Thursday, February 1st 2018

Any New Members?

Charity

- Casa de Peregrinos
- El Caldito Soup Kitchen

Events

- PI Day
  - March 14 (3/14)
  - Oh what to do, what to do (hint hint)
- A&S Council Meetings
  - Need to attend meetings to be recognized
  - Actually interesting and informative, worth going to
  - **Hardman Jacobs Room 125 @ 5:30, 2hrs bi-weekly**
- Future Meeting Topics
  - Serialization with JSON, YAML, XML, and Pickle Files
  - Python!!!!

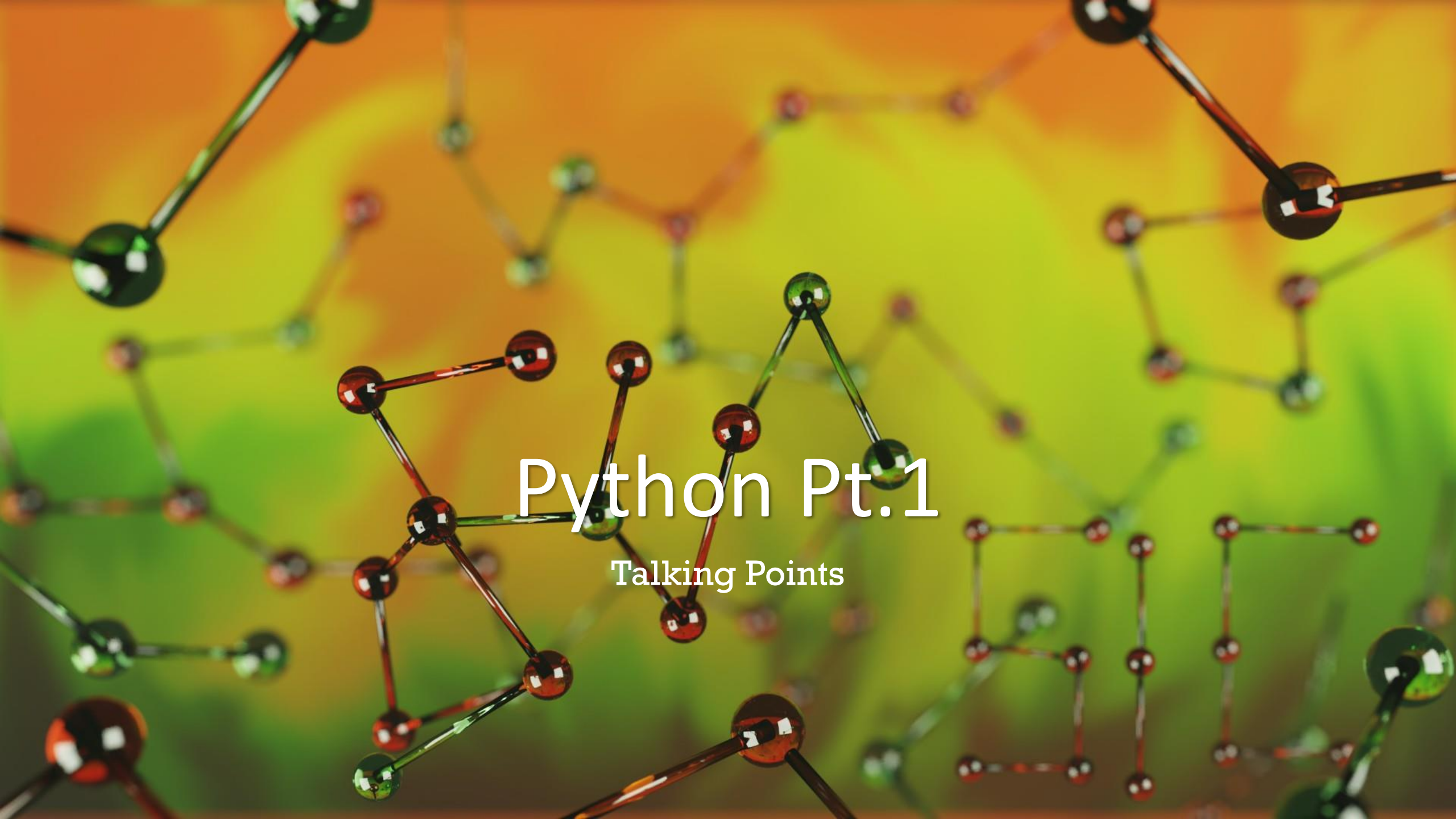
Nerd Moment

- Elon Musk is Crazy
  - Founder/Co-founder of PayPal, Tesla, The Boring Company (The bore holes)
  - Taking pre-orders for a Boring Company Flamethrower: https://youtu.be/2er6liWIufA
- 3Blue1Brown Covers the Fourier Transform
  - Previously covered Linear Algebra, Calculus, Neural Networks…
  - https://youtu.be/spUNpyF58BY

Nerd Moment

- L1 Techs: Car Manufactures tracking your location:
  - https://www.youtube.com/watch?v=dVVaMnlYSoo&t=2114s

- Fake YouTube ads run bitcoin mining scripts
  - https://diegobetto.com/crypto-miner-coinhive-youtube-ads/


Python Pt.1
Talking Points

Why Python

- In the modern world everyone needs to know some programming
- If there is one language that programmers and non-programmers should know, it's Python
- It's a modern flexible, easy to learn and hard to screw-up language, used in everything from web-services to machine learning to generating 3D content and even making you're homework easier

The Good

- Simple, consistent syntax
- Packages of useful code
  - + easy to get
- Popularity = Support Network
  - StackOverflow.com
- Scripting = Flexible
  - + easy to experiment with
  - + easy to bend the rules
- Can be found everywhere

The Bad

- Python is a scripting language
  - Easily can be slower than compiled languages
- Some people **still** use Python 2.7
  - ALWAYS use Python 3
- Poor type safety
- Most syntax is obvious, much of it is not

Sidebar: Types of Languages

- Assembly Languages:
  - Translate to CPU instructions, a.k.a., binary
- Compiled Languages:
  - Compiler will read code and translate it to an executable
  - C, C++, Fortran, Ada…
  - Special Cases: C#, Java
- Interpreted Languages
  - Parsed at runtime, and run by an existing executable runtime
  - Python, JavaScript

Lets Talk Python

- Python can be slow or fast, depending on how it's used
  - Many libraries run native, pre-compiled, optimized code
    - These libraries aren't 100% written with python
  - Parsing still slow, but the actual execution is fast!
- Python has tons of libraries, which are easy to get
  - PIP (or Conda) can be used to get more by name, without hassle
  - The is a package for everything (graphics, physics, neural nets…)
- Many programs have built-in Python interpreters.
  - Some are pre-wired with application specific packages
  - E.g. Maya, Blender

Python 2.7 vs 3.6

- There is a lot of older Python 2 code
  - Some older programmers prefer it
  - Iron Python (.NET version) is still based on Python 2.7
  - Many Python 3 features can still be used with special directives
- Python 3.6 should be considered the standard for all NEW python scripts and applications
  - Most embedded interpreters will support it
- Syntax wise, the two version are still very similar.
  Very little will not carry over

Anaconda

- Anaconda is special open-sourced science oriented distribution of Python (and R)
- Installing all of packages takes a while, but opens up a consistent environment for Machine Learning, Image Processing, Graphing, etc.
  - These means getting more done, faster, with less code
  - If you want to experiment with machine learning, start here
- You can use Conda to get more packages and stay updated

Developing in Python

- You can use a lot of different IDEs or even plain text editors to write Python
  - IDE = Integrated Development Environment = Time Saver
  - Note: for environments like Blender, you probably should use the built-in interpreter to the program for creature comforts like auto-complete

- If you plan on playing around with Python outside of specific environments, I recommend using PyCharm (tis free)

Python Enhancement Proposal (PEP)

- PEP is a set of Python stylistic and functional design standards that all Python programmers should adhere too.
- If you use a decent IDE, like PyCharm, you'll be notified whenever you don't adhere to the standards, even though your code still works
- PEP compliant code is pleasant to read and consistent, take time to clean up your code to keep it looking good

Next Time: Actual Python Coding

- Giving this is an intro meeting, I don't want to go too in depth
  - Pizza + potential keyboard usage is bad
- Pt.2 Topics
  - Programming For Beginners (For those not in CS)
  - Getting Python
  - Python Syntax
  - PyCharm
  - Blender Python (Maybe)
- Future Topics?
  - Machine Learning w/ Keras? OpenCV Image Processing?