```python
# Blurs an image

from PIL import Image, ImageFilter

# Blur image
before = Image.open("bridge.bmp")
after = before.filter(ImageFilter.BoxBlur(1))
after.save("out.bmp")
```

```python
# Blurs an image

from PIL import Image, ImageFilter

# Find edges
before = Image.open("bridge.bmp")
after = before.filter(ImageFilter.FIND_EDGES)
after.save("out.bmp")
```

```python
# Words in dictionary
words = set()


def check(word):
    """Return true if word is in dictionary else false"""
    if word.lower() in words:
        return True
    else:
        return False


def load(dictionary):
    """Load dictionary into memory, returning true if successful else false"""
    file = open(dictionary, "r")
    for line in file:
        words.add(line.rstrip())
    file.close()
    return True


def size():
    """Returns number of words in dictionary if loaded else 0 if not yet loaded"""
    return len(words)


def unload():
    """Unloads dictionary from memory, returning true if successful else false"""
    return True
```

```c
// A program that says hello to the world

#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

```python
# A program that says hello to the world

print("hello, world")
```

```c
// get_string and printf with %s

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    string answer = get_string("What's your name? ");
    printf("hello, %s\n", answer);
}
```

```python
# get_string and print, with concatenation

from cs50 import get_string

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

```python
# get_string and print, with format strings

from cs50 import get_string

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

```python
# input and print, with format strings

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

```c
// Addition with int

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Perform addition
    printf("%i\n", x + y);
}
```

```python
# Addition with int [using get_int]

from cs50 import get_int

# Prompt user for x
x = get_int("x: ")

# Prompt user for y
y = get_int("y: ")

# Perform addition
print(x + y)
```

```python
# Addition with int [using input]

# Prompt user for x
x = int(input("x: "))

# Prompt user for y
y = int(input("y: "))

# Perform addition
print(x + y)
```

```python
# Division with int

# Prompt user for x
x = int(input("x: "))

# Prompt user for y
y = int(input("y: "))

# Perform division
print(x / y)
```

```c
// Conditions and relational operators

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Compare x and y
    if (x < y)
    {
        printf("x is less than y\n");
    }
    else if (x > y)
    {
        printf("x is greater than y\n");
    }
    else
    {
        printf("x is equal to y\n");
    }
}
```

```python
# Conditions and relational operators

from cs50 import get_int

# Prompt user for x
x = get_int("x: ")

# Prompt user for y
y = get_int("y: ")

# Compare x and 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")
```

```c
// Logical operators

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    // Prompt user to agree
    char c = get_char("Do you agree? ");

    // Check whether agreed
    if (c == 'Y' || c == 'y')
    {
        printf("Agreed.\n");
    }
    else if (c == 'N' || c == 'n')
    {
        printf("Not agreed.\n");
    }
}
```

```python
# Logical operators

from cs50 import get_string

# Prompt user to agree
s = get_string("Do you agree? ")

# Check whether agreed
if s == "Y" or s == "y":
    print("Agreed.")
elif s == "N" or s == "n":
    print("Not agreed.")
```

```python
# Logical operators, using lists

from cs50 import get_string

# Prompt user to agree
s = get_string("Do you agree? ")

# Check whether agreed
if s.lower() in ["y", "yes"]:
    print("Agreed.")
elif s.lower() in ["n", "no"]:
    print("Not agreed.")
```

```c
// Opportunity for better design

#include <stdio.h>

int main(void)
{
    printf("meow\n");
    printf("meow\n");
    printf("meow\n");
}
```

```python
# Opportunity for better design

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

```c
// Better design

#include <stdio.h>

int main(void)
{
    for (int i = 0; i < 3; i++)
    {
        printf("meow\n");
    }
}
```

```python
# Better design

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

```c
// Abstraction

#include <stdio.h>

void meow(void);

int main(void)
{
    for (int i = 0; i < 3; i++)
    {
        meow();
    }
}

// Meow once
void meow(void)
{
    printf("meow\n");
}
```

```python
# Abstraction

def main():
    for i in range(3):
        meow()

# Meow once
def meow():
    print("meow")


meow()
```

```c
// Abstraction with parameterization

#include <stdio.h>

void meow(int n);

int main(void)
{
    meow(3);
}

// Meow some number of times
void meow(int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("meow\n");
    }
}
```

```python
# Abstraction with parameterization

def main():
    meow(3)


# Meow some number of times
def meow(n):
    for i in range(n):
        print("meow")


meow()
```

```c
// Abstraction and scope

#include <cs50.h>
#include <stdio.h>

int get_positive_int(void);

int main(void)
{
    int i = get_positive_int();
    printf("%i\n", i);
}

// Prompt user for positive integer
int get_positive_int(void)
{
    int n;
    do
    {
        n = get_int("Positive Integer: ");
    }
    while (n < 1);
    return n;
}
```

```python
# Abstraction and scope

from cs50 import get_int


def main():
    i = get_positive_int()
    print(i)


# Prompt user for positive integer
def get_positive_int():
    while True:
        n = get_int("Positive Integer: ")
        if n > 0:
            break
    return n


main()
```

```python
# Prints a column of 3 bricks with a loop

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

```python
# Prints a row of 4 question marks with a loop

for i in range(4):
    print("?", end="")
print()
```

```python
# Prints a row of 4 question marks without a loop

print("?" * 4)
```

```python
# Prints a 3-by-3 grid of bricks with loops

for i in range(3):
    for j in range(3):
        print("#", end="")
    print()
```

```python
# Integer non-overflow

# Iteratively double i
i = 1
while True:
    print(i)
    i *= 2
```

```python
# Averages three numbers using a list

# Scores
scores = [72, 73, 33]

# Print average
print(f"Average: {sum(scores) / len(scores)}")
```

```python
# Averages three numbers using an array and a loop

from cs50 import get_int

# Get scores
scores = []
for i in range(3):
    scores.append(get_int("Score: "))

# Print average
print(f"Average: {sum(scores) / len(scores)}")
```

```python
# Uppercases string one character at a time

from cs50 import get_string

s = get_string("Before: ")
print("After:  ", end="")
for c in s:
    print(c.upper(), end="")
print()
```

```python
# Uppercases string all at once

from cs50 import get_string

s = get_string("Before: ")
print(f"After:  {s.upper()}")
```

```python
# Prints a command-line argument

from sys import argv

if len(argv) == 2:
    print(f"hello, {argv[1]}")
else:
    print("hello, world")
```

```python
# Printing command-line arguments, indexing into argv

from sys import argv

for i in range(len(argv)):
    print(argv[i])
```

```python
# Printing command-line arguments

from sys import argv

for arg in argv:
    print(arg)
```

```python
# Exits with explicit value, importing sys

import sys

if len(sys.argv) != 2:
    print("missing command-line argument")
    sys.exit(1)

print(f"hello, {sys.argv[1]}")
sys.exit(0)
```

```python
# Implements linear search for numbers

import sys

# A list of numbers
numbers = [4, 6, 8, 2, 7, 5, 0]

# Search for 0
if 0 in numbers:
    print("Found")
    sys.exit(0)

print("Not found")
sys.exit(1)
```

```python
# Implements linear search for names

import sys

# A list of names
names = ["Bill", "Charlie", "Fred", "George", "Ginny", "Percy", "Ron"]

# Search for Ron
if "Ron" in names:
    print("Found")
    sys.exit(0)

print("Not found")
sys.exit(1)
```

```python
# Implements a phone book

import sys

from cs50 import get_string

people = {
    "Brian": "+1-617-495-1000",
    "David": "+1-949-468-2750"
}

# Search for name
name = get_string("Name: ")
if name in people:
    print(f"Number: {people[name]}")
```

```python
# Compares two strings

from cs50 import get_string

# Get two strings
s = get_string("s: ")
t = get_string("t: ")

# Compare strings
if s == t:
    print("Same")
else:
    print("Different")
```

```python
# Capitalizes a copy of a string

from cs50 import get_string

# Get a string
s = get_string("s: ")

# Capitalize copy of string
t = s.capitalize()

# Print strings
print(f"s: {s}")
print(f"t: {t}")
```

```python
# Swaps two integers

x = 1
y = 2

print(f"x is {x}, y is {y}")
x, y = y, x
print(f"x is {x}, y is {y}")
```

```python
# Saves names and numbers to a CSV file

import csv
from cs50 import get_string

# Open CSV file
file = open("phonebook.csv", "a")

# Get name and number
name = get_string("Name: ")
number = get_string("Number: ")

# Print to file
writer = csv.writer(file)
writer.writerow([name, number])

# Close file
file.close()
```

```python
# Saves names and numbers to a CSV file

import csv
from cs50 import get_string

# Get name and number
name = get_string("Name: ")
number = get_string("Number: ")

# Open CSV file
with open("phonebook.csv", "a") as file:

    # Print to file
    writer = csv.writer(file)
    writer.writerow([name, number])
```

```python
# Counts number of students in houses

import csv

# Numbers of students in houses
houses = {
    "Gryffindor": 0,
    "Hufflepuff": 0,
    "Ravenclaw": 0,
    "Slytherin": 0
}

# Count votes
with open("Sorting Hat - Form Responses 1.csv", "r") as file:
    reader = csv.reader(file)
    next(reader)
    for row in reader:
        houses[row[3]] += 1

# Print counts
for house in houses:
    print(f"{house}: {houses[house]}")
```

```python
# Logical operators, using regular expressions

import re

from cs50 import get_string

# Prompt user to agree
s = get_string("Do you agree? ")

# Check whether agreed
if re.search("^y(es)?$", s, re.IGNORECASE):
    print("Agreed.")
elif re.search("^no?$", s, re.IGNORECASE):
    print("Not agreed.")
```

```python
# Says hello

import pyttsx3

engine = pyttsx3.init()
engine.say("hello, world")
engine.runAndWait()
```

```python
# Says hello

import pyttsx3

engine = pyttsx3.init()
name = input("What's your name? ")
engine.say(f"hello, {name}")
engine.runAndWait()
```

```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

```

```python
# Display the resulting image
pil_image.show()
```

```python
# Generates a QR code
# https://github.com/lincolnloop/python-qrcode

import os
import qrcode

# Generate QR code
img = qrcode.make("https://youtu.be/oHg5SJYRHA0")

# Save as file
img.save("qr.png", "PNG")

# Open file
os.system("open qr.png")
```

```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.")
```