

Objectives

- Learning Linux
 - Linux practice
- Programming practice
 - Print statements
 - Numeric operations, assignments
 - Input statements

Reintroduce lab assistants

Jan 26, 2021

Sprenkle - CSCI111

1

1

Getting Started

- In-Lab
 - Accessing a terminal
 - Under Applications, search for Terminal
 - Add to your favorites by dragging into side dock
 - Find a Web browser
 - Add to your favorites
 - Go to the course web site
- Remote access
 - Ssh to fred and then to one of the “center” machines
 - logo, postscript, rexx, smalltalk, apl, cilk, forth, limbo
 - In a web browser (on your personal machine), go to the course web site

Jan 26, 2021

Sprenkle - CSCI111

2

2

After Logging In

- `runHelpClient`

- Everything you do in lab on these machines (if you save it), you can access remotely.
- Everything you do remotely (if you save it), you can see on the lab machines.

3

Lab 0 Feedback

- Overall, did well
 - Generally, lab grades should be high
- Canvas extra credit Easter egg
 - Great fun facts!

4

Intro to UNIX: File Structure

- Organize our files
- Hierarchy of *directories* or “folders”

Jan 26, 2021

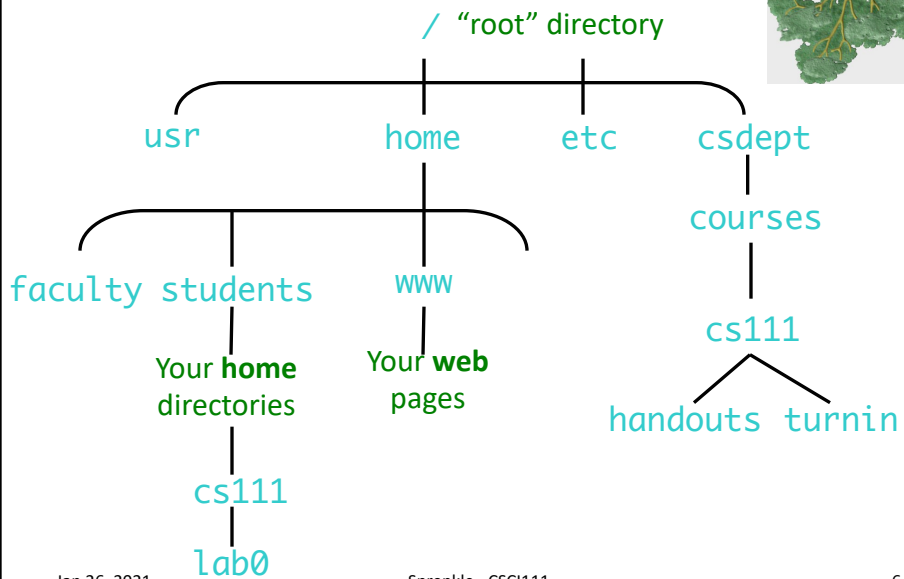
Sprenkle - CSCI111

5

5

(Partial) Linux File Structure

Paths through tree



Jan 26, 2021

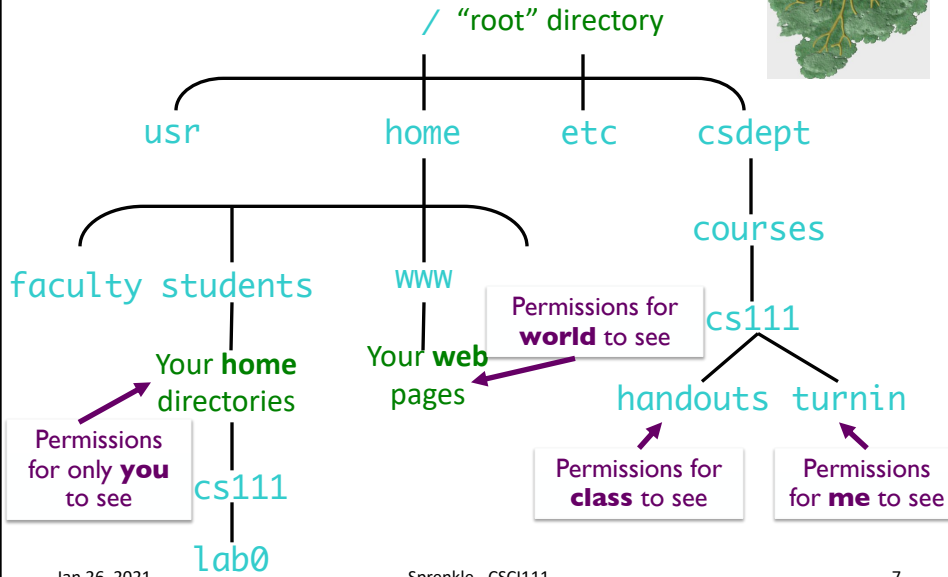
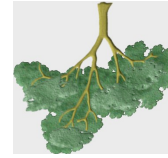
Sprenkle - CSCI111

6

6

(Partial) Linux File Structure

Paths through tree



7

Intro to UNIX: Essential Commands

- Manipulating Files

Command	What it does
ls	list the files, directories in a directory
mkdir dname	make a directory with the name "dname"
cp src dest	copy a src to a dest src and dest can be a file, set of files, or a directory
rm file	remove (delete) a file/directory

- Navigating Directories

pwd	print working directory
cd name	change to directory name

Jan 26, 2021

Sprenkle - CSCI111

8

8

Discuss (but don't do): What is the Unix command to do the following?

1. Find out what directory you're in
2. View the contents of the directory
3. Create a directory called cs111
4. View the contents of your directory (again)
5. Go into the cs111 directory
6. View the contents of cs111 directory

Jan 26, 2021

Sprenkle - CSCI111

9

9

What is the Unix command to do the following?

Now, execute those commands!

1. Find out what directory you're in
 - `pwd` You should be in your home directory
2. View the contents of the directory
 - `ls` What files are in your home directory?
3. Create a directory called cs111
 - `mkdir cs111`
 - View the contents of your directory again
4. Go into the cs111 directory
 - `cd cs111`
5. View the contents of cs111 directory
 - `ls`

Jan 26, 2021

Sprenkle - CSCI111

10

10

Where Am I ?

- It's easy to get lost as to where you are in the file system
- Some help:

```
sprenkle@fred:~/public_html/cs111/examples/03-development
sprenkle@fred:examples$ cd 03-development/
sprenkle@fred:03-development$ ls
arith_and_assign2.py  arith_and_assign.py  average2.py  index.html
sprenkle@fred:03-development$ more average2.py
# Finds the average of two numbers
# by CS111 class

print("This program computes the average of two numbers.")

# assign values to two numbers

num1=2.5
num2=-2.5

# find the average of the two numbers:
# add the two numbers together and then divide by 2

average=(num1 + num2)/2

# display the result
print("The average of", num1, "and", num2, "is", average)
sprenkle@fred:03-development$
```

Jan 26, 2021 Sprenkle - CSCI111 11

11

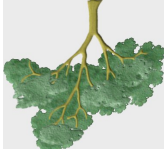
Useful Trick: Up Arrow

- Hit the up arrow. What happened?

- Hit the up arrow again? What happened?

12

Absolute vs Relative paths

Paths through tree 

/ "root" directory

```

graph TD
    Root[" / \"root\" directory "] --- usr
    Root --- home
    Root --- etc
    Root --- csdept
    home --- faculty
    home --- students
    home --- www
    csdept --- courses
    courses --- cs111
    cs111 --- handouts
    cs111 --- turnin
    cs111 --- lab0
  
```

usr home etc csdept

faculty students www

Your **home** directories Your **web** pages

cs111

handouts turnin

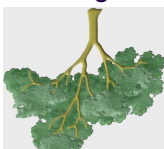
lab0

Absolute paths start from the root directory, work no matter where you are
→ What you see when you type pwd

Jan 26, 2021 Sprenkle - CSCI111 13

13

Absolute vs Relative paths

Paths through tree 

/ "root" directory

```

graph TD
    Root[" / \"root\" directory "] --- usr
    Root --- home
    Root --- etc
    Root --- csdept
    home --- faculty
    home --- students
    home --- www
    csdept --- courses
    courses --- cs111
    cs111 --- handouts
    cs111 --- turnin
    cs111 --- lab0
  
```

usr home etc csdept

faculty students www

Your **home** directories Your **web** pages

cs111

handouts turnin

lab0

Relative paths start from the current directory
Example: From your home directory, you can type `cd cs111` rather than `cd /home/students/username/cs111`

Jan 26, 2021 14

14

Intro to UNIX: Shortcuts

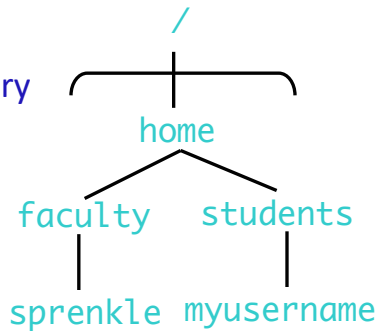
Shortcut	Meaning
.	Current Directory
..	Parent Directory

➤ Often used with **cp**, **mv**, **cd** commands

● **cd** or **cd ~**

➤ Change to *your* HOME directory

Example: /home/faculty
is the parent directory of
/home/faculty/sprenkle

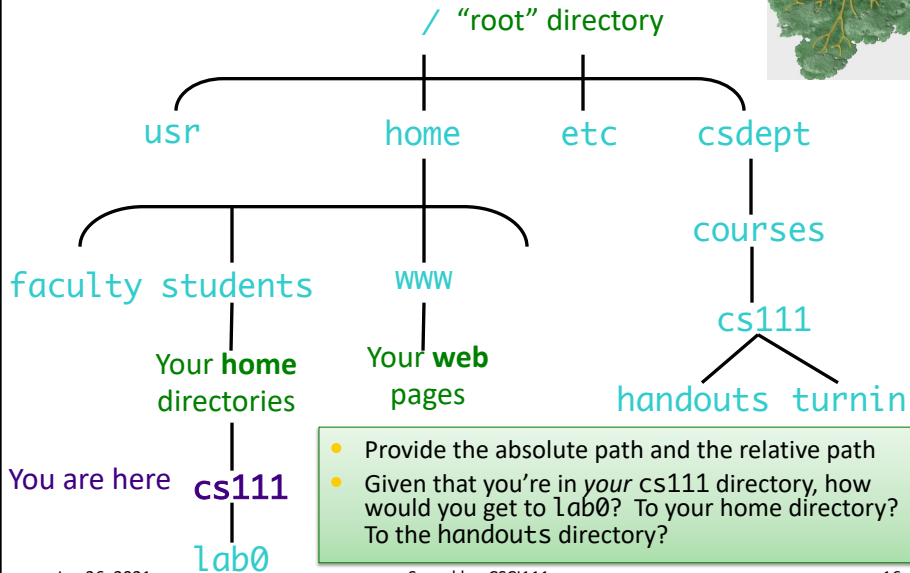
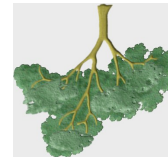


Jan 26, 2021

Sprenkle - CSCI111

Absolute vs Relative paths

Paths through tree



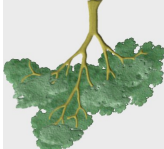
- Provide the absolute path and the relative path
- Given that you're in *your* cs111 directory, how would you get to lab0? To your home directory? To the handouts directory?

Jan 26, 2021

Sprenkle - CSCI111

16

Absolute vs Relative paths

Paths through tree 

/ "root" directory

usr home etc csdept

- To cs111
 - ∅ cd /home/students/username/cs111/lab0
 - ∅ cd lab0
- To home
 - ∅ cd /home/students/username
 - ∅ cd ..
 - ∅ cd
 - ∅ cd ~
- To handouts
 - ∅ cd /csdept/courses/cs111/handouts
 - ∅ cd ../../../../csdept/courses/cs111/handouts

Jan 26, 2021 Sprenkle - CSCI111 17

17

Practice, with Tab Completion

This is an absolute path

- Goal: go to the directory /csdept/courses/cs111
 - You can use **tab completion** to help complete commands
 - After typing the appropriate command, start to type /CS and then press tab.
 - What happens?
 - Use tab completion to help you complete the rest of the path
- What are the contents of this directory?
- How can you get to the directory /csdept/courses?
- How can you get back to *your* home directory?
 - (3 ways; which is fastest/most efficient?)

Jan 26, 2021 Sprenkle - CSCI111 18

18

jEdit: A text editor

Check: Are you are in your home directory?
Now, go into your CS111 directory.

- `jedit &`
 - Command to run
 - If command doesn't exist, you're probably still on fred; go to another machine
- Create a new file, add some text to it
 - e.g., "this is my file"
- Save the file in *your* `cs111` directory, naming it `test.txt`
- Exit jEdit, from the menu
- What are the contents of the directory now?

Jan 26, 2021

Sprenkle - CSCI111

19

19

More on the `cp` command: Copy

- `cp src dest`
 - `src`: what you want to copy
 - `dest`: to where you want to copy
 - If `dest` is a directory, copies `src` into that directory
 - If `dest` is a filename, makes a copy of `src` and names it `dest`
- Practice in the terminal: *First, discuss the steps*
 - Copy the file you just created and make a backup of it, e.g., named `test.txt.bkup`
 - Create a directory called `lab0`
 - Copy the backup file you just created into the `lab0` directory

Jan 26, 2021

Sprenkle - CSCI111

20

20

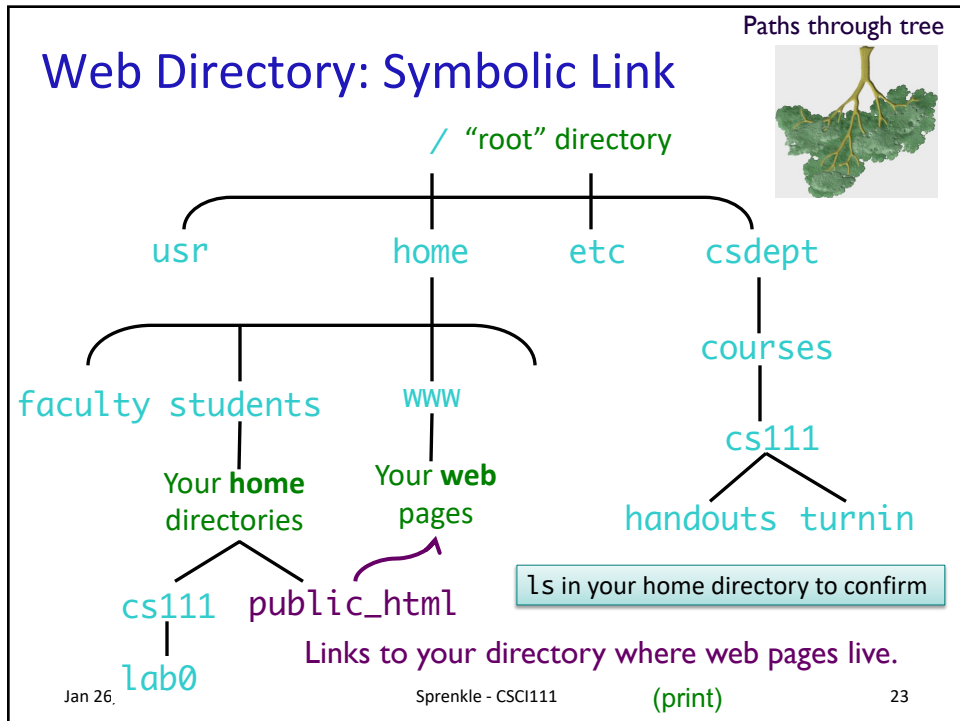
Using the Wildcard: *

- Go into `/csdept/courses/cs111/handouts/lab0`
 - What are the contents of this directory?
- Try executing
 - `ls *.py`
 - `ls example.*`

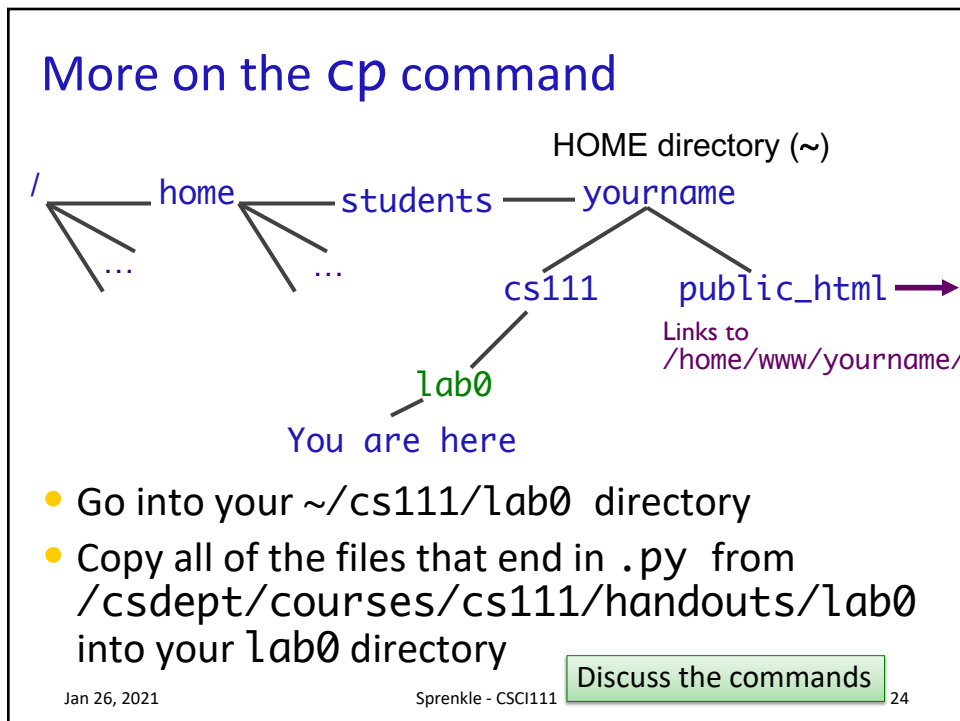
What does the * do?

Wildcard: *

- Matches 0 or more characters in filenames
- Used to operate on more than one file

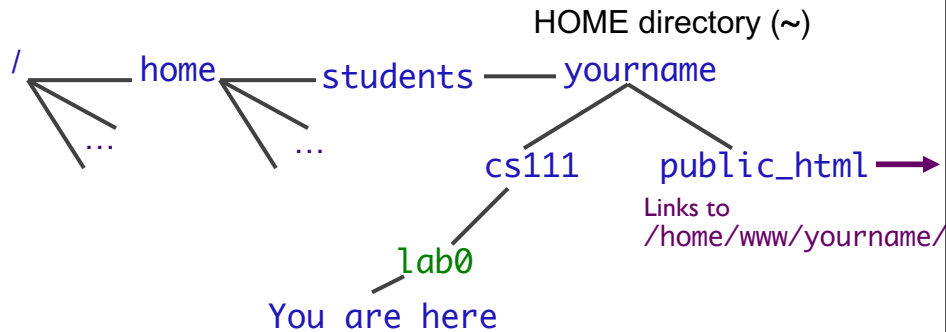


23



24

More on the cp command



- Go into your `~/cs111/lab0` directory
- Copy all of the files that end in `.py` from `/csdept/courses/cs111/handouts/lab0` into your `lab0` directory

```
cp /csdept/courses/cs111/handouts/lab0/*.py .
```

25

In-Lab Instructions

- You'll need a web browser

How can you launch a web browser?

- When you're done and leave lab, you should log out

➤ but not shutdown the machine

How do you log out?

26

Creating a Web Page

- Practical application of UNIX command skills
 - Practice commands you learned today
- Learning from following examples and adapting
- Learn what's "behind the curtain" of web pages

Go to Lab 1 in the browser and get started on this part.
Then, we'll get to programming Python on Linux.

PYTHON PROGRAMMING IN LINUX

Review

- What are the two ways we can run the Python interpreter?
- How do we assign values to variables?
- What arithmetic operators are available?
 - What rules do they follow?
- What is our development process?
 - What is the two-part verification process we need to do after we implement a program?

Jan 26, 2021

Sprenkle - CSCI111

29

29

Recap: Programming Fundamentals

- Most important data types (for us, for now):
int, float, str, bool
 - Use these types to represent various information
- Variables have identifiers, (implicit) types
 - Should have “good” names
 - Names: start with lowercase letter; can have numbers, underscores
- Assignments
 - $x = y$ means “x set to value y” or “x is assigned value of y”
 - Only variable on LHS of statement changes

Jan 26, 2021

Sprenkle - CSCI111

30

30

Review: Assignment statements

- Assignment statements are NOT math equations!

```
count = count + 1
```

- These are commands!

```
x = 2
```

```
y = x
```

```
x = x + 3
```

After these 3 statements execute,
what are the values of x, y?

Review: Numeric Arithmetic Operations

Symbol	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder ("mod")
**	Exponentiation (power)

Remember PEMDAS

Review: Development Process

1. Create a sketch of how to solve the problem (the algorithm)
2. Fill in the details in Python
3. Test code using *good, varied* test cases to try to find errors in code
4. If program's output does not match the expected output, debug to find the problem and fix it
 - Repeat testing and debugging until no more faults
5. Make code "better", test again
 - Fix variable names, better comments

Jan 25, 2021

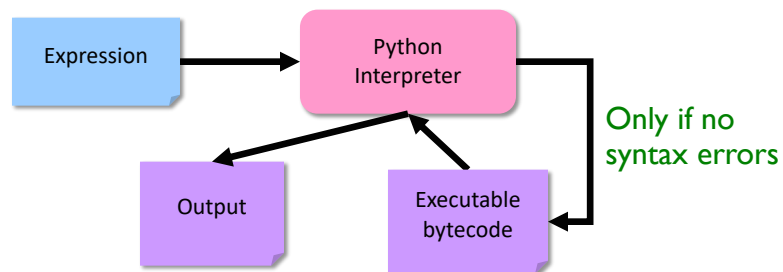
Sprenkle - CSCI111

33

33

Python Interpreter

1. Validates Python programming language expression(s)
 - Enforces Python syntax rules
 - Reports syntax errors ← Have a lot of these early on!
2. Executes expression(s)



Jan 26, 2021

Sprenkle - CSCI111

34

34

Two Modes to Execute Python Code

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

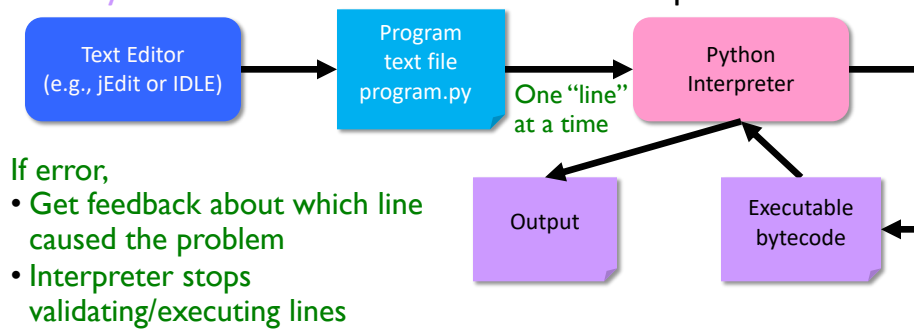
Python Interpreter: Interactive Mode

Run by typing `python3` in terminal

```
sprengle@Saras-MacBook-Pro: ~ % python3
Python 3.9.1 (v3.9.1:1e5d33e00, Dec 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
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
Traceback (most recent call last):
  File "<stdin>", line 1
    print 4+5
    ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(4+5)?
>>> print(4+5)
9
>>>
```

Batch Mode: Execute Scripts

1. Programmer save a **program/script** into a **text file** using a **text editor**.
2. An **interpreter** turns each expression in file into **bytecode** and then executes each expression



- If error,
- Get feedback about which line caused the problem
 - Interpreter stops validating/executing lines

Jan 26, 2021

Sprenkle - CSC111

37

37

Example Python Script

Text file named: `hello.py`

```
# A first program  
# by Sara Sprenkle, 01/26/2021  
print("Hello, world!")
```

Print statement

- What does this program do?
 - Validate your guess by executing the program
 - Go into your `lab0` directory
 - `python3 hello.py`

Jan 26, 2021

Sprenkle - CSC111

38

38

Example Python Script

```
# A first program
# by Sara Sprenkle, 01/26/2021
print("Hello, world!")
```

Documentation
-- good style

- Only **Hello, world!** is printed out
- Python ignores everything after the “#”
 - Known as “**comments**” or, collectively, as **documentation**

Your program should *always* start with a high-level description of what the program does, your name, and the date the program was written

Jan 26, 2021

Sprenkle - CSCI111

39

39

IDLE Development Environment

- Runs on top of Python interpreter
- Command: **idle3 &**
 - **&** Runs command in “background” so you can continue to use the terminal

IDLE
python

Since our programming language is named after Monty Python, what is the development environment named after?

- Can use IDLE to
 - Run Python in **interactive** mode
 - Write and execute scripts in **batch** mode

Jan 26, 2021

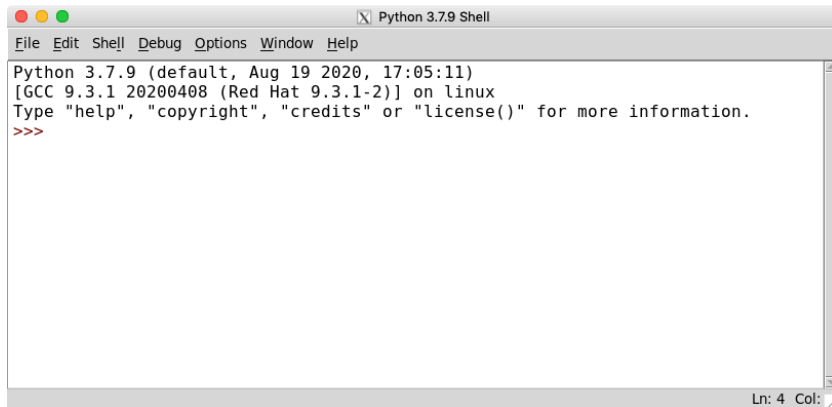
Sprenkle - CSCI111

40

40

IDLE

- IDLE first opens up a Python shell
 - i.e., the Python interpreter in interactive mode



```
Python 3.7.9 (default, Aug 19 2020, 17:05:11)
[GCC 9.3.1 20200408 (Red Hat 9.3.1-2)] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Jan 26, 2021

Sprenkle - CSC111

41

41

Your Turn in Interactive Mode...

- If you exited IDLE, run `idle3 &`
- Enter the following expressions and see what Python displays:
 - `3`
 - `4 * -2`
 - `-1+5`
 - `2 +`
 - `print("Hello!")`
- Alternatively, can use `python3`
 - If you used `python3`, to quit the interpreter, use `Control-D`

Jan 26, 2021

Sprenkle - CSC111

42

42

IDLE

- In IDLE, under the **File** menu
 - Use **New File** or **Open**, as appropriate, to open a window so that you can write your Python script.
- Practice:
 - Create a new file
 - Print out “hello!”
 - Save the file in your home directory
 - Execute the program (opens a new Python shell)
 - Run → Run Module or F5

Jan 26, 2021

Sprenkle - CSCI111

43

43

Recap: Executing Python

- Interactive Mode
 - Try out expressions
 - **python3**
- Batch Mode
 - Execute Python scripts
 - **python3 <pythonscript>**
- **IDLE** combines these two modes into one *integrated development environment (IDE)*
 - **idle3 &**

Jan 26, 2021

Sprenkle - CSCI111

44

44

Lab 1 Expectations

- Comments in programs
 - High-level comments, author
 - Notes for your algorithms, implementation
- Nice, readable, clearly labeled understandable output
 - User running your program needs to **understand** what the program is saying
- Honor System

Jan 26, 2021

Sprenkle - CSCI111

45

45

Lab 1: Programming Practice

- After the warm up problems
- Name program files **lab1.n.py**, where n is the problem you're working on
- After completed, demonstrate that your program works
 1. Close IDLE/Python interpreter, rerun program
 - Get rid of the output from when you were developing/debugging ("scratch work")
 2. Save output for each program in file named **lab1.n.out** where n is the problem you're working on

Jan 26, 2021

Sprenkle - CSCI111

46

46

Lab 1 Expectations: Example Output

- Your program should have clearly labeled output
 - Clear to user what is happening in program
- You will run some programs **multiple times** to demonstrate that the program works with different values of variables.
- Resulting output should be saved in a `.out` file

Lab 1 Expectations: Read the Directions

- To **completion**
- Often the answer to your question is in the next sentence
- Practice patience
 - Rushing → poor outcomes

Lab 1 Submission

- Electronic
 - I can execute your program, help find mistakes
 - Copy your lab directory into your turnin directory
- Instructions are in the lab

Honor

- You may discuss programming assignments *informally* with other students
 - Sharing the **code** is an honor violation
 - Do **not** share your password
- You should know where to draw the line between legitimate outside assistance with course material and outright cheating
 - Students who obtain too much assistance without learning the material ultimately cheat themselves
- If you have any uncertainty about what this means, consult with me before you collaborate.

Honor System: Rules of Thumb

- Discussion of problems/programs - OK
 - Clarification questions
 - Algorithm discussion (on paper, board)
- Do not look at another student's solution
 - "What did you do for that?"
- Debugging help
 - Programmer always "owns" keyboard, mouse
 - Helper can read other's program/debug/help, up to 5 minutes
 - Ask student assistant or me or email me for problems that require more time