Azure OpenAI

TheTechPlatform
Where Tech Meets.

Azure AI Series (Virtual)

23 Nov 2024, Saturday

AI Python for Beginners

Aroh Shukla

Regional Microsoft Cloud Architect

Microsoft MVP Alumni, MCT

TheTechPlatform
Where Tech Meets.

Aroh Shukla

MVP Alumni, MCT

Global Speaker

- Passionate to **learn**.
- Passionate to **share knowledge**.
- Passionate to work on **Microsoft Technologies**

Aroh.Shukla@gmail.com

/arohshukla

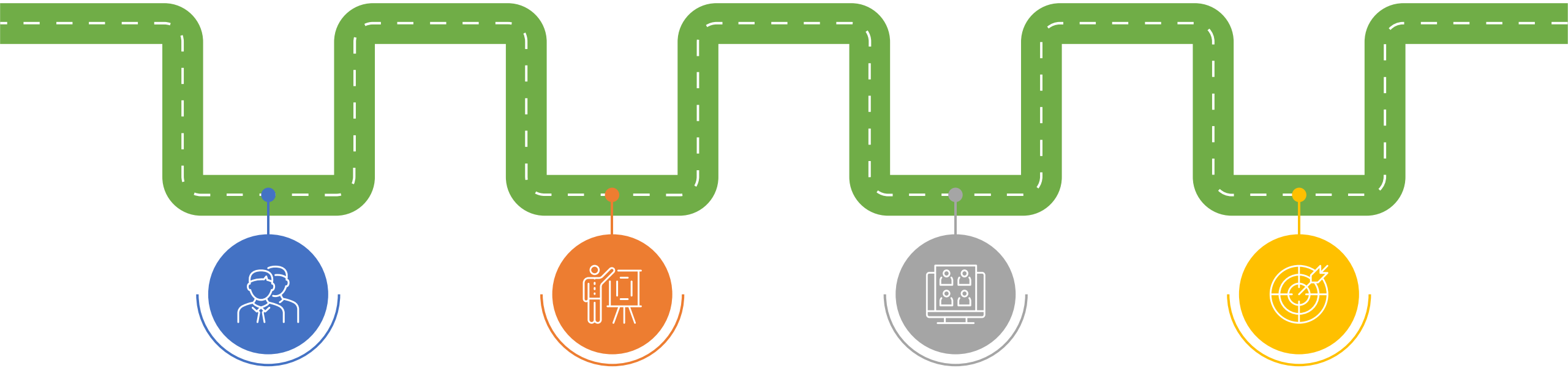
@aaroh_bits

TheTechPlatform
Where Tech Meets.

TheTechPlatform
Where Tech Meets.

Azure AI Course

8 Series Agenda



1. AI Python for Beginners

Learn Python with AI support. Develop skills in coding, testing, debugging, applications.

2. Prompt Engineering

Basics and advanced aspects of prompt engineering.

3. Responsible AI

The principles of Responsible AI and their role in AI development

4. Create own your private data

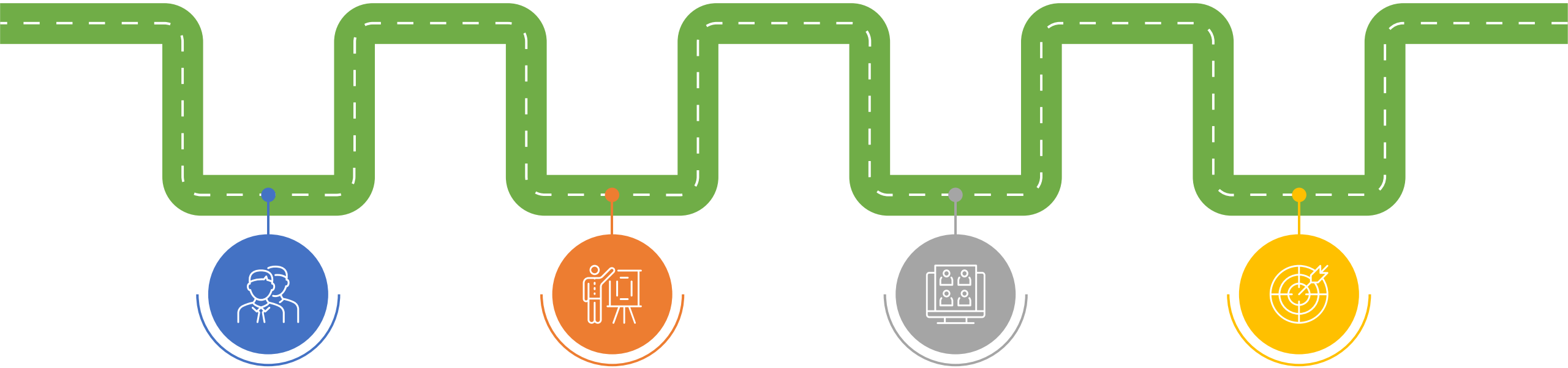
leverage your own data for AI development in Azure AI Studio using RAG

TheTechPlatform
Where Tech Meets.

TheTechPlatform
Where Tech Meets.

Azure AI Course

8 Series Agenda



5. Build Speech-to-Text App

Overview of Azure Open's Speech-to-Text Models

6. Prompt Flow – Basics

AI Studio Prompt Flow as no code solution

7. Prompt Flow – Advanced

Build a Custom Azure Prompt Flow

8 . Securing AI Services

Essential Strategies for Securing Azure AI Service.

TheTechPlatform
Where Tech Meets.

AI Python for Beginners

Chapter 1

01

AI Overview

What is AI overview

02

Why Python

Why Python is become more popular

03

Basics of Python

Vars, strings, function, Lists, files, module

04

Python External Libraries

TensorFlow, Pandas, Matplotlib, OpenAI

05

Demos

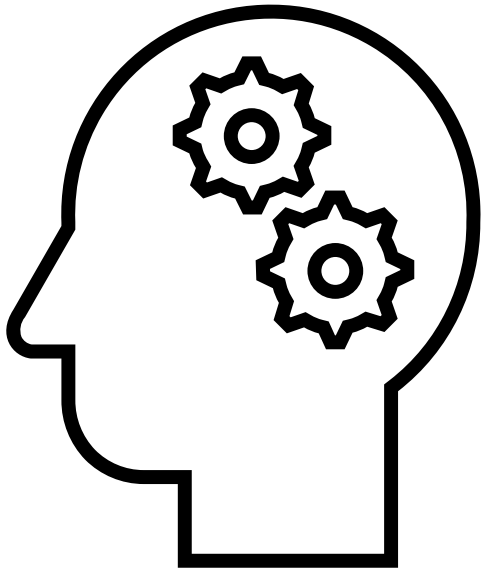
All as discussed & resources

TheTechPlatform
Where Tech Meets.

What is Artificial Intelligence?

Software that **imitates human capabilities**

- **Predicting outcomes** and **recognizing patterns** based on historic data.
- Recognizing **abnormal events** and **making decisions.**
- Interpreting **visual input.**
- **Understanding language** and engaging in conversations.
- **Extracting information** from sources to gain knowledge.



Common AI workloads

	Machine Learning	**Predictive models** based on data and statistics – the foundation for AI.
	Computer Vision	**Capabilities within AI** to interpret the world visually through cameras, video, and images.
	Natural Language Processing	**Capabilities within AI** for a computer to interpret **written or spoken** language and respond appropriately. (e.g. ChatGPT, Bing Copilot)
	Document Intelligence	**Capabilities within AI** that deal with **managing, processing**, and using high volumes of data found in forms and documents.
	Knowledge Mining	Capabilities within AI to extract information from **large volumes of often unstructured data t**o create a searchable knowledge store.
	Generative AI	**Capabilities within AI** that create original content in a variety of formats including natural **language, image, code, and more**.

TheTechPlatform
Where Tech Meets.

Principles of responsible AI (**more in Chapter 3**)

	Challenge or Risk	Example
 Fairness	Bias can affect results.	A loan-approval model **discriminates by gender due to bias in the data** with which it was trained.
 Reliability & safety	Errors may cause harm.	An **autonomous vehicle experiences** a system failure and causes a collision.
 Privacy & security	Private data could be exposed.	A medical diagnostic bot is trained using **sensitive patient data**, which is stored insecurely.
 Inclusiveness	Solutions may not work for everyone.	A predictive app provides **no audio output for visually impaired users.**
 Transparency	Users must trust a complex system.	An **AI-based financial tool** makes investment recommendations – what are they based on?
 Accountability	Who's liable for AI-driven decisions?	An **innocent person is convicted of a crime based on evidence** from facial recognition – who's responsible?

TheTechPlatform
Where Tech Meets.

Predicative AI vs Generative AI

	Predicative AI	Gen AI
Function	Predicts the most probable **output** given input data.	Creates **new data similar** to the **input** data distribution.
Training	Typically trained on **labeled data** for classification tasks.	Often trained on **unlabeled data** for generating new content.
Output Variety	**Limited to predicting** predefined classes or values.	Capable of **producing diverse outputs, sometimes novel.**

Python 101

- Will learn **Python in 1 hour** a Demo

TheTechPlatform
Where Tech Meets.

Why Python for Data Science, AI & ML

- **User Friendly** – User in mind
- **Widely Popular**: 1st Programming Language
- **Versatile** - Jack-of-all-trades in the programming world,
  a) Web Development
  b) Data Science
  c) AI
  d) ML

TheTechPlatform
Where Tech Meets.

How Python is Revolutionizing the Tech Industry

Data Science

- Pandas
- NumPy
- Matplotlib
- Jupyter Notebooks

AI

- TensorFlow & Keras
- PyTorch
- OpenAI
- Azure OpenAI

Machine Learning

- Scikit-learn
- TensorFlow & Keras

TheTechPlatform
Where Tech Meets.

Python Step up

Step by Step

01

Prerequisites

Locally setup Python on your machine.

02

Verify Installation

Verify the Python version

03

VS Code (Free)

How to use Python as a VS Code extensions

04

Create Virtual Environment(Free)

Why to use Python and how it used.

Demo

Demo 1 – Set up Demo Python

- Download Python,
- Set up Path,
- VS Code set up
- VS Code Python Extensions
- Python Virtual Environment

TheTechPlatform
Where Tech Meets.

Variables and Strings

Variables are:

- storage locations that have a name
- name-value pairs fruit = 'apple' fruit = 'orange'

Case sensitive. (Case matters!)

- Fruit and fruit are different variables.

Must start with a letter.

- Can contain numbers.

Underscores (_)allowed in variable names

- Not allowed
  - +
  - _

Valid Variable Names

- first3letters = 'ABC’
- first_three_letters = 'ABC’
- firstThreeLetters = 'ABC'

Variables and Strings

Strings

- Represent **text**
- Surrounded by **quote**
  - fruit = 'apple’
  - fruit = "apple"

Using Quotes within Strings.

- double = "She said, ****"That's a great tasting apple!\""“
- single = 'She said, "That****'s a great tasting apple!“’
- **Underscores** (_)allowed in variable names
- Not allowed
  - +
  - –

Indexing

- String: a p p l e
- Index: 0 1 2 3 4
  - a = 'apple'[0]
  - e = 'apple'[4]
- fruit = 'apple'
- first_character = fruit[0]

TheTechPlatform
Where Tech Meets.

Functions

Function

- section of **reusable code** that performs an action.
- A function **has a name** and is called, or executed, by that name.
- Optionally, functions can **accept arguments** and **return data**

Common Functions

- **print() function**
  - fruit = 'apple’
  - print(fruit)
  - print('orange’)

  Output
  - Apple
  - apple

len() function

- fruit = 'apple’
- fruit_len = len(fruit)
- print(fruit_len)

Output

- 5

Nesting function

- fruit = 'apple’
- **print(len**(fruit))

Output

- 5

TheTechPlatform
Where Tech Meets.

Basic OOP

- Everything in **Python is an object**.
- Every **object** has a **type**.
- 'apple' is an object of type "**str**".
- 'apple' is a **string object**.
- fruit = 'apple'.
  - fruit is a string object.
- **Methods** are functions run against an object.
  - object.method()

lower() String Method

- **fruit = 'Apple'**
- print(fruit.lower())

Output

- apple

upper() String Method

- **fruit = 'Apple'**
- print(fruit.upper())

Output

- APPLE

TheTechPlatform
Where Tech Meets.

Numbers

- Use numbers directly in your source code
  - **integer = 42**
  - **float = 4.2**

Numeric Operations

Symbol	Operation
+	Add
-	Subtract
*	Multiply
/	Divide
**	Exponentiate *2 ** 4 means "2 raised to the power of 4"* *2 * 2 * 2 * 2*
%	Modulo *3 % 2 = 1* *4 % 2 = 0*

Floating Point Numbers (Floats)

- 8 / 2 = 4.0
- 1 + 2.0 = 3.0

The Int() function

- quantity_string = '3'
- total = **int**(quantity_string) + 2
- print(total)

Output

- 5

The float() function

- quantity_string = '3'
- quantity_float = **float**(quantity_string)
- print(quantity_float)

Output

- 3.0

Comments – Single line

```
# This is a comment. Python ignores them.
# The following code:
#    Computes hosting costs.
#    Determines the duration of hosting
```

Comments – Multiple line

```
""" This is the start of the comment
This is another line.
This is the last line in the comment. """
```

TheTechPlatform
Where Tech Meets.

Boolean

- Can only be True or False
  - **a_boolean = True**
  - **the_other_boolean = False**
  - **print(a_boolean) // True**
  - **print(the_other_Boolean // False**

Boolean Operators

Operator	Description
and	Evaluates to **True** if both statements are true, otherwise evaluates to **False**.
or	Evaluates to **True** if ***either*** of the statements is true, otherwise evaluates to **False**
not	Evaluates to the opposite of the statement

Conditionals

```
if 37 < 40:
    print('Thirty-seven is less than forty.')
```

Code Blocks – 2 spaces

```
Block One
  Block Two
  Block Two
    Block Three
```

Code Blocks – 4 spaces

```
Block One
    Block Two
    Block Two
        Block Three
```

Spacing problems

- **IndentationError: expected an indented block**

TheTechPlatform
Where Tech Meets.

Lists

- A list is a data type that holds an ordered collection of items.
- The items can be of various data types.
- You can even have lists of lists!

Creating lists

- list_name = [item_1, item_2, item_N]
- llist_name = []
- list_name[index]

- animals = ['man', 'bear', 'pig’]
- print(animals[0])
- print(animals[1]) print(animals[2])
- Output
- Man
- Bear
- pi

TheTechPlatform
Where Tech Meets.

Dictionaries

- **Hold key-value pairs called items**.
- AKA associative arrays, hash tables and hashes.

Creating lists

- contacts = {'Jason': '555-0123', 'Carl': '555-0987'}
  jasons_phone = contacts['Jason']
- carls_phone = contacts['Carl']
- print('Dial {} to call Jason.'.format(jasons_phone))
  print('Dial {} to call Carl.'.format(carls_phone))

- Output
- Dial 555-0123 to call Jason.
- Dial 555-0987 to call Carl.

Loops

- for key_variable in dictionary_name:
- # Code block
- # dictionary_name[key_variable]

```
for contact in contacts:
  # Code block
for person in people:
  # Code block
```

Demo

Demo 2 – Basics of Python

- VS Code samples

TheTechPlatform
Where Tech Meets.

Learning Python

Free (Curated)

1. "Python Tutorial" by Corey Schafe (146 videos)

   (2156) Python Tutorials – YouTube

2. "Python Programming for Beginners" by Programming with Mosh (6 hours)

   Python Tutorial - Python Full Course for Beginners (youtube.com)

Paid (Udemy – Curated)

1. "Python for Beginners: Learn Python Programming (Python 3)

   https://www.udemy.com/course/python-programming-projects/

2. "Machine Learning, Data Science and Generative AI with Python

   https://www.udemy.com/course/data-science-and-machine-learning-with-python-hands-on/

Microsoft AI Certifications

Free (Curated)

1. AI – 900

   Microsoft Certified: Azure AI Fundamentals - Certifications | Microsoft Learn

2. AI-102

   Microsoft Certified: Azure AI Engineer Associate - Certifications | Microsoft Learn

3. YouTube Videos (search AI-900, AI-103)

Paid (Udemy – Curated)

1. AI-900

   AI-900 Azure AI Fundamentals Exam Prep In One Day - OCT 2023 | Udemy

2. AI-102

   AI-102 Microsoft Azure AI Solution Complete Exam Prep 2024 | Udemy

TheTechPlatform
Where Tech Meets.