```python
# Find faces in picture
# https://github.com/ageitgey/face_recognition/blob/master/examples/find_faces_in_picture.py

from PIL import Image
import face_recognition

# Load the jpg file into a numpy array
image = face_recognition.load_image_file("office.jpg")

# Find all the faces in the image using the default HOG-based model.
# This method is fairly accurate, but not as accurate as the CNN model and not GPU accelerated.
# See also: find_faces_in_picture_cnn.py
face_locations = face_recognition.face_locations(image)

for face_location in face_locations:

    # Print the location of each face in this image
    top, right, bottom, left = face_location

    # You can access the actual face itself like this:
    face_image = image[top:bottom, left:right]
    pil_image = Image.fromarray(face_image)
    pil_image.show()
```

```python
# Identify and draw box on David
# https://github.com/ageitgey/face_recognition/blob/master/examples/identify_and_draw_boxes_on_faces.py

import face_recognition
import numpy as np
from PIL import Image, ImageDraw

# Load a sample picture and learn how to recognize it.
known_image = face_recognition.load_image_file("toby.jpg")
encoding = face_recognition.face_encodings(known_image)[0]

# Load an image with unknown faces
unknown_image = face_recognition.load_image_file("office.jpg")

# Find all the faces and face encodings in the unknown image
face_locations = face_recognition.face_locations(unknown_image)
face_encodings = face_recognition.face_encodings(unknown_image, face_locations)

# Convert the image to a PIL-format image so that we can draw on top of it with the Pillow library
# See http://pillow.readthedocs.io/ for more about PIL/Pillow
pil_image = Image.fromarray(unknown_image)

# Create a Pillow ImageDraw Draw instance to draw with
draw = ImageDraw.Draw(pil_image)

# Loop through each face found in the unknown image
for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):

    # See if the face is a match for the known face(s)
    matches = face_recognition.compare_faces([encoding], face_encoding)

    # Use the known face with the smallest distance to the new face
    face_distances = face_recognition.face_distance([encoding], face_encoding)
    best_match_index = np.argmin(face_distances)
    if matches[best_match_index]:

        # Draw a box around the face using the Pillow module
        draw.rectangle(((left - 20, top - 20), (right + 20, bottom + 20)), outline=(0, 255, 0), width=20)

# Remove the drawing library from memory as per the Pillow docs
del draw

```

```
43  # Display the resulting image
44  pil_image.show()
```

```python
# Demonstrates a function with a positional argument

print("hello, world")
```

```python
# Demonstrates a function with a positional argument and a return value

name = input("What's your name? ")
print("hello,")
print(name)
```

```
# Demonstrates concatenation of strings

name = input("What's your name? ")
print("hello, " + name)
```

```python
# Demonstrates a function with two positional arguments

name = input("What's your name? ")
print("hello,", name)
```

```python
# Demonstrates a function with a positional argument and a named argument

name = input("What's your name? ")
print("hello, ", end="")
print(name)
```

```python
# Demonstrates a format string

name = input("What's your name? ")
print(f"hello, {name}")
```

```python
# Demonstrates str functions

name = input("What's your name? ").strip().title()
print(f"hello, {name}")
```

```python
# Demonstrates str functions

name = input("What's your name? ").strip().title()
first, last = name.split(" ")
print(f"hello, {first}")
```

```
# Demonstrates addition

x = 1
y = 2

z = x + y

print(z)
```

```python
# Demonstrates (unintended) concatenation of strings

# Prompt user for two integers
x = input("What's x? ")
y = input("What's y? ")

# Print sum
z = x + y
print(z)
```

```python
# Demonstrates conversion from str to int

x = input("What's x? ")
y = input("What's y? ")

z = int(x) + int(y)

print(z)
```

```
# Demonstrates nesting of function calls

x = int(input("What's x? "))
y = int(input("What's y? "))

z = x + y

print(z)
```

```python
# Demonstrates conversion of str to float

x = float(input("What's x? "))
y = float(input("What's y? "))

z = x + y

print(z)
```

```python
# Demonstrates rounding to nearest int

x = float(input("What's x? "))
y = float(input("What's y? "))

z = round(x + y)

print(z)
```

```
# Demonstrates fewer variables

x = float(input("What's x? "))
y = float(input("What's y? "))

print(round(x + y))
```

```python
# Demonstrates formatting with commas

x = float(input("What's x? "))
y = float(input("What's y? "))

z = round(x + y)

print(f"{z:,}")
```

```
# Demonstrates division

x = float(input("What's x? "))
y = float(input("What's y? "))

z = x / y

print(z)
```

```python
# Demonstrates rounding after the decimal point

x = float(input("What's x? "))
y = float(input("What's y? "))

z = round(x / y, 2)

print(z)
```

```python
# Demonstrates formatting after the decimal place

x = int(input("What's x? "))
y = int(input("What's y? "))

z = x / y

print(f"{z:.2f}")
```

```python
# Demonstrates defining a function without parameters


def hello():
    print("hello")


name = input("What's your name? ")
hello()
print(name)
```

```python
# Demonstrates defining a function with a parameter


def hello(to):
    print("hello,", to)


name = input("What's your name? ")
hello(name)
```

```python
# Demonstrates defining a function with a parameter with a default value


def hello(to="world"):
    print("hello,", to)


hello()
name = input("What's your name? ")
hello(name)
```

```python
# Demonstrates defining a main function


def main():
    name = input("What's your name? ")
    hello(name)


def hello(to="world"):
    print("hello,", to)


main()
```

```python
# Demonstrates defining a function with a return value


def main():
    x = int(input("What's x? "))
    print("x squared is", square(x))


def square(n):
    return n * n


main()
```

```
# Demonstrates conditionals

x = int(input("What's x? "))
y = int(input("What's y? "))

if x < y:
    print("x is less than y")
if x > y:
    print("x is greater than y")
if x == y:
    print("x is equal to y")
```

```python
# Demonstrates mutually exclusive conditions

x = int(input("What's x? "))
y = int(input("What's y? "))

if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
elif x == y:
    print("x is equal to y")
```

```python
# Demonstrates fewer conditions

x = int(input("What's x? "))
y = int(input("What's y? "))

if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x is equal to y")
```

```python
# Demonstrates inequalities and logical operator

x = int(input("What's x? "))
y = int(input("What's y? "))

if x < y or x > y:
    print("x is not equal to y")
else:
    print("x is equal to y")
```

```
# Demonstrates equality

x = int(input("What's x? "))
y = int(input("What's y? "))

if x == y:
    print("x is equal to y")
else:
    print("x is not equal to y")
```

```python
# Demonstrates inequality

x = int(input("What's x? "))
y = int(input("What's y? "))

if x != y:
    print("x is not equal to y")
else:
    print("x is equal to y")
```

```python
# Demonstrates inequalities and logical operators

score = int(input("Score: "))

if score >= 90 and score <= 100:
    print("Grade: A")
elif score >= 80 and score < 90:
    print("Grade: B")
elif score >= 70 and score < 80:
    print("Grade: C")
elif score >= 60 and score < 70:
    print("Grade: D")
else:
    print("Grade: F")
```

```python
# Demonstrates inequalities and logical operators

score = int(input("Score: "))

if 90 <= score and score <= 100:
    print("Grade: A")
elif 80 <= score and score < 90:
    print("Grade: B")
elif 70 <= score and score < 80:
    print("Grade: C")
elif 60 <= score and score < 70:
    print("Grade: D")
else:
    print("Grade: F")
```

```python
# Demonstrates chained comparisons

score = int(input("Score: "))

if 90 <= score <= 100:
    print("Grade: A")
elif 80 <= score < 90:
    print("Grade: B")
elif 70 <= score < 80:
    print("Grade: C")
elif 60 <= score < 70:
    print("Grade: D")
else:
    print("Grade: F")
```

```python
# Demonstrates fewer comparisons

score = int(input("Score: "))

if score >= 90:
    print("Grade: A")
elif score >= 80:
    print("Grade: B")
elif score >= 70:
    print("Grade: C")
elif score >= 60:
    print("Grade: D")
else:
    print("Grade: F")
```

```python
# Compares strings

answer = input("Do you agree? ")
if answer == "yes":
    print("Agreed")
else:
    print("Not agreed")
```

```python
# Strips string before comparing

answer = input("Do you agree? ").strip()
if answer == "yes":
    print("Agreed")
else:
    print("Not agreed")
```

```python
# Lowercases string before comparing

answer = input("Do you agree? ").strip().lower()
if answer == "yes":
    print("Agreed")
else:
    print("Not agreed")
```

```python
# Compares multiple strings

answer = input("Do you agree? ").strip().lower()
if answer == "yes" or answer == "y":
    print("Agreed")
else:
    print("Not agreed")
```

```python
# Compares multiple strings

answer = input("Do you agree? ").strip().lower()
if answer.startswith("y"):
    print("Agreed")
else:
    print("Not agreed")
```

```python
# Demonstrates modulo operator

x = int(input("What's x? "))

if x % 2 == 0:
    print("Even")
else:
    print("Odd")
```

```python
# Demonstrates a function that returns a bool


def main():
    x = int(input("What's x? "))
    if is_even(x):
        print("Even")
    else:
        print("Odd")


def is_even(n):
    if n % 2 == 0:
        return True
    else:
        return False


main()
```

```python
# Demonstrates conditional expressions (ternary operators)


def main():
    x = int(input("What's x? "))
    if is_even(x):
        print("Even")
    else:
        print("Odd")


def is_even(n):
    return True if n % 2 == 0 else False


main()
```

```python
# Demonstrates returning the value of a Boolean expression


def main():
    x = int(input("What's x? "))
    if is_even(x):
        print("Even")
    else:
        print("Odd")


def is_even(n):
    return n % 2 == 0


main()
```

```python
# Compares multiple strings with if/elif/else

name = input("What's your name? ")

if name == "Harry":
    print("Gryffindor")
elif name == "Hermione":
    print("Gryffindor")
elif name == "Ron":
    print("Gryffindor")
elif name == "Draco":
    print("Slytherin")
else:
    print("Who?")
```

```python
# Uses or

name = input("What's your name? ")

if name == "Harry" or name == "Hermione" or name == "Ron":
    print("Gryffindor")
elif name == "Draco":
    print("Slytherin")
else:
    print("Who?")
```

```python
# Uses match with case

name = input("What's your name? ")

match name:
    case "Harry":
        print("Gryffindor")
    case "Hermione":
        print("Gryffindor")
    case "Ron":
        print("Gryffindor")
    case "Draco":
        print("Slytherin")
    case _:
        print("Who?")
```

```python
# Uses |

name = input("What's your name? ")

match name:
    case "Harry" | "Hermione" | "Ron":
        print("Gryffindor")
    case "Draco":
        print("Slytherin")
    case _:
        print("Who?")
```

```
# Demonstrates multiple (identical) function calls

print("meow")
print("meow")
print("meow")
```

```
# Demonstrates a while loop, counting down

i = 3
while i != 0:
    print("meow")
    i = i - 1
```

```
# Demonstrates a while loop, counting up from 1

i = 1
while i <= 3:
    print("meow")
    i = i + 1
```

```
# Demonstrates a while loop, counting up from 0

i = 0
while i < 3:
    print("meow")
    i = i + 1
```

```
# Demonstrates (more succinct) incrementation

i = 0
while i < 3:
    print("meow")
    i += 1
```

```python
# Demonstrates a for loop, using a list

for i in [0, 1, 2]:
    print("meow")
```

```python
# Demonstrates a for loop, using range

for i in range(3):
    print("meow")
```

```python
# Demonstrates a for loop, with _ as a variable

for _ in range(3):
    print("meow")
```

```
# Demonstrates str multiplication

print("meow\n" * 3, end="")
```

```python
# Introduces continue, break

while True:
    n = int(input("What's n? "))
    if n <= 0:
        continue
    else:
        break

for _ in range(n):
    print("meow")
```

```python
# Removes continue

while True:
    n = int(input("What's n? "))
    if n > 0:
        break

for _ in range(n):
    print("meow")
```

```python
# Demonstrates defining functions


def main():
    meow(get_number())


def get_number():
    while True:
        n = int(input("What's n? "))
        if n > 1:
            return n


def meow(n):
    for _ in range(n):
        print("meow")


main()
```

```
# Demonstrates indexing into a list

students = ["Hermione", "Harry", "Ron"]

print(students[0])
print(students[1])
print(students[2])
```

```python
# Demonstrates iterating over a list

students = ["Hermione", "Harry", "Ron"]

for student in students:
    print(student)
```

```python
# Demonstrates iterating over and indexing into a list

students = ["Hermione", "Harry", "Ron"]

for i in range(len(students)):
    print(i + 1, students[i])
```

```python
# Demonstrates indexing into a dict

students = {
    "Hermione": "Gryffindor",
    "Harry": "Gryffindor",
    "Ron": "Gryffindor",
    "Draco": "Slytherin",
}

print(students["Hermione"])
print(students["Harry"])
print(students["Ron"])
print(students["Draco"])
```

```python
# Demonstrates iterating over and index into a dict

students = {
    "Hermione": "Gryffindor",
    "Harry": "Gryffindor",
    "Ron": "Gryffindor",
    "Draco": "Slytherin",
}

for student in students:
    print(student, students[student], sep=", ")
```

```python
# Demonstrates iterating over a list of dict objects

students = [
    {"name": "Hermione", "house": "Gryffindor", "patronus": "Otter"},
    {"name": "Harry", "house": "Gryffindor", "patronus": "Stag"},
    {"name": "Ron", "house": "Gryffindor", "patronus": "Jack Russell terrier"},
    {"name": "Draco", "house": "Slytherin", "patronus": None},
]

for student in students:
    print(student["name"], student["house"], student["patronus"], sep=", ")
```

```python
# Prints a column of bricks

print("#")
print("#")
print("#")
```

```python
# Prints column of bricks using a loop

for _ in range(3):
    print("#")
```

```python
# Prints column of bricks using a function with a loop


def main():
    print_column(3)


def print_column(height):
    for _ in range(height):
        print("#")


main()
```

```python
# Prints column of bricks using a function with str multiplication


def main():
    print_column(3)


def print_column(height):
    print("#\n" * height, end="")


main()
```

```python
# Prints row of coins using a function with str multiplication


def main():
    print_row(4)


def print_row(width):
    print("?" * width)


main()
```

```python
# Prints square of bricks using a function with nested loops


def main():
    print_square(3)


def print_square(size):
    for i in range(size):
        for j in range(size):
            print("#", end="")
        print()


main()
```

```python
# Prints square of bricks using a function with a loop and str multiplication


def main():
    print_square(3)


def print_square(size):
    for _ in range(size):
        print("#" * size)


main()
```

```python
# Prints square of bricks using a function with a loop and str multiplication


def main():
    print_square(3)


def print_square(size):
    for _ in range(size):
        print_row(size)


def print_row(width):
    print("#" * width)


main()
```

```python
# Demonstrates import and random.choice

import random

coin = random.choice(["heads", "tails"])
print(coin)
```

```
# Demonstrates from

from random import choice

coin = choice(["heads", "tails"])
print(coin)
```

```python
# Demonstrates randint

import random

number = random.randint(1, 10)
print(number)
```

```python
# Demonstrates shuffle

import random

cards = ["jack", "queen", "king"]
random.shuffle(cards)
for card in cards:
    print(card)
```

```
# Demonstrates statistics

import statistics

print(statistics.mean([100, 90]))
```

```python
# Demonstrates sys.argv

import sys

print("hello, my name is", sys.argv[1])
```

```python
# Demonstrates IndexError

import sys

try:
    print("hello, my name is", sys.argv[1])
except IndexError:
    print("Too few arguments")
```

```python
# Adds error checking

import sys

if len(sys.argv) < 2:
    print("Too few arguments")
elif len(sys.argv) > 2:
    print("Too many arguments")
else:
    print("hello, my name is", sys.argv[1])
```

```python
# Demonstrates sys.exit

import sys

if len(sys.argv) < 2:
    sys.exit("Too few arguments")
elif len(sys.argv) > 2:
    sys.exit("Too many arguments")

print("hello, my name is", sys.argv[1])
```

```python
# Demonstrates list slice

import sys

if len(sys.argv) < 2:
    sys.exit("Too few arguments")

for arg in sys.argv[1:]:
    print("hello, my name is", arg)
```

```python
# Demonstrates pip-installed package

import cowsay
import sys

if len(sys.argv) == 2:
    cowsay.cow("hello, " + sys.argv[1])
```

```
# Demonstrates a t-rex

import cowsay
import sys

if len(sys.argv) == 2:
    cowsay.trex("hello, " + sys.argv[1])
```

```python
# Demonstrates requests

import sys
import requests

if len(sys.argv) != 2:
    sys.exit()

response = requests.get(
    "https://itunes.apple.com/search?entity=song&limit=1&term=" + sys.argv[1]
)
print(response.json())
```

```python
# Demonstrates json

import json
import sys
import requests

if len(sys.argv) != 2:
    sys.exit()

response = requests.get(
    "https://itunes.apple.com/search?entity=song&limit=1&term=" + sys.argv[1]
)
print(json.dumps(response.json(), indent=2))
```

```python
# Demonstrates iterating over JSON

import json
import sys
import requests

if len(sys.argv) != 2:
    sys.exit()

response = requests.get(
    "https://itunes.apple.com/search?entity=song&term=" + sys.argv[1]
)
o = response.json()
for result in o["results"]:
    print(result["trackName"])
```

```
def hello(name):
    print(f"hello, {name}")


def goodbye(name):
    print(f"goodbye, {name}")
```

```
1  # Demonstrates own module
2
3  import sys
4
5  from sayings0 import hello
6
7  if len(sys.argv) == 2:
8      hello(sys.argv[1])
```

```python
# Doesn't check __name__


def main():
    hello("world")
    goodbye("world")


def hello(name):
    print(f"hello, {name}")


def goodbye(name):
    print(f"goodbye, {name}")


main()
```

```
# Demonstrates own module

import sys

from sayings1 import hello

if len(sys.argv) == 2:
    hello(sys.argv[1])
```

```python
# Check __name__


def main():
    hello("world")
    goodbye("world")


def hello(name):
    print(f"hello, {name}")


def goodbye(name):
    print(f"goodbye, {name}")


if __name__ == "__main__":
    main()
```

```python
# Demonstrates own module

import sys

from sayings2 import hello

if len(sys.argv) == 2:
    hello(sys.argv[1])
```

```
1  # Demonstrates own module
2
3  import sys
4
5  from sayings2 import goodbye
6
7  if len(sys.argv) == 2:
8      goodbye(sys.argv[1])
```

```python
# Recognizes a greeting

# Get input
words = input("Say something!\n").lower()

# Respond to speech
if "hello" in words:
    print("Hello to you too!")
elif "how are you" in words:
    print("I am well, thanks!")
elif "goodbye" in words:
    print("Goodbye to you too!")
else:
    print("Huh?")
```

```python
# Recognizes a voice
# https://pypi.org/project/SpeechRecognition/

import speech_recognition

# Obtain audio from the microphone
recognizer = speech_recognition.Recognizer()
with speech_recognition.Microphone() as source:
    print("Say something:")
    audio = recognizer.listen(source)

# Recognize speech using Google Speech Recognition
print("You said:")
print(recognizer.recognize_google(audio))
```

```python
# Responds to a greeting
# https://pypi.org/project/SpeechRecognition/

import speech_recognition

# Obtain audio from the microphone
recognizer = speech_recognition.Recognizer()
with speech_recognition.Microphone() as source:
    print("Say something:")
    audio = recognizer.listen(source)

# Recognize speech using Google Speech Recognition
words = recognizer.recognize_google(audio)

# Respond to speech
if "hello" in words:
    print("Hello to you too!")
elif "how are you" in words:
    print("I am well, thanks!")
elif "goodbye" in words:
    print("Goodbye to you too!")
else:
    print("Huh?")
```

```python
# Responds to a name
# https://pypi.org/project/SpeechRecognition/

import re
import speech_recognition

# Obtain audio from the microphone
recognizer = speech_recognition.Recognizer()
with speech_recognition.Microphone() as source:
    print("Say something:")
    audio = recognizer.listen(source)

# Recognize speech using Google Speech Recognition
words = recognizer.recognize_google(audio)

# Respond to speech
matches = re.search("my name is (.*)", words)
if matches:
    print(f"Hey, {matches[1]}.")
else:
    print("Hey, you.")
```