
```
1 # Says hello to the world
2
3 print("hello, world")
```

```
1  # Says hello to someone
2
3  answer = input("What's your name? ")
4  print("hello,", answer)
```

```
1  # Conditions and relational operators
2
3  # Prompt user for x
4  x = int(input("x: "))
5
6  # Prompt user for y
7  y = int(input("y: "))
8
9  # Compare x and y
10 if x < y:
11     print("x is less than y")
12 elif x > y:
13     print("x is greater than y")
14 else:
15     print("x is equal to y")
```

```
1 # Logical operators
2
3 # Prompt user for answer
4 c = input("Answer: ")
5
6 # Check answer
7 if c == "Y" or c == "y":
8     print("yes")
9 elif c == "N" or c == "n":
10    print("no")
```

```
1  # Logical operators
2
3  # Prompt user to agree
4  s = input("Do you agree? ")
5
6  # Check whether agreed
7  if s == "Y" or s == "y":
8      print("Agreed.")
9  elif s == "N" or s == "n":
10     print("Not agreed.")
```

```
1  # Logical operators, using lists
2
3  # Prompt user to agree
4  s = input("Do you agree? ")
5
6  # Check whether agreed
7  if s.lower() in ["y", "yes"]:
8      print("Agreed.")
9  elif s.lower() in ["n", "no"]:
10     print("Not agreed.")
```

```
1  # Abstraction and scope
2
3
4  def main():
5      i = get_positive_int("Positive integer: ")
6      print(i)
7
8
9  def get_positive_int(prompt):
10     while True:
11         n = int(input(prompt))
12         if n > 0:
13             break
14     return n
15
16
17 main()
```

```
1  # Generates a bar chart of three scores
2
3  # Get scores from user
4  score1 = int(input("Score 1: "))
5  score2 = int(input("Score 2: "))
6  score3 = int(input("Score 3: "))
7
8  # Generate first bar
9  print("Score 1: ", end="");
10 for i in range(score1):
11     print("#", end="")
12 print()
13
14 # Generate second bar
15 print("Score 2: ", end="");
16 for i in range(score2):
17     print("#", end="")
18 print()
19
20 # Generate third bar
21 print("Score 3: ", end="");
22 for i in range(score3):
23     print("#", end="")
24 print()
```

```
1  # Opportunity for better design
2
3  print("meow")
4  print("meow")
5  print("meow")
```

```
1 # Better design
2
3 for i in range(3):
4     print("meow")
```

```
1  # Abstraction
2
3  def main():
4      for i in range(3):
5          meow()
6
7  # Meow once
8  def meow():
9      print("meow")
10
11
12 main()
```

```
1  # Abstraction with parameterization
2
3  def main():
4      meow(3)
5
6
7  # Meow some number of times
8  def meow(n):
9      for i in range(n):
10         print("meow")
11
12
13  main()
```

```
1 # Floating-point imprecision
2
3 print(f"{1/10:.50f}")
```

```
1 # Prints a row of 4 question marks with a loop
2
3 for i in range(4):
4     print("?", end="")
5 print()
```

```
1 # Prints a row of 4 question marks without a loop
2
3 print("?" * 4)
```

```
1 # Prints a column of 3 bricks with a loop
2
3 for i in range(3):
4     print("#")
```

```
1  # Prints a column of 3 bricks without a loop
2
3  print("#\n" * 3, end="")
```

```
1  # Prints a 3-by-3 grid of bricks with loops
2
3  for i in range(3):
4      for j in range(3):
5          print("#", end="")
6      print()
```

```
1  # Says hello
2
3  import pyttsx3
4
5  engine = pyttsx3.init()
6  engine.say("hello, world")
7  engine.runAndWait()
```

```
1 # Says hello
2
3 import pyttsx3
4
5 engine = pyttsx3.init()
6 name = input("What's your name? ")
7 engine.say(f"hello, {name}")
8 engine.runAndWait()
```

```
1  # Find faces in picture
2  # https://github.com/ageitgey/face\_recognition/blob/master/examples/find\_faces\_in\_picture.py
3
4  from PIL import Image
5  import face_recognition
6
7  # Load the jpg file into a numpy array
8  image = face_recognition.load_image_file("office.jpg")
9
10 # Find all the faces in the image using the default HOG-based model.
11 # This method is fairly accurate, but not as accurate as the CNN model and not GPU accelerated.
12 # See also: find_faces_in_picture_cnn.py
13 face_locations = face_recognition.face_locations(image)
14
15 for face_location in face_locations:
16
17     # Print the location of each face in this image
18     top, right, bottom, left = face_location
19
20     # You can access the actual face itself like this:
21     face_image = image[top:bottom, left:right]
22     pil_image = Image.fromarray(face_image)
23     pil_image.show()
```

```
1  # Identify and draw box on David
2  # https://github.com/ageitgey/face\_recognition/blob/master/examples/identify\_and\_draw\_boxes\_on\_faces.py
3
4  import face_recognition
5  import numpy as np
6  from PIL import Image, ImageDraw
7
8  # Load a sample picture and learn how to recognize it.
9  known_image = face_recognition.load_image_file("toby.jpg")
10 encoding = face_recognition.face_encodings(known_image)[0]
11
12 # Load an image with unknown faces
13 unknown_image = face_recognition.load_image_file("office.jpg")
14
15 # Find all the faces and face encodings in the unknown image
16 face_locations = face_recognition.face_locations(unknown_image)
17 face_encodings = face_recognition.face_encodings(unknown_image, face_locations)
18
19 # Convert the image to a PIL-format image so that we can draw on top of it with the Pillow library
20 # See http://pillow.readthedocs.io/ for more about PIL/Pillow
21 pil_image = Image.fromarray(unknown_image)
22
23 # Create a Pillow ImageDraw Draw instance to draw with
24 draw = ImageDraw.Draw(pil_image)
25
26 # Loop through each face found in the unknown image
27 for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
28
29     # See if the face is a match for the known face(s)
30     matches = face_recognition.compare_faces([encoding], face_encoding)
31
32     # Use the known face with the smallest distance to the new face
33     face_distances = face_recognition.face_distance([encoding], face_encoding)
34     best_match_index = np.argmin(face_distances)
35     if matches[best_match_index]:
36
37         # Draw a box around the face using the Pillow module
38         draw.rectangle(((left - 20, top - 20), (right + 20, bottom + 20)), outline=(0, 255, 0), width=20)
39
40 # Remove the drawing library from memory as per the Pillow docs
41 del draw
42
```

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

```
1 # Generates a QR code
2 # https://github.com/lincolnloop/python-qrcode
3
4 import os
5 import qrcode
6
7 # Generate QR code
8 img = qrcode.make("https://youtu.be/oHg5SJYRHA0")
9
10 # Save as file
11 img.save("qr.png", "PNG")
12
13 # Open file
14 os.system("open qr.png")
```



```
1  # Recognizes a greeting
2
3  # Get input
4  words = input("Say something!\n").lower()
5
6  # Respond to speech
7  if "hello" in words:
8      print("Hello to you too!")
9  elif "how are you" in words:
10     print("I am well, thanks!")
11 elif "goodbye" in words:
12     print("Goodbye to you too!")
13 else:
14     print("Huh?")
```

```
1  # Recognizes a voice
2  # https://pypi.org/project/SpeechRecognition/
3
4  import speech_recognition
5
6  # Obtain audio from the microphone
7  recognizer = speech_recognition.Recognizer()
8  with speech_recognition.Microphone() as source:
9      print("Say something:")
10     audio = recognizer.listen(source)
11
12  # Recognize speech using Google Speech Recognition
13  print("You said:")
14  print(recognizer.recognize_google(audio))
```

```
1  # Responds to a greeting
2  # https://pypi.org/project/SpeechRecognition/
3
4  import speech_recognition
5
6  # Obtain audio from the microphone
7  recognizer = speech_recognition.Recognizer()
8  with speech_recognition.Microphone() as source:
9      print("Say something:")
10     audio = recognizer.listen(source)
11
12 # Recognize speech using Google Speech Recognition
13 words = recognizer.recognize_google(audio)
14
15 # Respond to speech
16 if "hello" in words:
17     print("Hello to you too!")
18 elif "how are you" in words:
19     print("I am well, thanks!")
20 elif "goodbye" in words:
21     print("Goodbye to you too!")
22 else:
23     print("Huh?")
```

```
1  # Responds to a name
2  # https://pypi.org/project/SpeechRecognition/
3
4  import re
5  import speech_recognition
6
7  # Obtain audio from the microphone
8  recognizer = speech_recognition.Recognizer()
9  with speech_recognition.Microphone() as source:
10     print("Say something:")
11     audio = recognizer.listen(source)
12
13 # Recognize speech using Google Speech Recognition
14 words = recognizer.recognize_google(audio)
15
16 # Respond to speech
17 matches = re.search("my name is (.*)", words)
18 if matches:
19     print(f"Hey, {matches[1]}.")
20 else:
21     print("Hey, you.")
```