APS106

UNIVERSITY OF TORONTO

introduction.

Week 1 | Lecture 1 (1.1)

if nothing else, write #cleancode

This Week's Content

- **Lecture 1.1**
  - Part 1 - Introduction
  - Part 2 - Variables, Expressions, and Operators
  - Reading: Chapters 1, 2

Teaching Team

Ben
Instructor

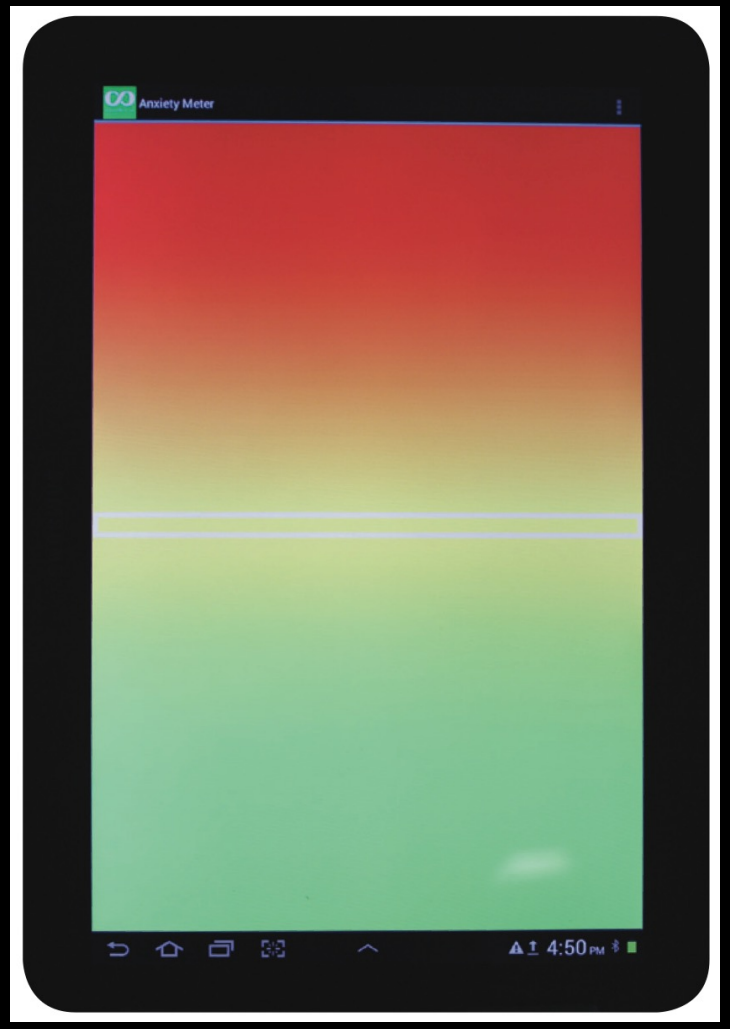
Craig
Tutorial TA

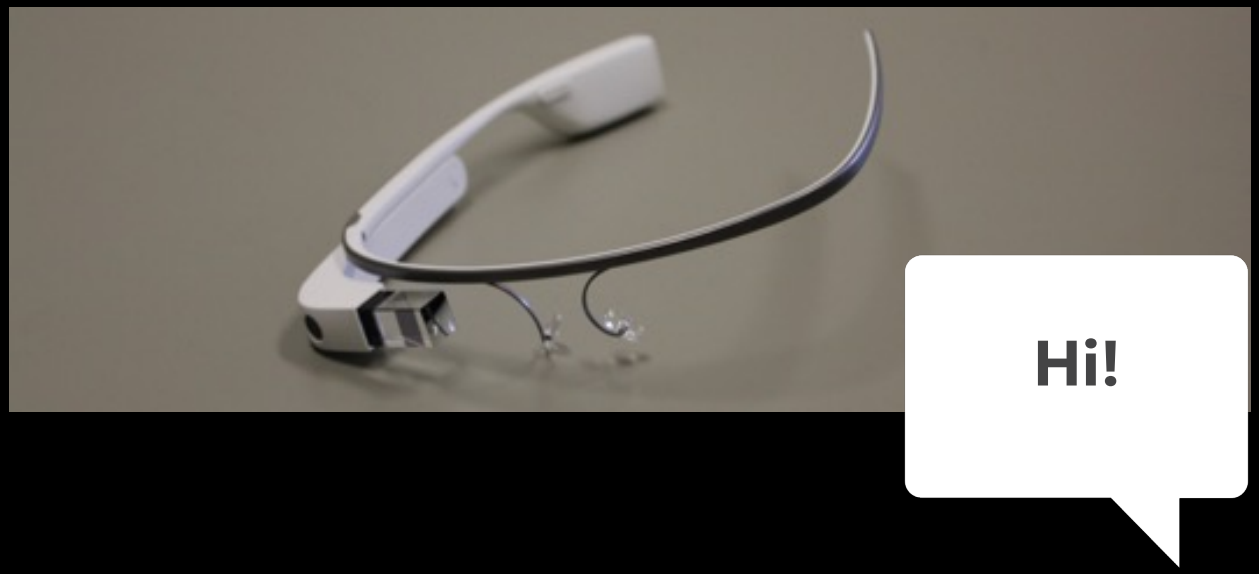
Behrang
Lab TA

Ben

- Bachelor of Electrical & Biomedical Engineering @ McMaster University 2015
  - Realized the power of programming (and how it can be used for good, not evil)
- Master of Biomedical Engineering @ University of Toronto 2020
  - Hold up… people will pay me for ‘playing’ on my computer?
- In between obtaining degrees I spent time programming a few different projects…

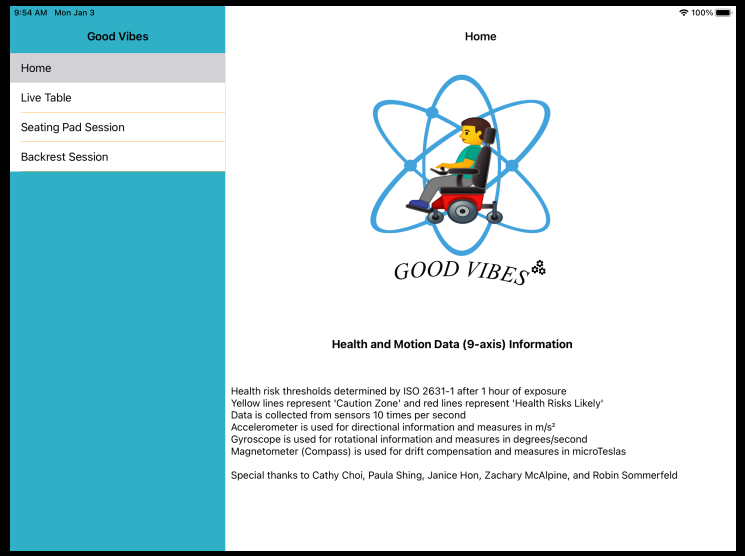
Projects for children with autism spectrum disorder

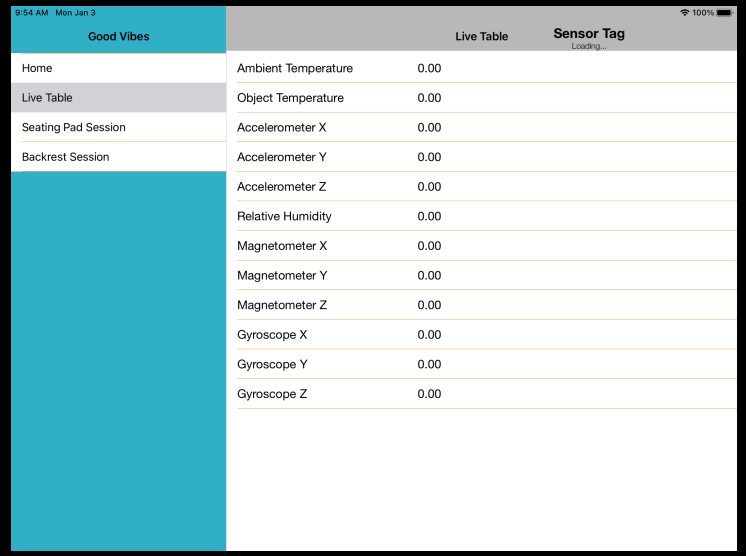


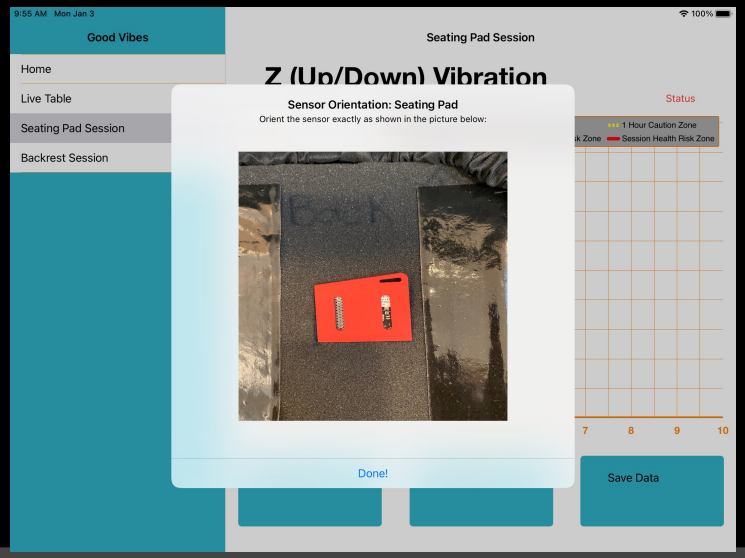


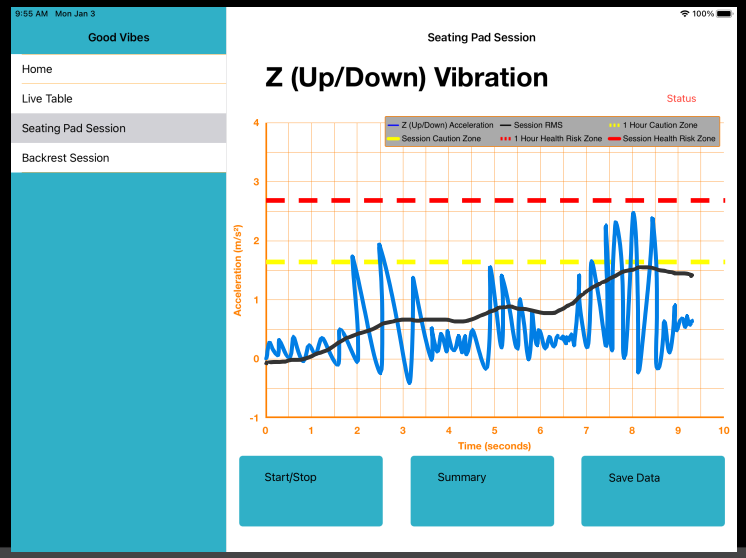
Good Vibes

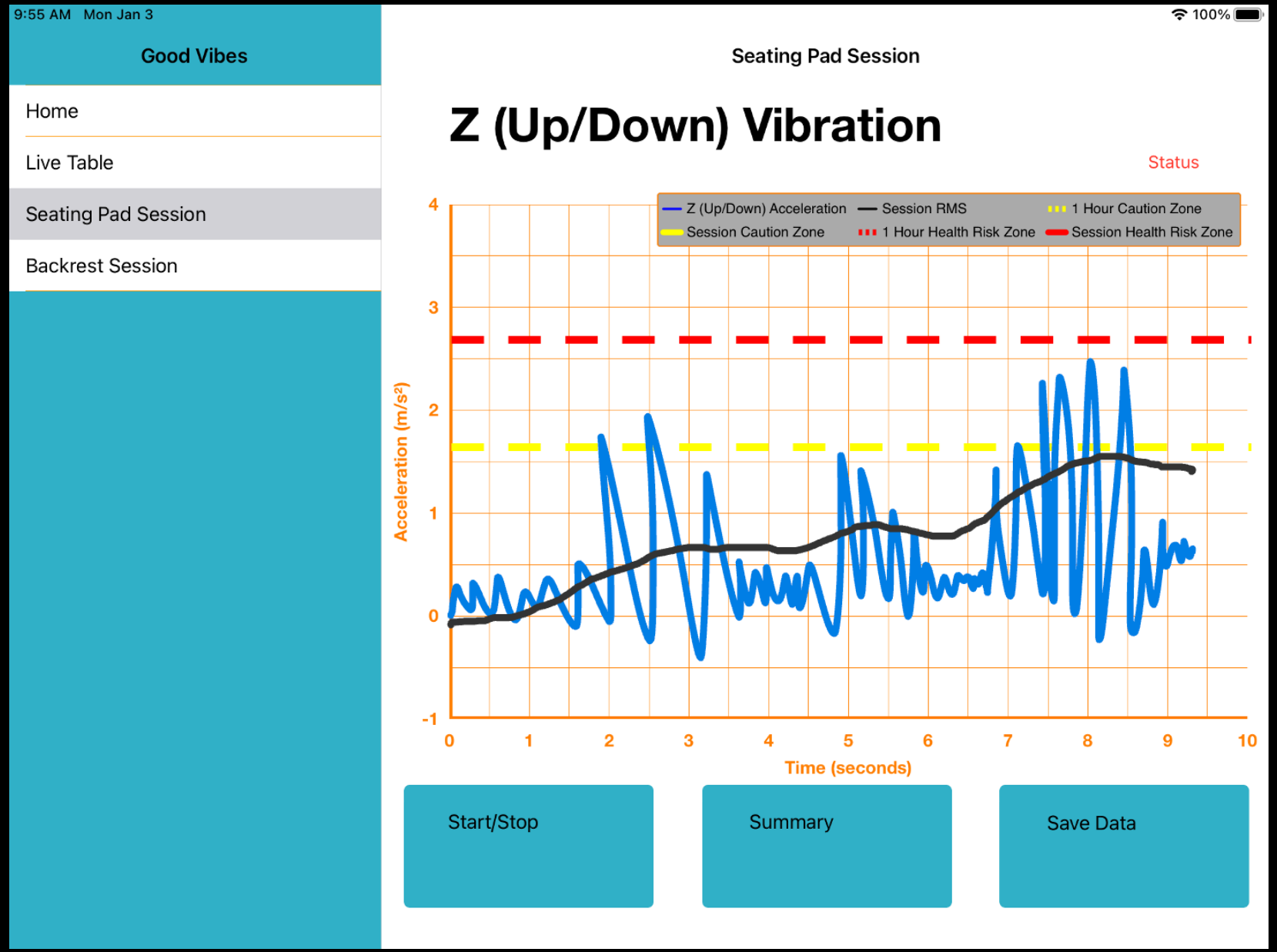
A wheelchair vibration measurement tool







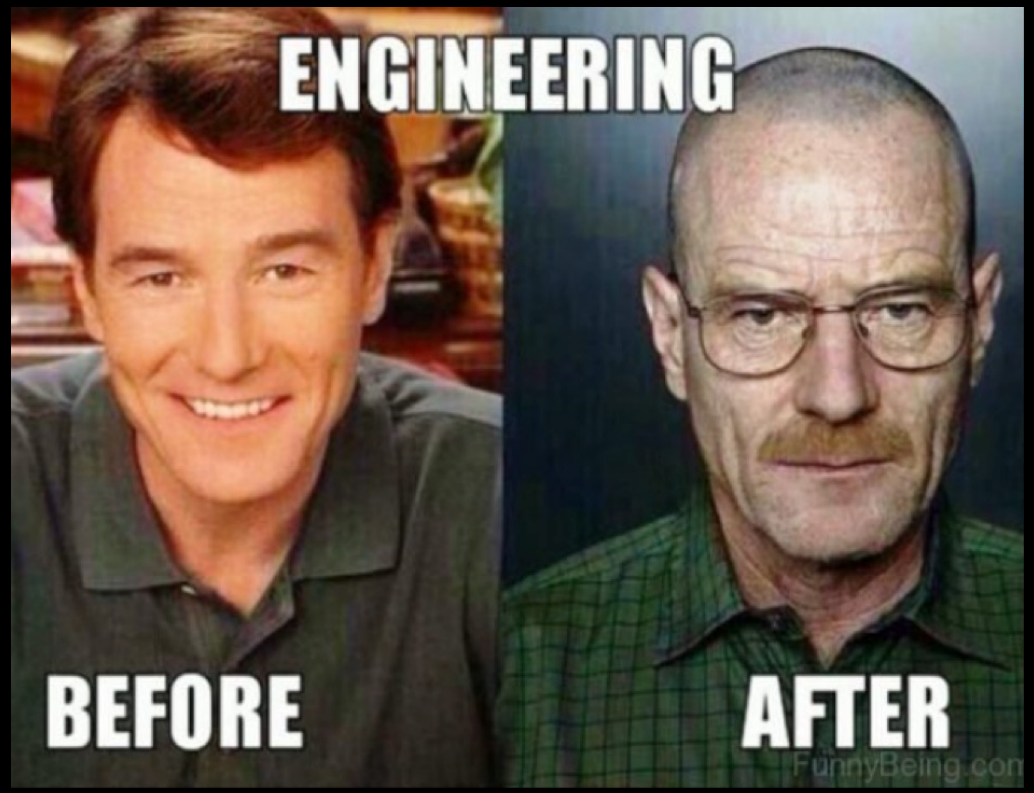



9:55 AM Mon Jan 3
100%
Good Vibes
Home
Live Table
Seating Pad Session
Backrest Session
Seating Pad Session
Z (Up/Down) Vibration
Status
Z (Up/Down) Acceleration
Session RMS
1 Hour Caution Zone
Session Caution Zone
1 Hour Health Risk Zone
Session Health Risk Zone
Acceleration (m/s²)
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)
Start/Stop
Summary
Save Data

More about Ben…

University is HARD

- Engineering at university is harder.
- "**Everyone you meet is fighting a hard battle.**"



Being respectful

- Talking
- Cell Phones
- Eating
- Being Late for Lectures

Being respectful

- Talking
- Cell Phones
- Eating
- Being Late for Lectures

Ask Questions!

- Ask questions as much as possible.
- Ask **as soon as** you feel confused!
- Ask each other questions in-person or on Piazza and help if you think you know the answer
- Never forget - if you have a question, there's always other people in the room wondering the same thing. They will be thankful you asked it!

Course Information

- **Quercus:**
  - APS106H1 S LEC0101 (Website contains all information).
- **Textbook:**
  - ***Programming in Python 3***.
  - A link to the UofT bookstore where you can purchase the textbook can be found on the Syllabus page.

Grading

Petitions: First-year Office.

Labs (5 in total)	total: 15%
Reflections (6 in total)	total: 5%
Midterm (in-lecture)	total: 30%
Final Exam	total: 50%

Lecture Schedule

Section	Monday	Tuesday	Wednesday	Thursday	Friday
LEC01		9:00-11:00		9:00-11:00	9:00-11:00

- Lectures will be in person in **GB221 (for now…)**.
- If we need to move online, Microsoft Teams links for lectures will be found on Quercus.
- Lecture content can be found on Quercus.

Tutorial Schedule

- Tutorials will be in person in the same room starting this week
- Links for tutorial content will be found on Quercus.

Week	Tut Content	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1	Tutorial 1		**TUT0101** 11:00-1:00 Location: TBD-GB 119?			

Lab Schedule

- Labs will be in person in GB 144/150 starting this week
- Links for lab content will be found on Quercus.

Week	Lab Content	Released	Due
Week 1 (May 6 - May 10)	**Lab 1**	Tuesday May 7	Tuesday May 14 @ 11:00 **AM**
Week 2 (May 13 - May 17)	**Lab 2**	Friday May 10	Friday May 17 @ 23:00
Week 3 (May 20 - May 24)	**Lab 3**	Friday May 17	Friday May 27 @ 23:00

Click here for content.

TENTATIVE

Reflections

- There are 6 weekly reflections throughout the term.
- Reflections are worth a total of 5% of your final grade.
- Full marks for successful completion of all reflections.
- Reflections are used to take the pulse of students in the course.
- Insights from reflections are used to iteratively improve the course week-by-week.
- Reflections are there to help you, so please answer honestly.

Reflections

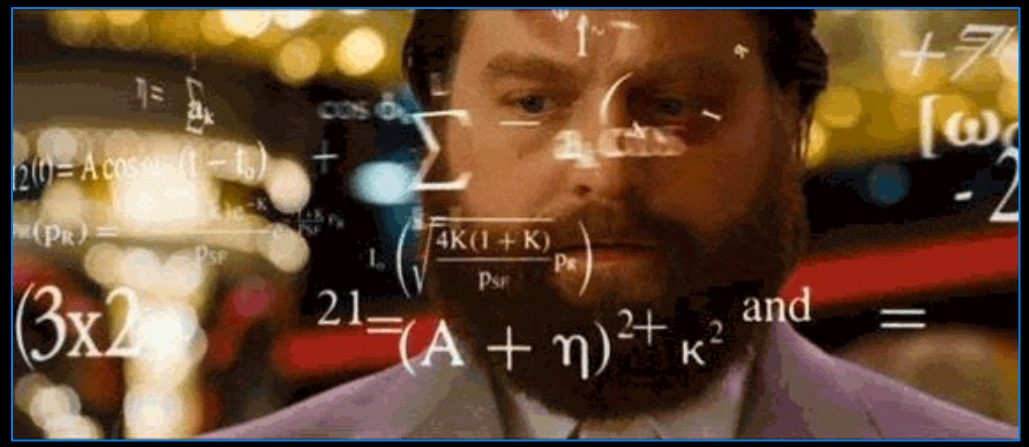
- Reflection include question like:
  - Did you attend lecture this week? If not, why?
  - Do you understand the topics covered this week?
  - Did you attend tutorial this week?
  - Which lecture section did you attend?
  - What will this code output?
  - Write one word to describe how you feel about the course.

Reflections Schedule

Week	Reflection	Released	Due
Week 1 (May 6 - May 10)	Reflection 1	Friday May 10 @ 11:00	Monday May 13 @ 17:00
Week 2 (May 13 - May 17)	Reflection 2	Friday May 17 @ 11:00	Monday May 20 @ 17:00
Week 3 (May 20 - May 24)	Reflection 3	Friday May 24 @ 11:00	Monday May 27 @ 17:00
Continued…			

Click here for content.

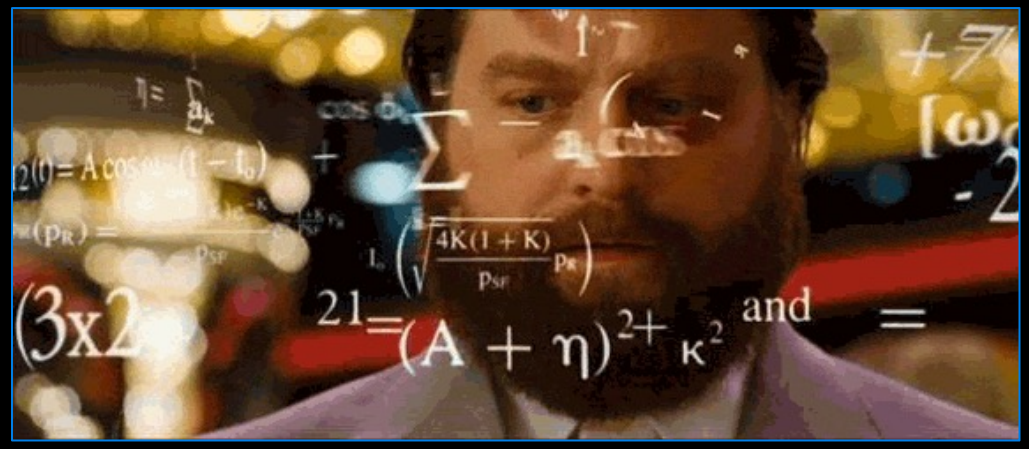
Tips for Success



- Do not wait until the last minute.
- Programming is a Language
- Work Together (except during the test/exam):
- Work Smart (effort doesn't necessarily equal progress).
- Time Management.
- Know when (and how) to ask for help.

Tips for Success

- **80%** of your final grade is based on two term tests and an exam.
- For these tests and exams, you'll need to write code.
- You will not be able to achieve a high mark by reading the textbook and lecture notes. You need to PRACTICE, PRACTICE, PRACTICE!
- How can you practice writing code?
  - **Lecture Breakout Sessions.**
  - **Labs (think of this as studying).**
  - **Practice Problems.**



Writing A Lot Of Code During The Term

=

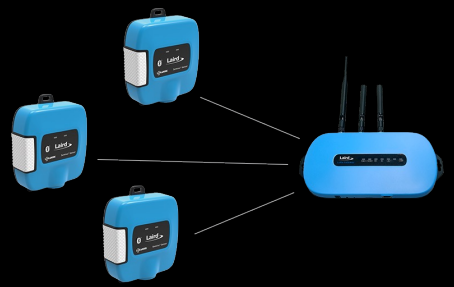
Higher Grade

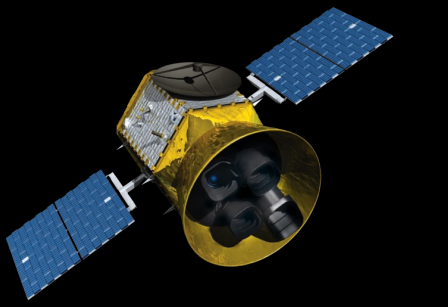
What is this course all about?

- Learn the **fundamentals** of programming.
- **Programming as a tool** used in engineering.
- **Engineering design** through programming.

Why Programming?

StartUp HERE





Why Programming?

- Computers are everywhere:
  - Learn to interact with them.
  - Simplifies our jobs.
- Entrepreneurial opportunities ($$$).
- Necessary for engineering jobs.
- Research and development.
- Because it's fun!

KORE GEOSYSTEMS

$1,000,000 Investment.

What is Programming?

- A way of telling a computer what to do.
- A computer can't infer (…yet).
  - Need to tell a computer every single step it needs to do in a language it can understand.
  - How would you request an egg for breakfast to a chef and to a computer/robot?

- **To a Chef**
  1. Sunny-side up, please!

- **To a Computer**
  1. "Turn on stove"
  2. "Take out pan"
  3. "Take one egg out of fridge"
  4. "Crack egg"
  5. "Pour egg into pan"
  6. "Wait 5 minutes"

How to Program a Computer.

Programmer

English

Pseudocode

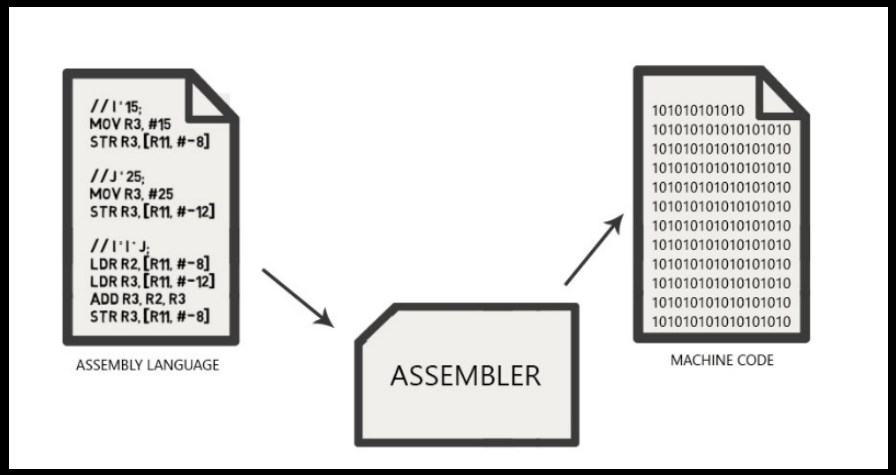
Programing Language

Compiler

Assembly Code

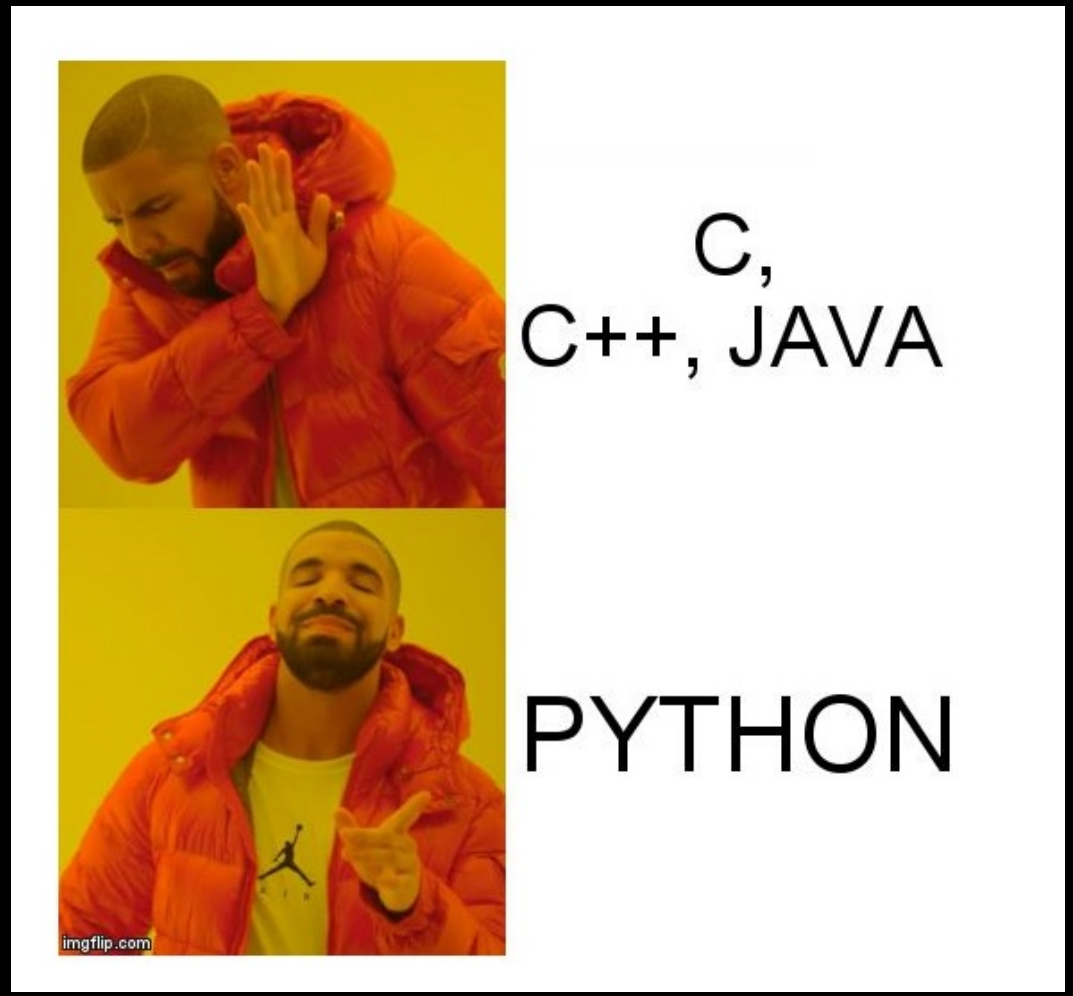
Assembler

Machine Code



Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Java

Lower-Level

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

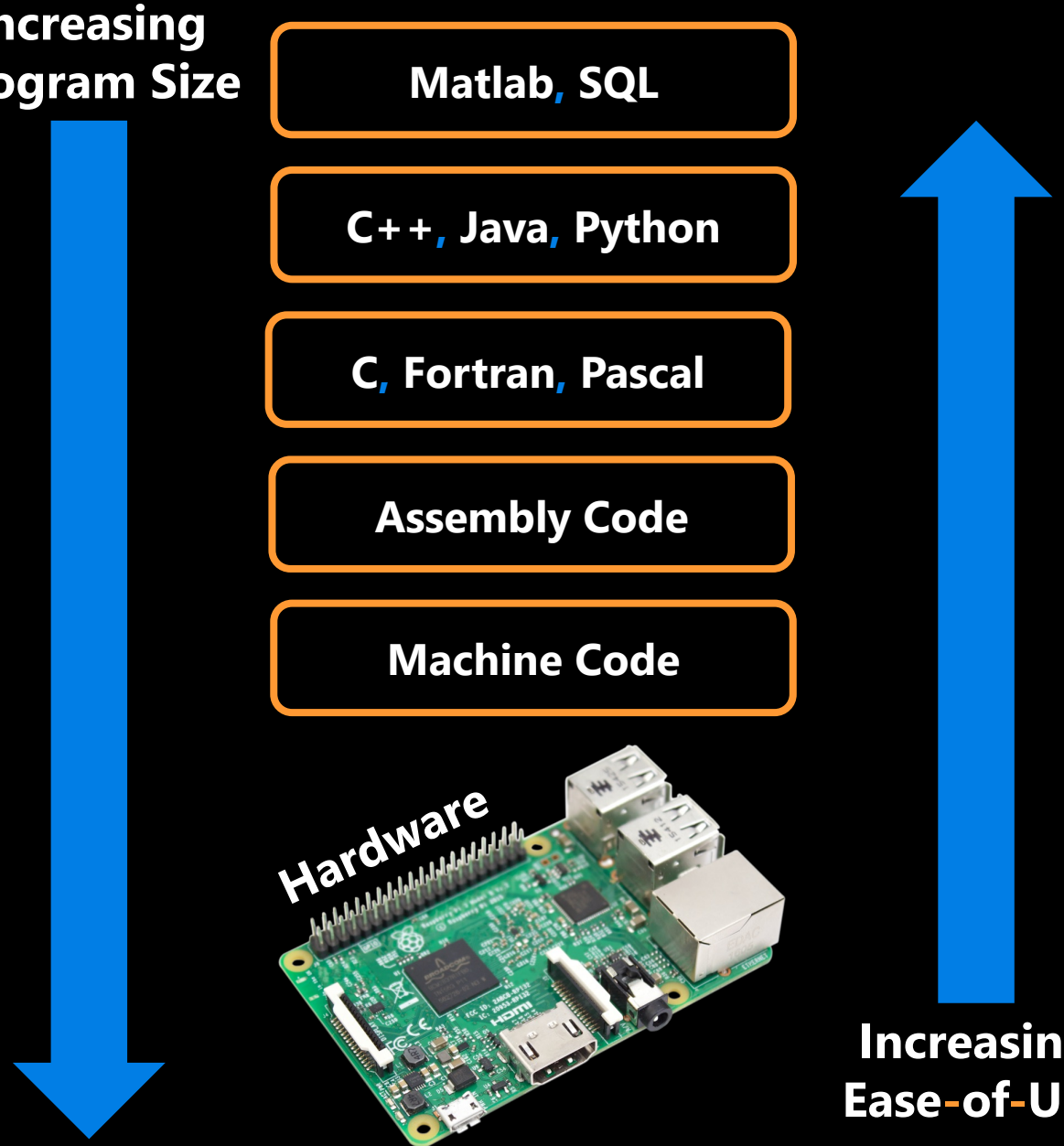
python

Higher-Level

```
print("Hello World!")
```

Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

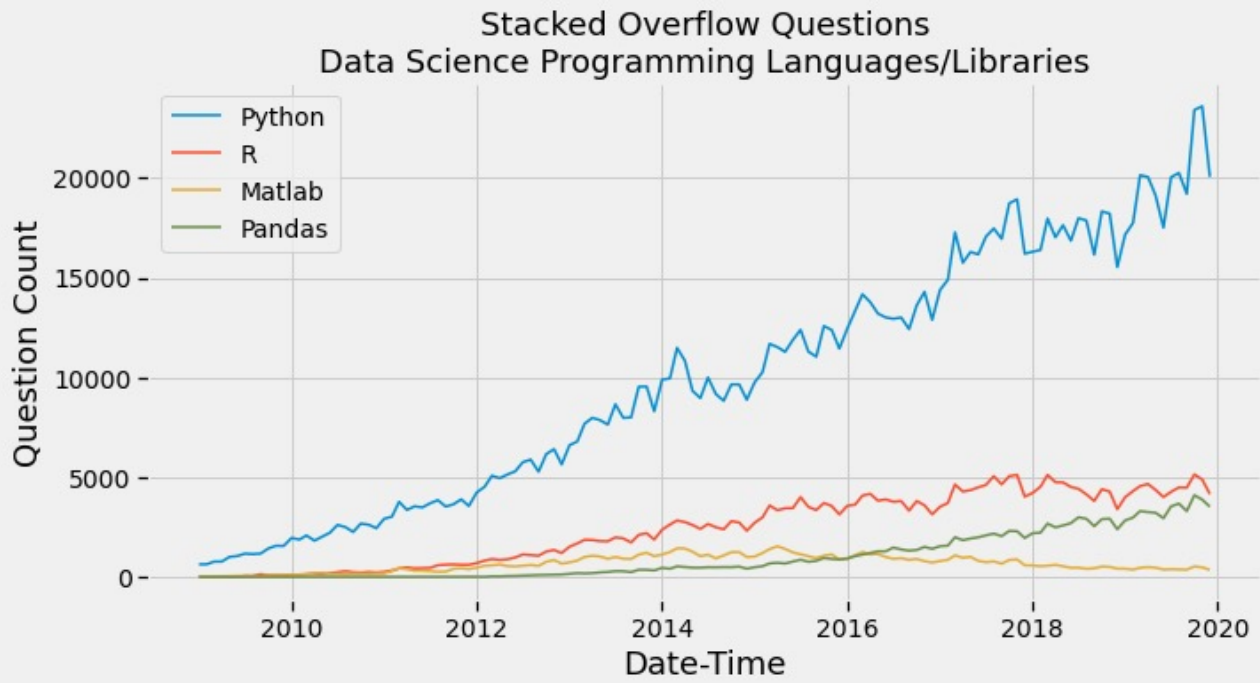
python

```
if x > 10:
    print("x is greater than 10")
```

You don't know Python.
Can you figure out what this code does?

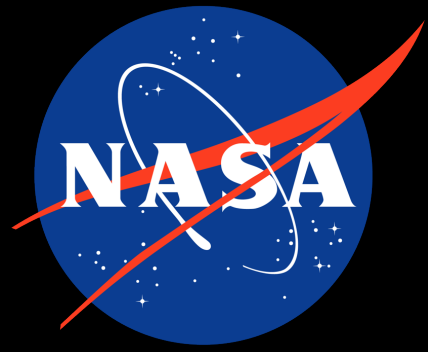
Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Why Engineers?

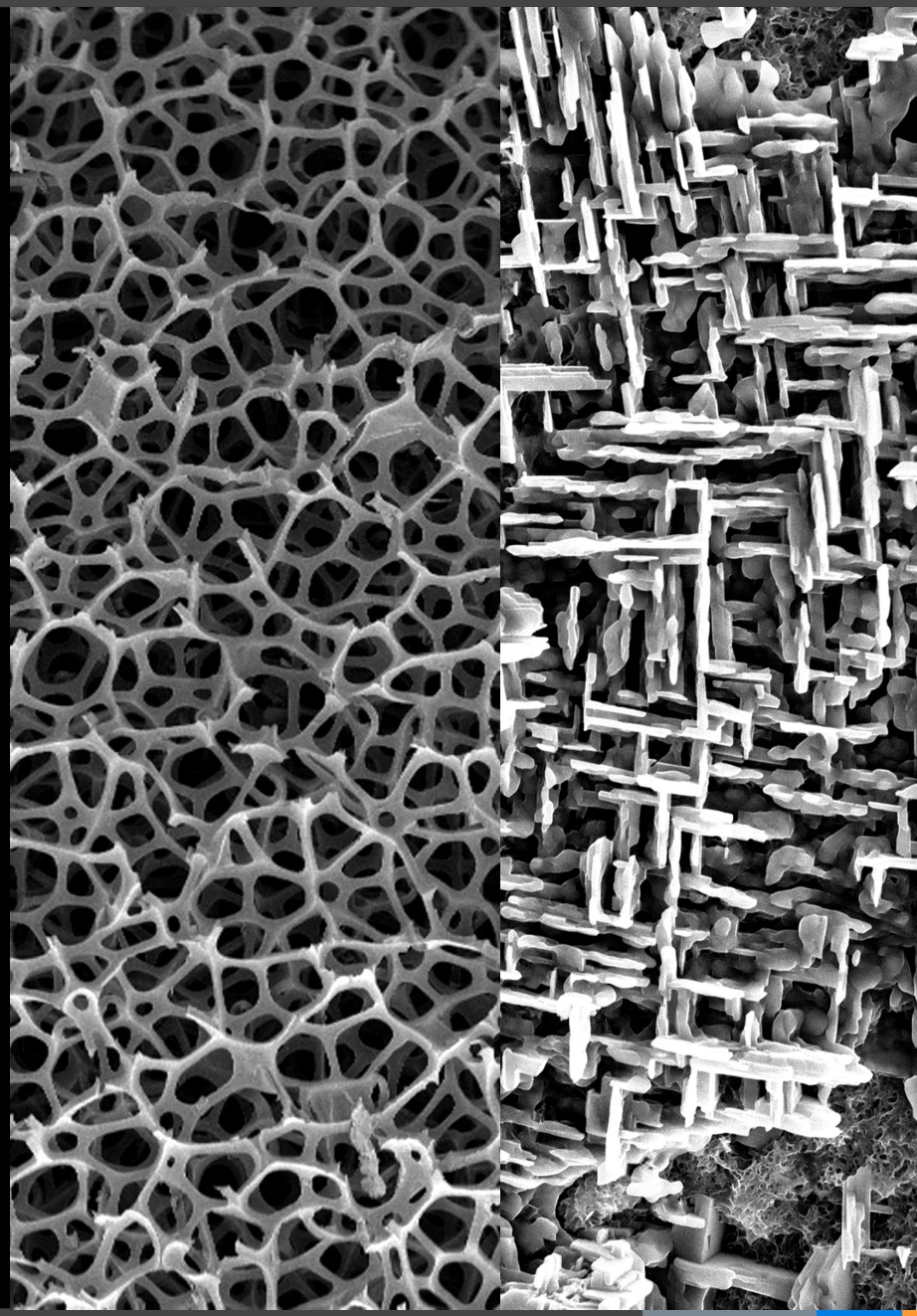
- Ok, so we've explained why coding is important and why we're learning to code in Python.
- **BUT**, you're all different types of engineers, right?
  - Mechanical
  - Materials
  - Civil
  - Chemical
  - Mineral
- Why do these kinds of engineers need to know how to code?

Mechanical Engineers

- Design Airplanes
- Micro Fluid Systems
- Automotive Engines
- Hydraulic Dams

Materials Engineers

- Biomaterials
- Design of Materials
- Manufacturing with Materials
- Sustainable Materials Processing

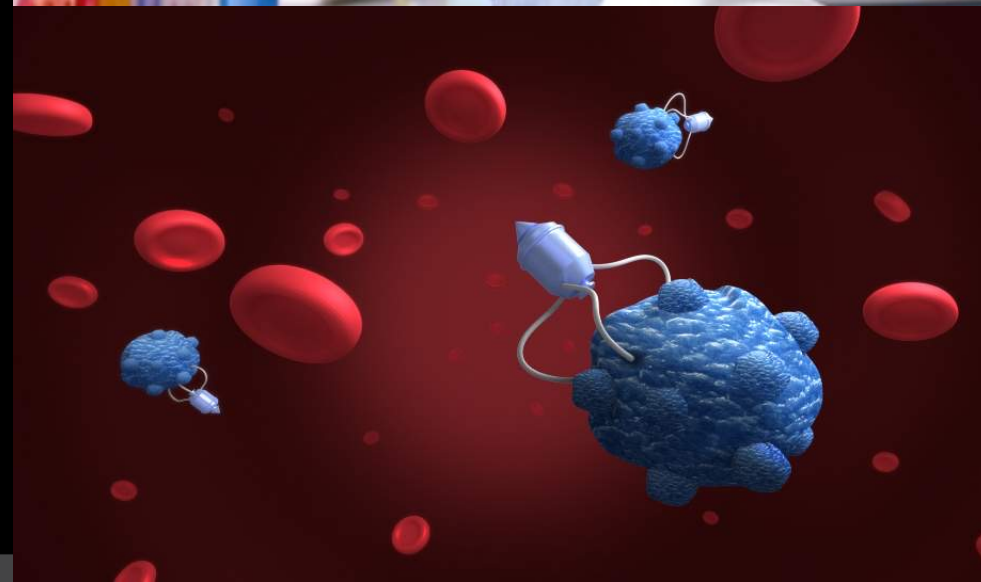


Civil Engineers

- Design Buildings
- Test Concrete
- Manage Water Supply System
- Design Bridges

Chemical Engineers

- Chemical Manufacturing Processes
- Petroleum Extraction
- Nanotechnology
- Manufacturing Computer Components

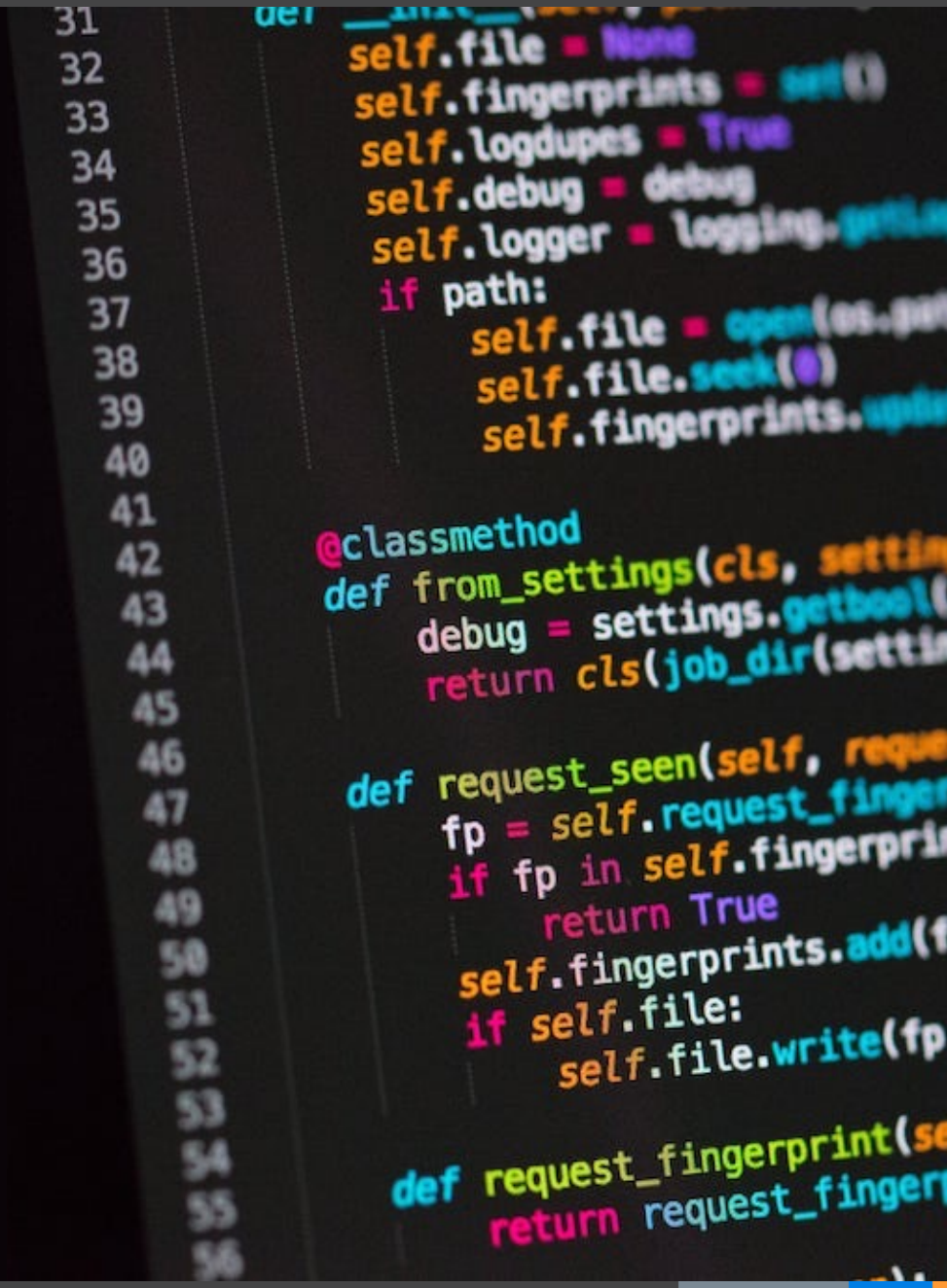


Mineral Engineers

- Mine Design
- Mineral Processing
- Mineral Exploration
- Mining Finance
- Mineral Extraction

Why Engineers?

- Again, why do engineers need to know how to code?
- In the 20th century, engineers didn't need to know how to code.
- In the 21th century, coding is an **essential skill** and any engineer without solid coding abilities will be at a major disadvantage.

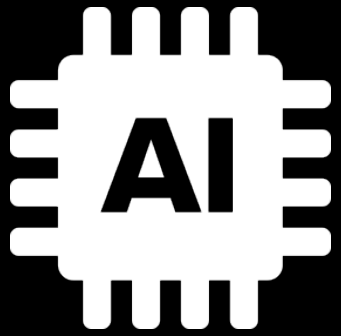


Why Engineers?

- Five technology trends have changed the game for engineers.

The Cloud

Cheap Sensors



Artificial Intelligence

Big Data

The Internet Of Things

Mechanical Engineers

Realtime Diagnostics & Tuning

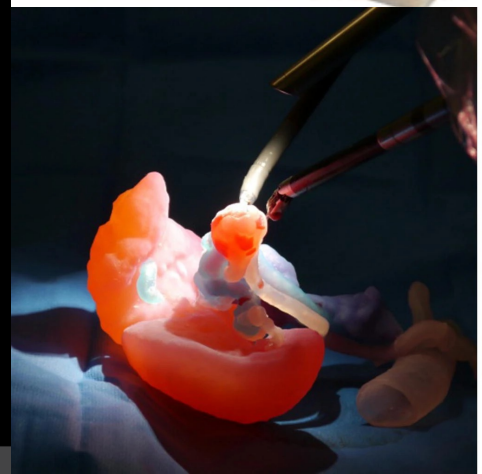
CNET Tech Money Home Wellness Home Internet

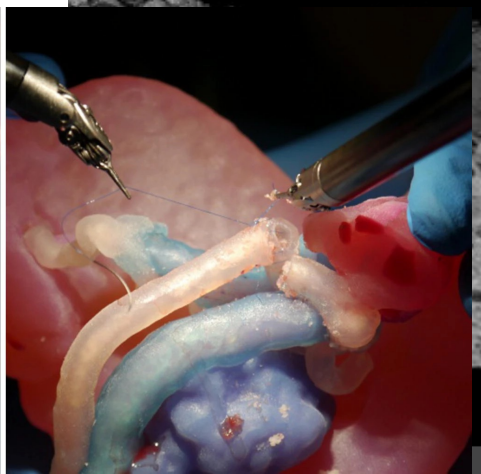
Cars > Auto Tech

Tesla can tune your Model S or X's suspension via software update

Materials Engineers

Digital Image Analysis of Material Microstructure



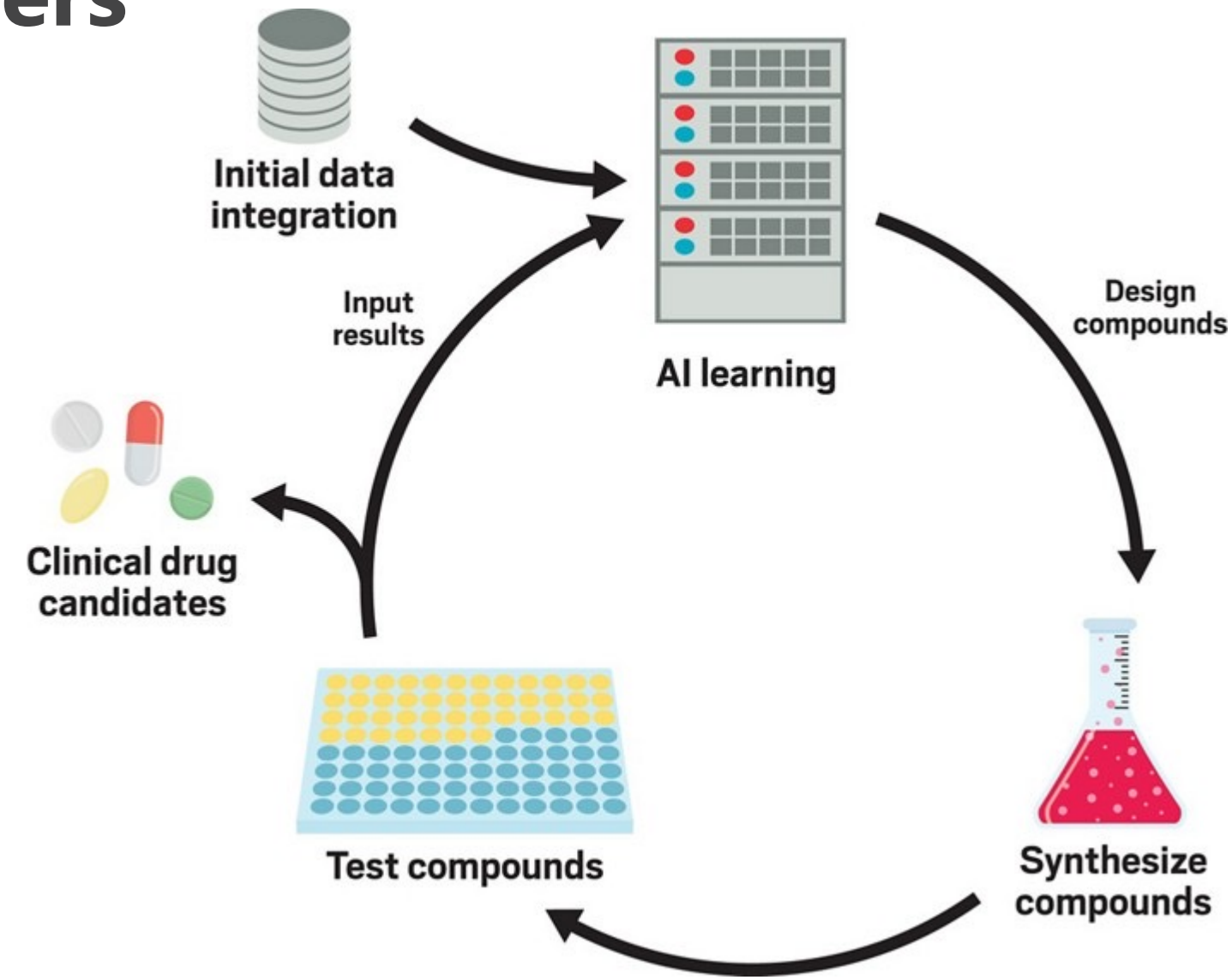


Civil Engineers

Urban Transportation Disrupted

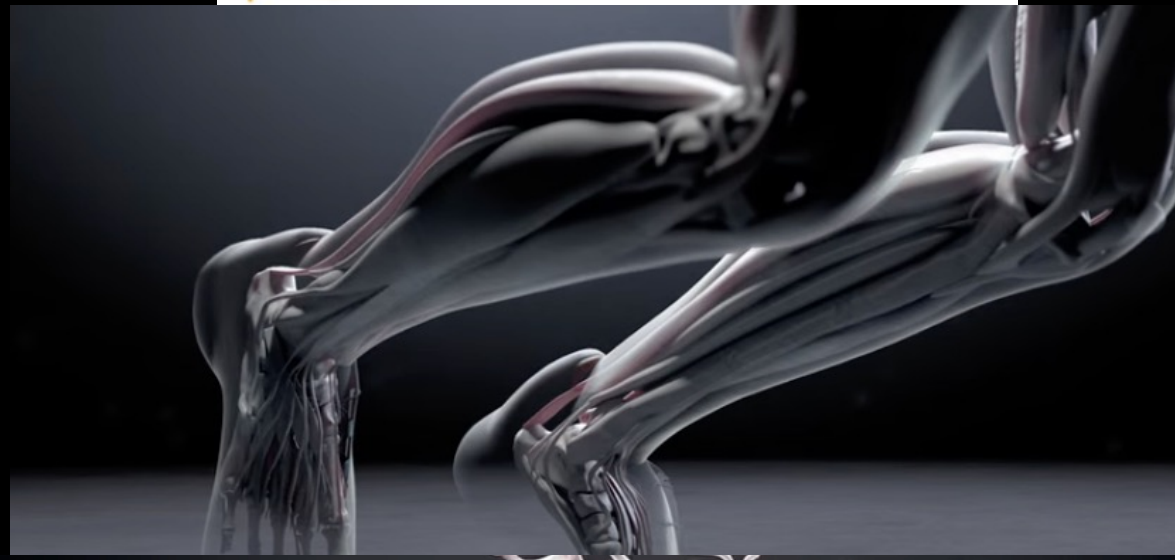
Chemical Engineers

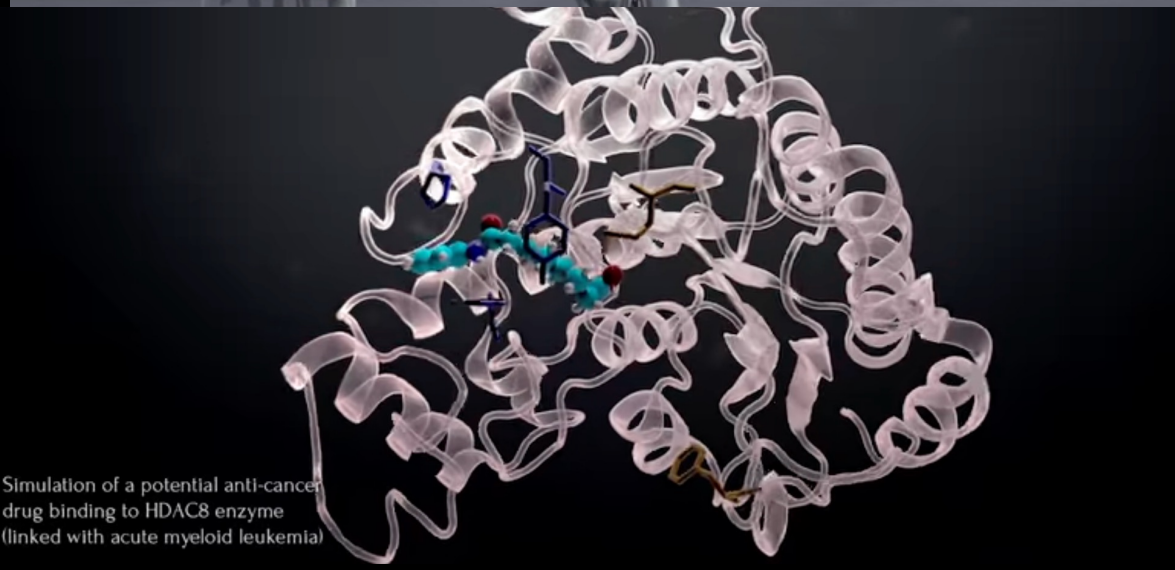
Synthesizing new compounds



Sci-fi Inspiration

CompBioMed





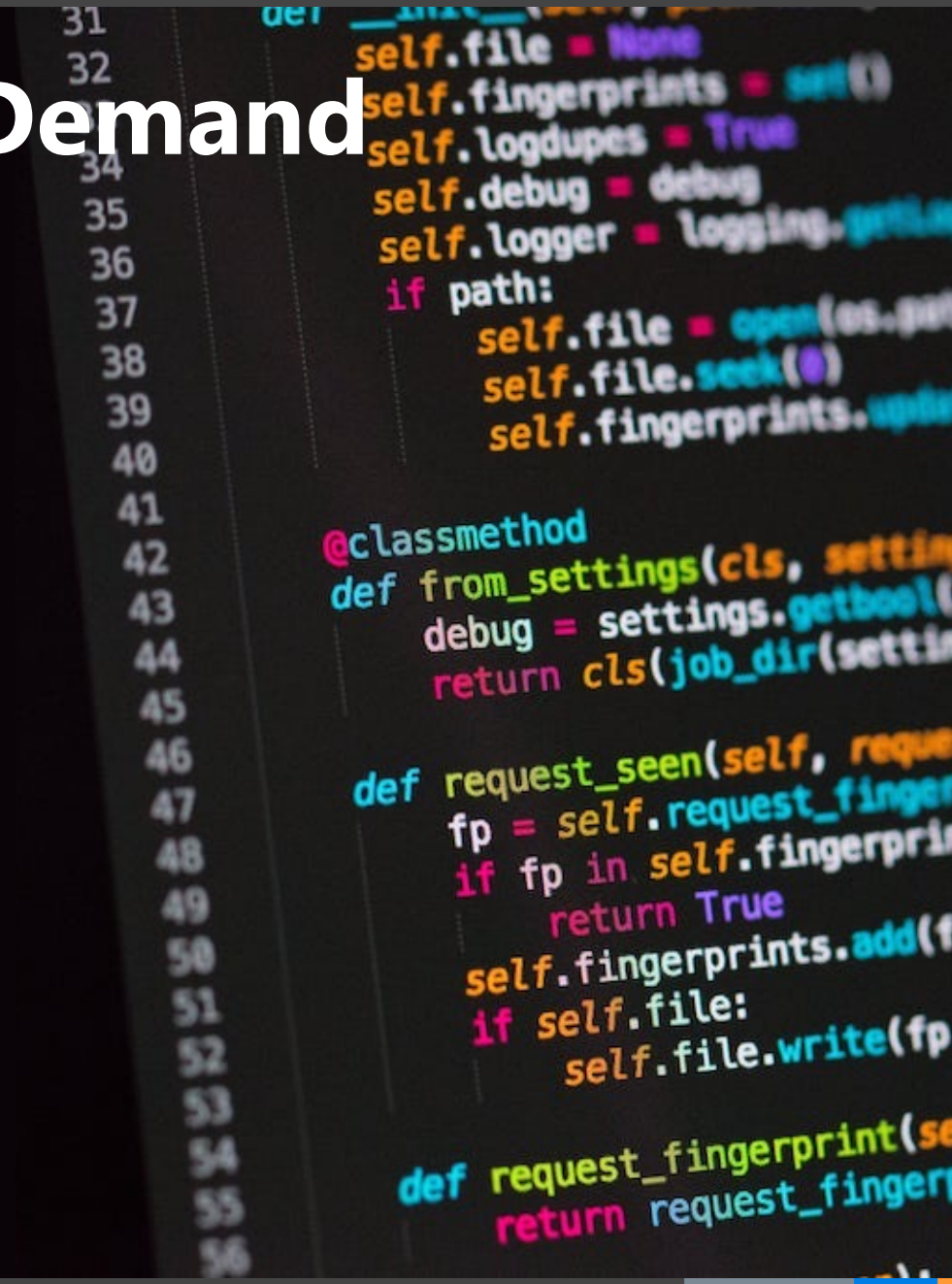
Simulation of a potential anti-cancer drug binding to HDAC8 enzyme (linked with acute myeloid leukemia)

Mineral Engineers

Semi-Autonomous Mines

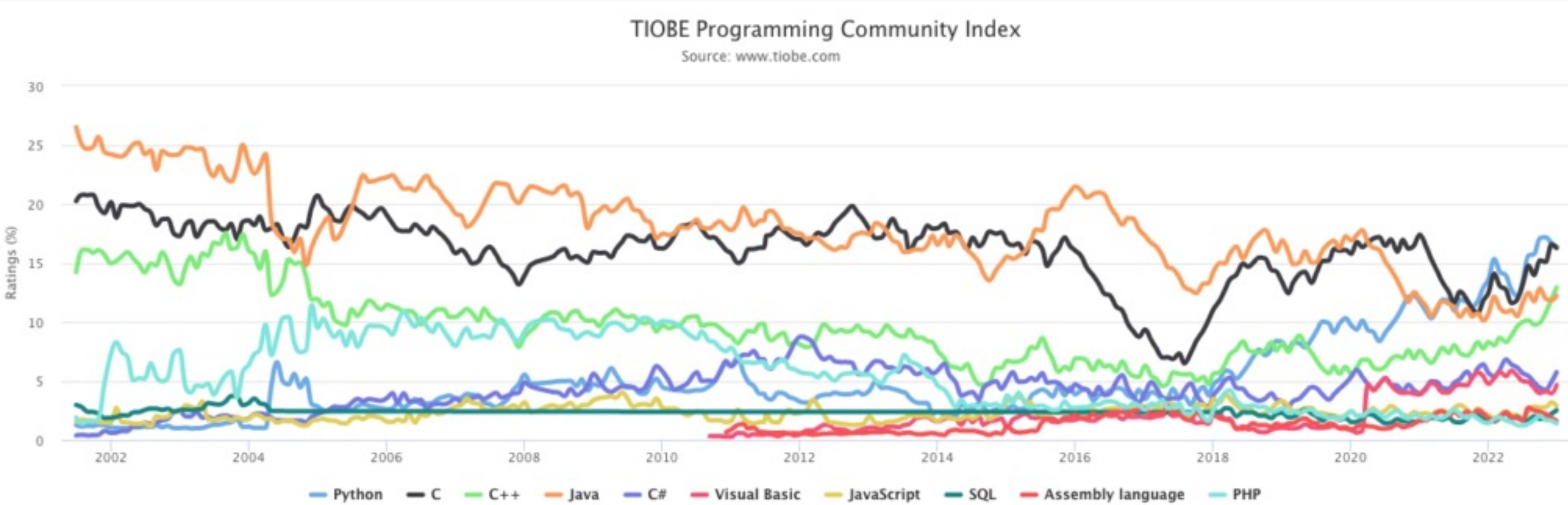
Programming Skills Are In Demand

- According to the *Developer Survey by StackOverflow*, Python was one of the most in-demand technologies of 2018, 2019, 2020, and 2021.
- As of 2021, it is ranked as the world's 2nd most popular programming language among professional software developers as well as the first most wanted programming language.



Python Skills Are In Demand

- According to the *TIOBE Index*, Python overtook C and Java in 2022 as the world's most popular programming language



One of the reasons for choosing Engineering!

- Finally! A course that isn't Chem, Bio, Physics, or Math
- This (as well as your engineering design course) is what separates you from many other sciences
  - And makes you **MUCH** more valuable….

Programmer


What my friends think I do


What my mom thinks I do


What society thinks I do

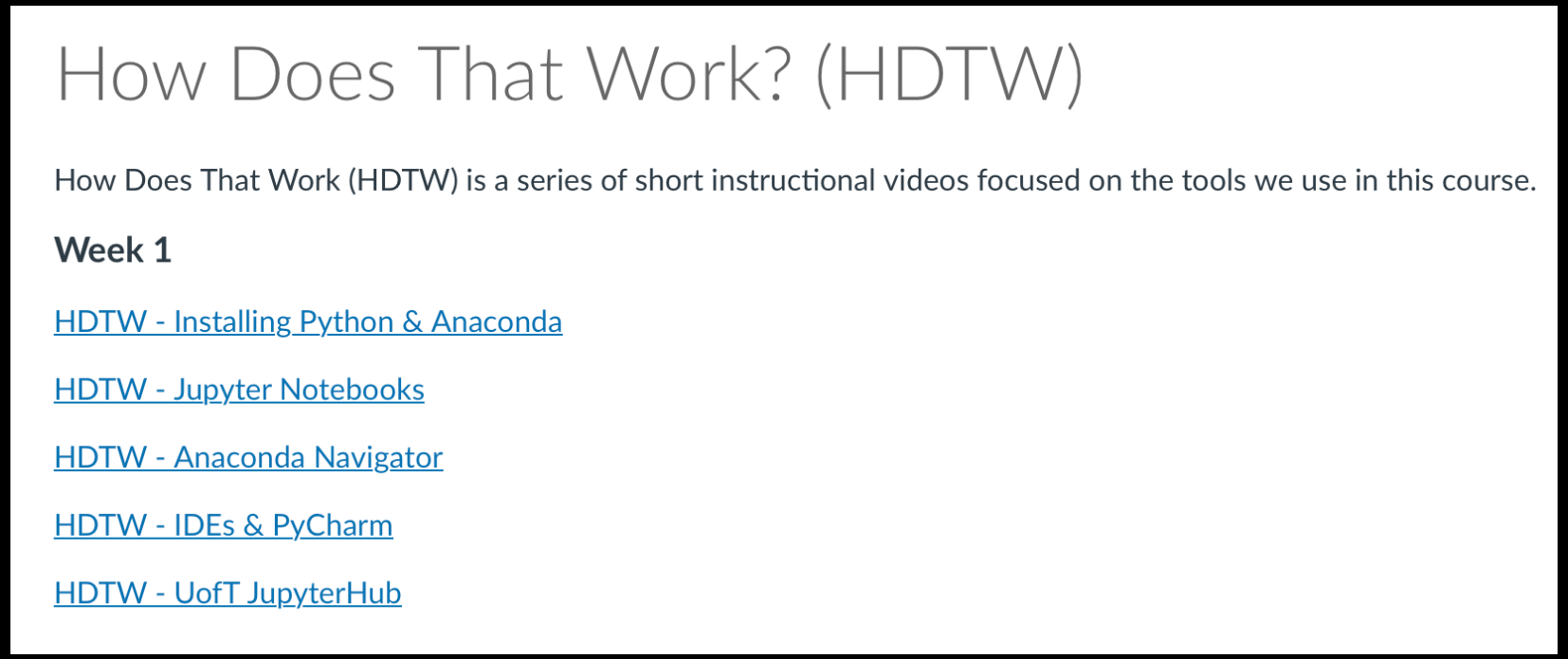

What my boss thinks I do


What I think I do


What I actually do

Wondering how something works?

- How Does That Work?
  - Modules -> Course Resources -> How Does That Work? (HDTW)



How Does That Work? (HDTW)

How Does That Work (HDTW) is a series of short instructional videos focused on the tools we use in this course.

Week 1

HDTW - Installing Python & Anaconda

HDTW - Jupyter Notebooks

HDTW - Anaconda Navigator

HDTW - IDEs & PyCharm

HDTW - UofT JupyterHub

APS106

UNIVERSITY OF TORONTO

introduction.

Week 1 | Lecture 1 (1.1)

if nothing else, write #cleancode