

Good morning!!!

We'll start at 9:30. Until then, make yourself comfortable!

All the exercises will be performed on your own systems, so if you'd like to follow along you can perform the following setup steps to get ready

- Install Python (python.org)
- Install Azure CLI (<https://docs.microsoft.com/cli/azure/install-azure-cli>)
- Clone the docs (<https://github.com/GeekTrainer/ai-reactor-workshop>)



AI for Good

September 2019

Microsoft Reactor | Sydney NSW,
Australia

led by player to
s.load_image("kg.png")
(self):
ialize Dog object and create Text o
g, self).__init__(image = Dog.image
x = games.mouse.x
bottom = games.screen.
re = games.Text(value = 0, size = 2
top = 5, right = game
reen.add(self.score)
a = games.Text(value = 0, size = 2
top = 5, left = game



Shana Matthews

Senior Program Manager
Microsoft
@shanamatthews

Today's workshop is a Microsoft Reactor event.

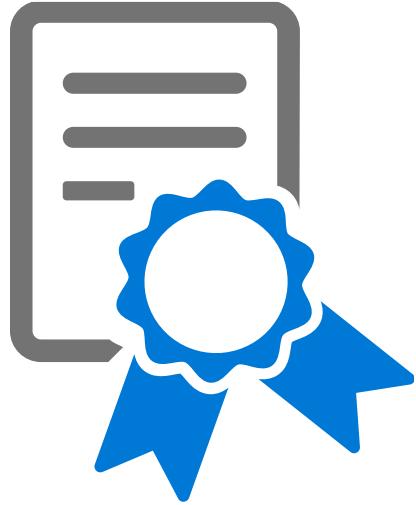


developer.microsoft.com/reactor/
@MSFTReactor on Twitter

Workshop agenda

09:30 am	Event registration
10:00 am	Introductions
10:10 am	Introduction to Microsoft AI
11:00 am	5-minute break
11:05 am	Microsoft Vision and Language Cognitive Services Hands-on
1:00 pm	Lunch break
1:50 pm	Facial Recognition (part 1)
2:35 pm	10-minute break
2:45 pm	Facial Recognition (part 2)
3:30 pm	Conclusion
4:00 pm	Event end

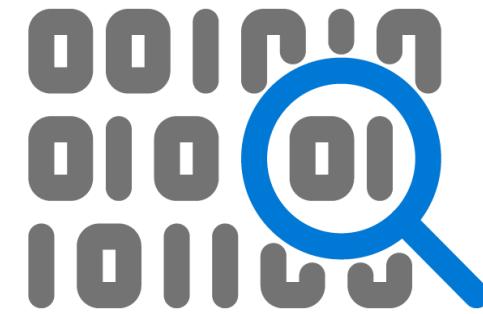
What should you expect from this workshop?



What should you be able to do when
you have completed this course?



What is this workshop like?



How will you be using real AI tools
along the way?

Workshop sections

Section 1: What is Artificial Intelligence (AI)?

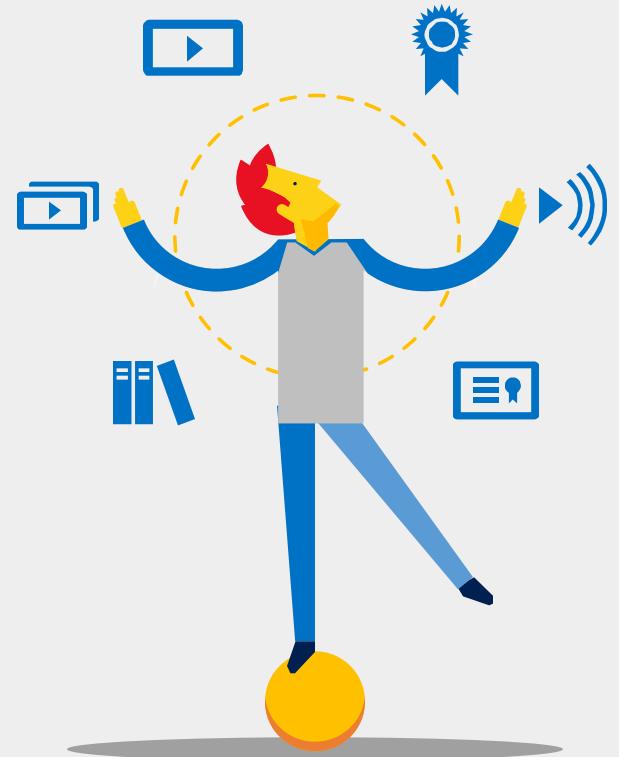
Section 2: Azure Cognitive Vision and Language Services

Section 3: Facial Recognition

Section 4: Conclusion and Next Steps

What is Artificial Intelligence?

Section 1



Section overview

What you will know by the end of the section

What AI is and how it differs from machine learning (ML)

Microsoft's goals for opening up AI

Why you should consider using pretrained AI tools before training your own ML models

What live examples of Azure Cognitive Services can do

Assumptions

Experience with a programming language

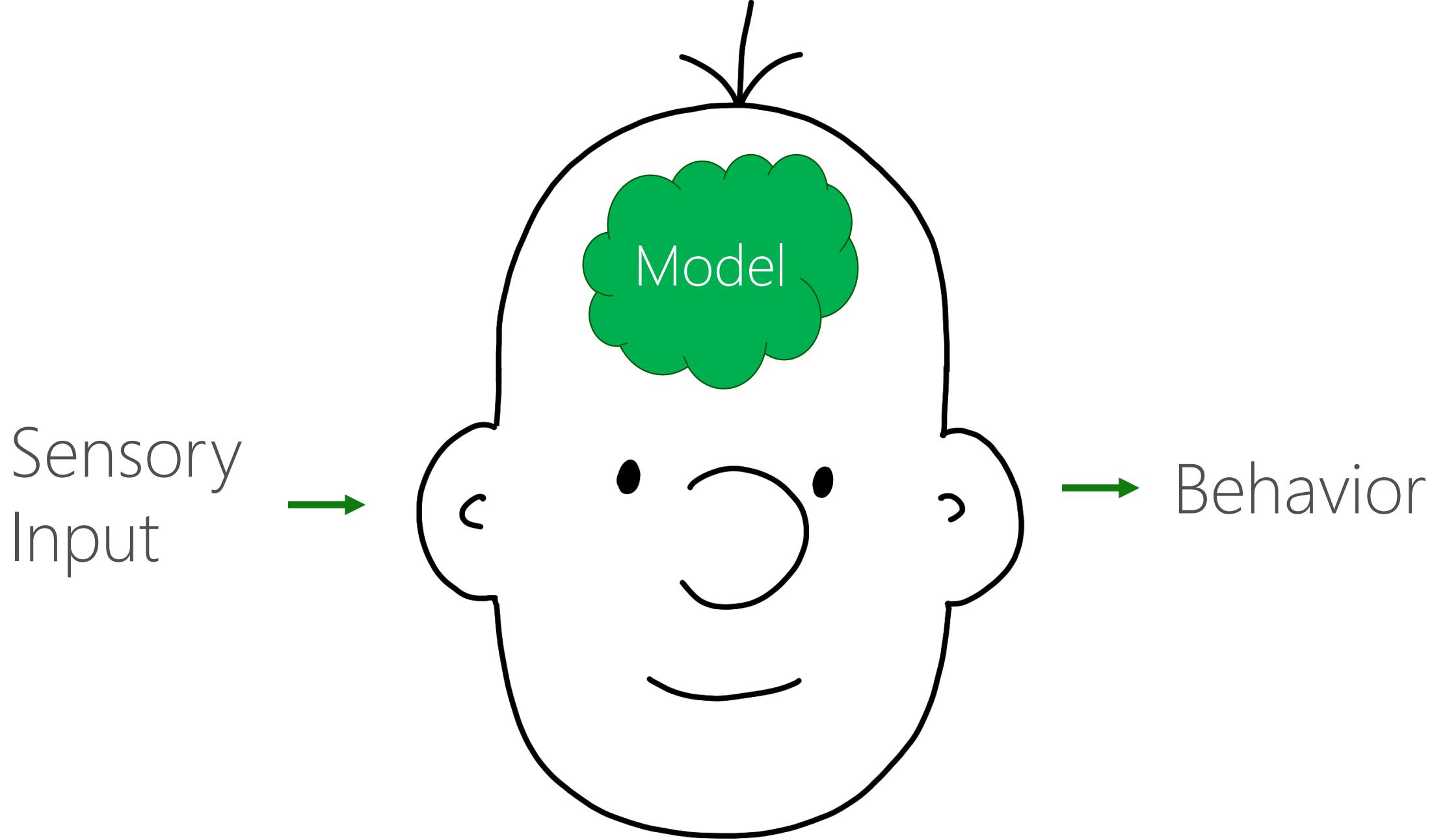
We will use Python (don't worry – we'll cover the basics)

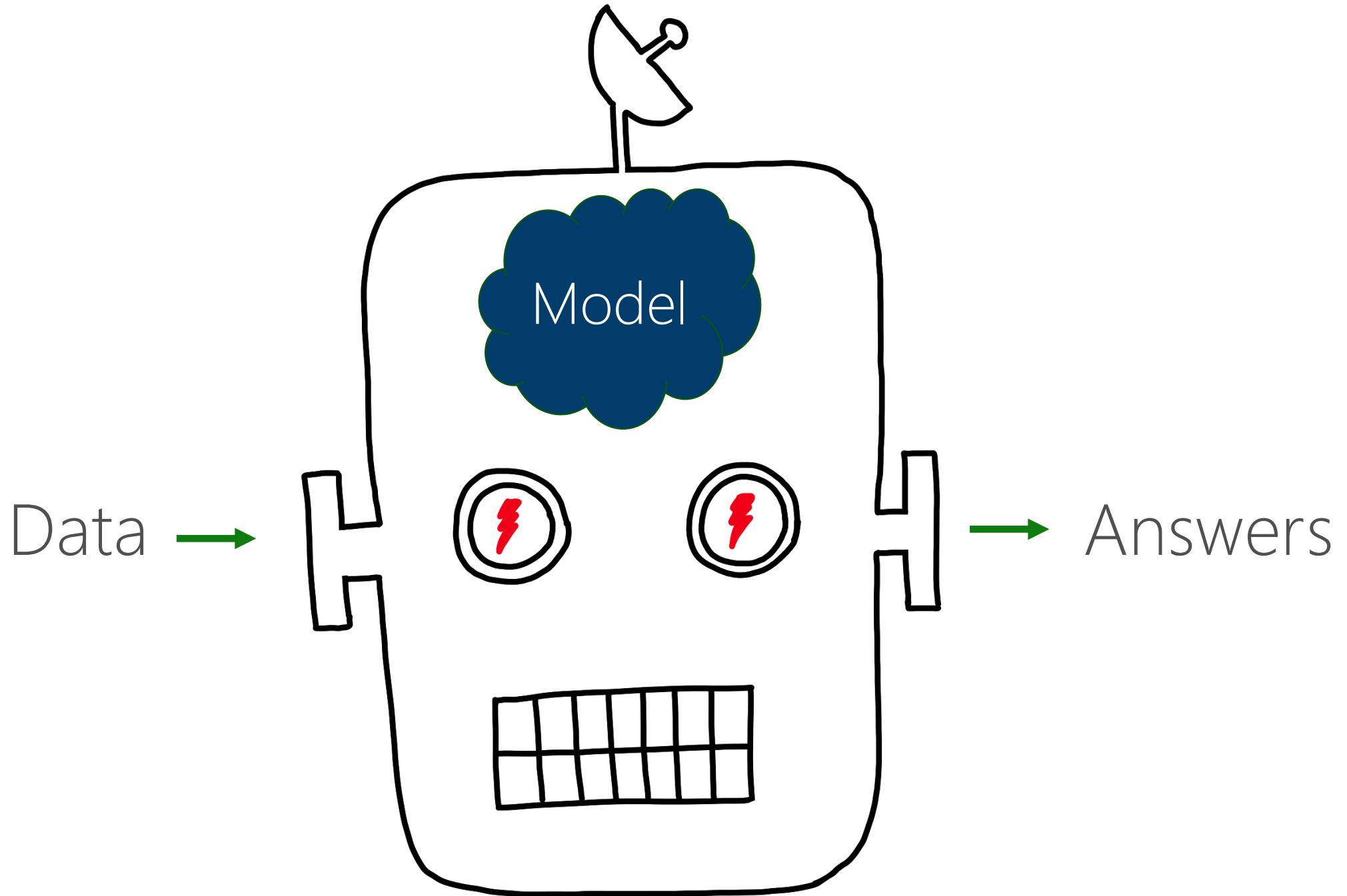
*"Artificial Intelligence is just
whatever a computer can't
do yet..."*

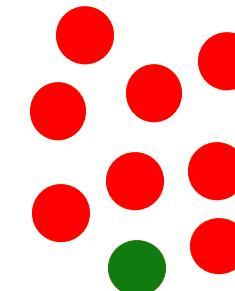
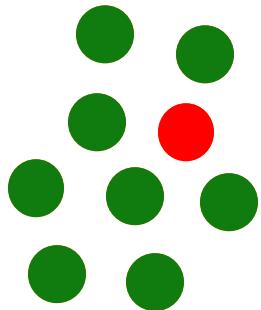
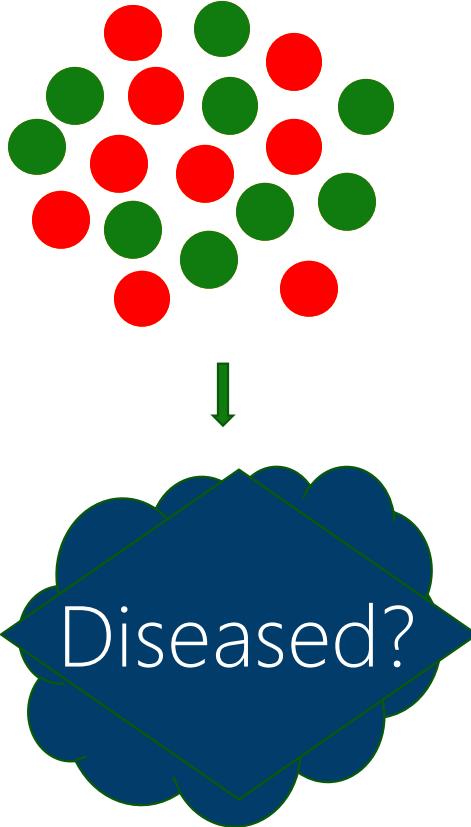
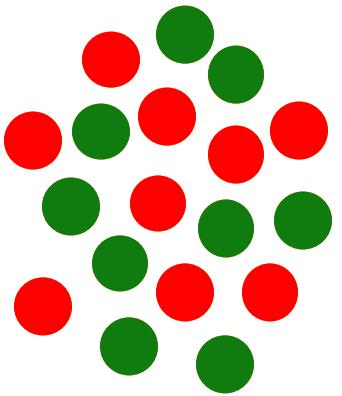
Chris Bishop, Microsoft Research

*"Artificial Intelligence is
computers doing things that we
would normally think of as
intelligent in humans."*

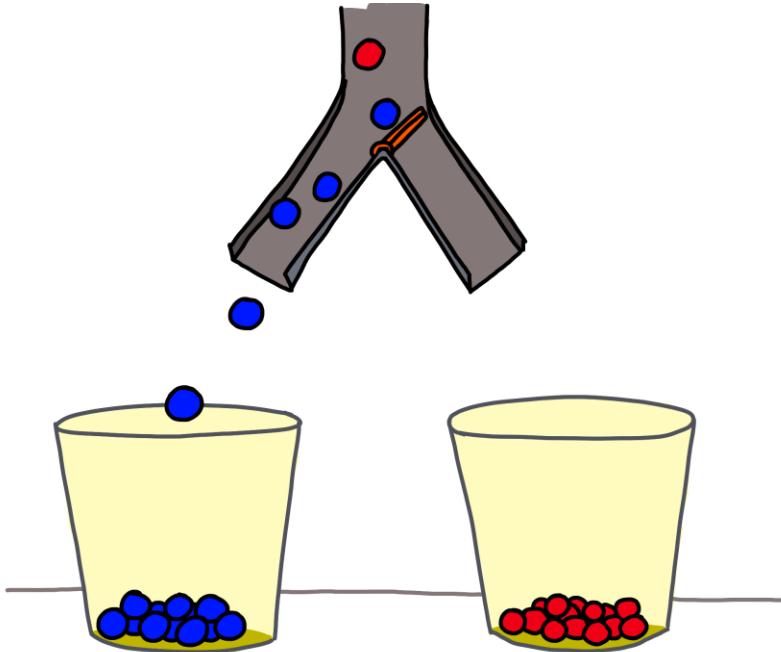
Rick Barazza, Microsoft DX



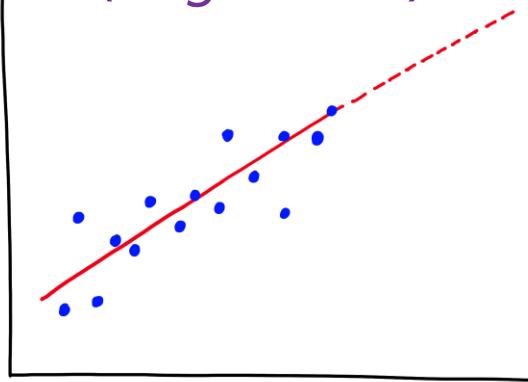




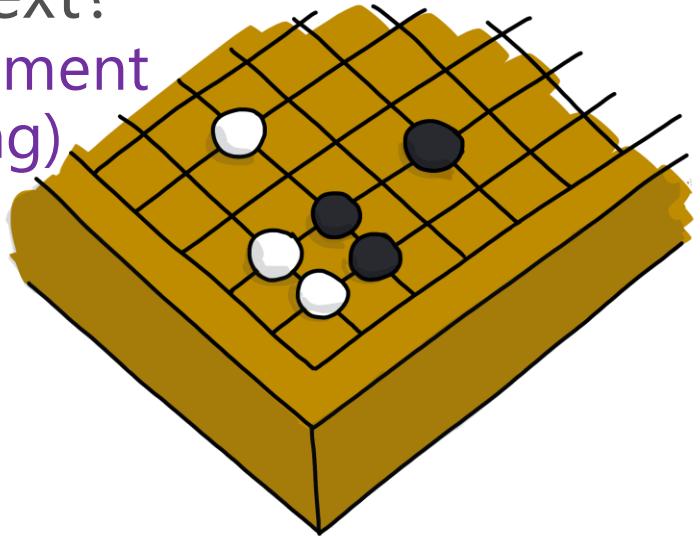
Which category?
(Classification)



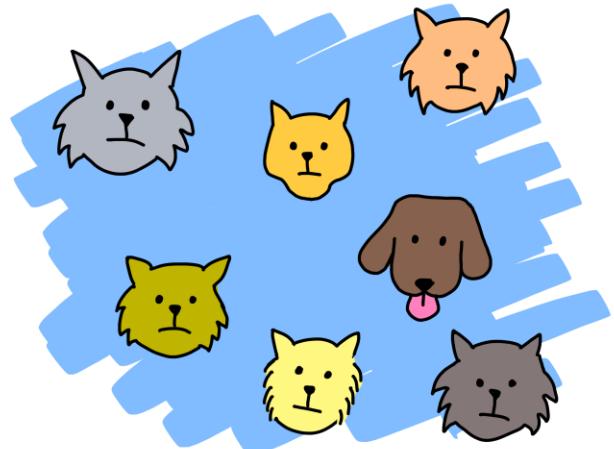
Predict how much/many
(Regression)



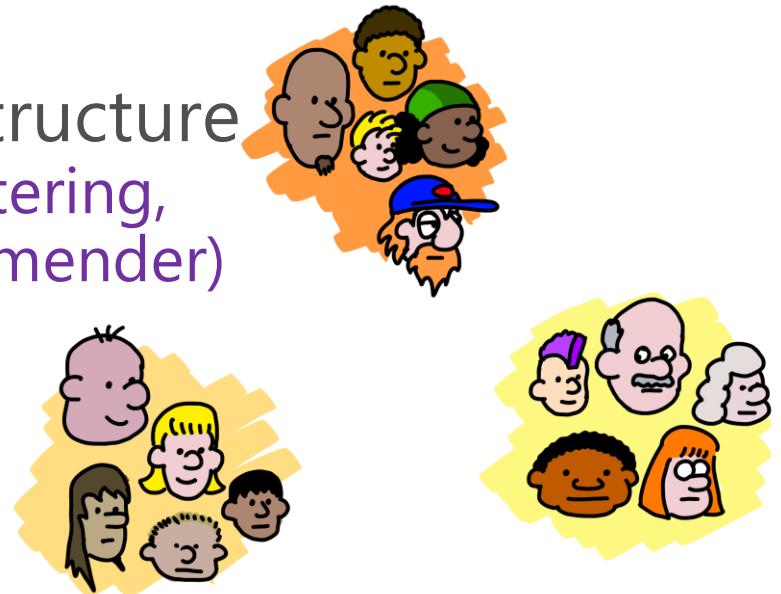
What next?
(Reinforcement Learning)



Is it weird?
(Anomaly)



Data structure
(Clustering, Recommender)



Where is AI being used?



Wild Me + Microsoft

Image analysis: identify species

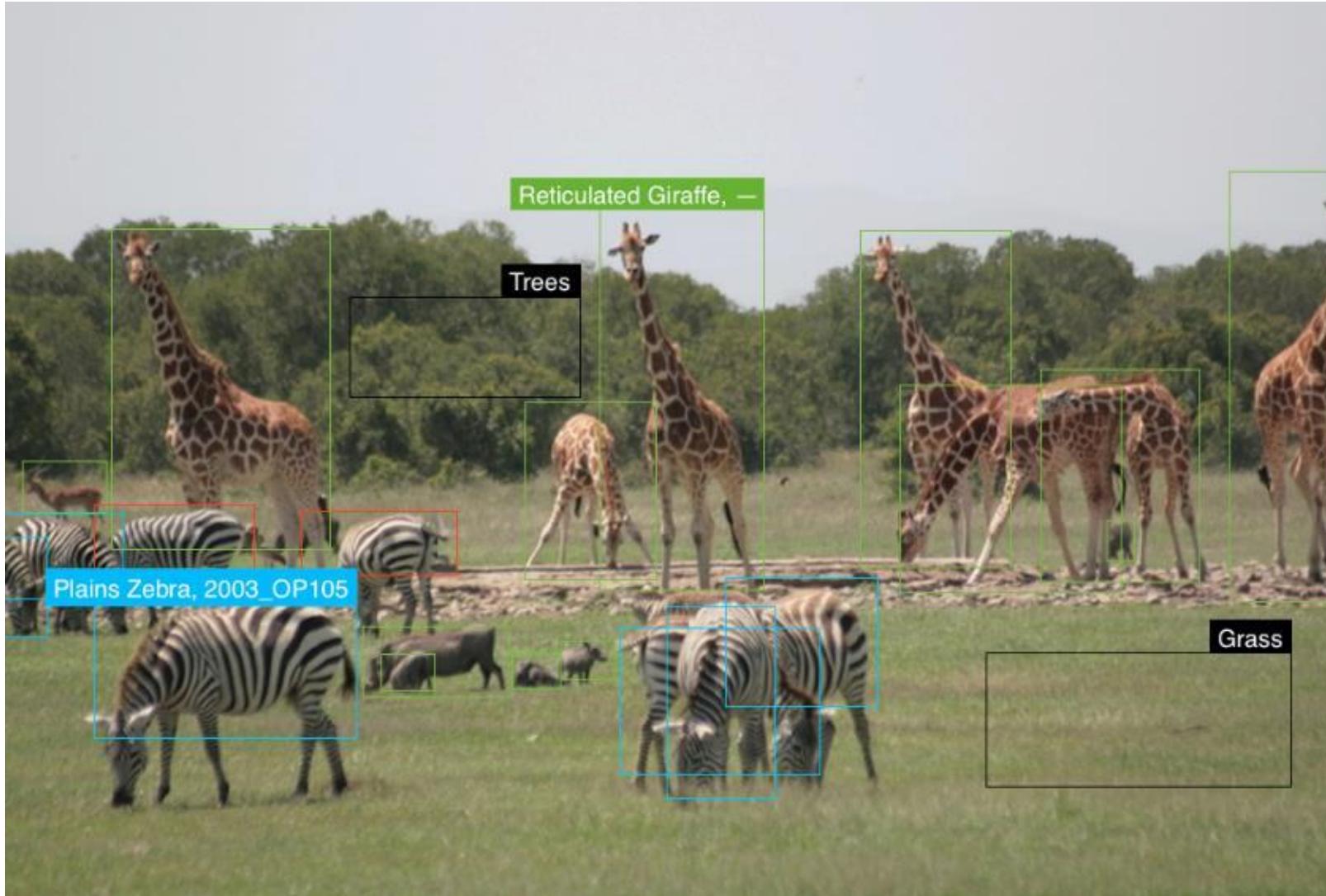
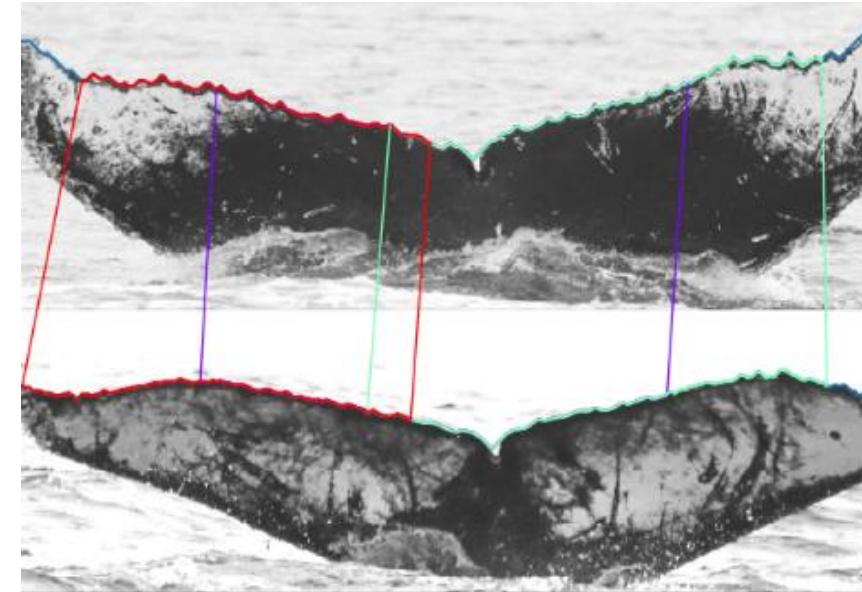
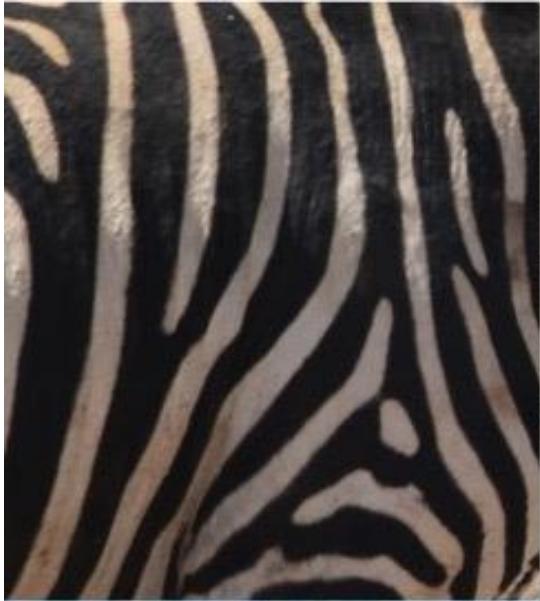


Image analysis: identify unique individuals



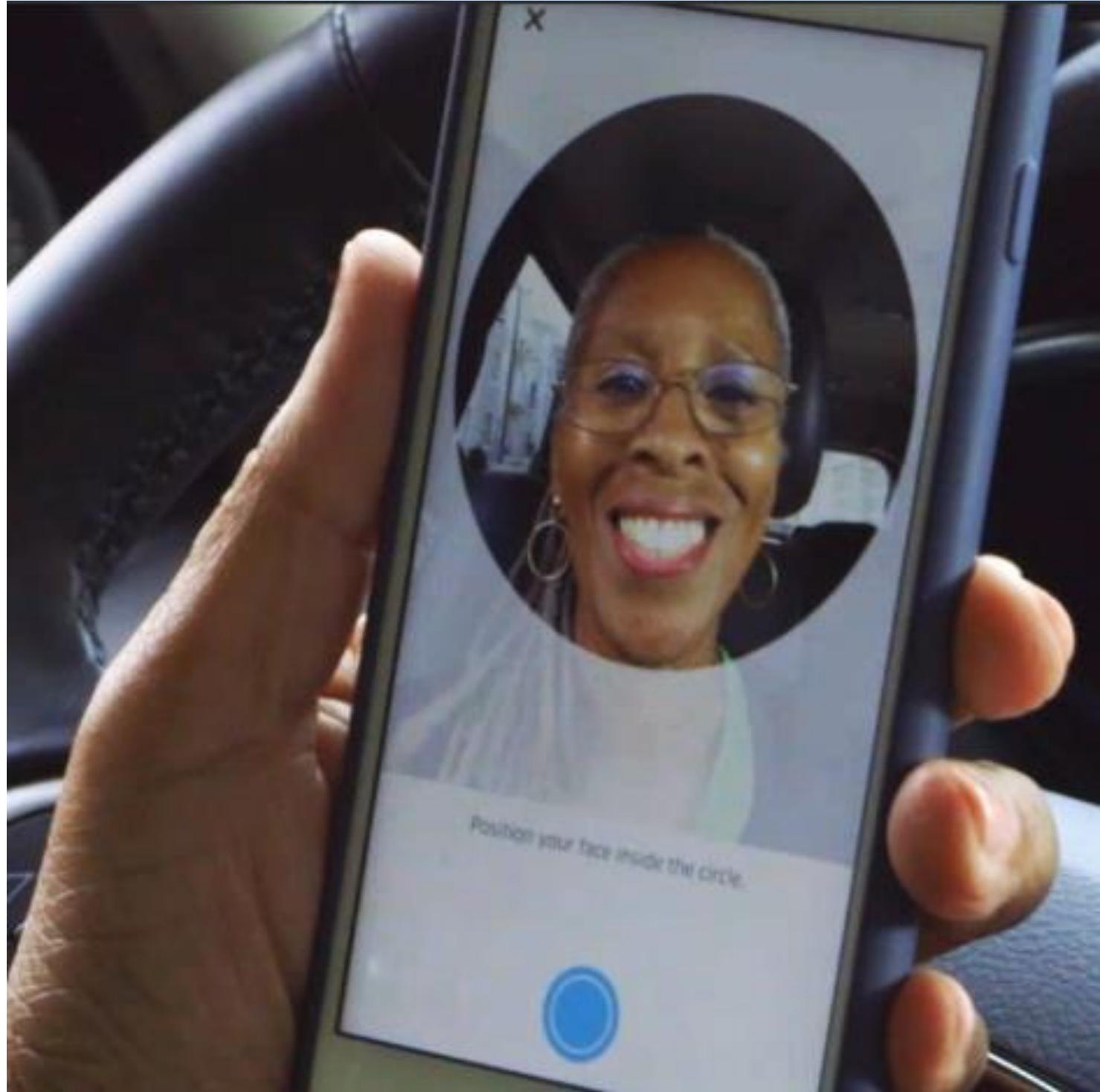


PROGRESSIVE®

Progressive used Microsoft Azure Bot Service and Cognitive Services to quickly and easily build the Flo Chatbot to answer customer questions and provide quotes.

UBER

Uber uses the Microsoft Cognitive Services Face API to help ensure that the driver using the app matches the account on file.





Microsoft seeks to combine the capabilities of computers with human capabilities to enable people to achieve more.





Microsoft's aim:

Making AI
available to
everyone



We are pursuing AI to empower every person and every institution ... so that they can go on to solve the most pressing problems of our society and our economy.

– Satya Nadella, CEO, Microsoft

Defining terms

Artificial intelligence

"In computer science, an artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving." – Wikipedia

Machine learning

"Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task." - Wikipedia

For our purposes

Artificial intelligence

Synonymous with Cognitive Services

Pre-trained (or customizable) models for you to use

Machine learning

Models you will build and train on your own

Useful for more advanced or specialized scenarios

Why work with AI before building ML models?

AI based on deep learning can provide good general-purpose tools for handling a wide variety of challenges:

- Microsoft has already done the hard part by building and training the deep-learning models.
- Easily access these models for whatever your needs are.
- Focus on building applications without getting bogged down with AI models.

With Microsoft Cognitive Services, extremely sophisticated deep neural networks are only an API call away.



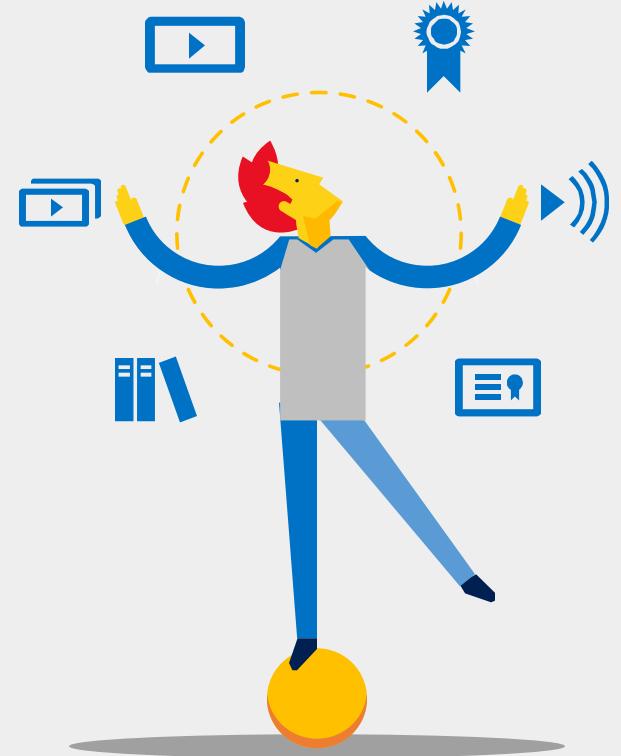
Language Understanding (LUIS)

Custom Speech Service demo

Placeholder

Microsoft Vision and Language Services Hands-on

Section 2



Azure Cognitive Vision and Language Services

What you will accomplish

Set up a development environment to use Azure Cognitive Services.

Learn (or refresh) enough Python and Flask to build the web application.

Use Flask to build a page that accepts photo uploads.

Call the Computer Vision API to extract text from photos.

Use the Translator Text API to translate text extracted from photos.

Deploy the application to Azure.

Assumptions

Experience programming in some computer language



Dual-use AI: can support business or pleasure travel

What is Python?

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. ([Wikipedia](#))

Open source

Flexible for all programming models

Object oriented

Procedural

Functional

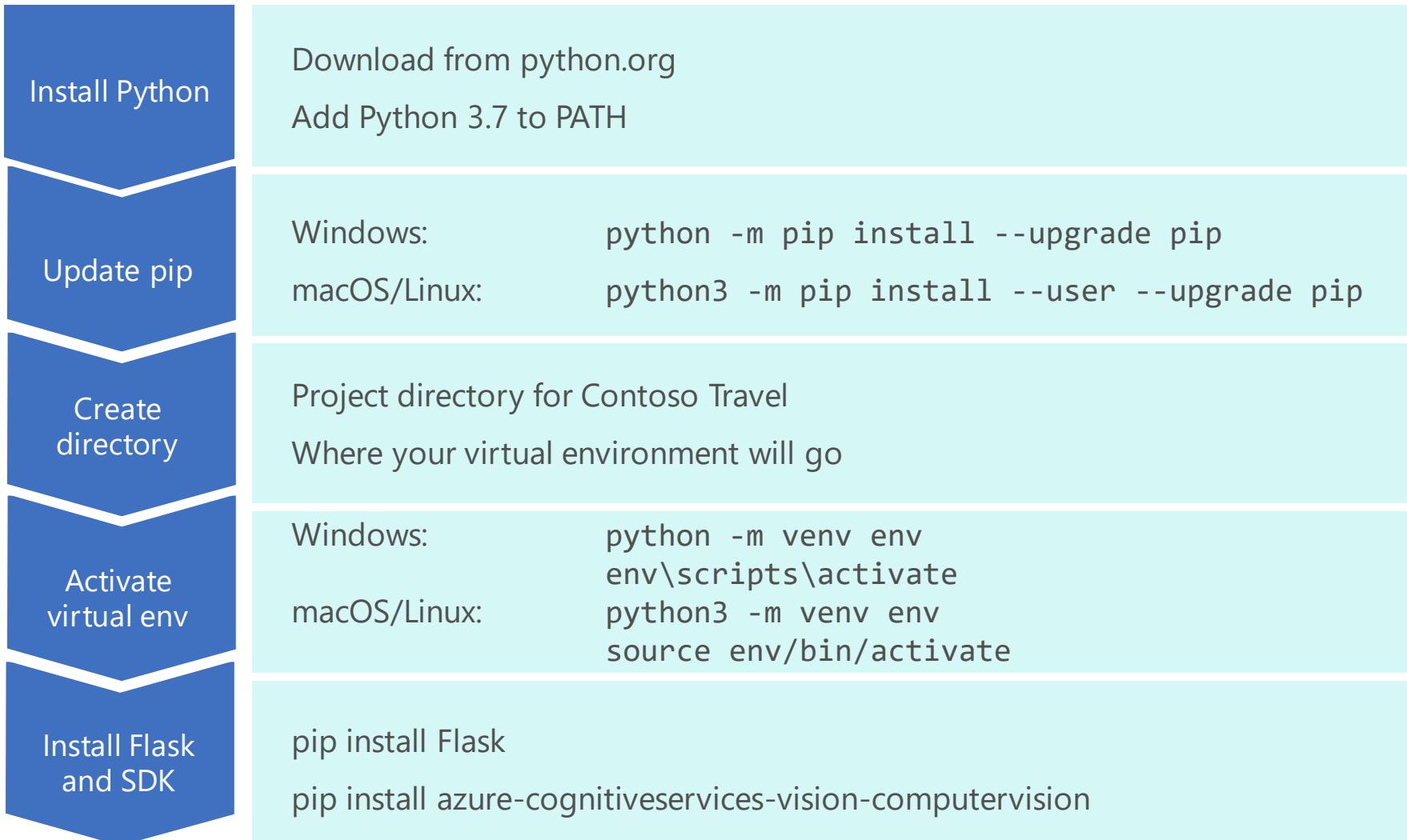
All sorts of uses

Common first language

Science & machine learning

Web development

Create a virtual Python environment



Python overview: variables and data types

- Variables

```
>>> country = "France"
```

- Dictionaries

```
>>> capitals = {"France": "Paris",  
                 "Germany": "Berlin"}
```

- Lists

```
>>> continents = ["Africa", "Asia"]
```

- Add to lists with append ()

```
>>> continents.append("Europe")  
>>> continents  
["Africa", "Asia", "Europe"]
```

Python overview: control flow expressions

- If statements

```
>>> y = 5
>>> if y % 2 == 0:
...     print("Even")
... else:
...     print("Odd")
...
Odd
```

- For loops

```
>>> for continent in continents:
...     print(continent)
...
Africa
Asia
Europe
```

Python overview: functions and exception handling

- Functions

```
>>> def list_length(l):
...     return len(l)
...
>>> list_length(continents)
3
```

- Exception handling

```
>>> while True:
...     try:
...         x = int(input("Please enter a number:
"))
...         break
...     except ValueError:
...         print("Invalid number. Try again...")
...
```

Python overview: importing functions

- Import an entire module:

```
>>> import flask
```

- Import specific functions from a module:

```
>>> from flask import render_template
```

Flask overview: 'Hello, world'

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "<h1>Hello, world</h1>"
```

Flask overview: routing

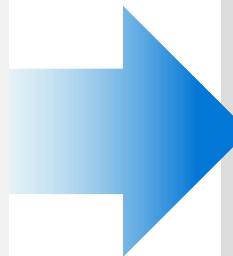
```
from flask import Flask

app = Flask(__name__)

# Define a route for the app's home page
@app.route("/")
def index():
    return "<h1>This is the home page</h1>

# Define a route for the app's About page
@app.route("/about")
def about():
    return "<h1>This is the About page</h1>

# Define a route for the app's Contact Us page
@app.route("/contact")
def contact():
    return "<h1>This is the Contact Us page</h1>"
```



If app hosted at www.contoso.com, it now supports these URLs:

- www.contoso.com/
- www.contoso.com/about
- www.contoso.com/contact

Flask overview: HTML templates

```
from flask import Flask, render_template

app = Flask(__name__)

# Define a route for the app's home page
@app.route("/")
def index():
    return render_template("This is the home page")

# Define a route for the app's About page
@app.route("/about")
def about():
    return render_template("This is the About page")

# Define a route for the app's Contact Us page
@app.route("/contact")
def contact():
    return render_template("This is the Contact Us page")
```

Three templates:

- **index.html**
- **about.html**
- **contact.html**

Flask overview: HTML templates with defined variables

```
from flask import Flask, render_template  
  
app = Flask(__name__)  
  
# Define a route for the app's home page  
@app.route("/")  
def index():  
    return render_template("master.html", message="This is the home page")  
  
# Define a route for the app's About page  
@app.route("/about")  
def about():  
    return render_template("master.html", message="This is the About page")  
  
# Define a route for the app's Contact Us page  
@app.route("/contact")  
def contact():  
    return render_template("master.html", message="This is the Contact Us page")
```

One template, **master.html**:

- <h1>{{ message }}</h1>

Flask overview: Jinja control flow expressions

- If statement

```
{% if message %}  
    <h1>{{ message }}</h1>  
{% else %}  
    <h1>No message specified</h1>  
{% endif %}
```

- If statement to conditionally execute JavaScript code:

```
{% if message %}  
    <h1>{{ message }}</h1>  
{% else %}  
    <script type="text/javascript">  
        window.alert("No message specified");  
    </script>  
{% endif %}
```

Flask overview: flashing messages with control of flow

- app.py

```
import os
from flask import Flask, render_template, flash

app = Flask(__name__)
app.secret_key = os.urandom(1)

# Define a route for the app's home page
@app.route("/")
def index():
    flash("This is an error message")
    return render_template("index.html")
```

- index.html

```
{% with messages = get_flashed_messages() %}
  {% if messages %}
    <script type="text/javascript">
      window.alert("{{ messages[0] }}");
    </script>
  {% endif %}
  {% endwith %}
```

Flask overview: static files

- Referenced via HTML:

```
<link rel="stylesheet" href="/static/main.css">

```

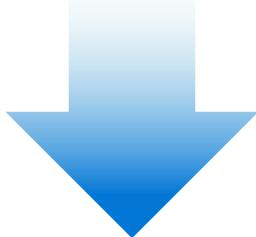
- Resolve URLs via Flask's url_for() function:

```
<link rel="stylesheet"
      href=url_for('static',filename='main.css')>
<img src=url_for('static', filename='banner.jpg')>
```

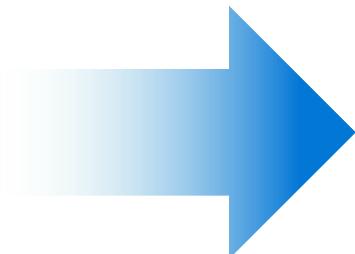
Create a site that supports photo uploads

Download starter files for development:

```
git clone https://github.com/MicrosoftDocs/mslearn-ai-web-app-flask.git .
```



- app.py
- templates/index.html
- static/main.css
- static/banner.jpg
- static/placeholder.jpg



app.py

```
from flask import Flask, render_template  
  
app = Flask(__name__)  
  
# Define route for the app's one and only page  
@app.route("/")  
def index():  
    return render_template("index.html")
```

Create a site that supports photo uploads



- Set Flask to run in dev mode:

Windows: set FLASK_ENV=development

macOS/Linux: export FLASK_ENV=development

- Start Flask:

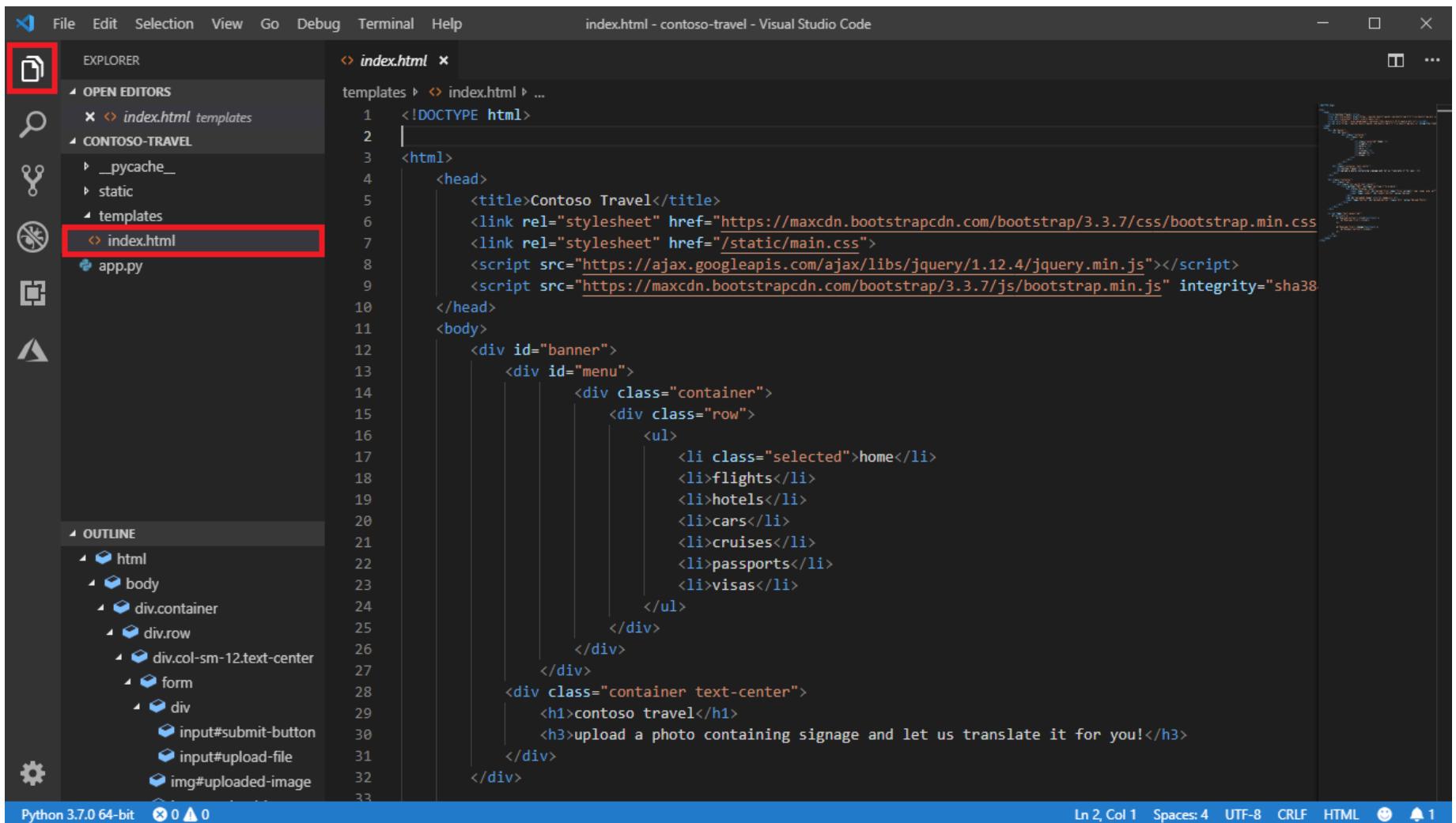
```
flask run
```

- Open website locally:

<http://localhost:5000>

Create a site that supports photo uploads

Edit **index.html**
and **app.py** in
Visual Studio
Code to make
the site
functional.



Create a site that supports photo uploads

- Paste into **index.html** immediately before the closing </body> tag:

```
<script type="text/javascript">
$(function() {
    $("#upload-button").click(function() {
        $("#upload-file").click();
    });

    $("#upload-file").change(function() {
        $("#submit-button").click();
    });
});
</script>
```

- Use jQuery to display an open-file dialog user clicks **Upload Photo**:

```
<div style="display: none">
    <input type="file" id="upload-file" name="file"
accept=".jpg,.jpeg,.png,.gif">
    <input type="submit" id="submit-button"
value="Upload">
</div>
```

Create a site that supports photo uploads

- Open **app.py** in Visual Studio Code and replace its contents with the following statements:

```
import base64
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":
        # Display the image that was uploaded
        image = request.files["file"]
        uri = "data:;base64," + base64.b64encode(image.read()).decode("utf-8")

    else:
        # Display a placeholder image
        uri = "/static	placeholder.png"

    return render_template("index.html", image_uri=uri)
```

Create a site that supports photo uploads

- Replace /static/placeholder.png on that line with {{ image_uri }} in **index.html**:

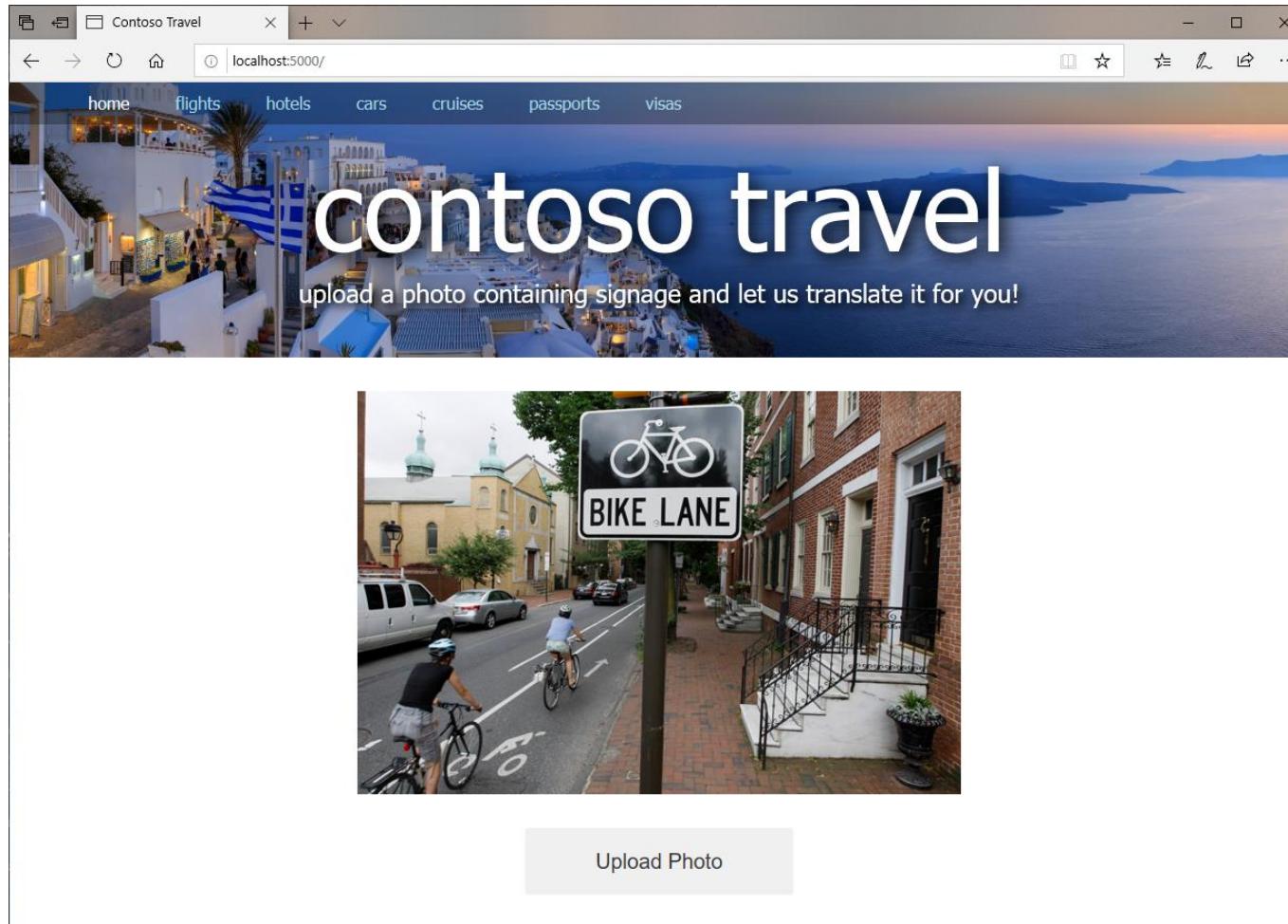
```

```

- When the page is first requested, image_uri points to the placeholder image.
- When the page is requested again because an image was uploaded, image_uri holds the data URI created from the image.
- Consequently, when the user uploads a photo, the photo replaces the placeholder image on the page.

Create a site that supports photo uploads

Upload a photo from your file system:



Extract text from photos with Azure Cognitive Services

From the command prompt:



```
az group create --name contoso-travel-rg --location northcentralus
```

```
az cognitiveservices account create --resource-group contoso-travel-rg --name computer-vision --location northcentralus --kind ComputerVision --sku F0 --yes
```

```
az cognitiveservices account keys list --resource-group contoso-travel-rg --name computer-vision --query key1 --output tsv
```

```
az cognitiveservices account keys list --resource-group contoso-travel-rg --name computer-vision --query endpoint --output tsv
```

Be sure to save your API key and endpoint URL to use later.

Extract text from photos with Azure Cognitive Services

Modify the site to use the Computer Vision API

- Make changes to **app.py** and **index.html** specified in your manual.
- Create environment variables containing your API key and endpoint URL:

Windows

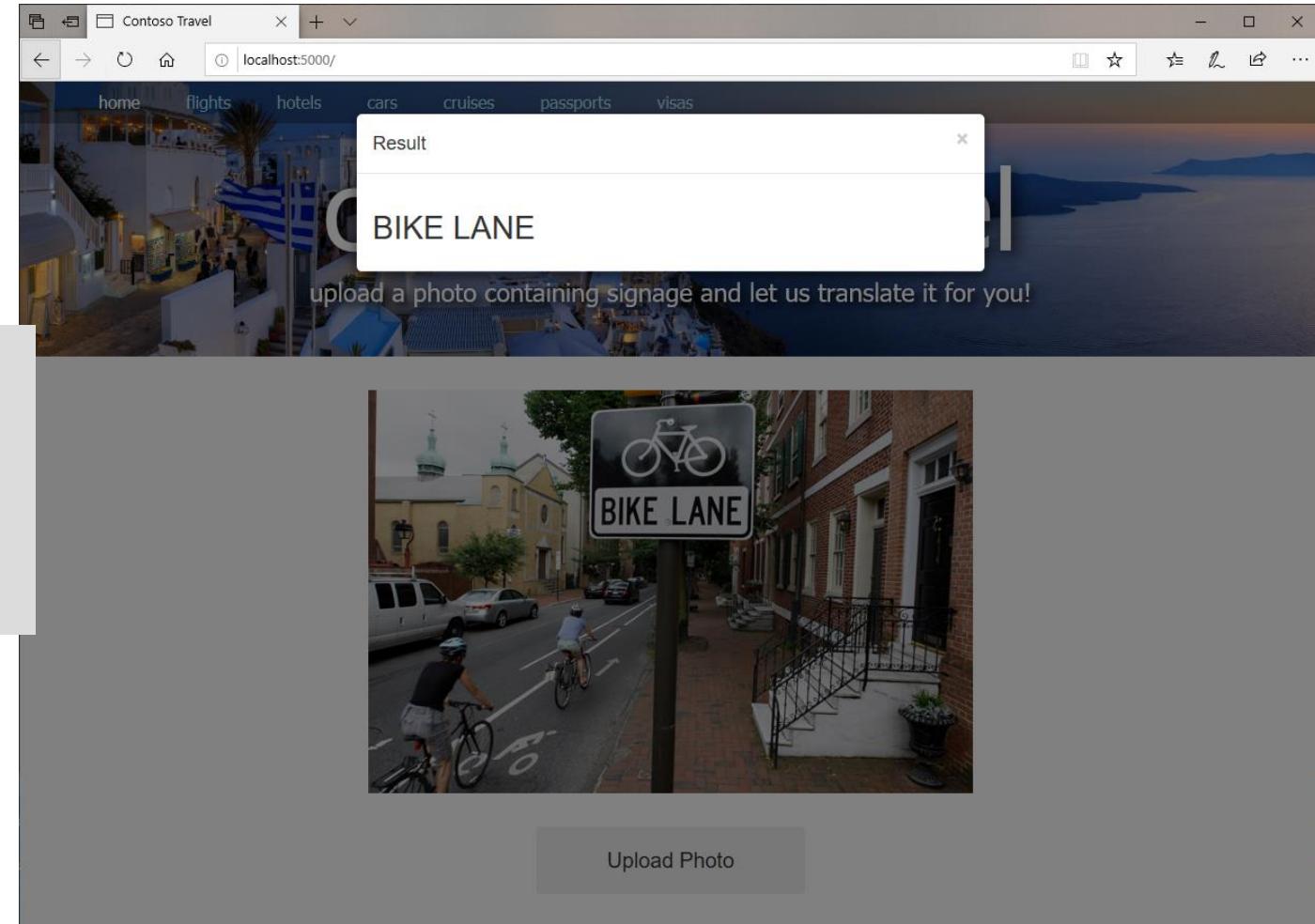
```
set VISION_KEY=computer_vision_api_key  
set VISION_ENDPOINT=computer_vision_endpoint
```

macOS/Linux

```
export VISION_KEY=computer_vision_api_key  
export VISION_ENDPOINT=computer_vision_endpoint
```

Extract text from photos with Azure Cognitive Services

- Open website locally:
<http://localhost:5000>
- Verify page extracts text



Use Azure Cognitive Services to translate text

- Subscribe to the Translator Text API and place the resulting resource in your resource group.

```
az cognitiveservices account create --resource-group contoso-travel-rg  
--name translator-text --location global --kind TextTranslation --sku F0 --yes
```

```
az cognitiveservices account keys list --resource-group contoso-travel-rg  
--name translator-text --query key1 --output tsv
```

Be sure to save your API key to use later.

Use Azure Cognitive Services to translate text

Modify the site to use the Translator Text API

- Make changes to **app.py** and **index.html** specified in your manual.
- Create environment variables containing your API key:

Windows

```
set TRANSLATE_KEY=translator_text_api_key
```

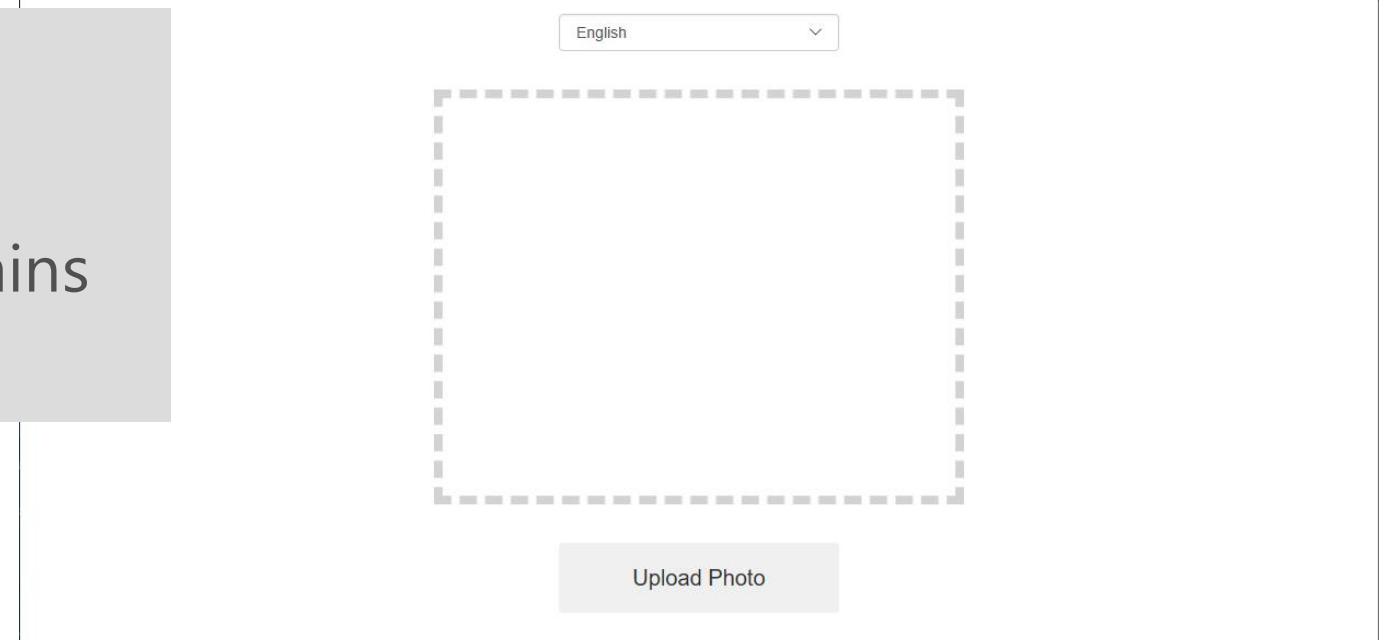
macOS/Linux

```
export TRANSLATE_KEY=translator_text_api_key
```

Use Azure Cognitive Services to translate text

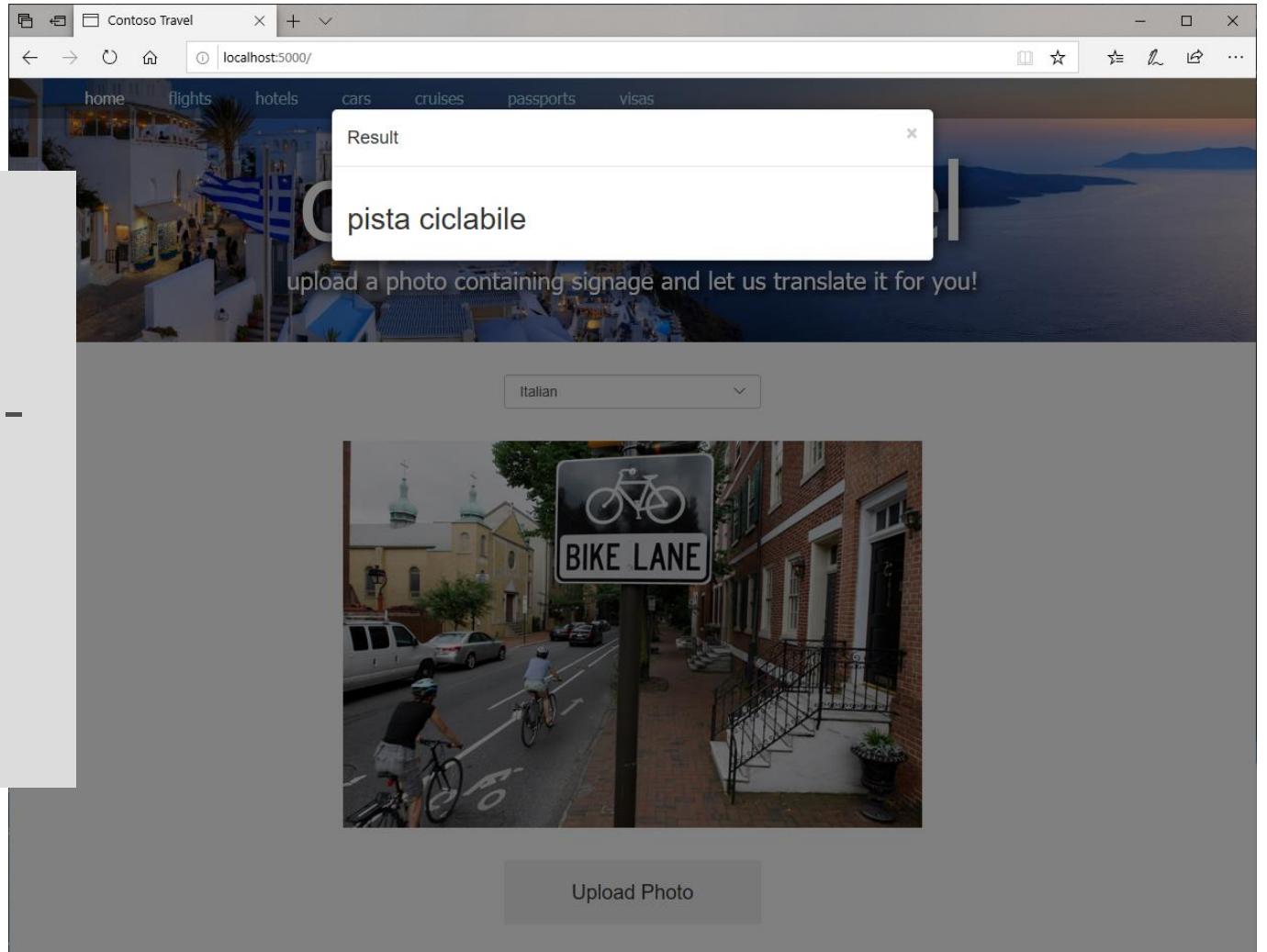


- Open website locally:
`http://localhost:5000`
- Verify that the page now contains a drop-down list.



Use Azure Cognitive Services to translate text

- Open website locally:
<http://localhost:5000>
- Select a language from the drop-down list.
- Upload picture containing text.
- Verify that text is translated.



Deploy the website to Azure

- Create an Azure App Service
 - In the project directory, create a text file named **requirements.txt** containing the following:

```
requests  
Flask  
azure-cognitiveservices-vision-computervision
```

Note: APP_NAME must be unique with Azure.

- From the command line, run the following:

```
az webapp up -n APP_NAME --resource-group contoso-travel-rg --location northcentralus
```

Add application settings

- From the command line, run the following (replacing APP_NAME and the key):

```
az webapp config appsettings set -g contoso-travel-rg -n APP_NAME  
--settings VISION_KEY=computer_vision_api_key  
az webapp config appsettings set -g contoso-travel-rg -n APP_NAME  
--settings VISION_ENDPOINT=computer_vision_endpoint  
az webapp config appsettings set -g contoso-travel-rg -n APP_NAME  
--settings TRANSLATE_API_KEY=translate_api_key
```

Deploy the website to Azure

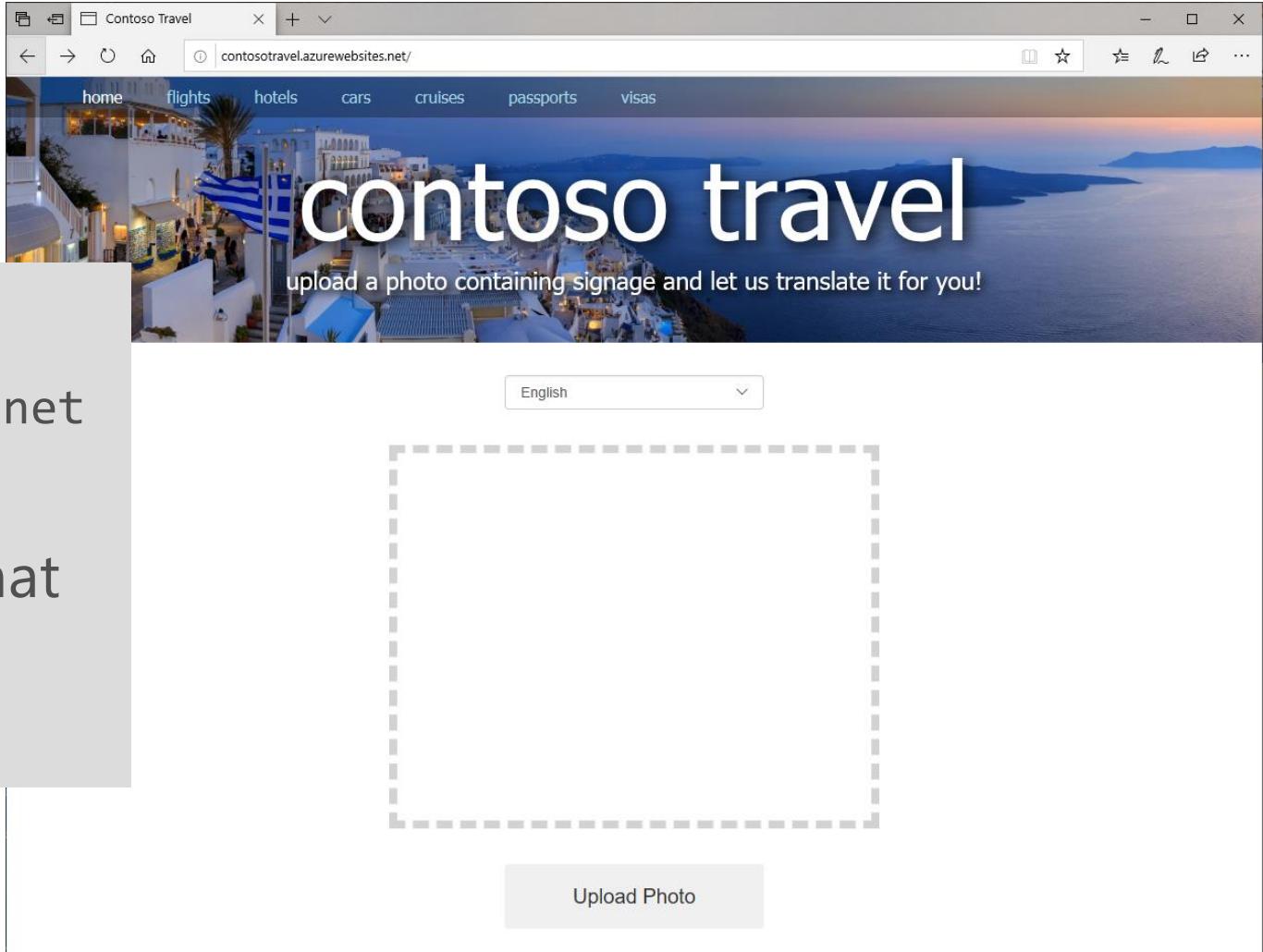
The screenshot shows the 'Application settings (Classic)' page for an Azure App Service named 'contosotranslate'. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Security. Under Deployment, there is a 'Quickstart' link. The main area displays a message about the deprecation of Application Settings (Classic). It features a 'Show values' button and a table of application settings.

APP SETTING NAME	VALUE	SLOT SETTING	DELETE
SCM_DO_BUILD_DURING...	Hidden value. Click to edit.	<input type="checkbox"/>	
TRANSLATE KEY	Hidden value. Click to edit.	<input type="checkbox"/>	
	Click to edit.	<input type="checkbox"/>	
	Click to edit.	<input type="checkbox"/>	
	Click to edit.	<input type="checkbox"/>	

- Optional: go to <http://portal.azure.com>
- Open Azure App Service created by az webapp up

Deploy the website to Azure

- Open the website on Azure:
`https://APP_NAME.azurewebsites.net`
- Verify that the site appears and functions in your browser and that it looks exactly as it did when running locally.

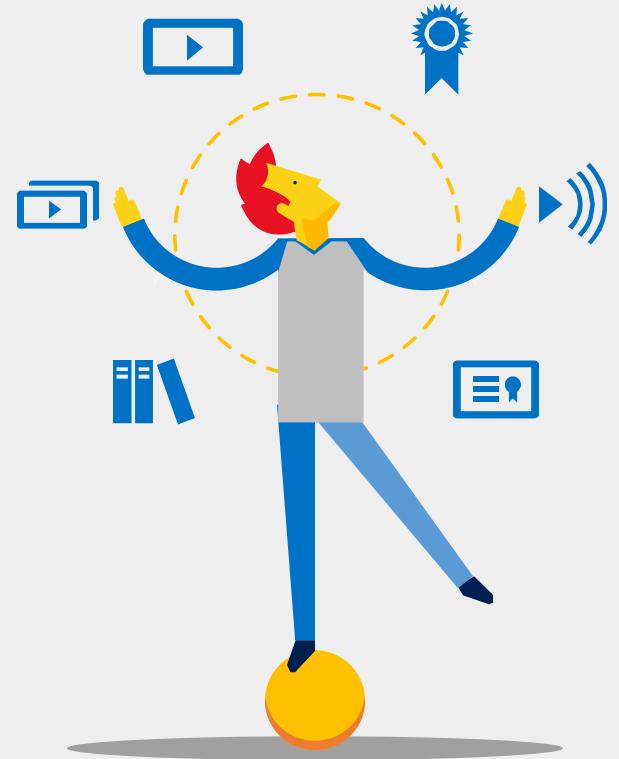


Section summary

- You learned the basics of building Web sites in Python using the Flask framework
 - Python code goes in a file named app.py and that code in that file calls Flask functions such as render_template() and flash()
 - Programmatically inject content into HTML templates using Flask expressions, and how to use @app.route() to connect functions in your code to routes in Flask
- You discovered that you can use Azure Cognitive Services to build intelligent applications infused with AI
- You experienced how easy it is to use the Azure CLI's az webapp up command to upload a Web site that you have written and tested locally to an Azure App Service

Facial Recognition

Section 3



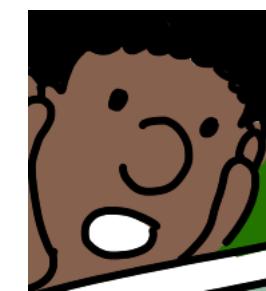
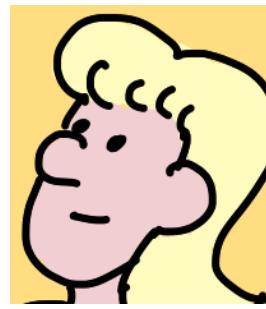
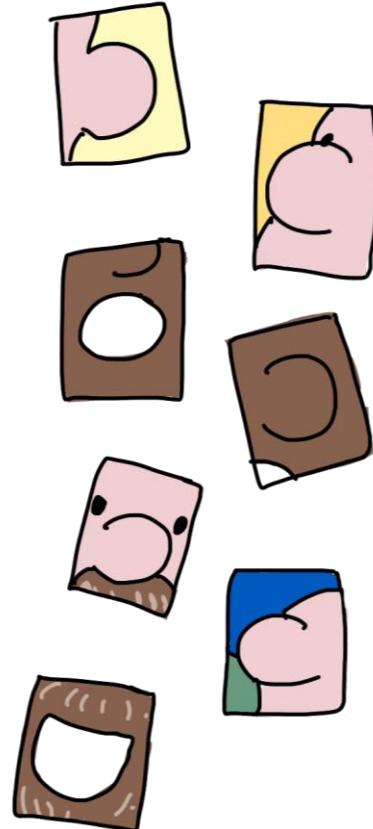
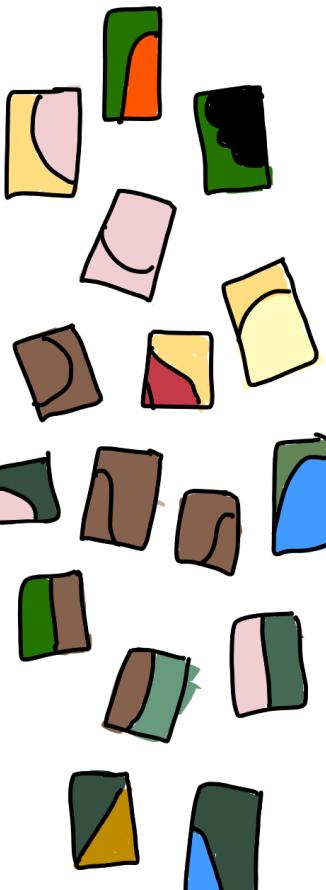
Facial Recognition

What you will accomplish by the end of this section

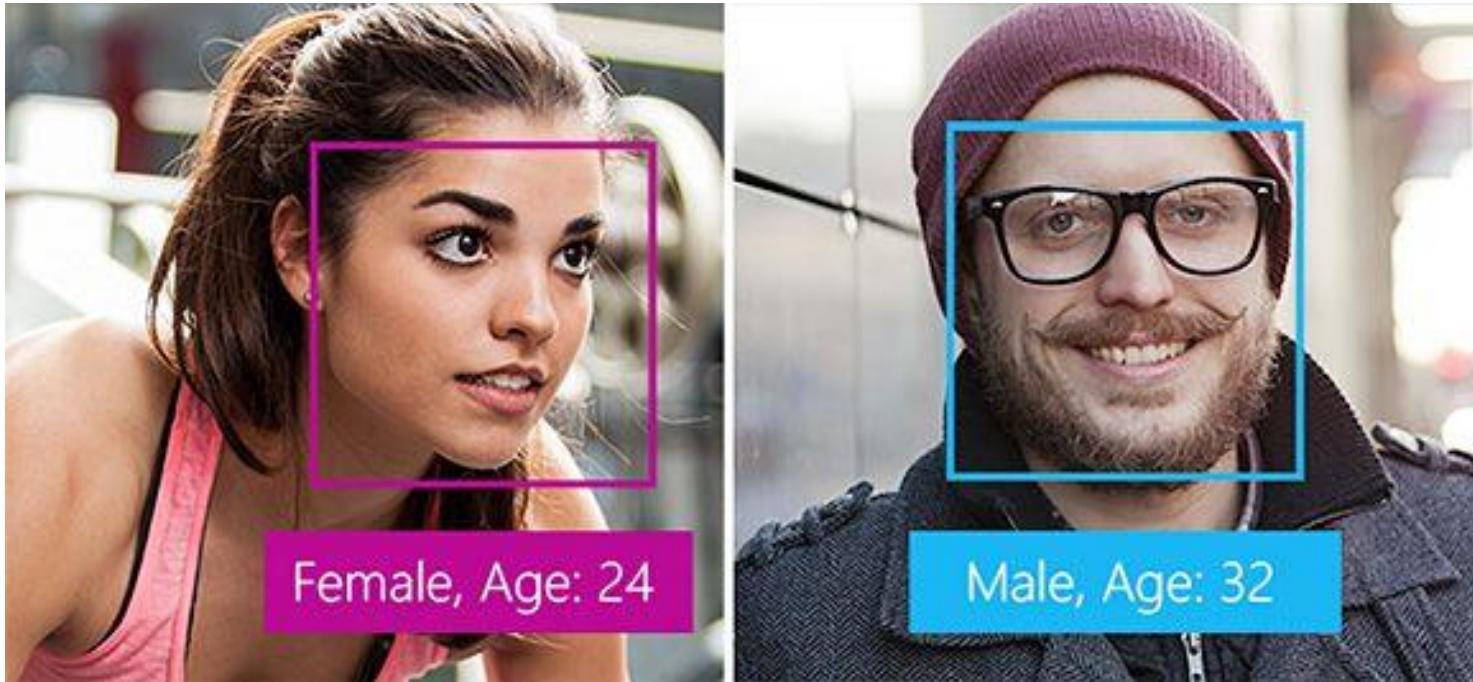
- Learn about the Face API and AI-driven facial recognition in general.
- Discuss ethical considerations about AI in the context of facial recognition.
- Use the Face API hands-on.

Assumptions for this section

- None. Although this section assumes that you have completed the Microsoft Vision and Language Cognitive Services Hands-on section, that is not a prerequisite for getting the most out of this section.



The Face API: How does it work?

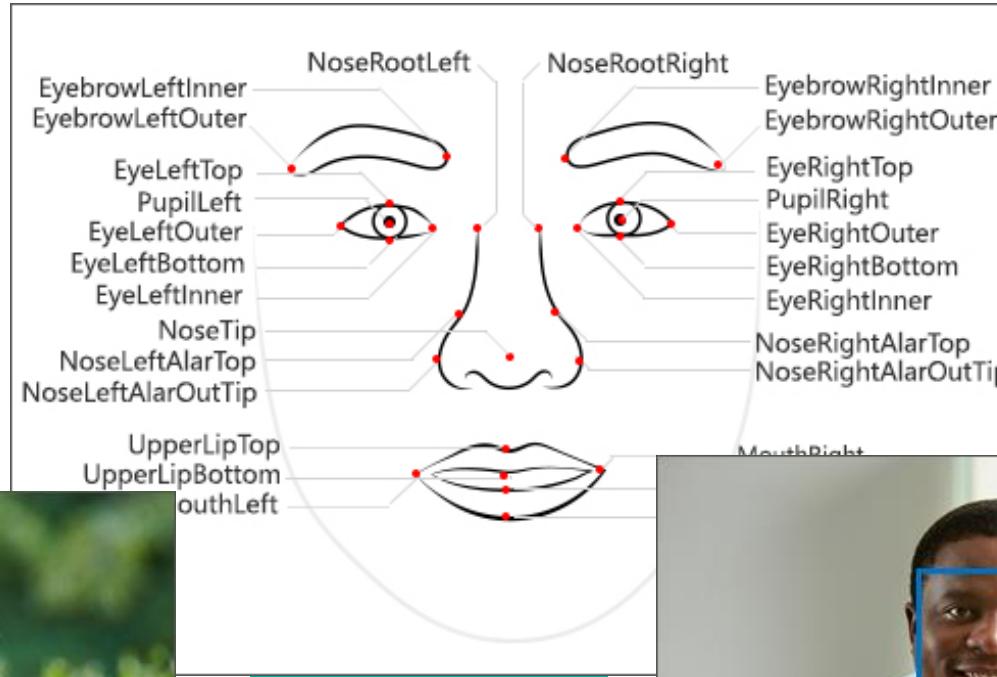


Types of facial data

Locations



TOP: 54, LEFT: 364, BOTTOM: 131, RIGHT: 438



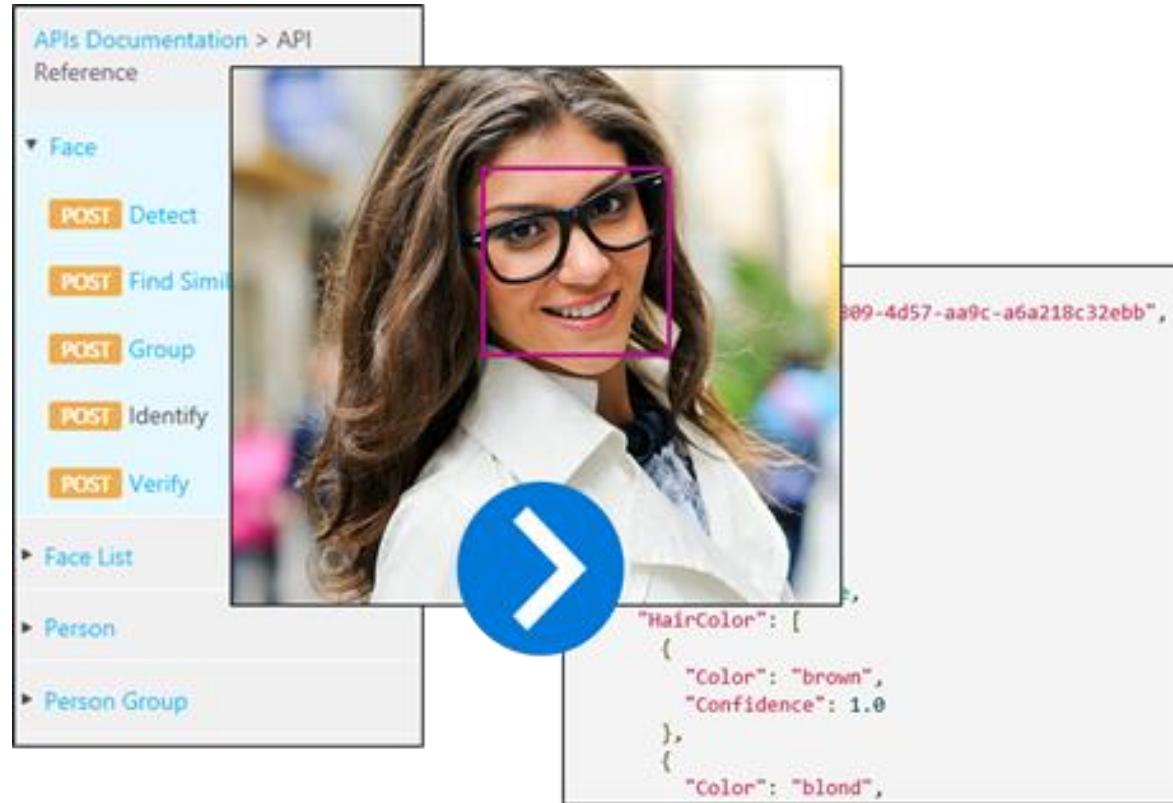
Landmarks

Attributes

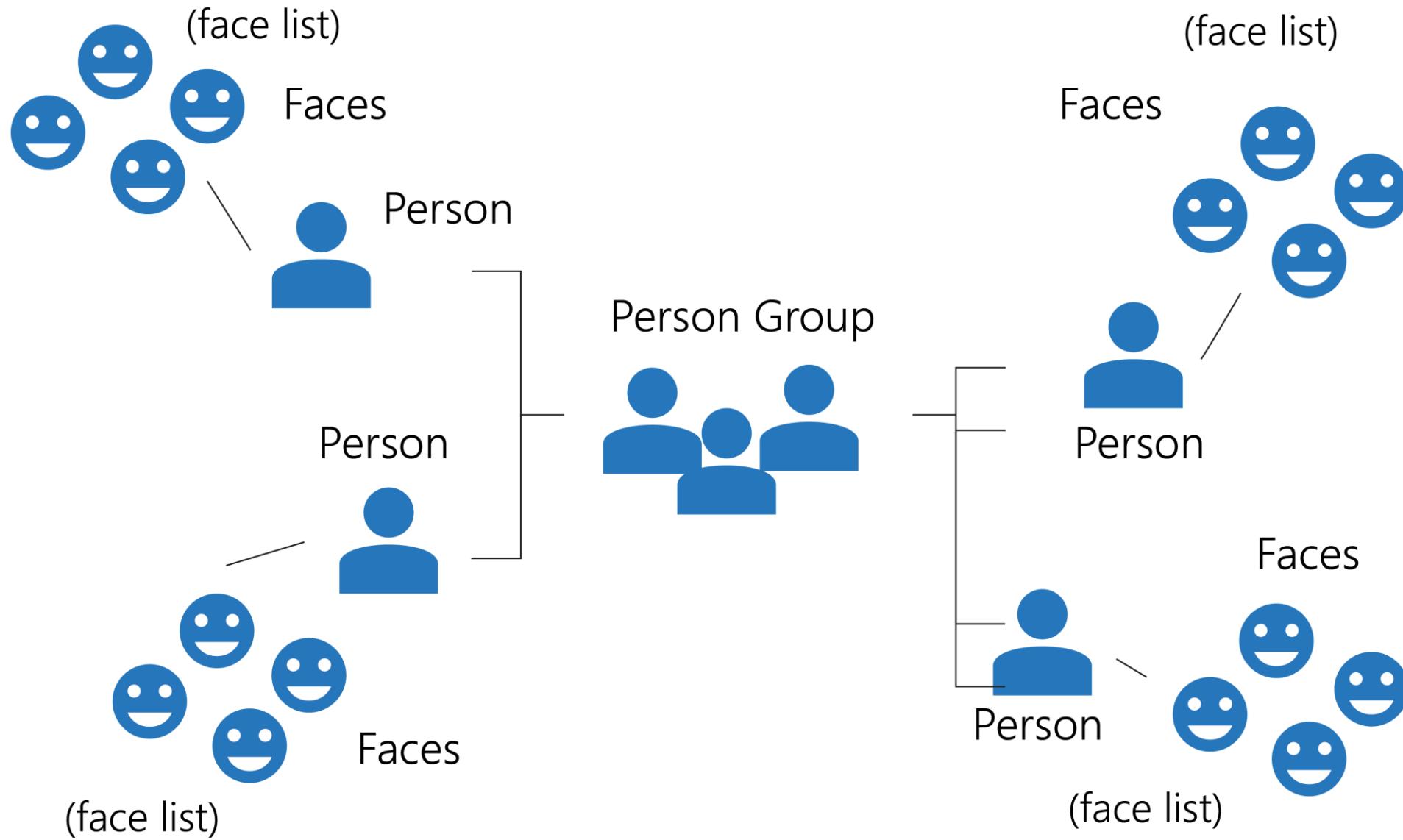


HAIR COLOR: BLACK, AGE: 29.7 YEARS

Overview of facial recognition



Introduction to face lists



More on face lists

PERSON GROUP: **myInnerCircle**

Doris	Hank	Tia
		
		
		

Face identification



Face similarity

Considerations for face data



Test the Face Detection API

Send

Response status

200 OK

Response latency

1364 ms

Response content

```
Pragma: no-cache
apim-request-id: b51e799c-b0db-4015-9c11-d789e1f8b67d
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
x-content-type-options: nosniff
Cache-Control: no-cache
Date: Mon, 07 Jan 2019 20:26:58 GMT
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Content-Length: 115
Content-Type: application/json; charset=utf-8
Expires: -1
```

```
[{
  "faceId": "bfd7c13c-9f81-41b9-842a-184c08cc82f5",
  "faceRectangle": {
    "top": 303,
    "left": 920,
    "width": 161,
    "height": 161
  }
}]
```

Your HTTP response should look something like this.

Section summary

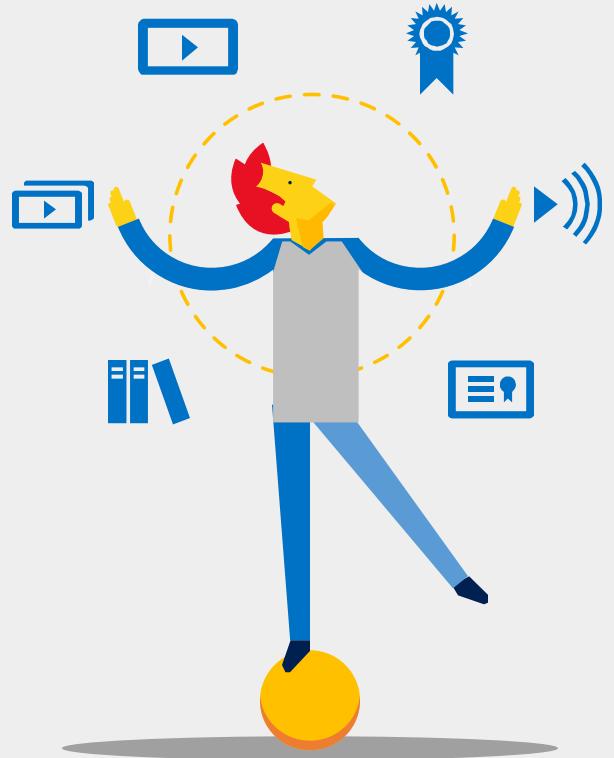
You learned the basics of how facial recognition works.

You also got to see the Emotion API in action and use the Face API.

You were introduced to the ethics of AI in the context of facial recognition.

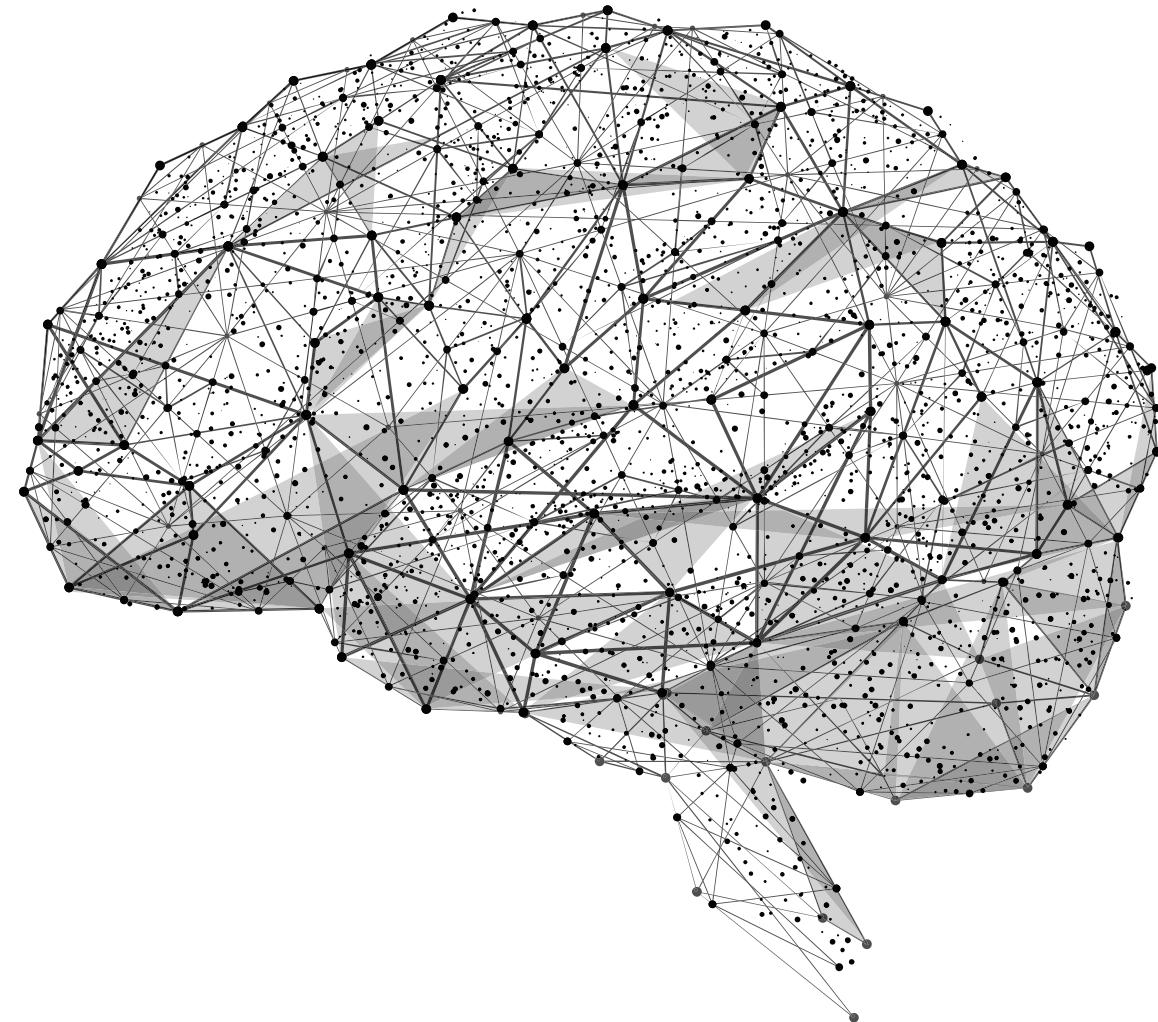
Conclusion

Section 4



AI can be your most powerful tool

- AI based on deep learning can produce excellent results in a variety of use cases.
- With Microsoft Cognitive Services, the cost, time, and expertise usually required by deep learning is not an issue.
- You can easily incorporate AI into your apps via APIs.

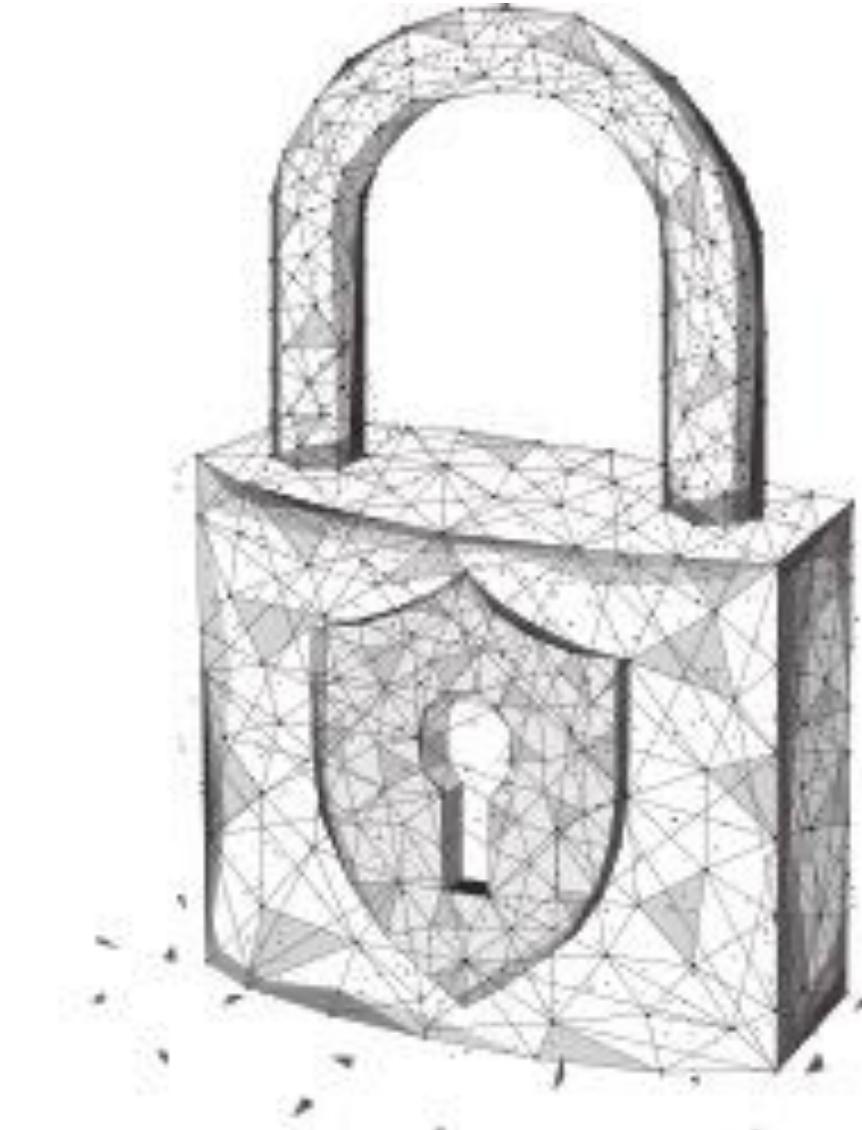


AI ethics matter



AI data and privacy considerations

- Data is the fuel for effective AI; more data = better predictions.
- Some AI services automatically store data.
- Data retention should be kept in mind for regulatory compliance.
- **None of this is legal advice, merely things to keep in mind as you tap into the power of AI.**



Further training from Microsoft

Microsoft Learn: <https://docs.microsoft.com/en-us/learn/browse/>

@geektrainer

chrhar@microsoft.com

<https://github.com/geektrainer/ai-reactor-workshop>

<https://github.com/geektrainer/ai-workshop-starter>

<https://github.com/geektrainer/ai-workshop-solution>



© 2016 Microsoft Corporation. All rights reserved. The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.