

Programming for Everybody

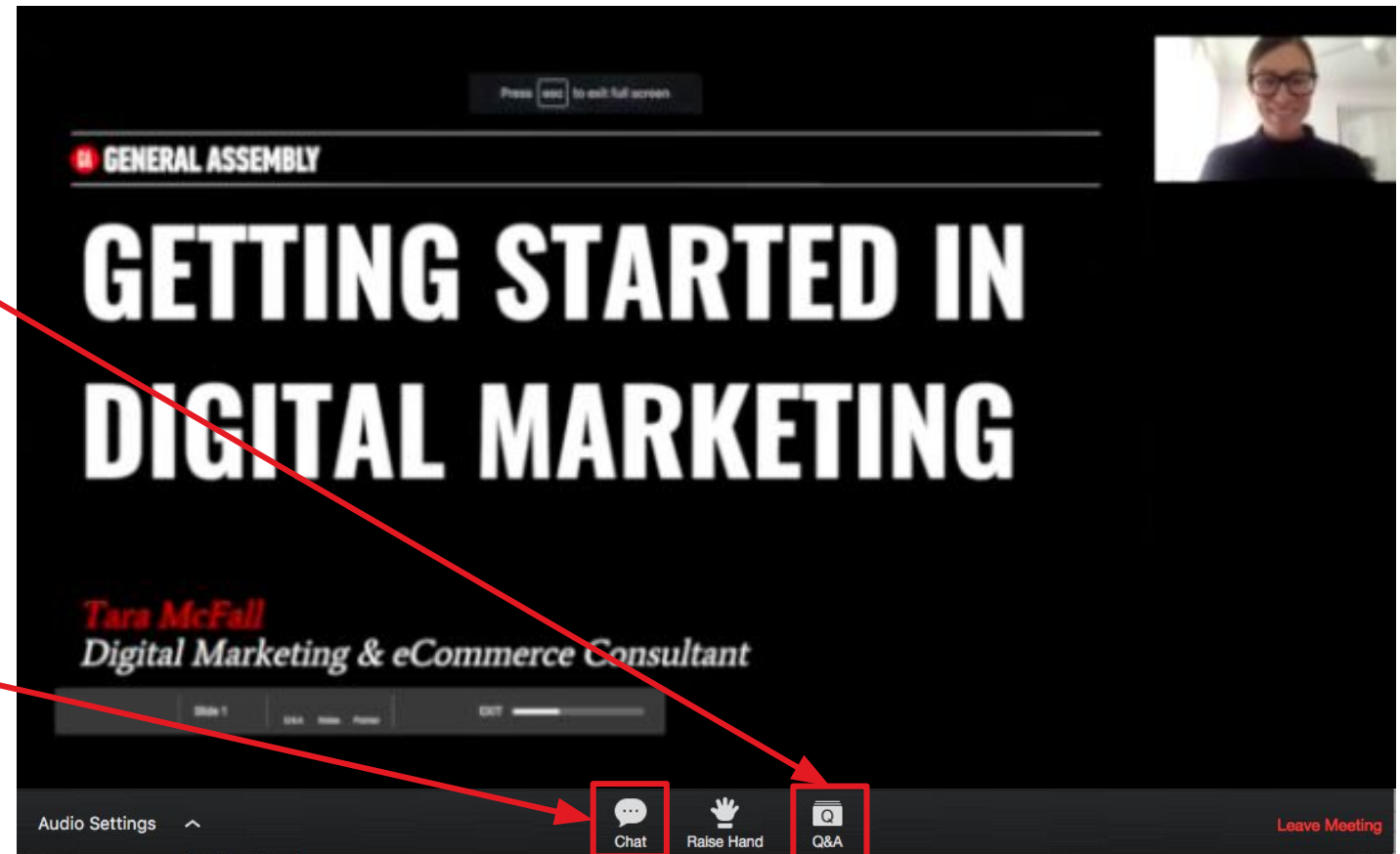
---

# Getting started with Python

# How to Participate Today

Ask questions in the Q&A section -  
We'll get to these at the end.

Make comments and answer  
questions in the Chat - everyone can  
see this!



# Meet Your Instructor



## Arwa Lokhandwala

Developer, Instructor, Speaker & Blogger

---

- Full Stack Developer with over 5+ years of experience developing scalable web applications
- Ex Lead Instructor for Software Engineering Immersive @GA
- Ex Telstra, Reliance Jio, BookMyShow
- Technical Blogger & Speaker



[Arwa Lokhandwala](#)



[@code.with.arwa](#)





# Who is here today?

# Agenda

- About Python (WHY)
- Functions (using)
- DataTypes and variables
- Control Structures with If/else
- How to Loop
- Functions (defining)
- Packages

# What is Python?

- Created by Guido Van Rossum in 1991
- Emphasizes **productivity** and code **readability**
- **Easy** to pick up and learn
- Easier for many to contribute to **production level code**
- **Readable code** means that almost anyone can read and understand what the code is doing



# JavaScript

```
let alphabets=["a","b","c"];

for(let i=0;i<alphabets.length;i++) {
    if(alphabets[i]=== "a") {
        console.log("Found a!");
        break;
    } else {
        console.log("Still looking!");
    }
}
```

# Python

```
for i in ["a","b","c"]:
    if i is "a":
        print("Found a!")
        break
    else:
        print("Still looking!")
```

# Why is Python readable?



- **Interpreted language:**
  - Step by step execution for easier programming ideation
  - Write once, run anywhere
  - Performance tradeoff
- **Object-oriented (OO)**
  - Code with objects that contain data and functions to manipulate it in predefined ways
- **Dynamically typed**
  - No need to explicitly specify type of data stored
  - Type is evaluated at runtime based on the value assigned to variable



# Typical Programs using Python (REAL WORLD)

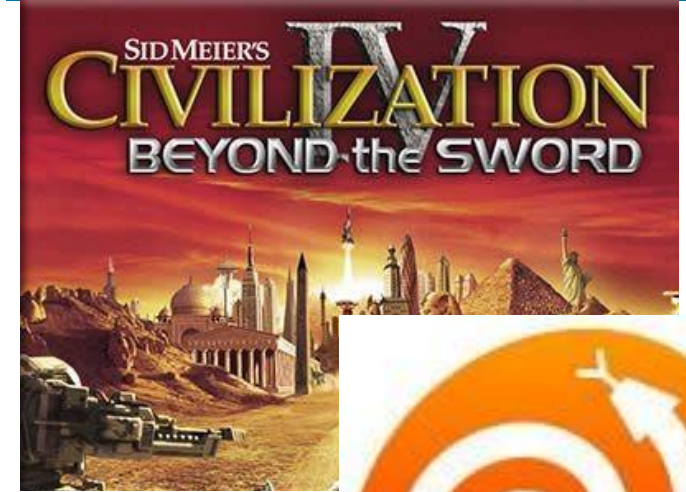
- **Data Science / Machine Learning:**
  - Analyse data and create predictive models.
    - [Pandas, ScikitLearn, Tensorflow](#)
- **Web Applications:**
  - Backends for website or mobile apps.
    - [Django, Flask.](#)
- **Data Engineering:**
  - Prepare data for machine learning / big data applications
    - [ETL Scripts, Data Pipelines, Data Analysis](#)
- **DevOps/SysOps:**
  - Maintain fleet of servers and live applications -
    - [Orchestration tools like Ansible](#)



# Real Cases: Who uses Python?



- **Industry & Academia**
  - AstroPy
  - BioPython
- **Web Development**
  - Youtube
  - DropBox
- **Game Development**
  - Civilization IV
- **Standalone Applications**
  - BitTorrent



BitTorrent™



# How to WRITE and RUN Python

## WHAT

SHELL

NOTEBOOK

CODE EDITOR

## WHY

FOR DISCOVERY /  
QUICK EXPERIMENTS

FOR LEARNING AND  
DATA SCIENCE

FOR BIGGER  
PROGRAMS

## HOW TO EXECUTE CODE

AT THE COMMAND LINE RUN "PYTHON".  
THEN TYPE CODE. ENTER EXECUTES.

CODE IS TYPED IN CELLS.  
TO EXECUTE A CELL, TYPE SHIFT + ENTER

TYPE CODE IN FILE. SAVE THE FILE.  
THEN EXECUTE "PYTHON <FILENAME>" AT THE  
COMMAND LINE.

# JUPYTER NOTEBOOK

- Most convenient way to learn – easy to run and re-run code
- Typically used by data scientists/ data analysts
- The Notebook consists of "Cells" that are space to write and execute code
- We can see the results immediately
- We're using an online notebook today, Google Colab




# Google Colab

- Jupyter Notebook hosted by Google Colab
- Completely online, no-installation required
- Collaboration is very easy
- It's like Google docs for your Jupyter Notebooks
- Easy to use Tensorflow and other ML libraries

<https://colab.research.google.com/notebooks/intro.ipynb>

# Python Workshop Notebook

1. Go to <https://github.com/arwalokhandwala/intro-to-python>
2. Click on [python-beginner.ipynb](#)
3. Click on  Open in Colab

—  
**What's Next?**

# Create a Learning Plan

Solidify your learning:

- Go through the parts of [Learn How to Think Like a Computer Scientist](#).
- Familiarize yourself with the language by going through [A Beginner's Python Tutorial](#).

Practice Practice Practice! Problems to expand your skills are available at:

- [HackerRank](#)
- [CodeWars](#)





# Real Cases: Expanding python

## Common Packages

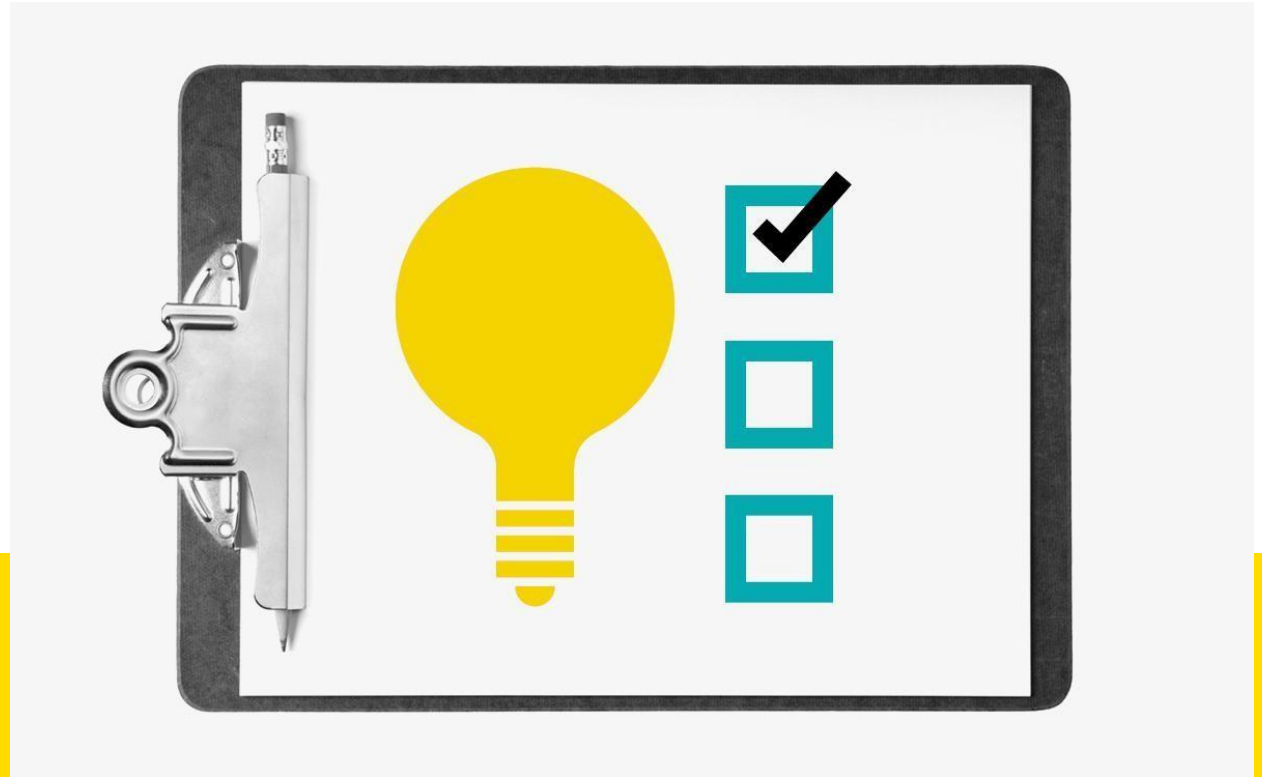
- Data manipulation: pandas, Numpy, scipy
- Machine Learning: scikit-learn, nltk
- Databases: psycopg2, sqlalchemy
- Visualizations: matplotlib, plotly, bokeh
- API calls / web scraping: requests, BeautifulSoup, Scrapy
- Web development: Django, Flask, Twisted, Scapy
- Game Development: Pygame, Pyglet
- Desktop App: PyQt, Tkinter

[More](#)



# A Few Good References

1. [Official Python Documentation](#)
2. [PEP-8 Official Guide](#)
3. [Anaconda Tutorials](#)
4. [Jupyter Documentation](#)
5. [Example Notebooks](#)



See you next time!



**Thank you!**

