

introduction.

Week 1 | Lecture 1 (1.1)

if nothing else, write `#cleancode`

This Week's Content

- **Lecture 1.1**
 - **Introduction**
- **Lecture 1.2**
 - Variables, Expressions, and Operators
 - Reading: Chapters 1, 2, 3
- **Lecture 1.3**
 - The Programming Process

Teaching Team



Ben
Instructor



Seb
Instructor



Joseph
TA



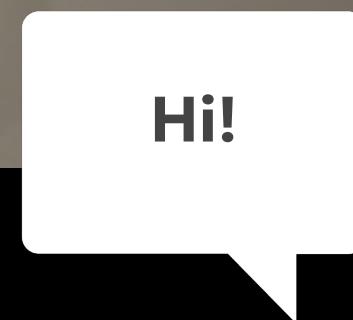
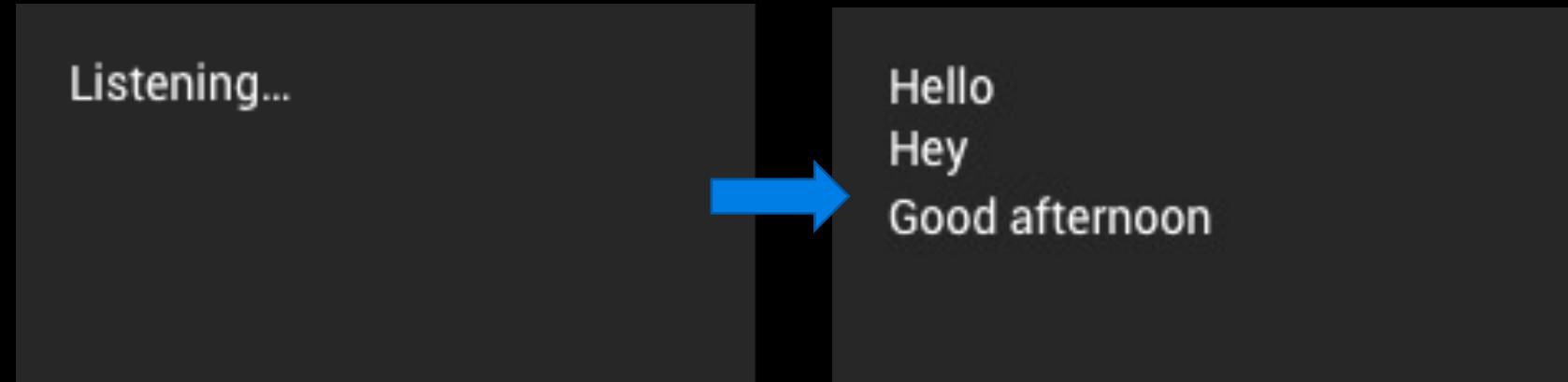
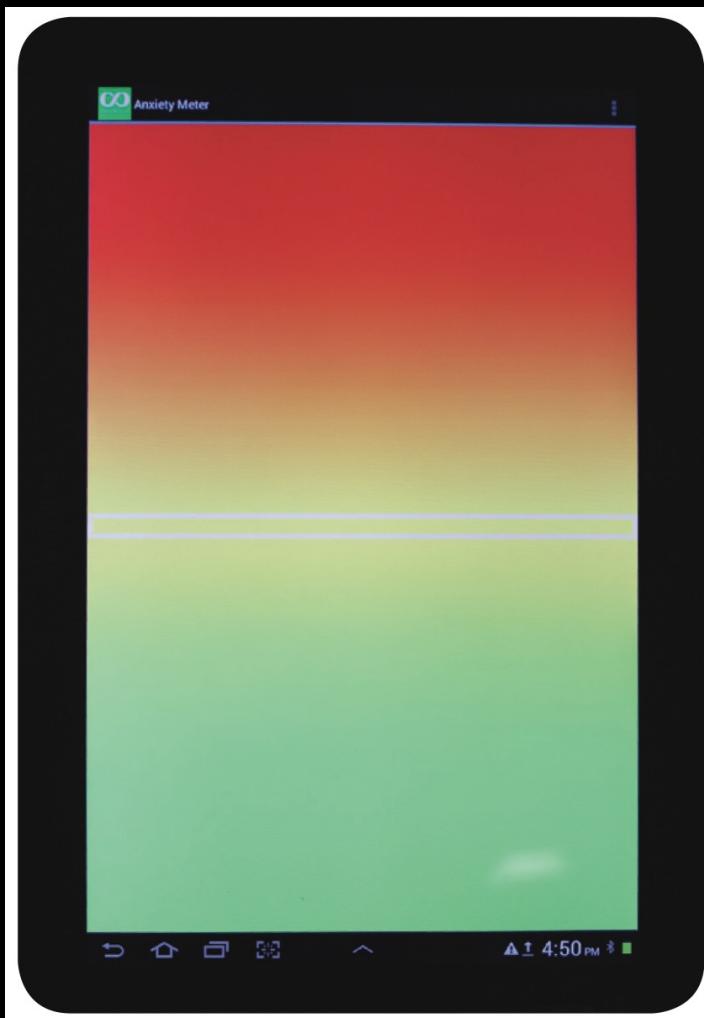
Katia
TA

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...

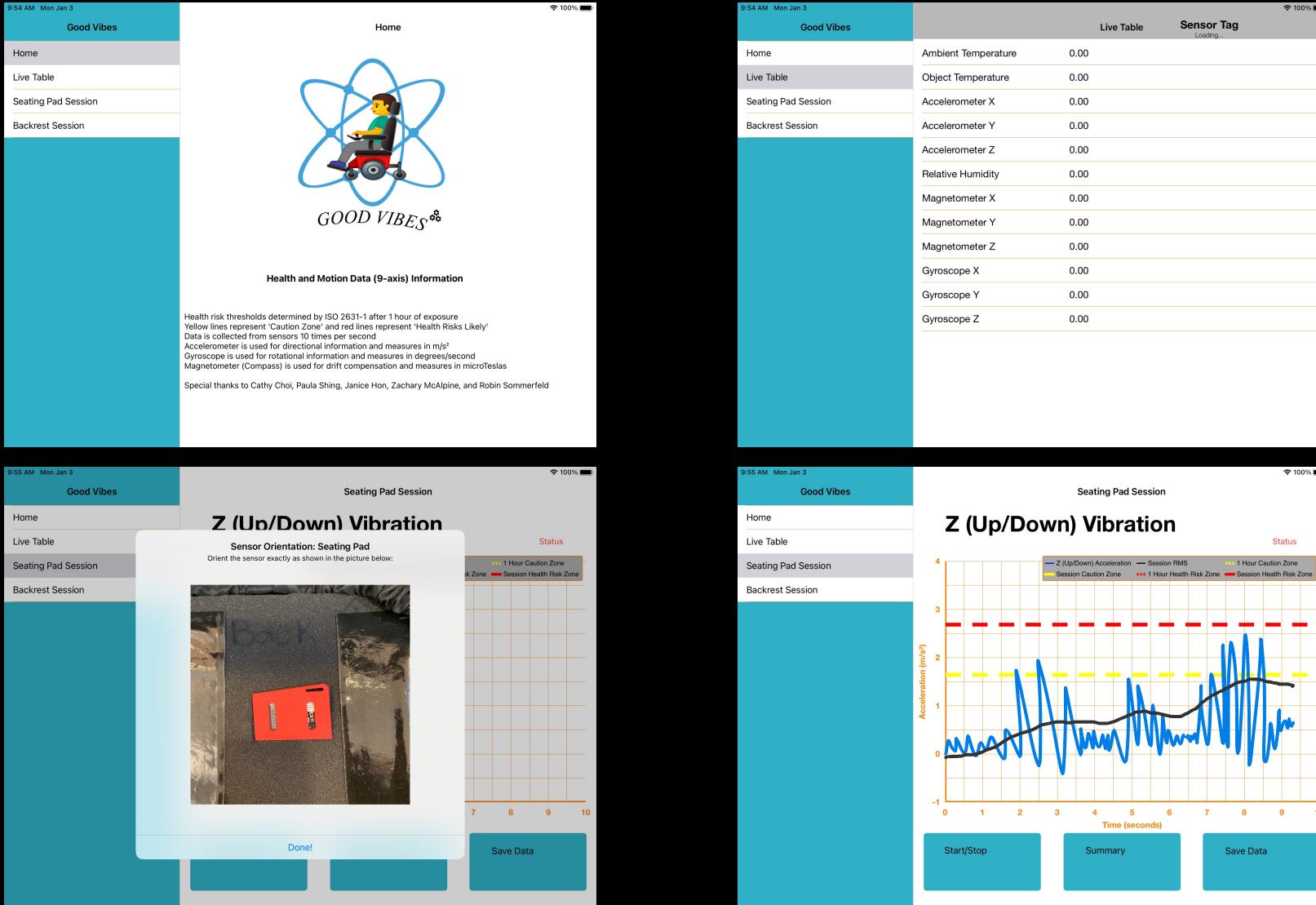


Projects for children with autism spectrum disorder



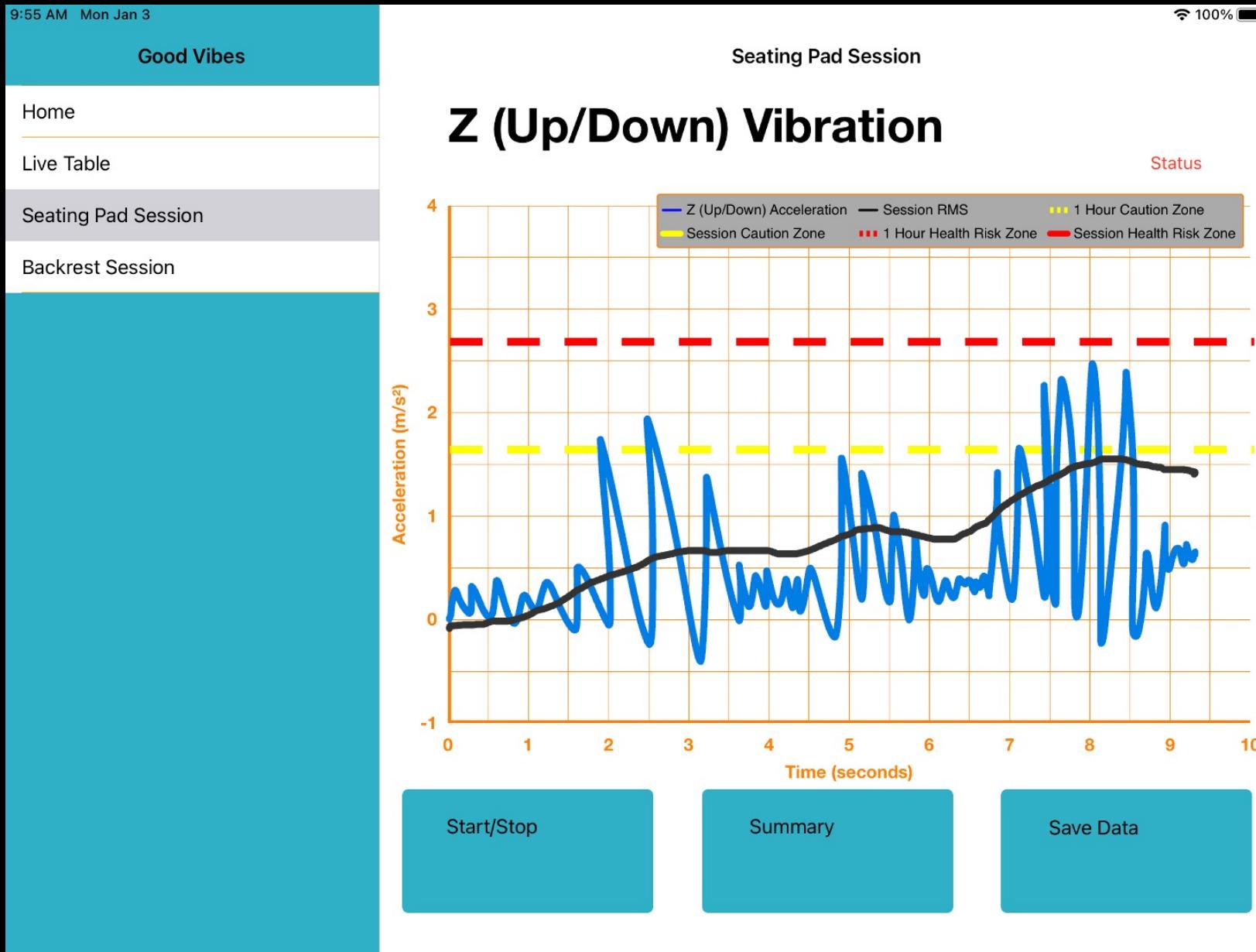
Good Vibes

A wheelchair vibration measurement tool



The image displays four screenshots of the Good Vibes mobile application, showing its user interface and data collection features.

- Home Screen:** Shows a sidebar menu with "Good Vibes", "Home", "Live Table", "Seating Pad Session", and "Backrest Session". The main area features a stylized atom icon with a person in a wheelchair, and the text "GOOD VIBES" below it. A note at the bottom explains the 9-axis sensor data and health risk thresholds.
- Seating Pad Session Setup:** Shows the "Seating Pad Session" screen with a camera view of a red sensor tag placed on a dark surface. Instructions for orientation are provided, along with a status bar and "Done!" and "Save Data" buttons.
- Sensor Tag Loading:** Shows the "Sensor Tag Loading..." screen with a table listing various sensors and their current values (e.g., Ambient Temperature, Object Temperature, Accelerometer X, etc.).
- Z (Up/Down) Vibration Analysis:** Shows a graph of "Z (Up/Down) Vibration" over time (0 to 10 seconds). The graph plots Acceleration (m/s²) against Time (seconds), showing a blue line for "Z (Up/Down) Acceleration" and a black line for "Session RMS". It includes yellow dashed lines for "1 Hour Caution Zone" and red dashed lines for "1 Hour Health Risk Zone". Buttons for "Start/Stop", "Summary", and "Save Data" are at the bottom.



More about Ben...

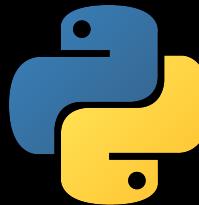


Seb

- Completed CivMin PhD in 2014 where I studies rock fracture and seismology ([RFDF](#)).
- AI Lead at KORE Geosystems, a mining tech startup.
- Senior Research Scientist at SickKids.  [SickKids](#)®
- Joined UofT CivMin in January 2020.
- Research topics: rock mechanics, ultrasonics, signal processing, computer vision, applied machine learning in mining.



AI 4 Mining



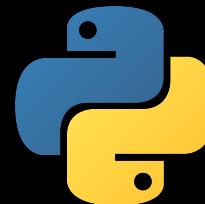
At KORE,
I use code
to do this!



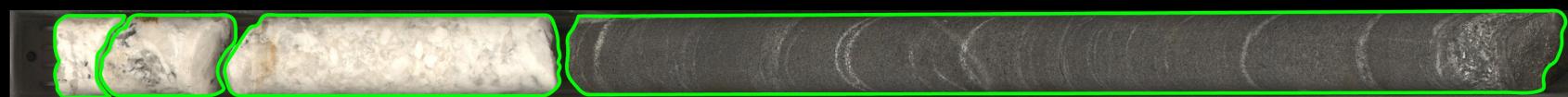
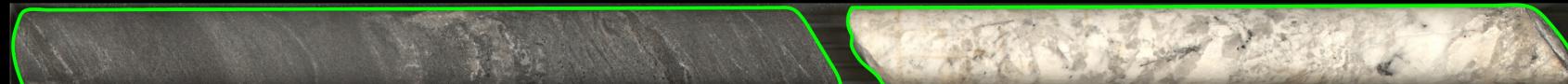
- Mining companies drill thousands of meters every year and the recovered rock (**core**) needs to be visually described by humans.



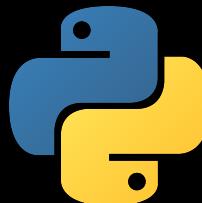
AI 4 Mining



At KORE,
I use code
to do this!



AI 4 Mining



At KORE,
I use code
to do this!



Alpha and Beta Angles

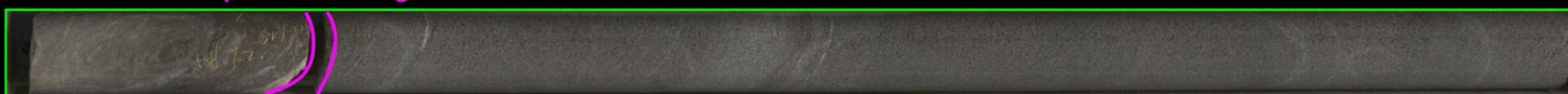


Alpha and Beta Angles

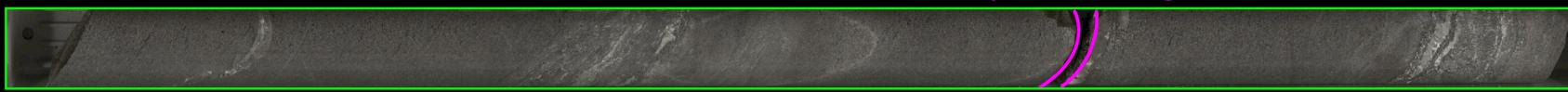
Alpha and Beta Angles



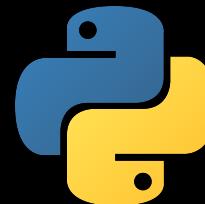
Alpha and Beta Angles



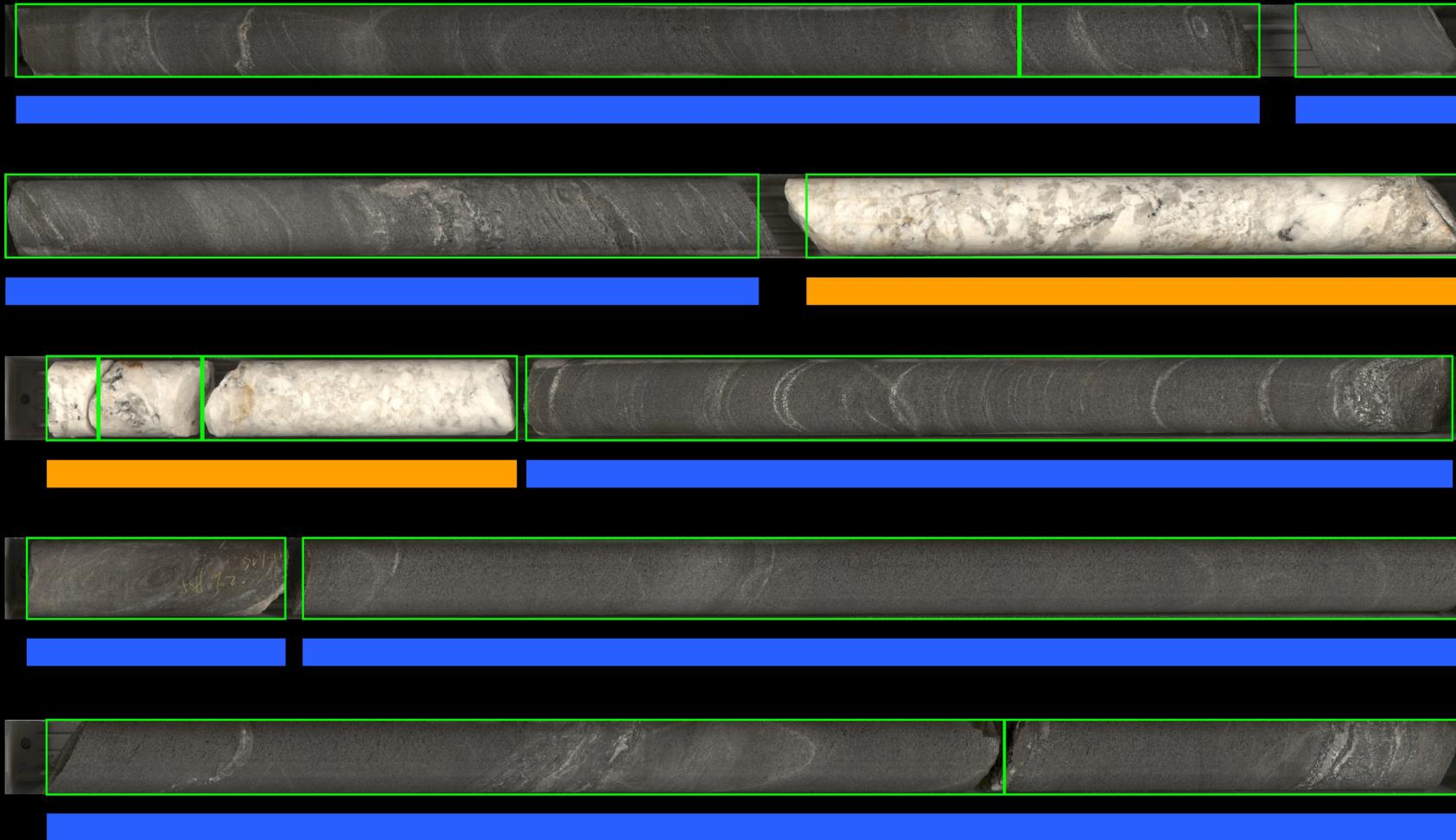
Alpha and Beta Angles



AI 4 Mining



At KORE,
I use code
to do this!



AI 4 Healthcare

AI
Detects
JET
Pages
Mjaye



Staff Physician

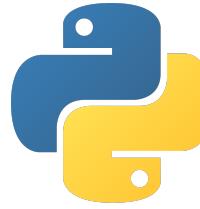


JET
Onset

Nurse

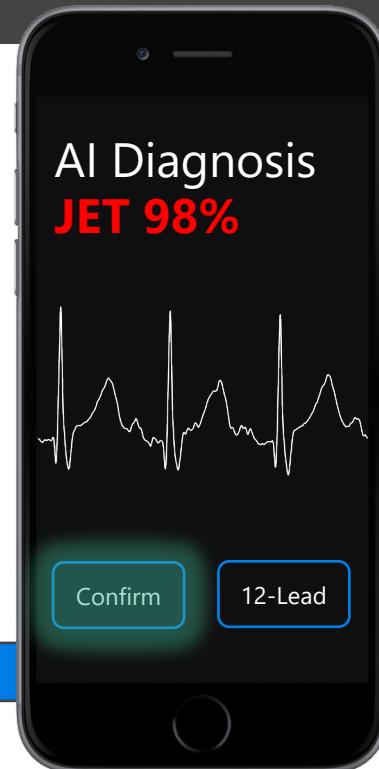


5 minutes later



Treatment
begins

At SickKids,
I use code
to do this!



AI  **JET: 5 %**



SickKids®

Seb

- I love travelling and have visited over 20 countries including France, Switzerland, Italy, Austria, Croatia, Costa Rica, Nicaragua, Panama, Peru, Bolivia, United States, Israel, Jordan, West Bank, Egypt, Turkey, Morocco, South Africa, and Namibia.
- I was a founding member of Ottawa's premiere Beatles cover band.
- I love camping, hiking, and surfing.
- I have a wonderful baby boy named Avery Goodfellow and baby girl name August.
- I love building things with code ( Github: Seb-Good).



Katia

- Doing a Master's in Civil Engineering.
- Worked as a software application engineer-in-training (EIT) for 3.5 years.
- Working with Python to analyze and clean data!
- Using Python to work with images from digital and optical microscopes of concrete and other building materials. As well, analyzing fracture behavior from Finite Discrete Element simulation models (FDEM).
- Martial arts fanatic :) involved in both Muay Thai and karate.
- Also play piano, working towards Royal Conservatory certification.



Joseph



- Doing a PhD in biomedical engineering.
- A marathon runner and currently training for a triathlon.
- Interweaves Python and MATLAB to analyze ultrasound wave propagation in artificial hearts made from stem cells through image and signal analysis.

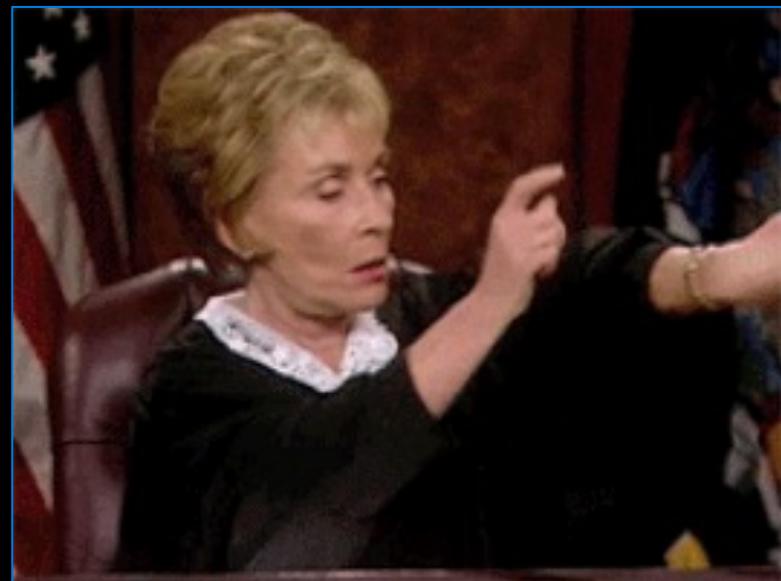
In Person (hopefully 😞)

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



In Person (hopefully ☹)

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



Online

- Ask questions in the chat or raise your hand if you'd like to unmute yourself to ask a question.
- Answer other students' questions 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 happy you asked it!

Course Information

- **Quercus:**

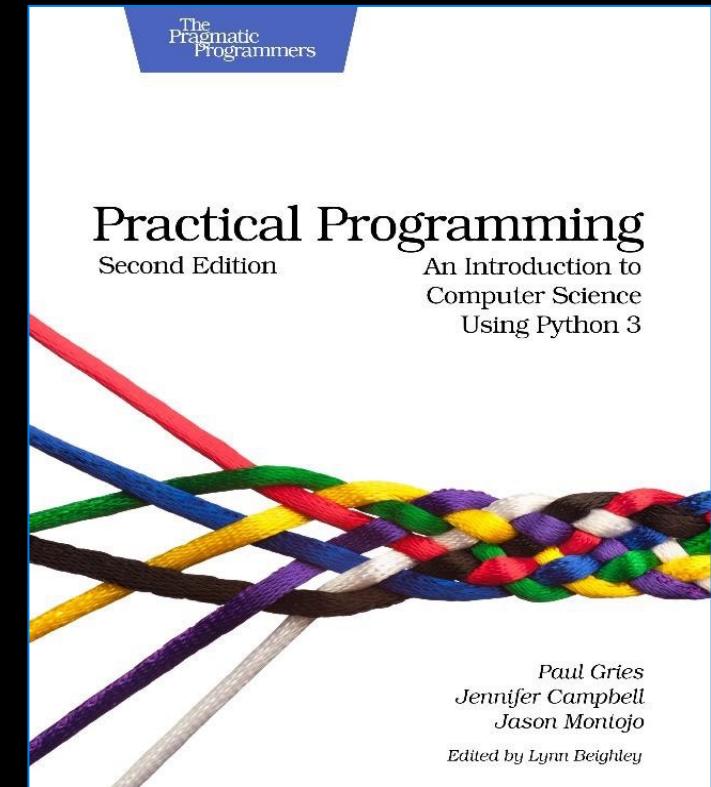
- APS106H1 S LEC0101 (One site contains all information).

- **Textbook:**

- ***Practical Programming: An Introduction to Computer Science Using Python 3***, 2nd edition, Paul Gries, Jennifer Campbell, Jason Montojo, Pragmatic Programmers.

- **Piazza:**

- piazza.com/utoronto.ca/winter2019/aps106



Print \$38 (e-book \$25)

Grading

Labs (9 in total)

total: 10%

Midterm Test (s)

total: 40%

Final Exam

total: 50%

Petitions: First-year Office.

Lecture Schedule

Section	Monday	Tuesday	Wednesday	Thursday	Friday
LEC01 & LEC02		15:00-16:00		15:00-16:00	15:00-16:00

- Lectures will be online until Jan 31 when we will (hopefully) transition to in person in **MYHAL 150**.
- Microsoft Teams links for lectures can be found on Quercus.
- Lecture content can be found on Quercus using the [Kinsella & Goodfellow](#) link.

Tutorial Schedule

- Tutorials will be online until Jan 31 when we will (hopefully) transition to in person.
- Zoom links for tutorials can be found on Quercus.

Week	Tut Content	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1 (Jan 10 - Jan 14)	No Tutorial					
Week 2 (Jan 17 - Jan 21)	Tutorial 1	TUT0104 11:00-12:00 MY 380	TUT0106 10:00-11:00 WB 130	TUT0102 12:00-13:00 MY 380	TUT0107 11:00-12:00 GB 303	TUT0105 16:00-17:00 MY 380
			TUT0108 15:00-16:00 SS 1087			TUT0101 09:00-10:00 GB 304
						TUT0103 13:00-14:00 WB 130

Click here
for content.

Lab Schedule

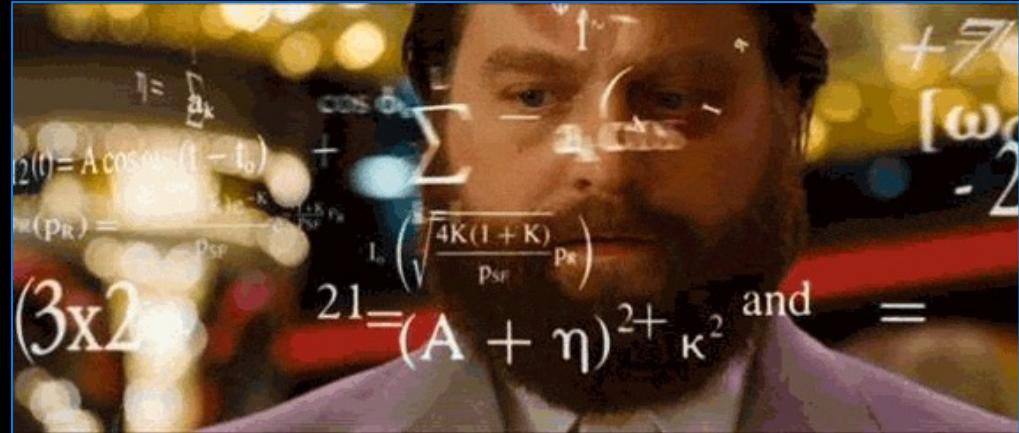
- Labs will be online until Jan 31 when we will (hopefully) transition to in person.
- Zoom links for labs can be found on Quercus.

Week	Lab Content	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1 (Jan 10 - Jan 14)	No Lab			Lab 1 Released 18:00		
Week 2 (Jan 17 - Jan 21)	Lab 1 	PR0106 09:00-11:00 MY 030 PR0108 09:00-11:00 GB 144	PR0107 09:00-11:00 MY 030 PR0102 16:00-18:00 GB 144	PR0104 09:00-11:00 GB 150 PR0103 13:00-15:00 GB 144 PR0101 16:00-18:00 GB 144 Lab 2 Released 18:00	Lab 1 Due 23:00	PR0105 11:00-13:00 WB 255

Click here for content.

Tips for Success

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



```
31     def __init__(self, path=None, debug=False):
32         self.file = None
33         self.fingerprints = set()
34         self.logduplicates = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(os.path.join(path, 'seen_requests'), 'a')
39             self.file.seek(0)
40             self.fingerprints.update([x.strip() for x in self.file])
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('SUPERVISOR_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

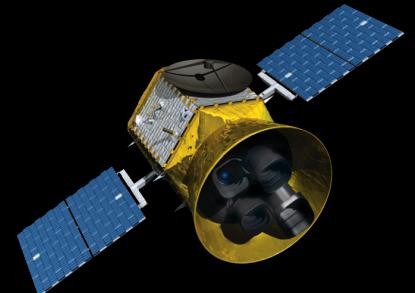
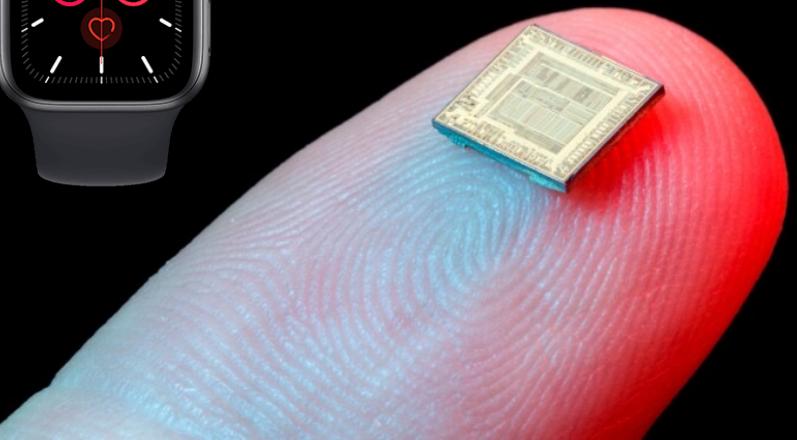
What is this course all about?

- 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!



#DISRUPTMINING

PRESENTED BY EGOLDCORP

CO-HOSTED BY KPMG



GEOSYSTEMS

\$1,000,000 Investment.

Owner of McEwen
Mining Inc.CEO of Franco-
Nevada CorporationPresident of Cisco
Systems Canada

COO of Goldcorp

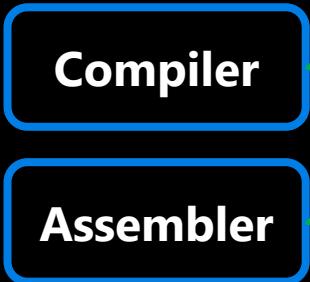
SHARK on ABC's
Shark Tank

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



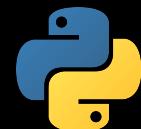
English

Pseudocode

Programming Language

Assembly Code

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.



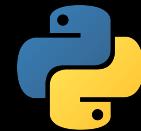
```
public class HelloWorld
{
    public static void main(String[ ] args)
    {
        system.out.println("Hello World!");
    }
}
```



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

Lower-Level

Higher-Level



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.

Increasing
Program Size

Matlab, SQL

C++, Java, Python

C, Fortran, Pascal

Assembly Code

Machine Code

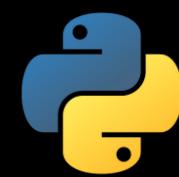


Hardware

Increasing
Ease-of-Use

 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

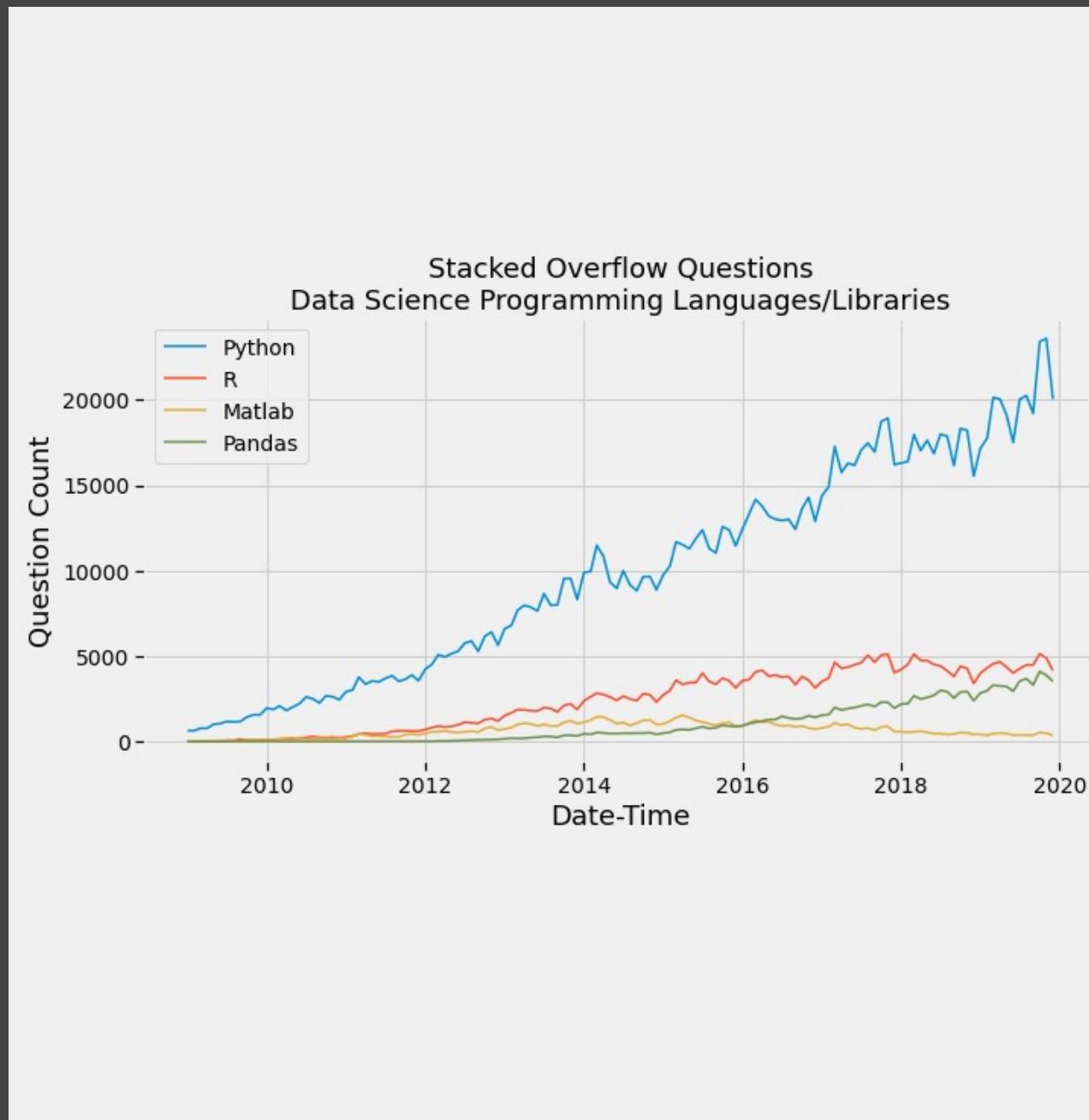
```
if x > 10:  
    print("x is greater than 10")
```



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?

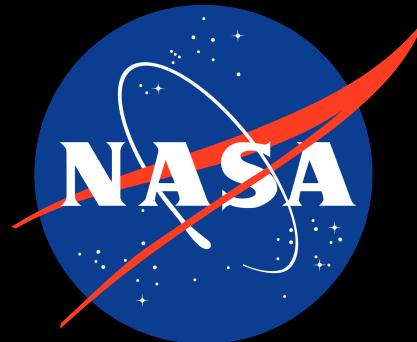
- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses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

SPACEX

yahoo!



Google

KORE
GEOSYSTEMS

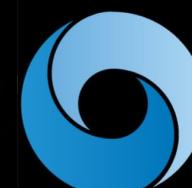
Spotify®

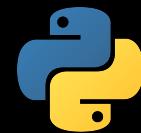


TESLA

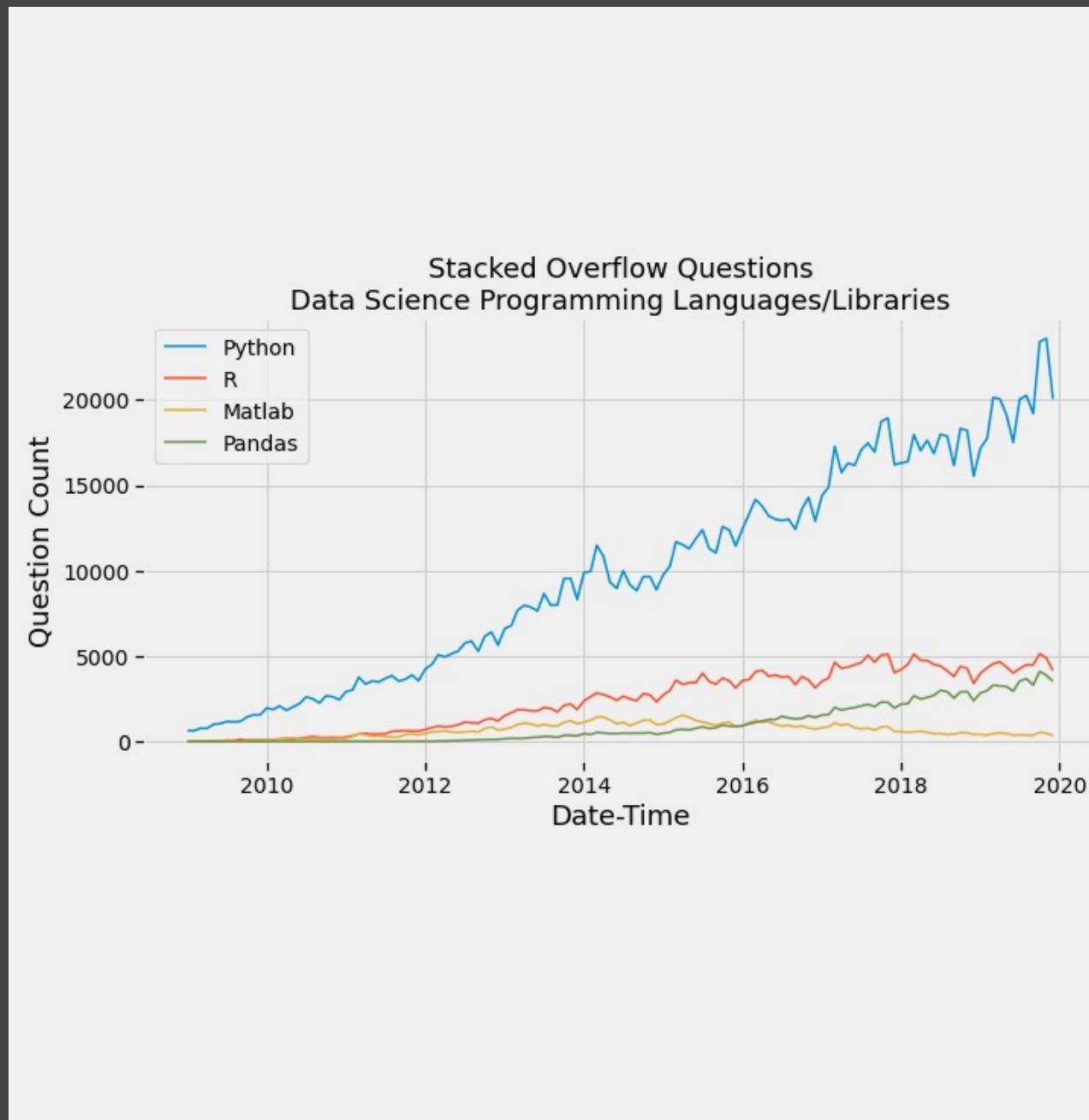
 Why Python?

- 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.

 PyTorch LANDING AI Azure
Machine Learning XGBoost TensorFlow scikit
learn DeepMind

 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 engineers, right?
 - Mechanical
 - Materials
 - Civil
 - Chemical
 - Mineral
- Why do these kinds of engineers need to know how to code?



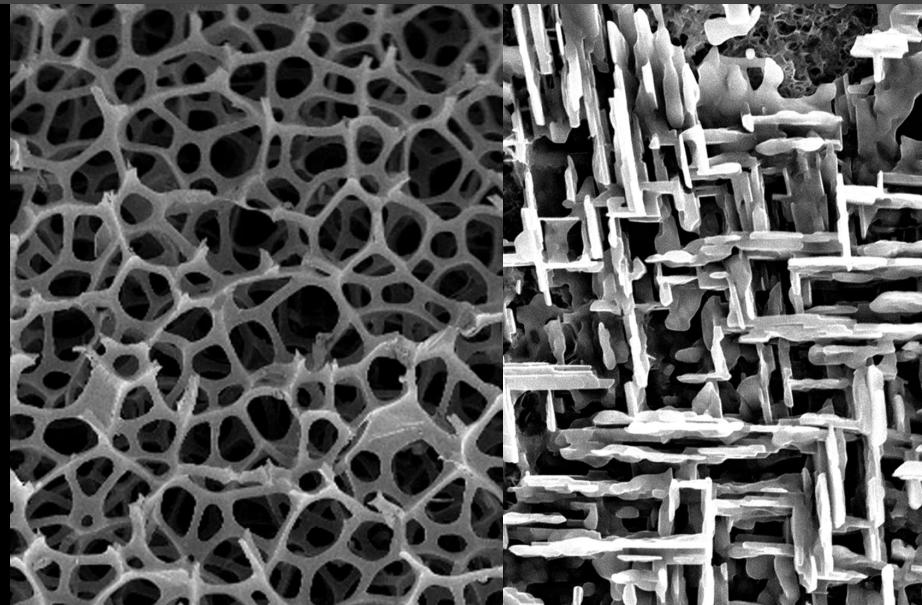
Mechanical Engineers

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



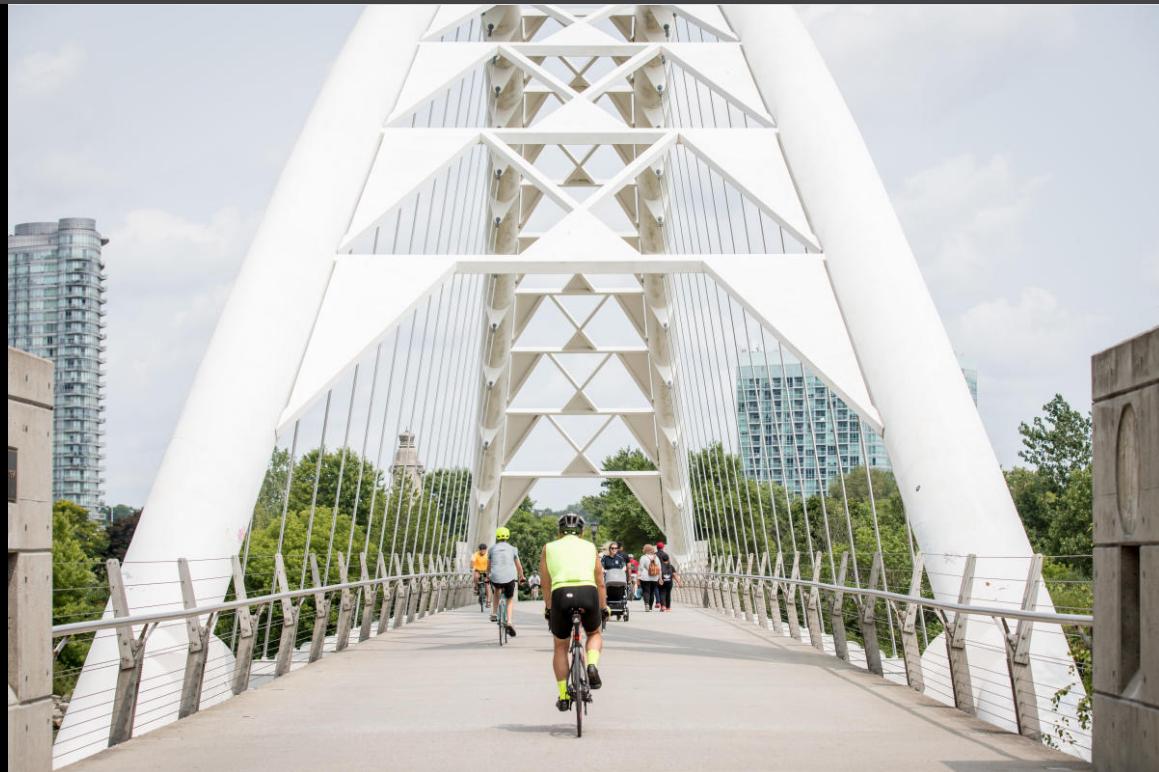
Materials Engineers

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



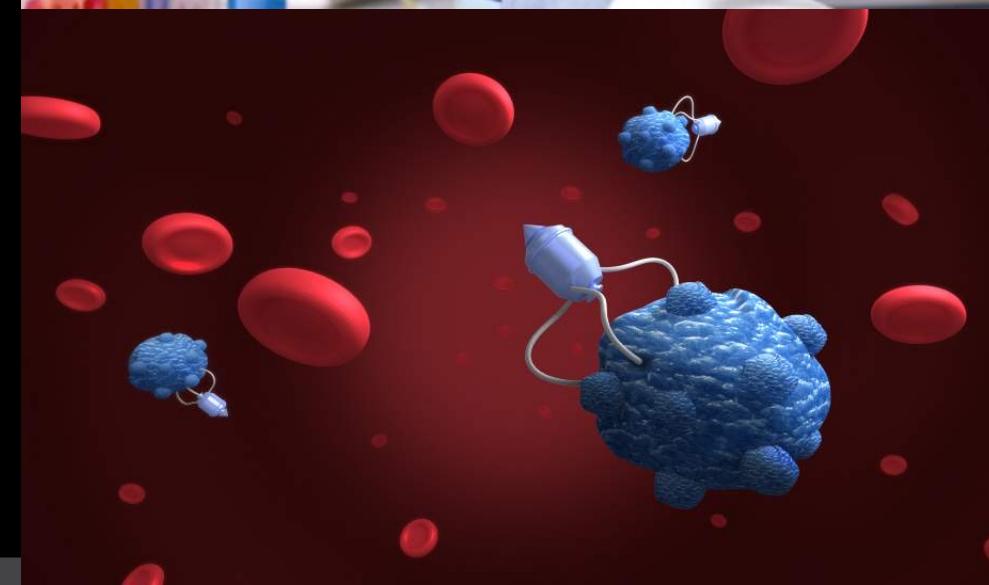
Civil Engineers

- 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.

```
31     dev = None
32     self.file = None
33     self.fingerprints = set()
34     self.logduplicates = True
35     self.debug = debug
36     self.logger = logging.getLogger()
37     if path:
38         self.file = open(os.path.join(path, 'fp.log'), 'w')
39         self.file.seek(0)
40         self.fingerprints.update(self._load_fingerprints())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getboolean('debug')
45         return cls(job_dir=settings['job_dir'],
46                    request_log=settings['request_log'],
47                    request_fp=settings['request_fp'],
48                    logduplicates=settings['logduplicates'],
49                    debug=debug,
50                    logger=logging.getLogger())
51
52     def request_seen(self, request_fp):
53         fp = self.request_fingerprint(request_fp)
54         if fp in self.fingerprints:
55             return True
56         self.fingerprints.add(fp)
57         if self.file:
58             self.file.write(fp + '\n')
59
60     def request_fingerprint(self, request_fp):
61         return request_fp
```

Why Engineers?

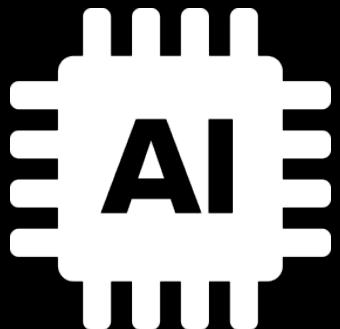
- Five technology trends have changed the game for engineers.



The Cloud



Cheap Sensors



Artificial
Intelligence



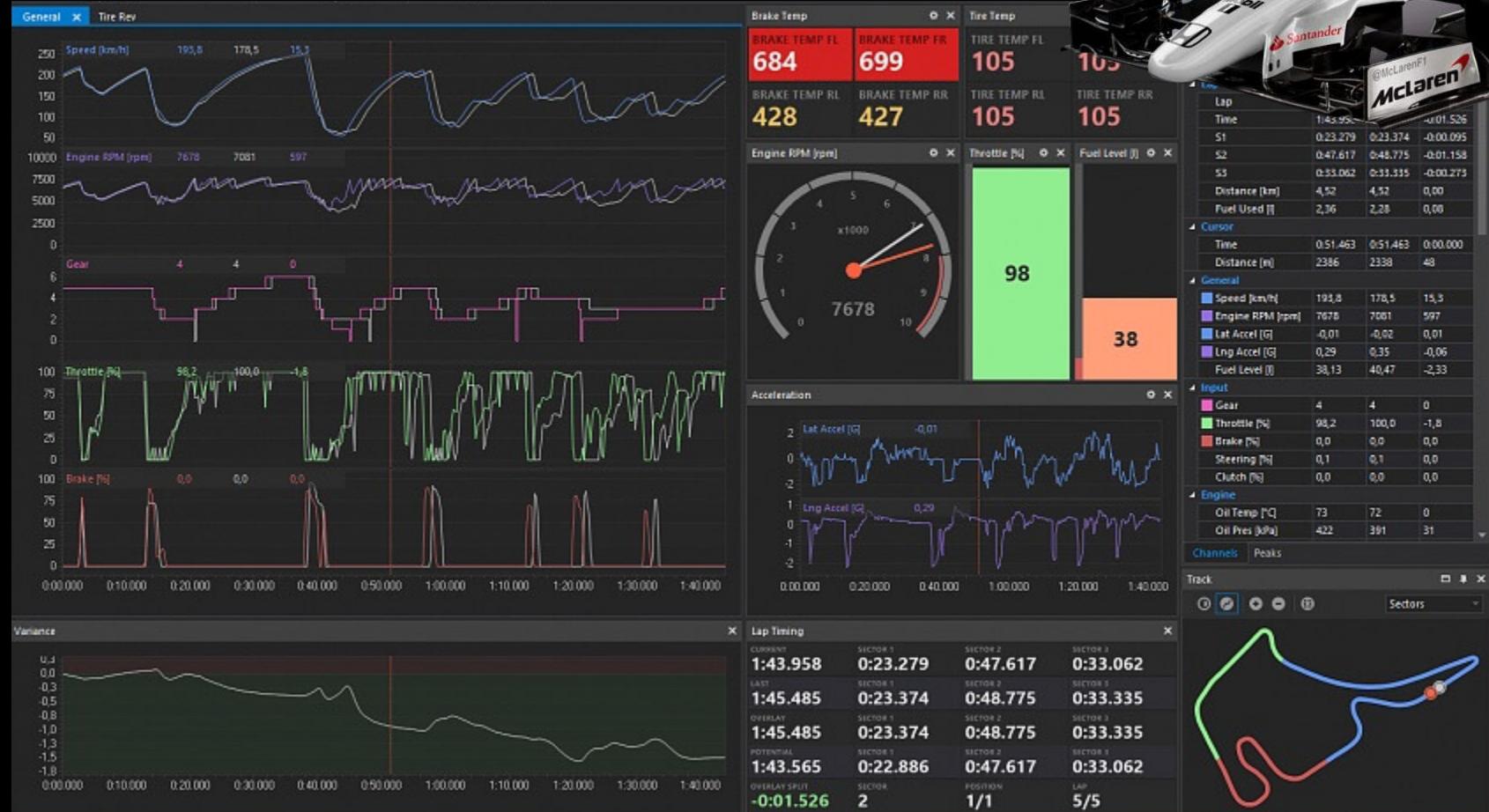
Big Data



The Internet
Of Things

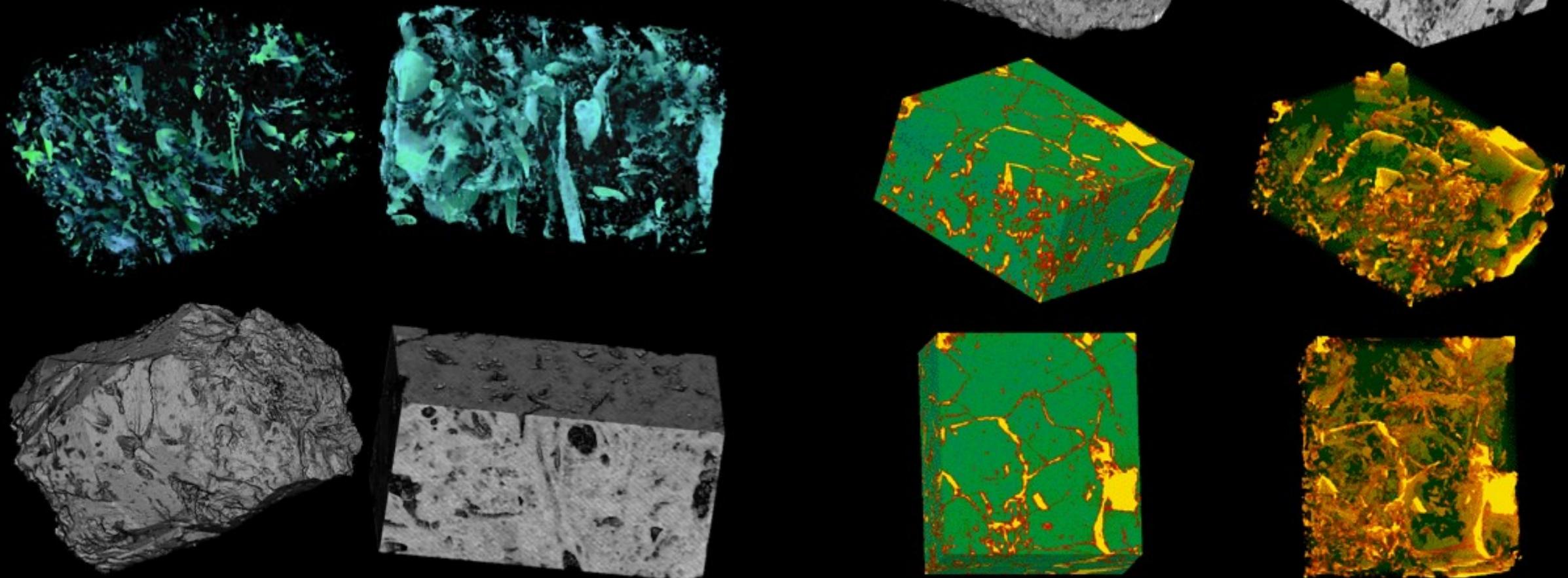
Mechanical Engineers

Realtime Diagnostics & Tuning



Materials Engineers

Digital Image Analysis of Material Microstructure



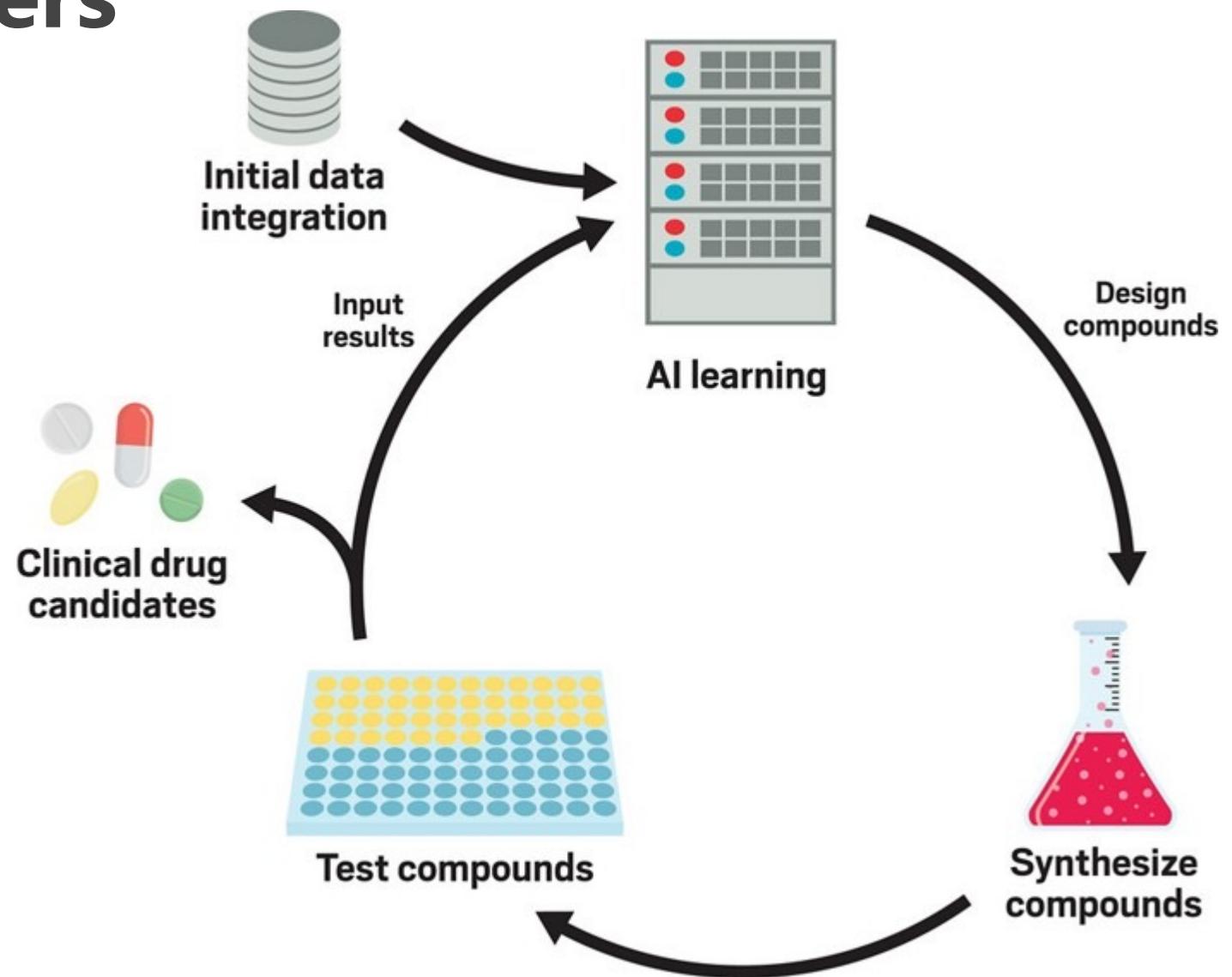
Civil Engineers

Urban Transportation Disrupted

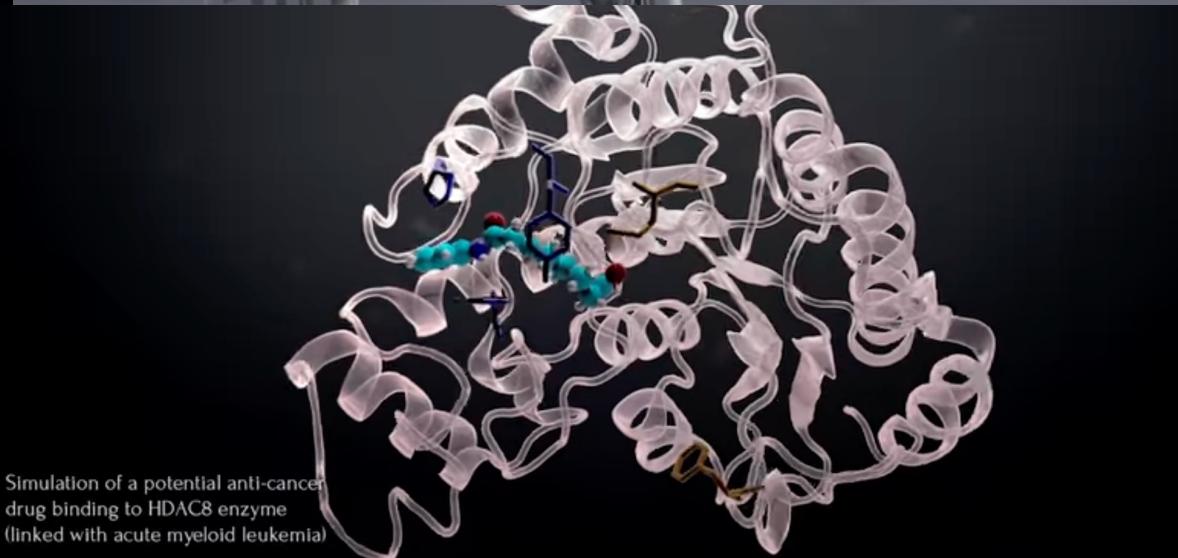


Chemical Engineers

Synthesizing new compounds



Sci-fi Inspiration



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.

```
31     dev = None
32     self.file = None
33     self.fingerprints = set()
34     self.logduplicates = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'fingerprint.log'), 'w')
39         self.file.seek(0)
40         self.fingerprints.update(self.read())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getboolean('debug', False)
45         return cls(job_dir=settings['job_dir'],
46                    request_log=settings['request_log'],
47                    request_fingerprint_log=settings['request_fingerprint_log'],
48                    logduplicates=settings.getboolean('logduplicates', True),
49                    file=settings['file'],
50                    fingerprints=settings['fingerprints'],
51                    logger=settings['logger'],
52                    dev=settings['dev'],
53                    max_fingerprints=settings['max_fingerprints'],
54                    max_requests=settings['max_requests'],
55                    max_time=settings['max_time'],
56                    max_size=settings['max_size'])
57
58     def request_seen(self, request):
59         fp = self.request_fingerprint(request)
60         if fp in self.fingerprints:
61             return True
62         self.fingerprints.add(fp)
63         if self.file:
64             self.file.write(fp + '\n')
65
66     def request_fingerprint(self, request):
67         return request_fingerprint(self, request)
```

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

introduction.

Week 1 | Lecture 1 (1.1)

if nothing else, write `#cleancode`