something happens (e.g., repeat *until* input is valid)
- **Conditions/Decisions** = *maybe do something*
  - Check something first; i.e., If there is a file present, read it
- **Named actions**
  - Grouping many lower-level actions in to **one** higher level **name**
  - We think of a named action in 2 different ways
    - Definition of the name action
    - Use of the named action
  - i.e., "`Get Lunch`" is comprised of several smaller actions (walk to restaurant, order food, pay, etc.)

Sequence

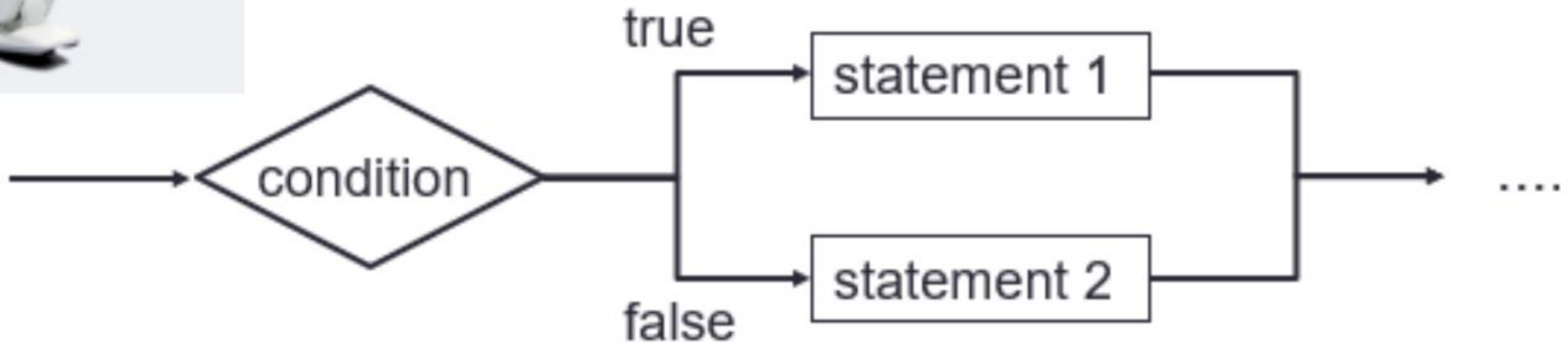
```
statement1
statement2
statement3
statement4
...
```



walk, walk, walk, walk, walk, walk, right-turn-180-degree, sit

Conditional

```
if (condition):
    statement1
    statement2
else:
    statement3
    statement4
    statement5
```

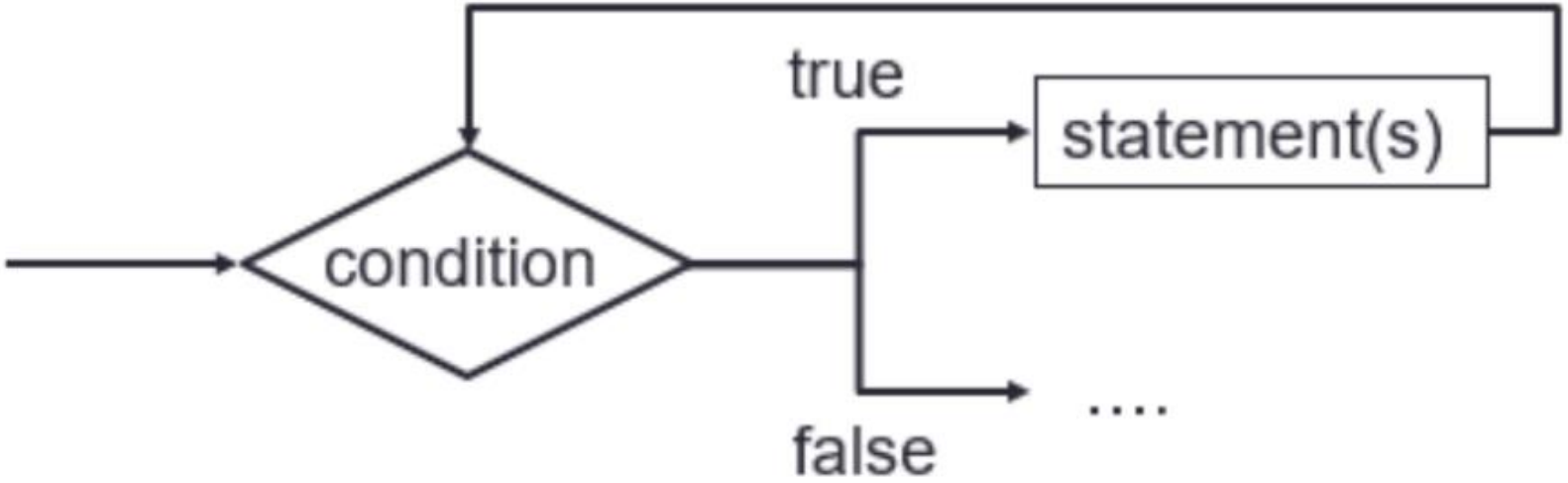


Repetition

```
while (condition):
    statement1
    statement2
statement3
```

? steps

- ~~Repeatedly walk 6 steps~~
- Repeatedly walk until you are in front of the chair
- Right-turn-180-degree
- Sit



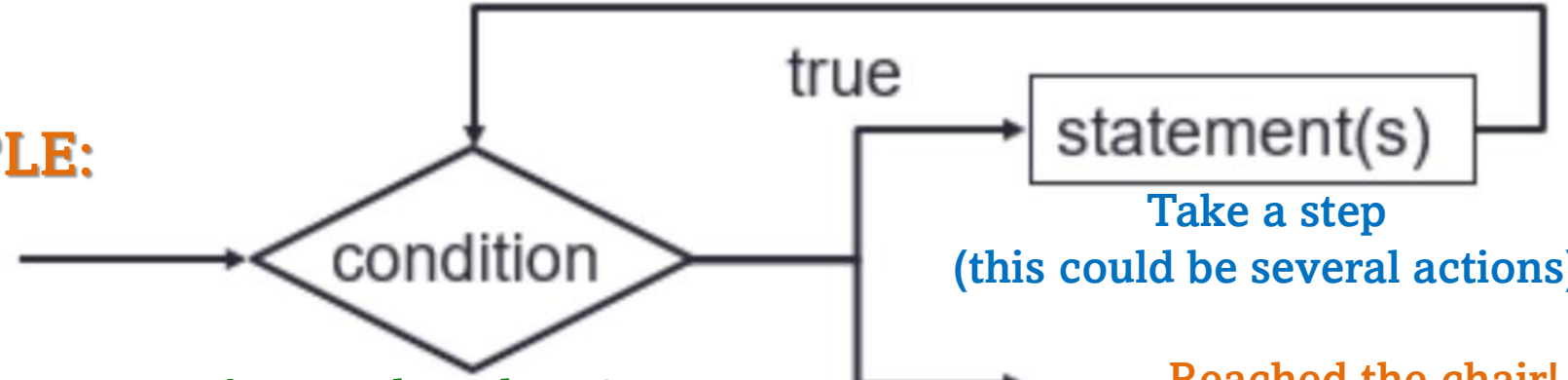
Repetition

```
while (condition):
    statement1
    statement2
statement3
```

? steps

- ~~Repeatedly walk 6 steps~~
- Repeatedly walk until you are in front of the chair
- Right-turn-180-degree
- Sit

EXAMPLE:

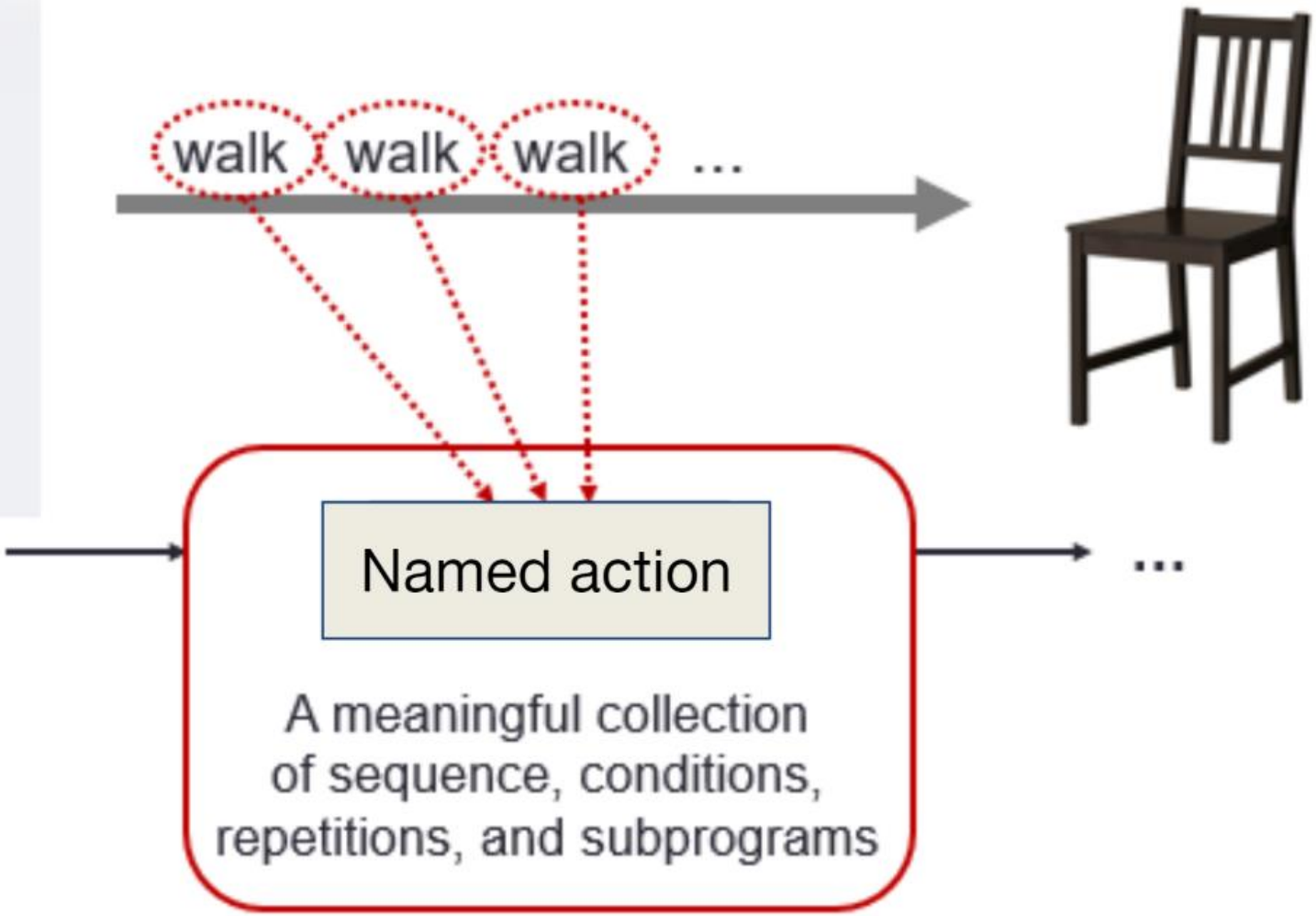


Take a step
(this could be several actions)

Are you less than 6 steps away from the chair?

….Reached the chair!
(So now turn around and sit)

Named Action



Algorithms (revisited)

- An algorithm is a step-by-step list of instructions to solve a problem
  - These steps must be followed EXACTLY
    - If you ever find yourself shouting at the computer *"Come on, you know what I mean!"*, **it doesn't.** Computers do exactly what you tell them, but will be VERY passive-aggressive about it

- Ways to describe an **algorithm**
  - Pseudocode ("*kinda*" code)
  - Flowchart (Diagram)
- Think of the **general solution** first before you try to write code to solve the problem!

What Makes a Good Algorithm?

- UNAMBIGUOUS
  - There are precise instructions that cannot be misinterpreted, that explain what to do each step AND what step to go to next
- EXECUTABLE
  - Each step can be carried out in practice
- TERMINATING
  - It will eventually come to and end
- DETERMINISTIC
  - It will do the "same thing" each time it is run
- CORRECT
  - Produces the right answer

- When describing an algorithm, don't think about implementation (coding) yet, focus on *"how do I solve this problem."*

What Makes a Good Algorithm?

- UNAMBIGUOUS
  - There are precise instructions that cannot be misinterpreted, that explain what to do each step AND what step to go to next
- EXECUTABLE
  - Each step can be carried out in practice
- TERMINATING
  - It will eventually come to and end
- DETERMINISTIC
  - It will do the "same thing" each time it is run
- CORRECT
  - Produces the right answer

Ambiguity

Prevalent in human language

No place for it in programming!

"She saw a woman on a hill with a telescope."

(who had the telescope?)

"A man walks into a bar." ***(ouch!)***

- When describing an algorithm, don't think about implementation (coding) yet, focus on *"how do I solve this problem."*

Bad Algorithm: Ambiguous

- "Bake it for a ***few*** minutes"

?

"How many minutes?"

Bad Algorithm: Not Executable

- "Bake it at ***10,000*** F"

!

"I can't do that!"

Bad Algorithm: Non-Terminating

- "Stir the batter (forever?!)"

∞

"…that's a long time!"

Bad Algorithm: Incorrect

- "Burnt Cookies"

By contrast: Hello World in Java

```
import java.io.*;

public class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
```

This is a simple yet complete Java program. It does one thing: Prints "**Hello World!**"
Output:

```
Hello World!
```

By contrast: Hello World in Java (with Comments)

```
/* Below is an import statement
 * it is used if you want to use code from other packages
 */
/* Java.io.* is all of Java's input/output stuff */
import java.io.*;

public class HelloWorld { // Class declaration (common single-line comment)
  /**
   * The main method of the program.
   * This is a Java doc comment, note the " /** "
   * @param args - variable for the input array of Strings
   */
  public static void main (String[] args) {
    /* This is how you print to the console */
    System.out.println("Hello World!");
  }
}
```

(This is a multi-line comment, note the " /* ")

What is System.out.println() ???

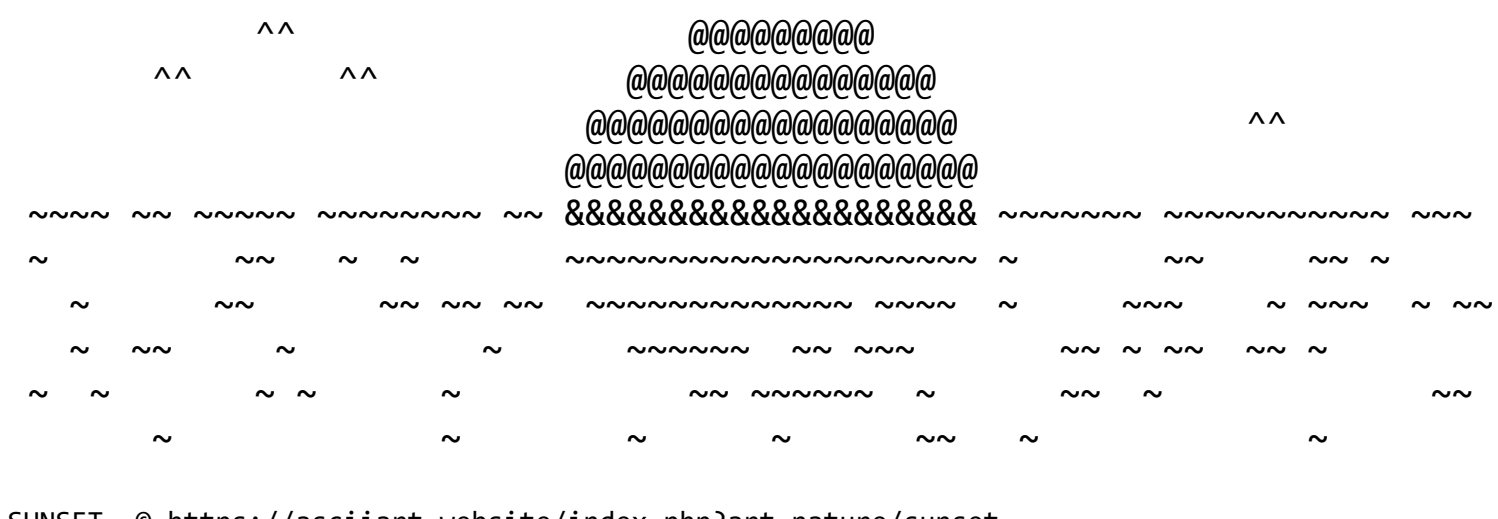
Python Demonstration

Introducing Basics of Python

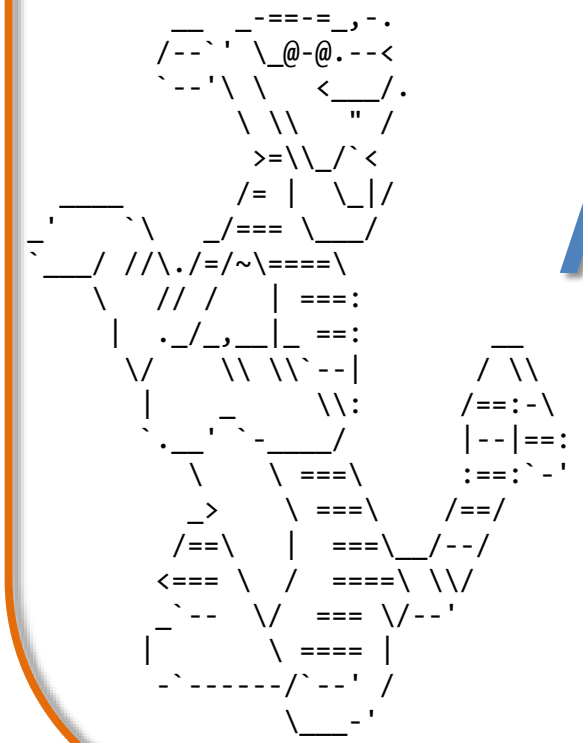
PyCharm environment (brief)

Simple Printing (using `print()` function) and Commenting

In-Class "lab" Activity: ASCII Art



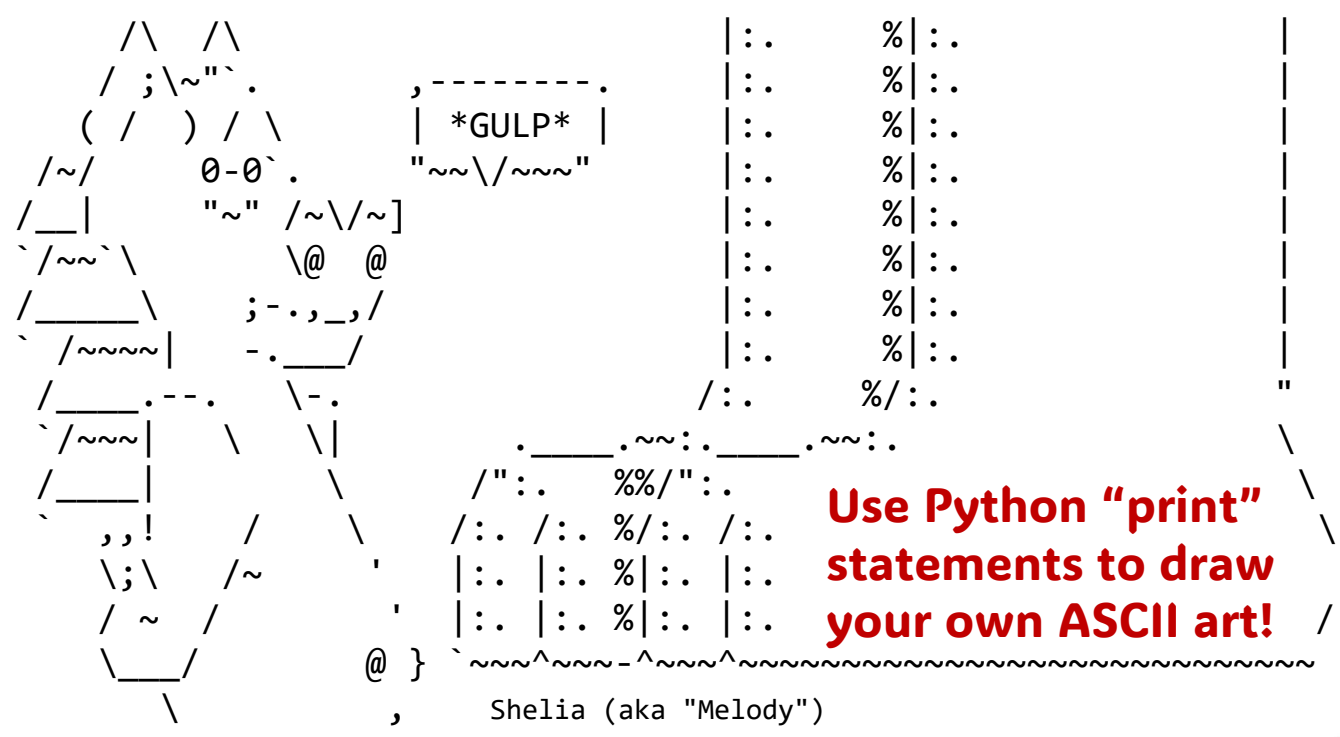
SUNSET © https://asciiart.website/index.php?art=nature/sunset



© https://asciiart.website/index.php?art=books/winnie%20the%20pooh

ASCII ART!

ACTIVITY



© https://asciiart.website/index.php?art=creatures/monsters

Notes/Reminders…

CS Laptop Loaner Program

- This course requires students to have a laptop
- I realize that not everybody might have one (nor necessarily need on for their desired major / path…)
- If you do not have a laptop for any reason… ***not to worry!***
- The CS department's Systems staff has a notebook / laptop loaner program and will be able to loan you a notebook / laptop computer for the duration of the semester if you don't have one or if you cannot afford one.
  - Also available if your laptop is broken and under repair, we can arrange for you to receive a loaner laptop for a week or two until your own laptop is fixed

Interested? Link: https://www.cs.virginia.edu/wiki/doku.php?id=cs_laptop_loaner

I am happy to be your sponsor. Please let me know.

Tools: Piazza

- We will use **Piazza** in the following way:
  - Website: https://piazza.com/ [Linked through Collab]
  - Piazza is a great tool for asking questions about **course content**, **policies**, or getting help on **homework** assignments
  - While you are waiting for an answer, see if there's an answer you can provide to someone else's question. We're all in this together! CS is a team sport! ☺
  - TAs will monitor and answer questions throughout the semester
  - Not a means to help you debug your code! (See more below)

It is very important to remember the following:

- **Do not post** complete or partial code solutions (for Homework) on Piazza when seeking answers to your question unless it is in a **PRIVATE** post
- **Do not post** complete or partial quiz solutions (code or short-answer) when seeking answers to your question unless it is in a **PRIVATE** post

Tools: Gradescope

- We will use **Gradescope** in the following way:
  - Website: https://www.gradescope.com/
  - Homework assignments will be **submitted**
    - Most programming assignments are autograded
    - Some aspects of programming assignments may be manually graded