

# WHAT TO ADD NEXT TIME YOU ARE UPDATING THESE SLIDES

- Update slides to have more animation in the bullet lists
- Verify that each slide has stand alone speaker notes



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™



**STEAM CLOWN™ PRODUCTIONS**

# **PYTHON 3 INTRODUCTION**

A Python class for my Mechatronics Engineering @ SVCTE. Last Updated for 2017 – 2018 school year



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**  
© Copyright 2017 STEAM Clown™



# STEAM CLOWN™ PRODUCTIONS



Attribution-NonCommercial-ShareAlike  
3.0 Unported (CC BY-NC-SA 3.0)

These slides are an adaption, to better target my SVCTE High School Mechatronics Engineering class, primarily from Dr. Charles R. Severance's Python for Everybody class <https://www.py4e.com/> ... but from other sources as well. See Appendix A

## SEE APPENDIX A, FOR LICENSING & ATTRIBUTION INFORMATION

by-nc-sa-3.0

<https://creativecommons.org/licenses/by-nc-sa/3.0/>

<https://creativecommons.org/faq/#what-does-some-rights-reserved-mean>

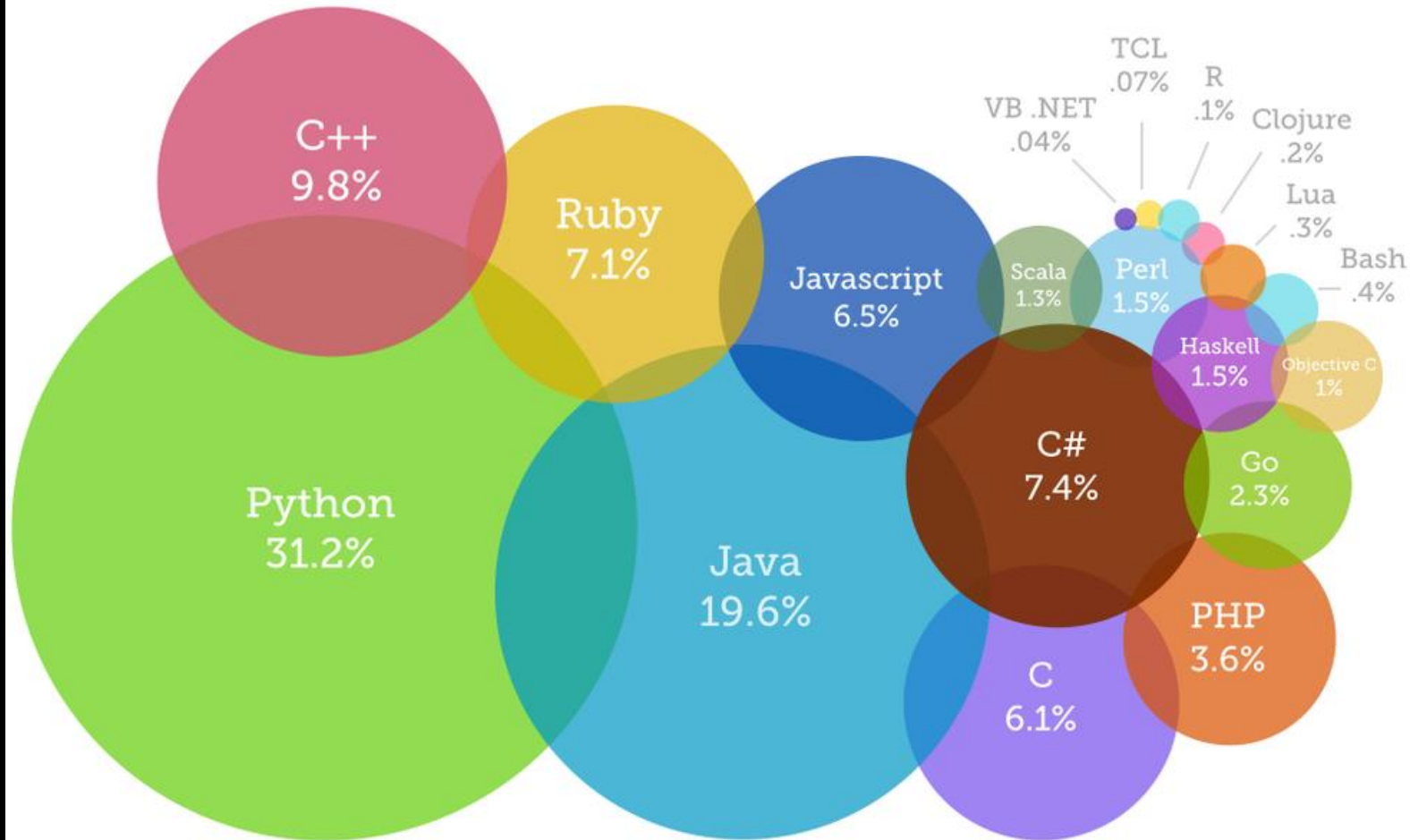


**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# WHY PYTHON?

Most Popular Coding Languages of 2015



# HISTORY OF PYTHON

- Python was conceived in the late 1980's by **Guido van Rossum**
- He started seriously writing and deploying code in December 1989
- Open Source



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# WHERE DID THE NAME COME FROM?



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# VIDEO HISTORY



The History of Python



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# PYTHON IS...

- Widely used general-purpose, high-level programming language
  - Easy to learn
- A design philosophy that emphasizes code readability
- A syntax that allows programmers to express concepts in fewer lines of code
  - Code Simplicity (Codability)
- While all languages have limitations, Python is robust and can handle most programming challenges





# FEATURES OF THE PYTHON LANGUAGE

- Clear, readable syntax
- Object orientation
- Natural expression of procedural code
- Full modularity, supporting hierarchical packages
- Exception-based error handling
- High level dynamic data types
- Extensive standard libraries and third party modules for virtually every task
- Extensions and modules easily written in C, C++ (or Java for Jython)
- Embeddable within applications as a scripting interface



# CORE PHILOSOPHY

- **Beautiful** is better than ugly
- **Explicit** is better than implicit
- **Simple** is better than complex
- **Complex** is better than complicated
- **Readability** counts
  - Indentation is the key to everything
  - Don't need to wrap code in {}
  - But you do need to watch your indentations

# PYTHON 3

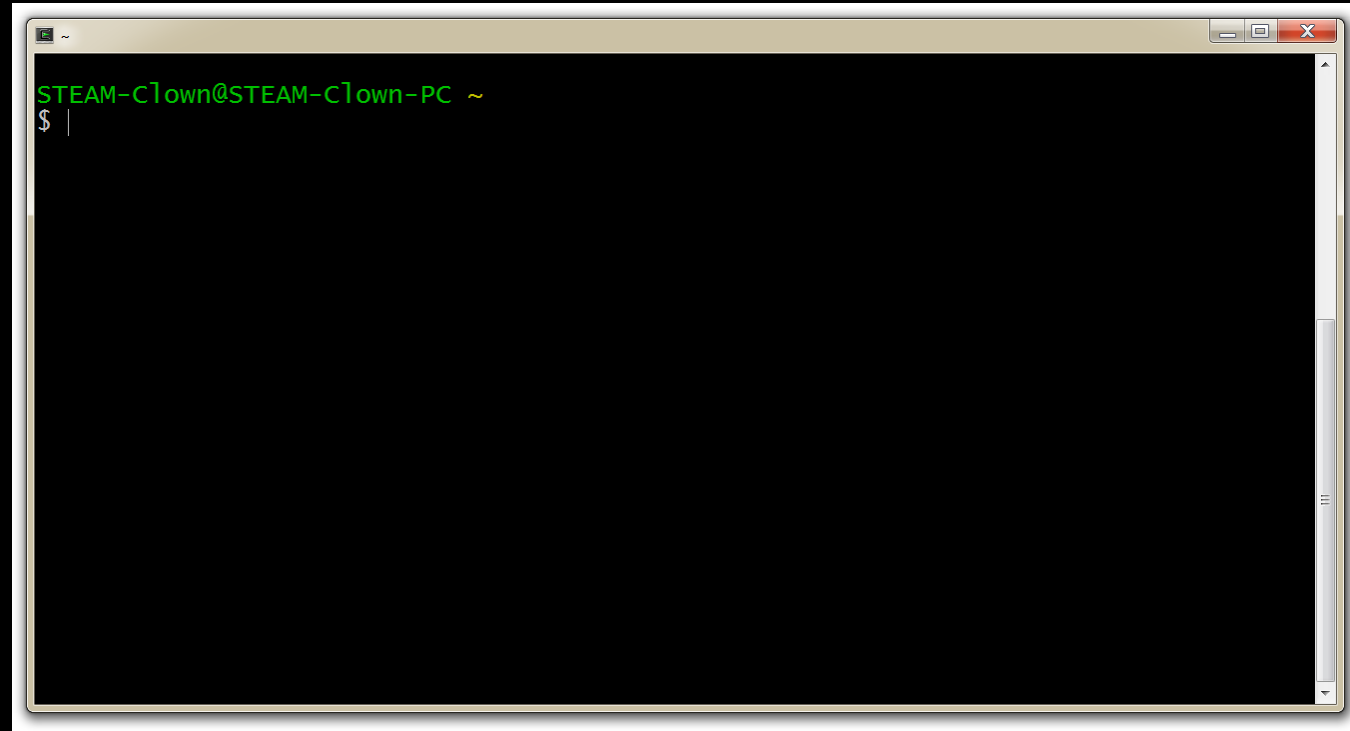
- This class will target Python 3. All posted code will be targeting a python 3 compiler/interpreter

# WHERE TO GET SOME HELP

- SVCTE Mechatronics Python Resource link
  - [Python Resources](#)

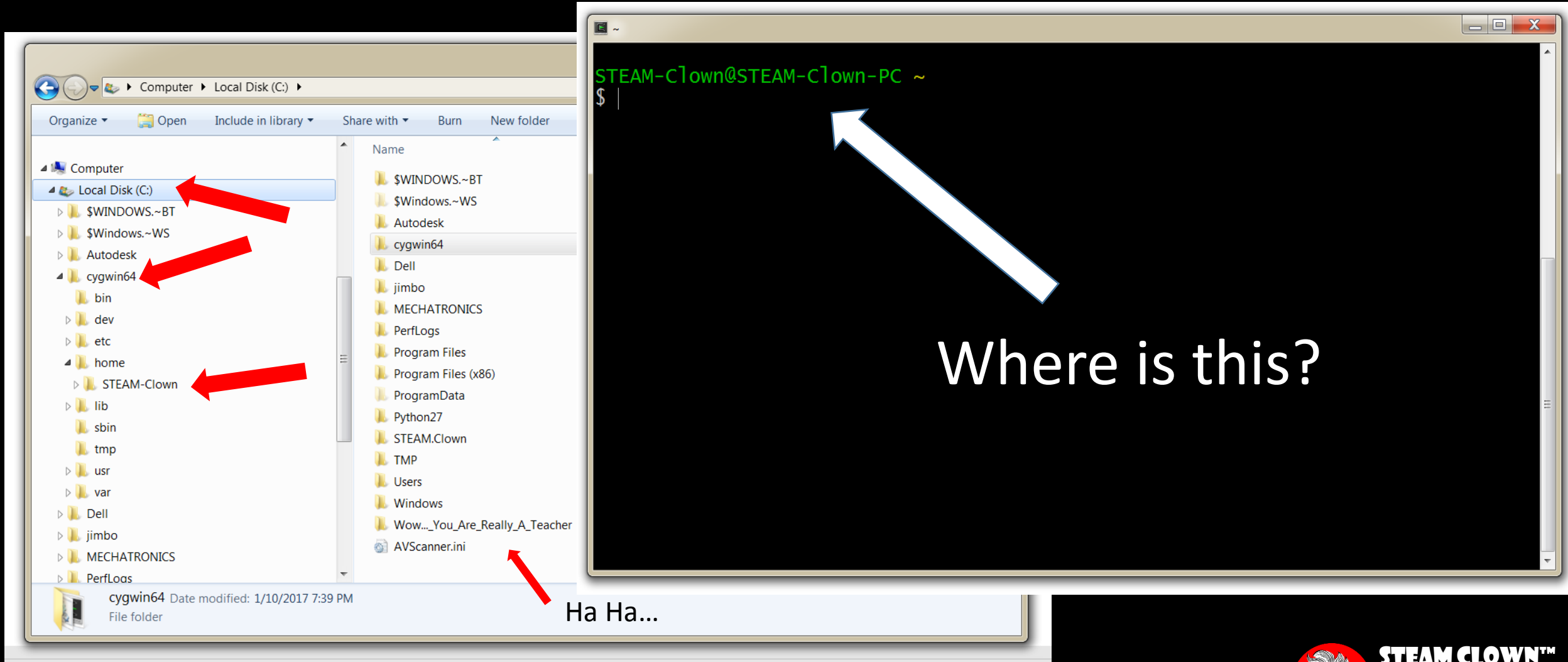
# HOW ARE WE GOING TO ACCESS PYTHON ON A PC? CYGWIN

- Open a Cygwin bash shell
- Yes... we are going to learn about Cygwin & Command line commands



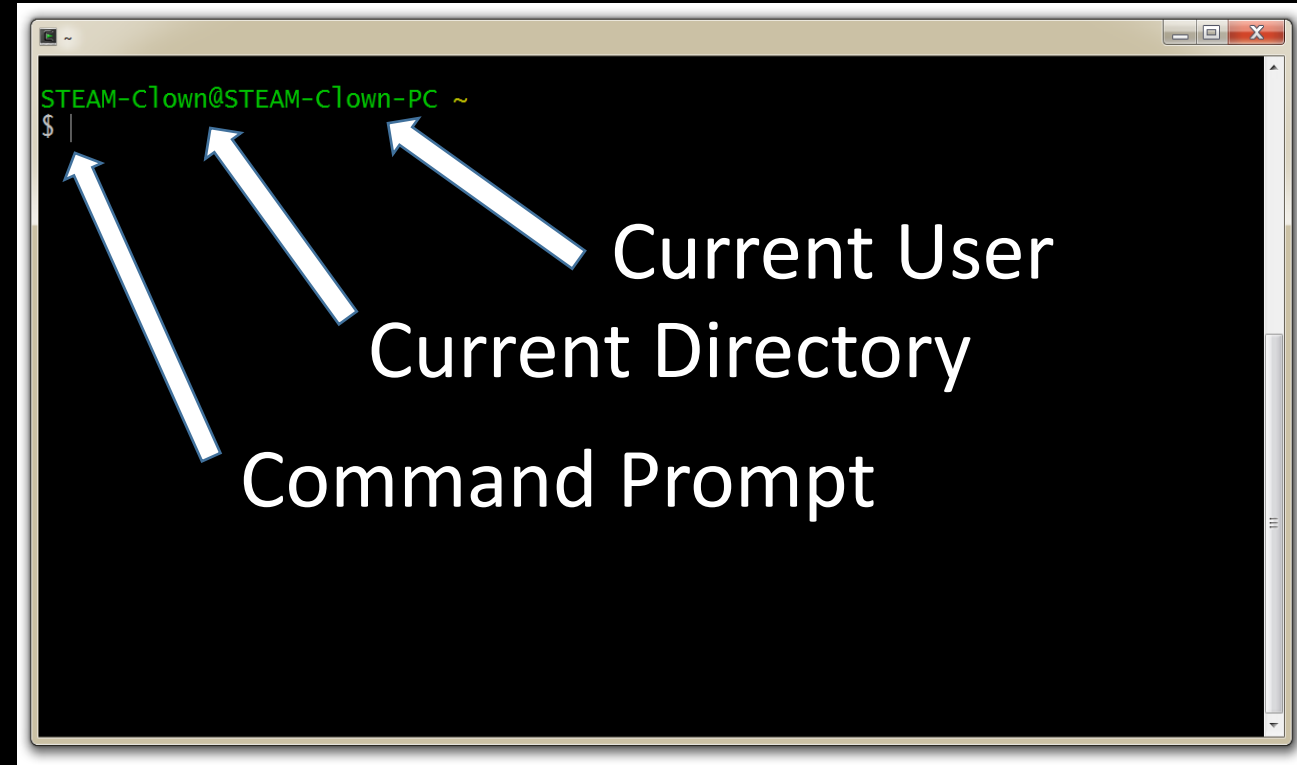
```
STEAM-Clown@STEAM-Clown-PC ~  
$
```

# GETTING AROUND A BASH SHELL



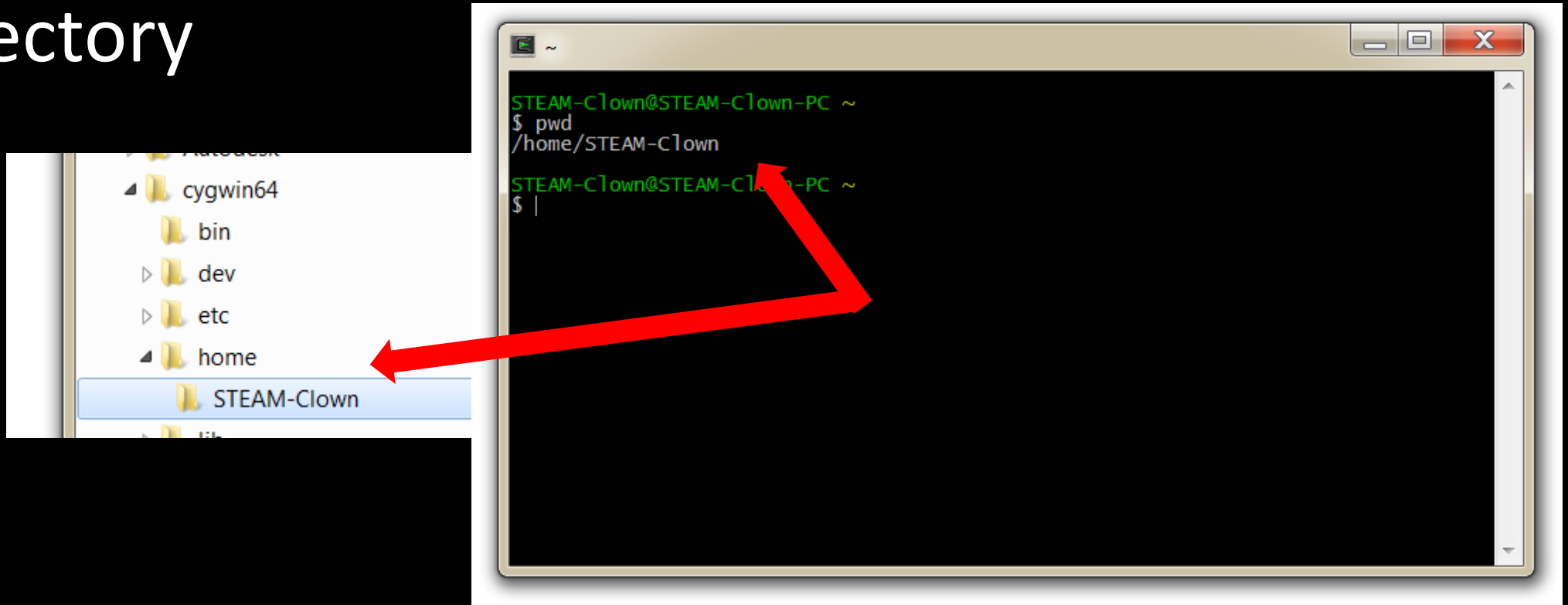
# JUST ENOUGH LINUX COMMANDS... 2 B DANGEROUS

- In my case, “home” user directory is STEAM-Clown and the user is @STEAM-Clown-PC ~
- The “\$” is the command prompt



# PWD

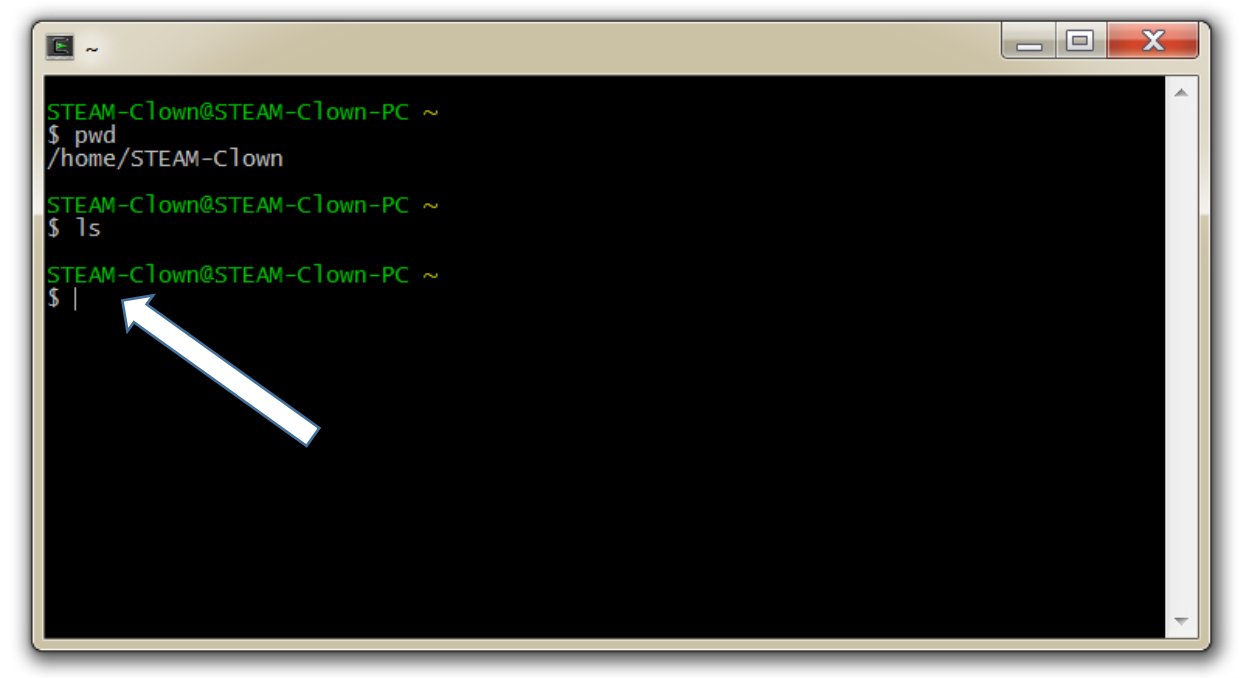
- `pwd` = Print Working Directory





# LS


- `ls` = List or List Directory
- Notice there are no sub-directories or files



```
STEAM-Clown@STEAM-Clown-PC ~  
$ pwd  
/home/STEAM-Clown  
STEAM-Clown@STEAM-Clown-PC ~  
$ ls  
STEAM-Clown@STEAM-Clown-PC ~  
$ |
```

# MKDIR

- `mkdir` = Make Directory
- Type `mkdir myPython`
- Now type `ls` again and see that a new directory `myPython` has been created

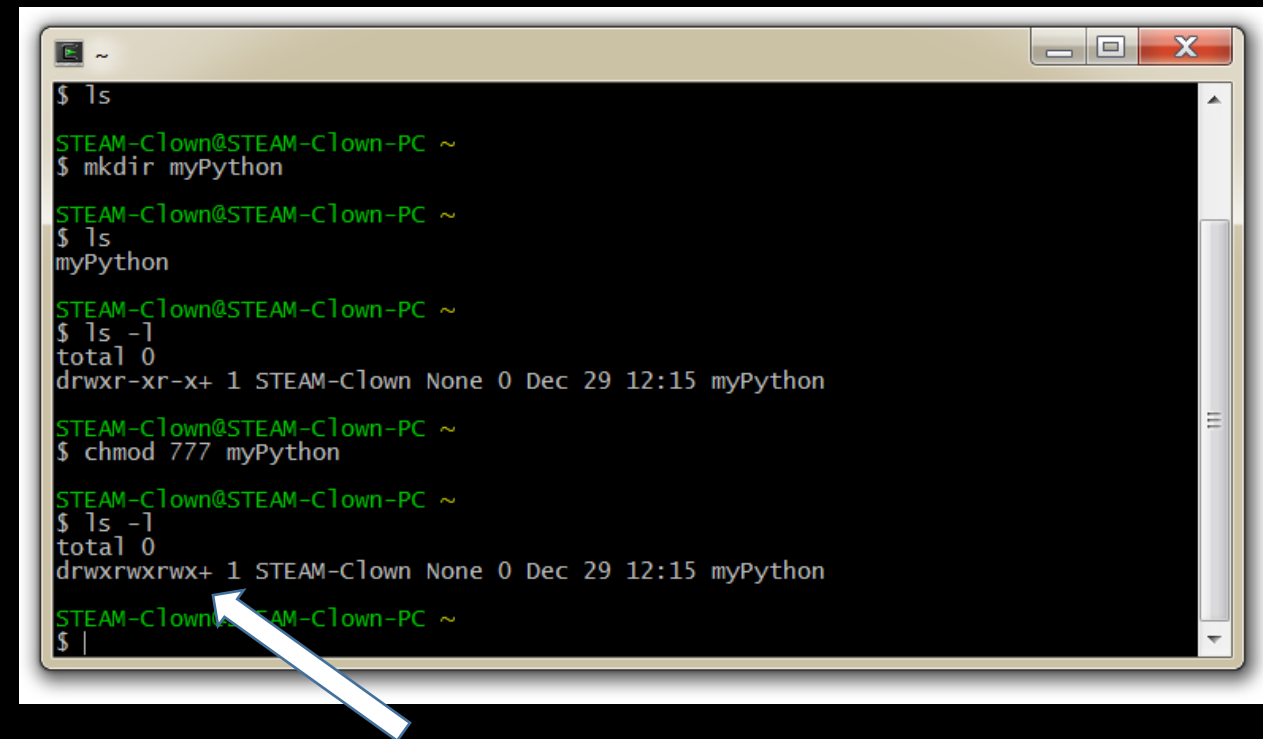


```
STEAM-Clown@STEAM-Clown-PC ~  
$ pwd  
/home/STEAM-Clown  
STEAM-Clown@STEAM-Clown-PC ~  
$ ls  
STEAM-Clown@STEAM-Clown-PC ~  
$ mkdir myPython  
STEAM-Clown@STEAM-Clown-PC ~  
$ ls  
myPython  
STEAM-Clown@STEAM-Clown-PC ~  
$ |
```

A terminal window showing the execution of the `mkdir` command to create a directory named `myPython`. The window title is `~`. The prompt is `STEAM-Clown@STEAM-Clown-PC ~`. The first `pwd` command returns `/home/STEAM-Clown`. The second `ls` command shows the current directory contents. The `mkdir myPython` command is executed, and the third `ls` command shows that `myPython` has been added to the directory listing. A white arrow points to the `myPython` entry in the output of the third `ls` command.

# A FEW MORE COMMANDS

- `ls -l` = List with more info
- `chmod` = Change Directory permissions
- `chmod 777 <dirName>`
- **Read/Write/Execute**

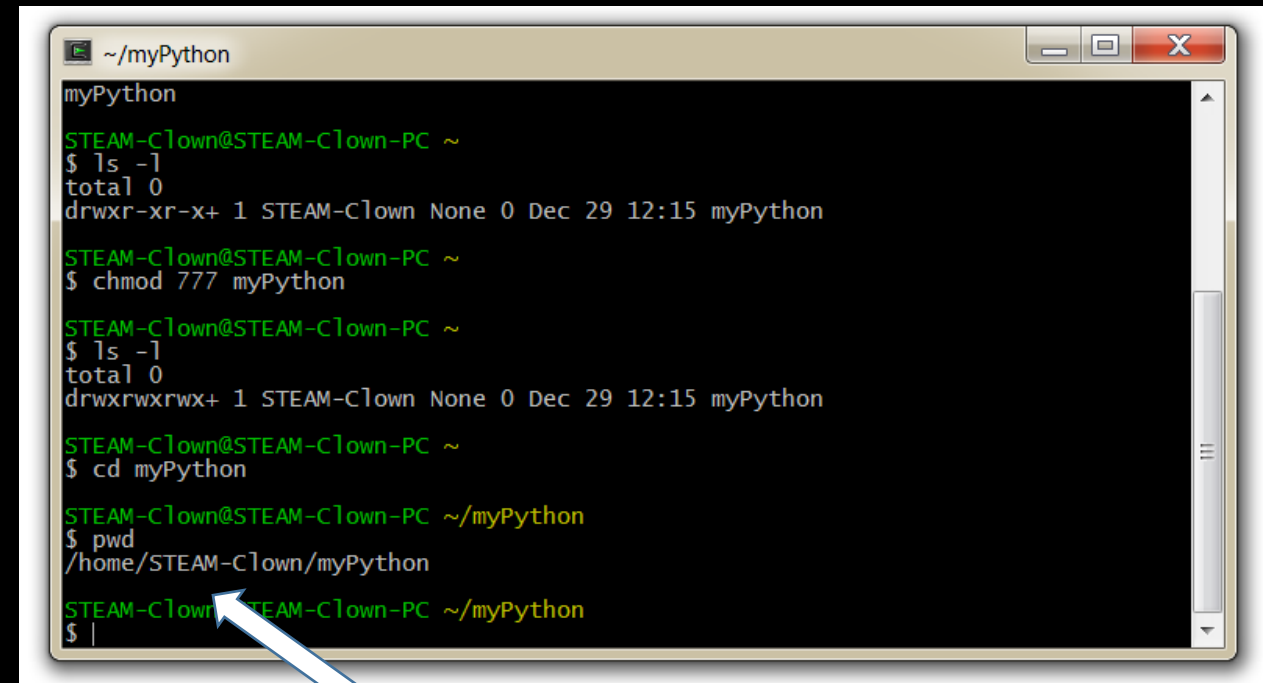


```
~  
$ ls  
STEAM-C1own@STEAM-C1own-PC ~  
$ mkdir myPython  
STEAM-C1own@STEAM-C1own-PC ~  
$ ls  
myPython  
STEAM-C1own@STEAM-C1own-PC ~  
$ ls -l  
total 0  
drwxr-xr-x+ 1 STEAM-C1own None 0 Dec 29 12:15 myPython  
STEAM-C1own@STEAM-C1own-PC ~  
$ chmod 777 myPython  
STEAM-C1own@STEAM-C1own-PC ~  
$ ls -l  
total 0  
drwxrwxrwx+ 1 STEAM-C1own None 0 Dec 29 12:15 myPython  
STEAM-C1own@STEAM-C1own-PC ~  
$ |
```

A screenshot of a terminal window showing the execution of several Linux commands. The user first runs `ls`, then `mkdir myPython`. They then run `ls` again to see the new directory. Next, they run `ls -l` to see detailed permissions for `myPython`, which are `drwxr-xr-x+ 1 STEAM-C1own None 0 Dec 29 12:15 myPython`. Finally, they run `chmod 777 myPython` to change permissions. A subsequent `ls -l` shows the updated permissions: `drwxrwxrwx+ 1 STEAM-C1own None 0 Dec 29 12:15 myPython`. A white arrow points to the `+` symbol in the final `ls -l` output.

# AND A FEW MORE...

- `cd` = Change Directory
- Now we are inside the directory `myPython`



A terminal window titled `~/myPython` showing the following commands and output:

```
myPython
STEAM-Clown@STEAM-Clown-PC ~
$ ls -l
total 0
drwxr-xr-x+ 1 STEAM-Clown None 0 Dec 29 12:15 myPython

STEAM-Clown@STEAM-Clown-PC ~
$ chmod 777 myPython

STEAM-Clown@STEAM-Clown-PC ~
$ ls -l
total 0
drwxrwxrwx+ 1 STEAM-Clown None 0 Dec 29 12:15 myPython

STEAM-Clown@STEAM-Clown-PC ~
$ cd myPython

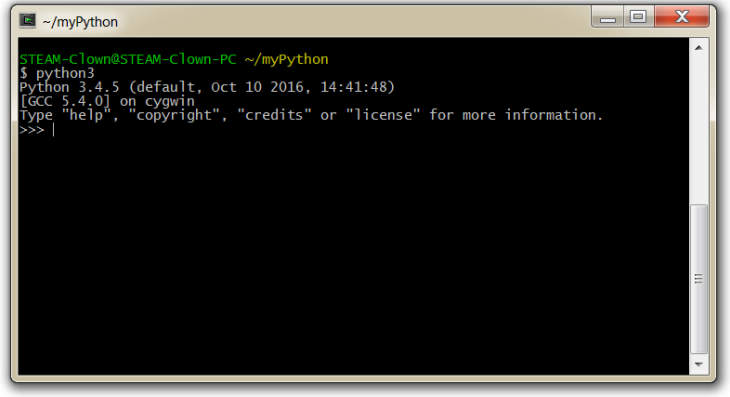
STEAM-Clown@STEAM-Clown-PC ~/myPython
$ pwd
/home/STEAM-Clown/myPython

STEAM-Clown@STEAM-Clown-PC ~/myPython
$ |
```

A white arrow points to the prompt `$ |` at the bottom of the terminal window.

# AND NOW... BACK TO PYTHON

- Check to see that you have python3 installed;
  - For this class, on a PC, we are going to run it in a Cygwin terminal
  - Install Cygwin with Python and python3
- In the Cygwin terminal
  - Type “python3” not just “python”
    - This starts the python3 interpreter



```
~/myPython
STEAM-Clown@STEAM-Clown-PC ~/myPython
$ python3
Python 3.4.5 (default, Oct 10 2016, 14:41:48)
[GCC 5.4.0] on cygwin
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

~/myPython

```
total 0
drwxr-xr-x+ 1 STEAM-C1own None 0 Dec 29 12:15 myPython
```

```
STEAM-C1own@STEAM-C1own-PC ~
$ chmod 777 myPython
```

```
STEAM-C1own@STEAM-C1own-PC ~
$ ls -l
total 0
drwxrwxrwx+ 1 STEAM-C1own None 0 Dec 29 12:15 myPython
```

```
STEAM-C1own@STEAM-C1own-PC ~
$ cd myPython
```

```
STEAM-C1own@STEAM-C1own-PC ~/myPython
$ pwd
/home/STEAM-C1own/myPython
```

```
STEAM-C1own@STEAM-C1own-PC ~/myPython
$ python3
Python 3.4.5 (default, Oct 10 2016, 14:41:48)
[GCC 5.4.0] on cygwin
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

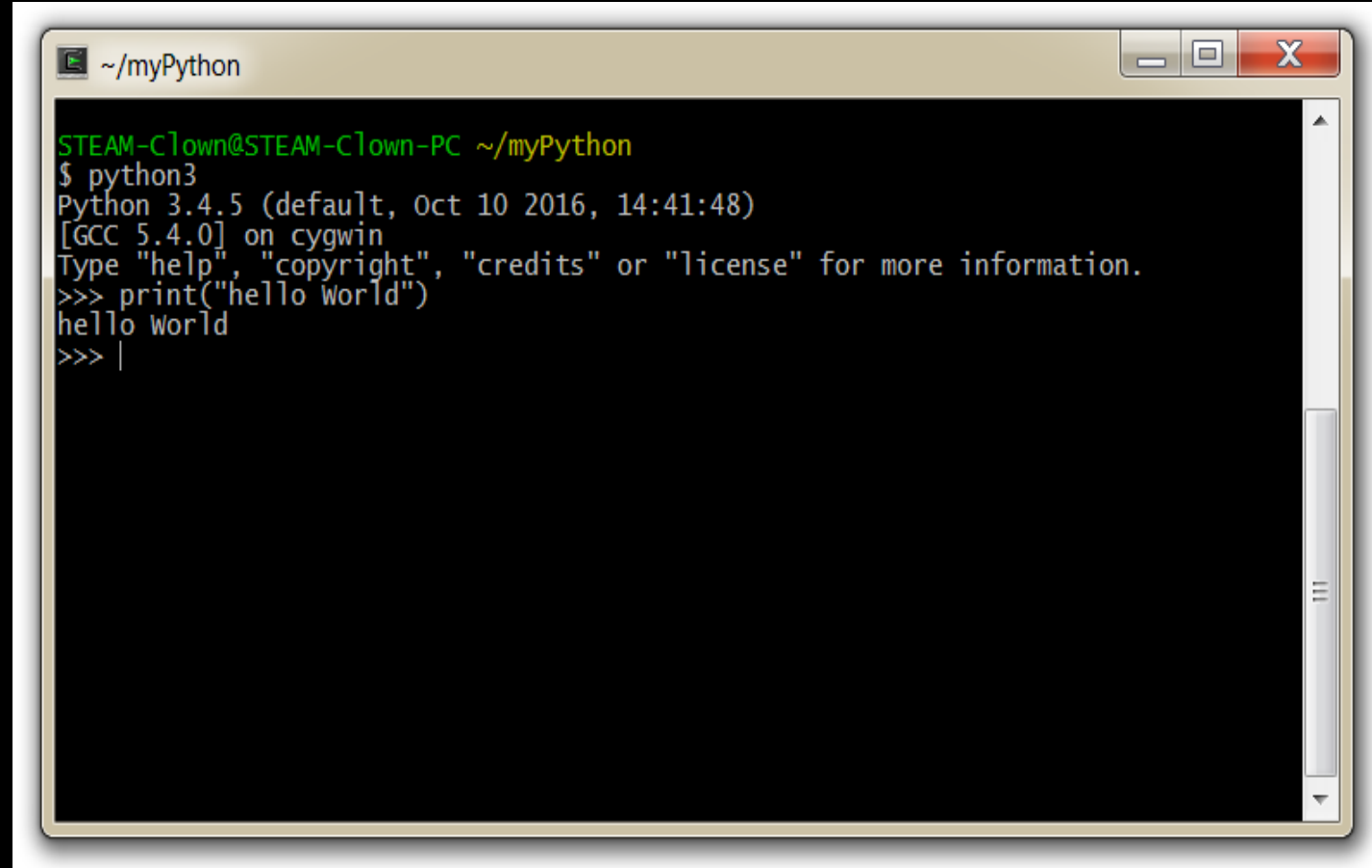


**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# HELLO WORLD

- Like C++ Python has functions
- Print("hello World")



```
~/myPython
STEAM-Clown@STEAM-Clown-PC ~/myPython
$ python3
Python 3.4.5 (default, Oct 10 2016, 14:41:48)
[GCC 5.4.0] on cygwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello World")
hello world
>>> |
```

# THE MAGIC OF PYTHON

- The “>>>” is a Python *prompt* indicating that Python is ready for us to give it a command. These commands are called *statements*

python

```
>>> print "Hello World"
Hello World
>>> print 2+3
5
>>> print "2+3=", 2+3
2+3= 5
>>>
```

python3

```
>>> print("Hello World")
Hello World
>>> print(2+3)
5
>>> print("2+3=", 2+3)
2+3= 5
>>>
```



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™



# OK, BEFORE WE GET TOO DEEP... LET'S GET SOME HELP FROM DR. CHARLES R. SEVERANCE

- We are going to use a few resources on the internet...
- Bookmark and remember a few sites...
  - SVCTE Mechatronics Python Resource link
    - [Python Resources](#)
- Python 4 Everybody - <https://www.py4e.com/>



## Python for Everybody

Hello and welcome to my site where you can work through my course materials related to my free [Python for Everybody](#) text book. You can take this course for a certificate as the [Python for Everybody Specialization](#) on Coursera.

You can use this web site many different ways:

You browse my videos and course materials under [Lessons](#). The materials I have developed for this class are all provided with a Creative Commons license so you can download or link to them to incorporate them into your own teaching if you like.

If you [log in](#) to this site it is as if you have joined a free, global open and online course. You have a grade book, autograded assignments, a discussion forum, and can earn badges for your efforts.

We take your privacy seriously on this site, you can review our [Privacy Policy](#) for more details.

If you want to use these materials in your own classes you can download or link to the artifacts on this site, [export the course material](#) as an IMS Common Cartridge®, or apply for an IMS Learning Tools Interoperability® (LTI®) [key and secret](#) to launch the autograded from your LMS.

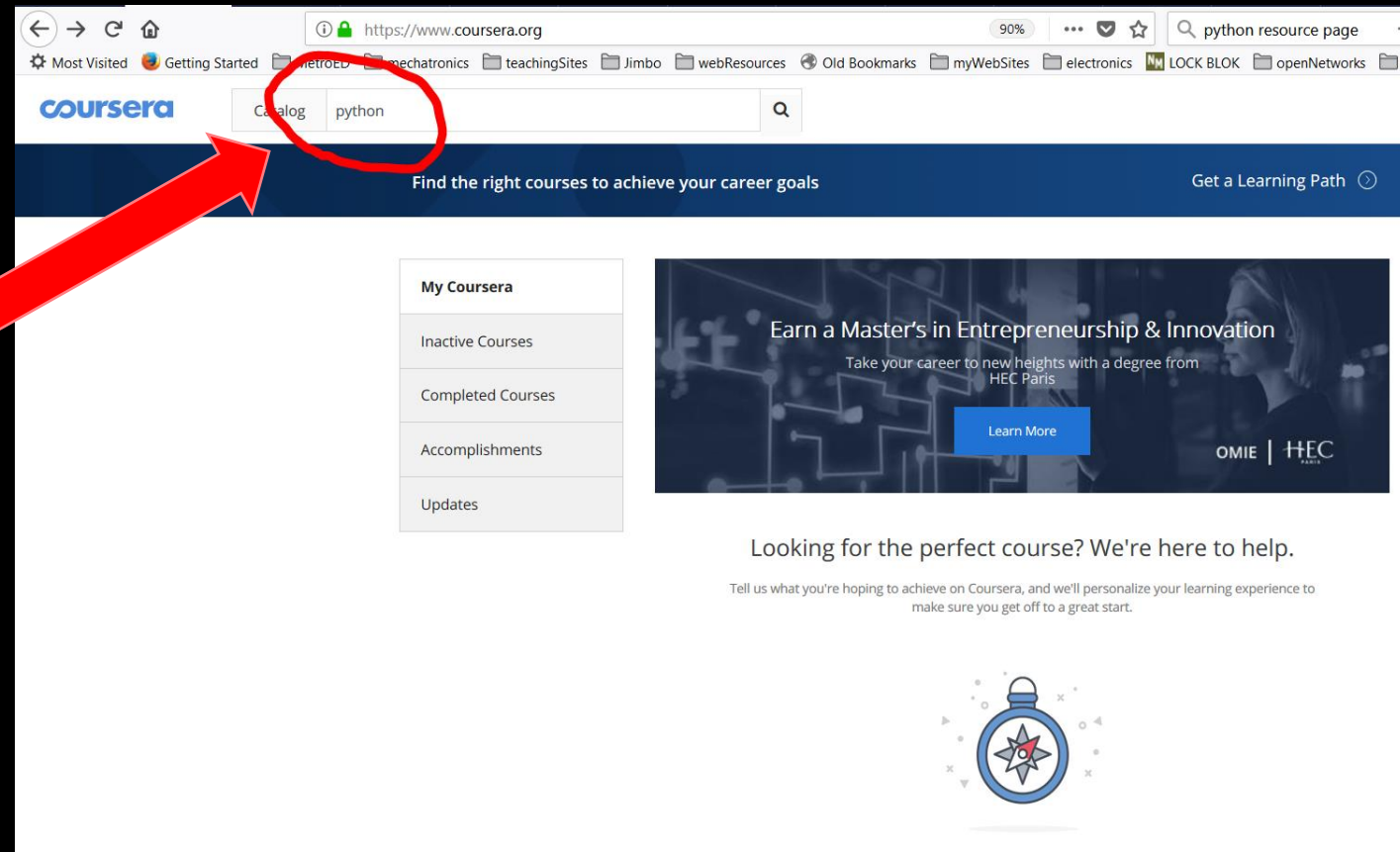
The code for this site including the autograded, slides, and course content is all available on [GitHub](#). That means you could make your own copy of the course site, publish it and remix it any way you like. Even more exciting, you could translate the entire site (course) into your own language and publish it. I have provided some [instructions on how to translate this course](#) in my GitHub repository.

This site uses [Tsugi](#) framework to embed a learning management system into this site and provide the autograded. If you are interested in collaborating to build these kinds of sites for yourself, please see the [tsugi.org](#) website and/or contact me.



# REGISTER FOR A COURSERA CLASS

- Go to <https://www.coursera.org/>



Python



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# PROGRAMMING FOR EVERYBODY

**coursera** Catalog python

Find the right courses to achieve your career goals Get a Learning Path

You searched for **python**. 153 matches

Active filters: **English**

### Course Languages

<input type="checkbox"/> English	143
<input type="checkbox"/> French	5
<input type="checkbox"/> Spanish	2

[Show More](#)





### Subtitle Languages

<input checked="" type="checkbox"/> English	153
<input type="checkbox"/> Chinese (Simplified)	22
<input type="checkbox"/> Spanish	21

### All Topics

<input type="checkbox"/> Computer Science	101
<input type="checkbox"/> Data Science	61
<input type="checkbox"/> Physical Science and Engineering	15

### Courses and Specializations

-  **Python for Everybody**  
5-course Specialization · University of Michigan
-  **Programming for Everybody (Getting Started with Python)**  
University of Michigan
-  **Python Data Structures**  
University of Michigan
-  **Applied Data Science with Python**  
5-course Specialization · University of Michigan

- Go to <https://www.codeschool.com/courses/try-python>

The screenshot shows the top navigation bar of the Code School website. The logo is on the left, followed by links for 'Learn', 'Pricing', and 'Business'. A 'Create' button is on the right. The main content area features a large white box with the text 'PYTHON Try Python' and a description: 'Explore the basics of Python and learn what it means to store and manipulate numbers and words as well as make decisions with your program.' Below this are two buttons: 'Play Course for Free' and 'Add bookmark'. At the bottom of the white box, there is a 'Course Overview' section and a 'Watch Course Intro' button. The footer of the white box shows 'LEVEL 1 | FREE LEVEL | 2 Videos | 6 Challenges'.

The screenshot shows the 'Sign In' page of the Code School website. The logo is on the left. The main heading is 'Sign In' with a link to 'Create a free account' for users who don't have an account. There are two input fields for 'Email' and 'Password'. Below the password field is a link for 'Forgot your password?'. A large blue 'Sign In' button is centered. Below the button are social media icons for 'Sign in with:'. At the bottom, there is a copyright notice: '© 2017 Code School LLC. Privacy Policy Terms of Use'.

# SUMMARY

- Intro to Python
- Checked that Python is installed
- Can run a Python `print("Hello World")` statement
- Registered for PY4E
- Registered for CodeSchool.com
- Have a bookmark for where to find resources



# STEAM CLOWN™ PRODUCTIONS

# APPENDIX



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™

# APPENDIX A: LICENSE & ATTRIBUTION

- These slides are an adaption, primarily from Dr. Charles R. Severance's Python for Everybody class
- <https://www.py4e.com/>
- Additionally this interpretation is primarily the Intellectual Property of Jim Burnham, Top STEAM Clown, at [www.STEAMClown.org](http://www.STEAMClown.org) contact @ [topClown@steamclown.org](mailto:topClown@steamclown.org)
- This presentation and content is distributed under the Creative Commons License CC-by-nc-sa-3.0
- My best attempt to properly attribute, or reference any other sources or work I have used are listed in Appendix B



## Under the following terms:



**Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



**NonCommercial** — You may not use the material for commercial purposes.



**ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

**No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**

© Copyright 2017 STEAM Clown™



# APPENDIX B: ATTRIBUTION FOR SOURCES USED

- Charles R. Severance slides can be found on the <https://www.py4e.com/> site are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.
  - Initial Development: Charles Severance, University of Michigan School of Information
  - Modifications and Adaptions by Jim Burnham, Top Clown @ [www.steamclown.org](http://www.steamclown.org)
- Another great Python site is Barbara Saurette AKA [mechanicalgirl](#) and her [Github site](#)
- Additionally used some content from slide deck from Mr Ganesh Bhosale found <https://github.com/gdbhosale/python-rpi/blob/master/python1.pdf>





# STEAM CLOWN™ PRODUCTIONS

# REFERENCE SLIDES



**STEAM CLOWN™**  
& **Squeaky Hinge**  
**PRODUCTIONS**  
© Copyright 2017 STEAM Clown™



