Programming the RPi & Python Primer

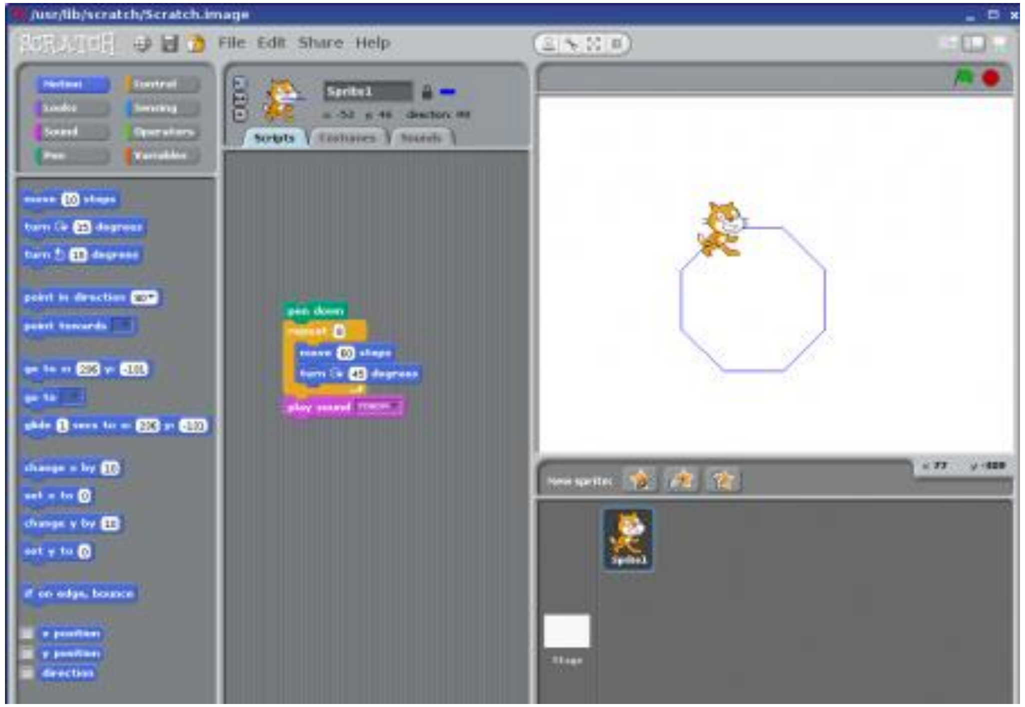
With help from

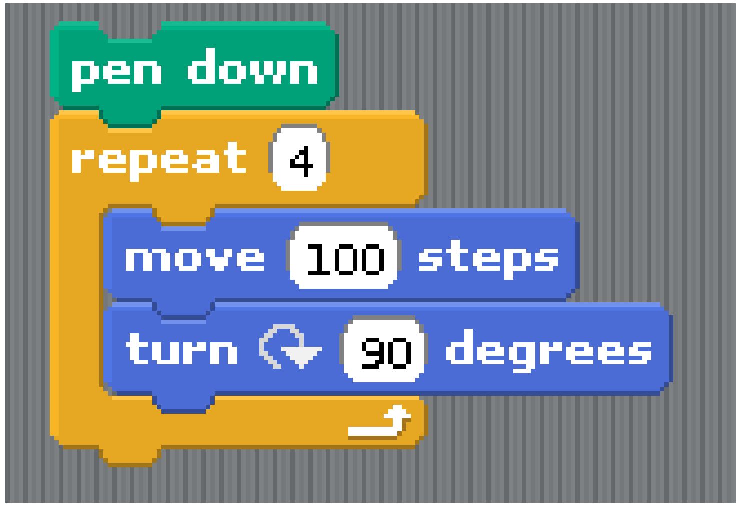
https://lobster1234.github.io/2017/05/25/python-java-primer/

Programming the RPi

- By default, supporting Python as the educational language
- Any language which will compile for ARMv6 can be used with the Raspberry Pi
- For primary age SCRATCH game maker is bundled
- Raspbian also contains Java SE platform

Programming – SCRATCH

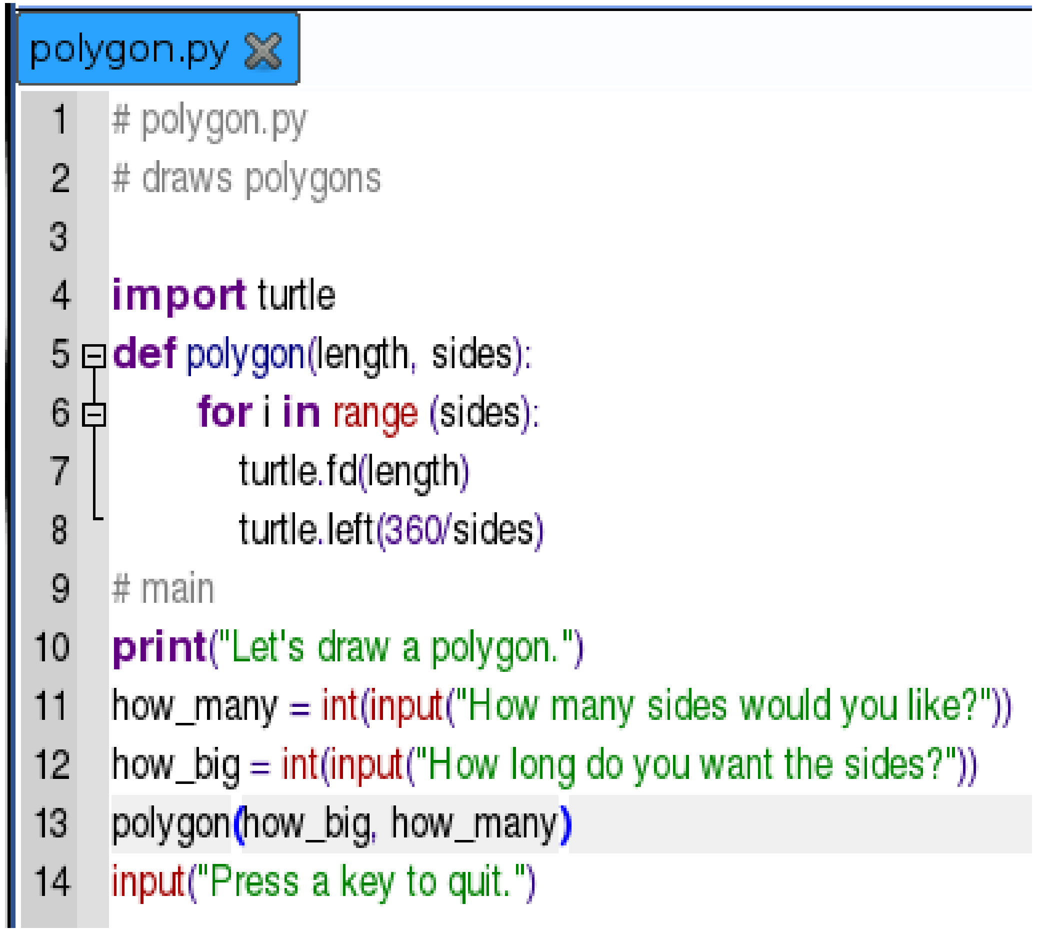




```
pen down
repeat 4
    move 100 steps
    turn ↻ 90 degrees
```

scratch is free from MIT

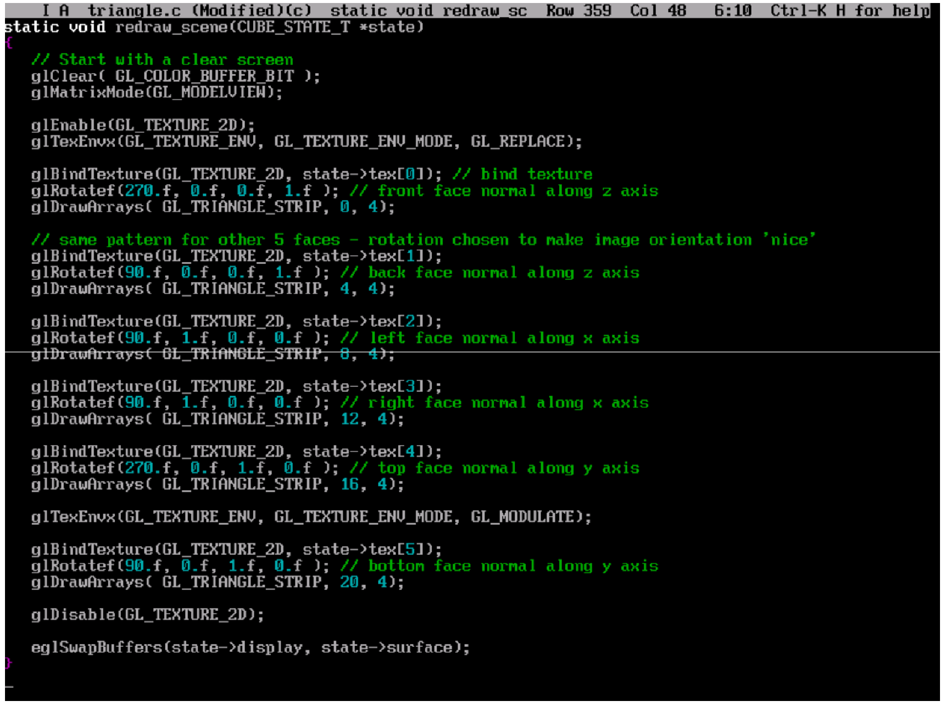
Programming – Python



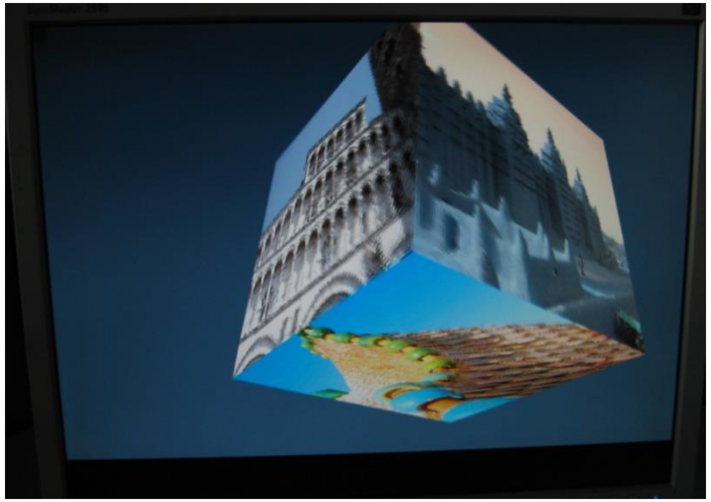
```
# polygon.py
# draws polygons

import turtle
def polygon(length, sides):
    for i in range (sides):
        turtle.fd(length)
        turtle.left(360/sides)
# main
print("Let's draw a polygon.")
how_many = int(input("How many sides would you like?"))
how_big = int(input("How long do you want the sides?"))
polygon(how_big, how_many)
input("Press a key to quit.")
```

Programming - OpenGL

- Raspberry Pi incorporates a powerful graphics accelerator – OpenGL
  - Examples, including Quake 3 at
    - https://github.com/raspberrypi/quake3



```
   I A  triangle.c (Modified)(c)  static void redraw_sc  Row 359  Col 48   6:10  Ctrl-K H for help
static void redraw_scene(CUBE_STATE_T *state)
{
   // Start with a clear screen
   glClear( GL_COLOR_BUFFER_BIT );
   glMatrixMode(GL_MODELVIEW);

   glEnable(GL_TEXTURE_2D);
   glTexEnvx(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);

   glBindTexture(GL_TEXTURE_2D, state->tex[0]); // bind texture
   glRotatef(270.f, 0.f, 0.f, 1.f ); // front face normal along z axis
   glDrawArrays( GL_TRIANGLE_STRIP, 0, 4);

   // same pattern for other 5 faces - rotation chosen to make image orientation 'nice'
   glBindTexture(GL_TEXTURE_2D, state->tex[1]);
   glRotatef(90.f, 0.f, 0.f, 1.f ); // back face normal along z axis
   glDrawArrays( GL_TRIANGLE_STRIP, 4, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[2]);
   glRotatef(90.f, 1.f, 0.f, 0.f ); // left face normal along x axis
   glDrawArrays( GL_TRIANGLE_STRIP, 8, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[3]);
   glRotatef(90.f, 1.f, 0.f, 0.f ); // right face normal along x axis
   glDrawArrays( GL_TRIANGLE_STRIP, 12, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[4]);
   glRotatef(270.f, 0.f, 1.f, 0.f ); // top face normal along y axis
   glDrawArrays( GL_TRIANGLE_STRIP, 16, 4);

   glTexEnvx(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);

   glBindTexture(GL_TEXTURE_2D, state->tex[5]);
   glRotatef(90.f, 0.f, 1.f, 0.f ); // bottom face normal along y axis
   glDrawArrays( GL_TRIANGLE_STRIP, 20, 4);

   glDisable(GL_TEXTURE_2D);

   eglSwapBuffers(state->display, state->surface);
}
_
```

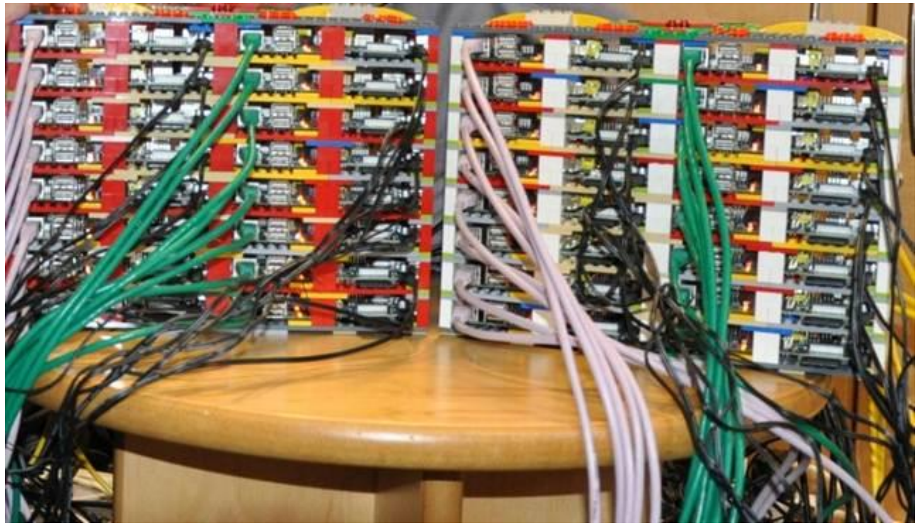


Some Other Things

- Can run Kodi software
  - Turn your TV into a Smart TV!
- Emulators for tons of old platforms (C64, Atari, NES, etc.)
  - I have a mame emulator.
- Brew your own beer!
  - I've been doing this with the help of an RPi since 2014
- Has its own app store
  - http://store.raspberrypi.com/

Projects Using Raspberry Pi

- Raspberry Pi Cluster
  - https://arstechnica.com/information-technology/2012/09/university-builds-cheap-supercomputer-with-raspberry-pi-and-legos/
  - http://www.youtube.com/watch?v=Jq5nrHz9I94



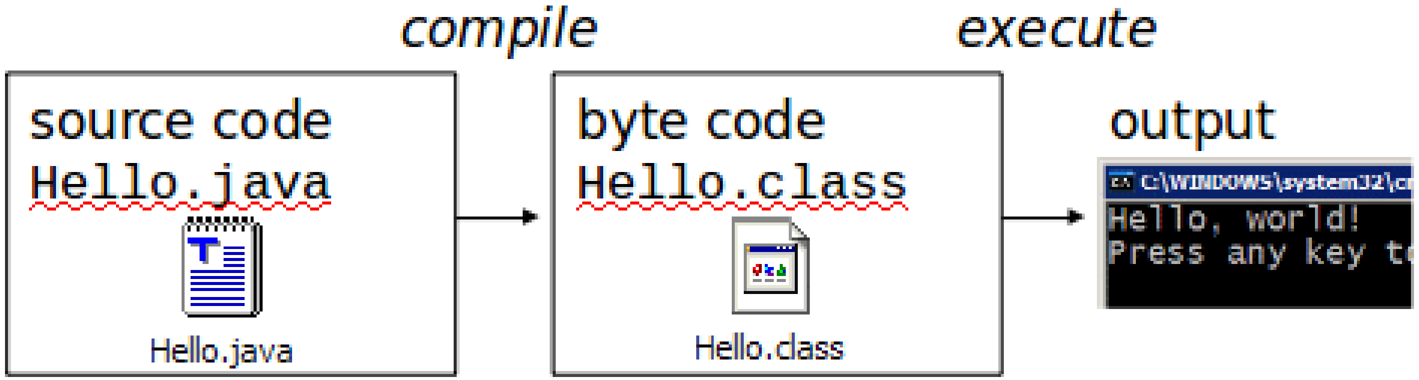
What’s Python

- Created 1991
- Used for Web Dev, Software Dev, Maths, Embedded Devices (micro python), Scripting
- Why Python
  - Works on many platforms
  - Easy to learn(simple syntax)
- One thing to always remember about Python:

  Relies on indentation(using white space) to define scope for if statements, loops, functions. Java uses curly-brackets for this.

Python is interpreted

- Many languages require compilation of program to a form the machine understands



compile *execute*

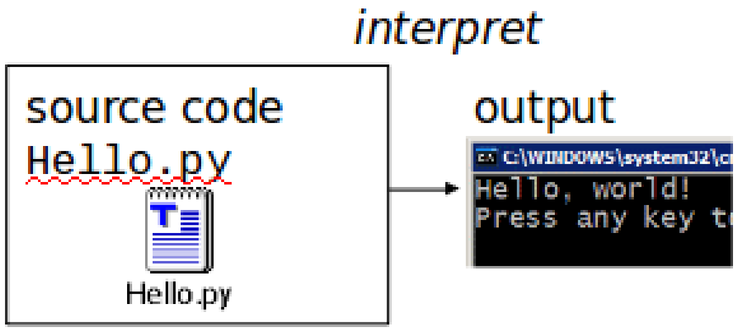
source code
Hello.java

byte code
Hello.class

output

```
Hello, world!
Press any key t
```

- Python is interpreted directly to machine instructions



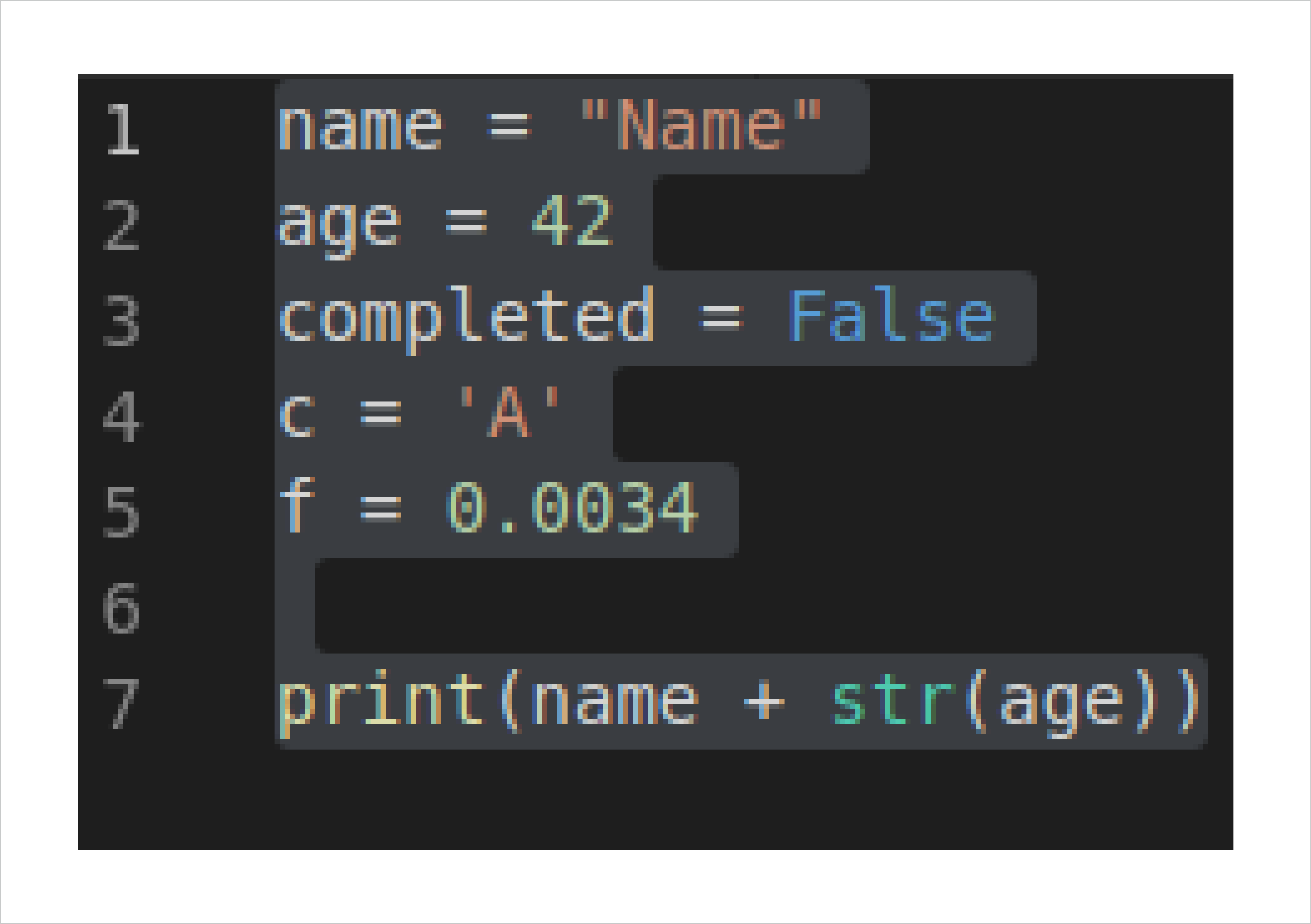
interpret

source code
Hello.py

output

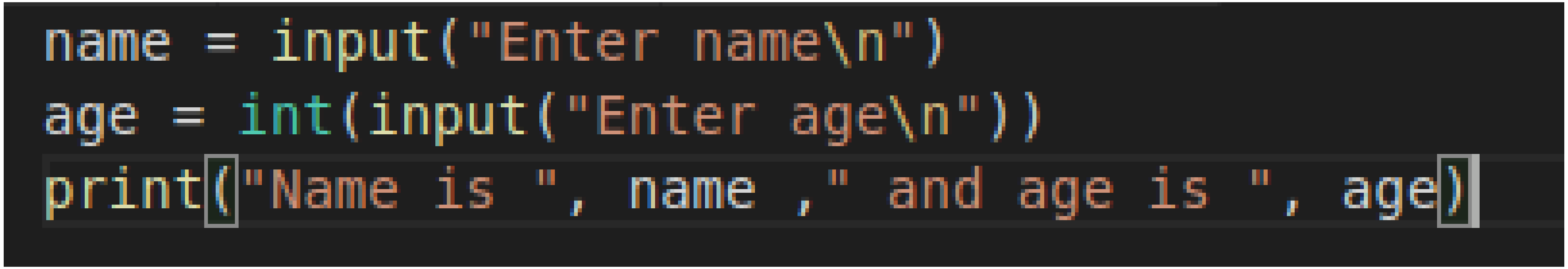
```
Hello, world!
Press any key t
```

Variables



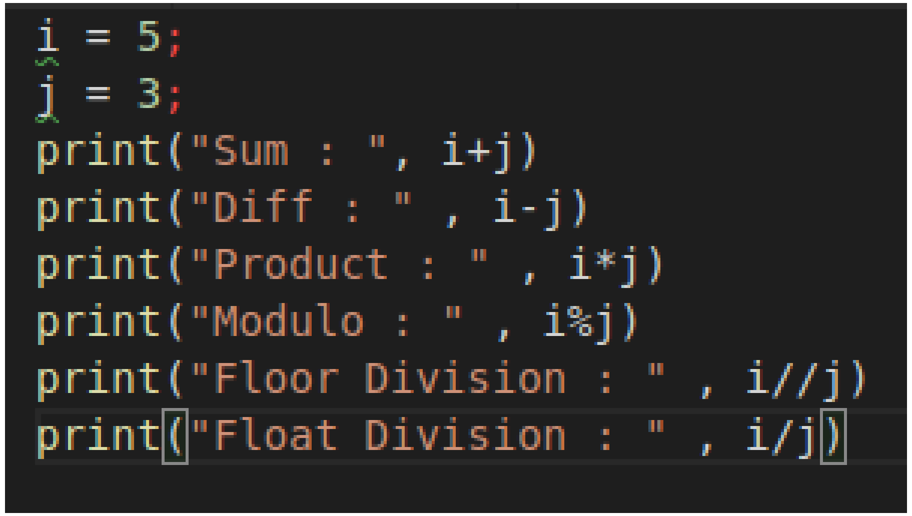
```
name = "Name"
age = 42
completed = False
c = 'A'
f = 0.0034

print(name + str(age))
```

Read Input

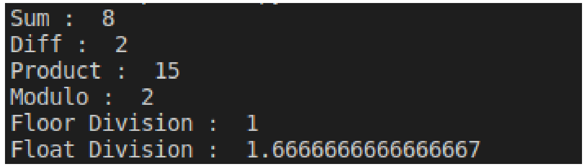


```python
name = input("Enter name\n")
age = int(input("Enter age\n"))
print("Name is ", name ," and age is ", age)
```

Operators

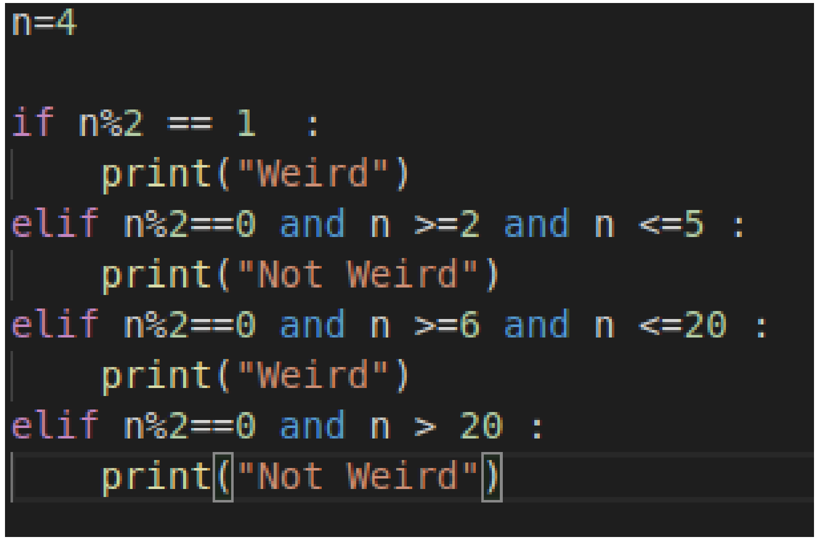


```
i = 5;
j = 3;
print("Sum : ", i+j)
print("Diff : " , i-j)
print("Product : " , i*j)
print("Modulo : " , i%j)
print("Floor Division : " , i//j)
print("Float Division : " , i/j)
```



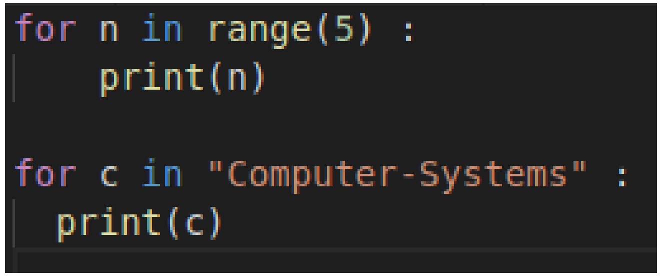
```
Sum :  8
Diff :  2
Product :  15
Modulo :  2
Floor Division :  1
Float Division :  1.6666666666666667
```

If-then-else

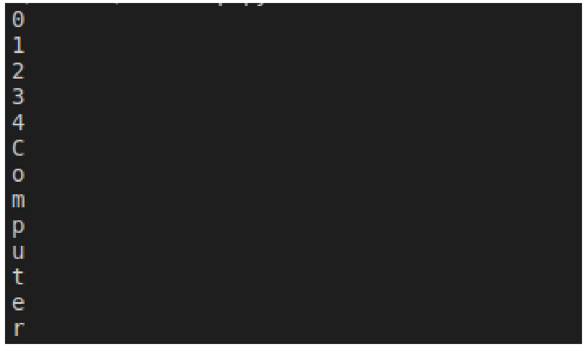


```
n=4

if n%2 == 1  :
    print("Weird")
elif n%2==0 and n >=2 and n <=5 :
    print("Not Weird")
elif n%2==0 and n >=6 and n <=20 :
    print("Weird")
elif n%2==0 and n > 20 :
    print("Not Weird")
```

For loop

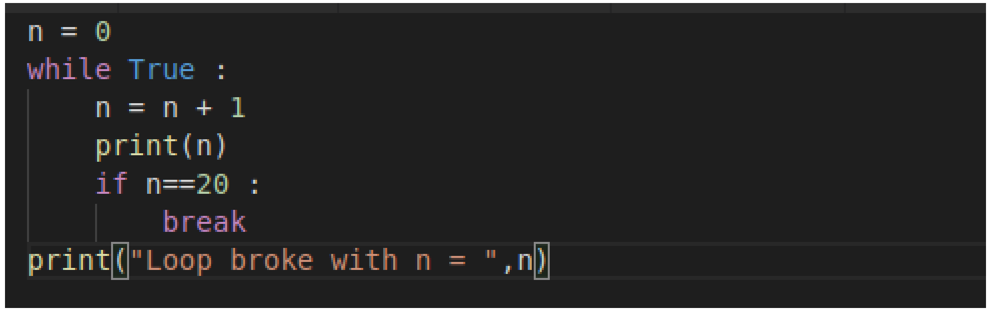


```
for n in range(5) :
    print(n)

for c in "Computer-Systems" :
  print(c)
```

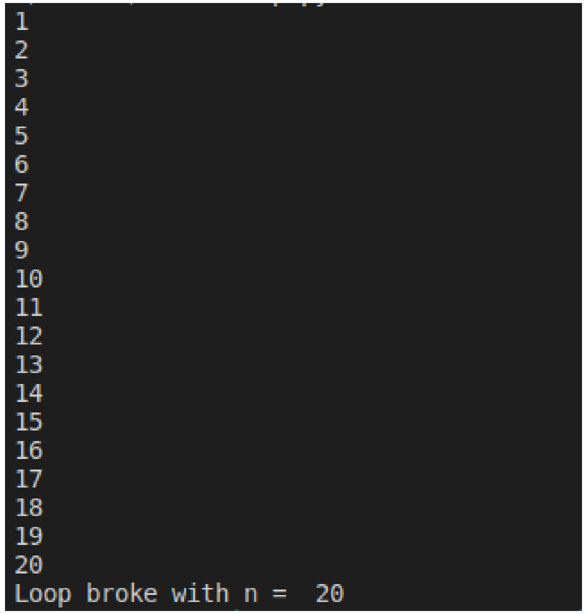


```
0
1
2
3
4
C
o
m
p
u
t
e
r
```

While loop

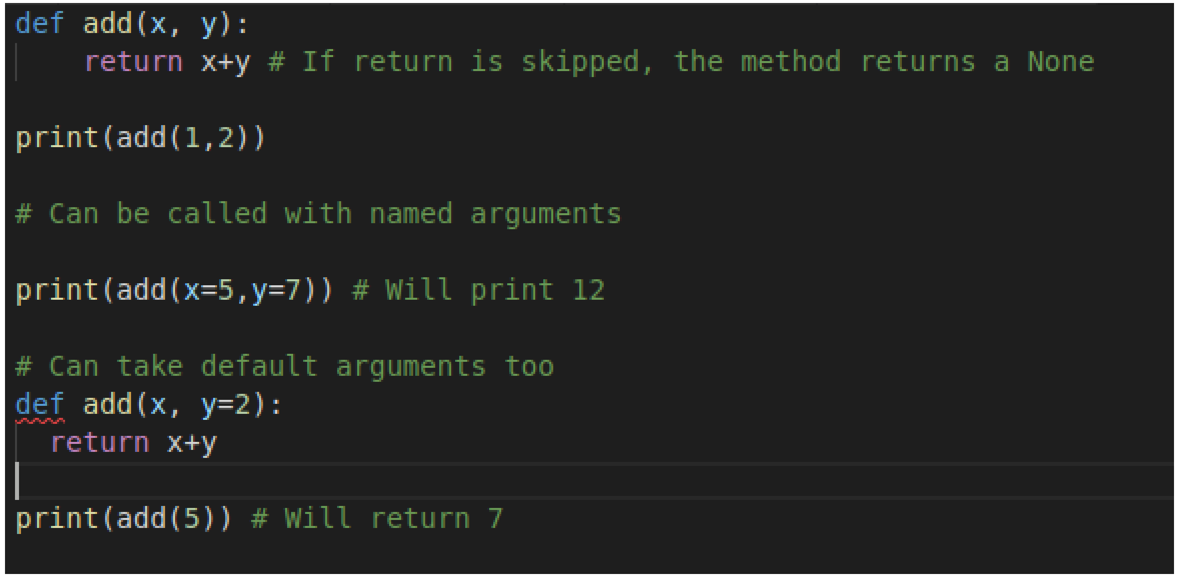


```python
n = 0
while True :
    n = n + 1
    print(n)
    if n==20 :
        break
print("Loop broke with n = ",n)
```

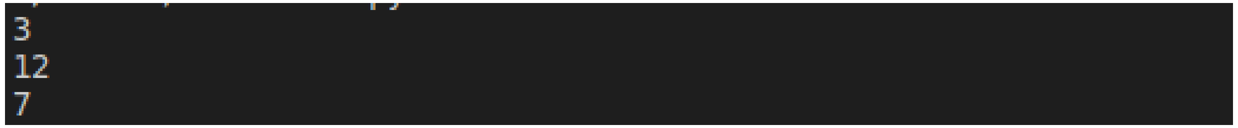


```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Loop broke with n =  20
```

Functions

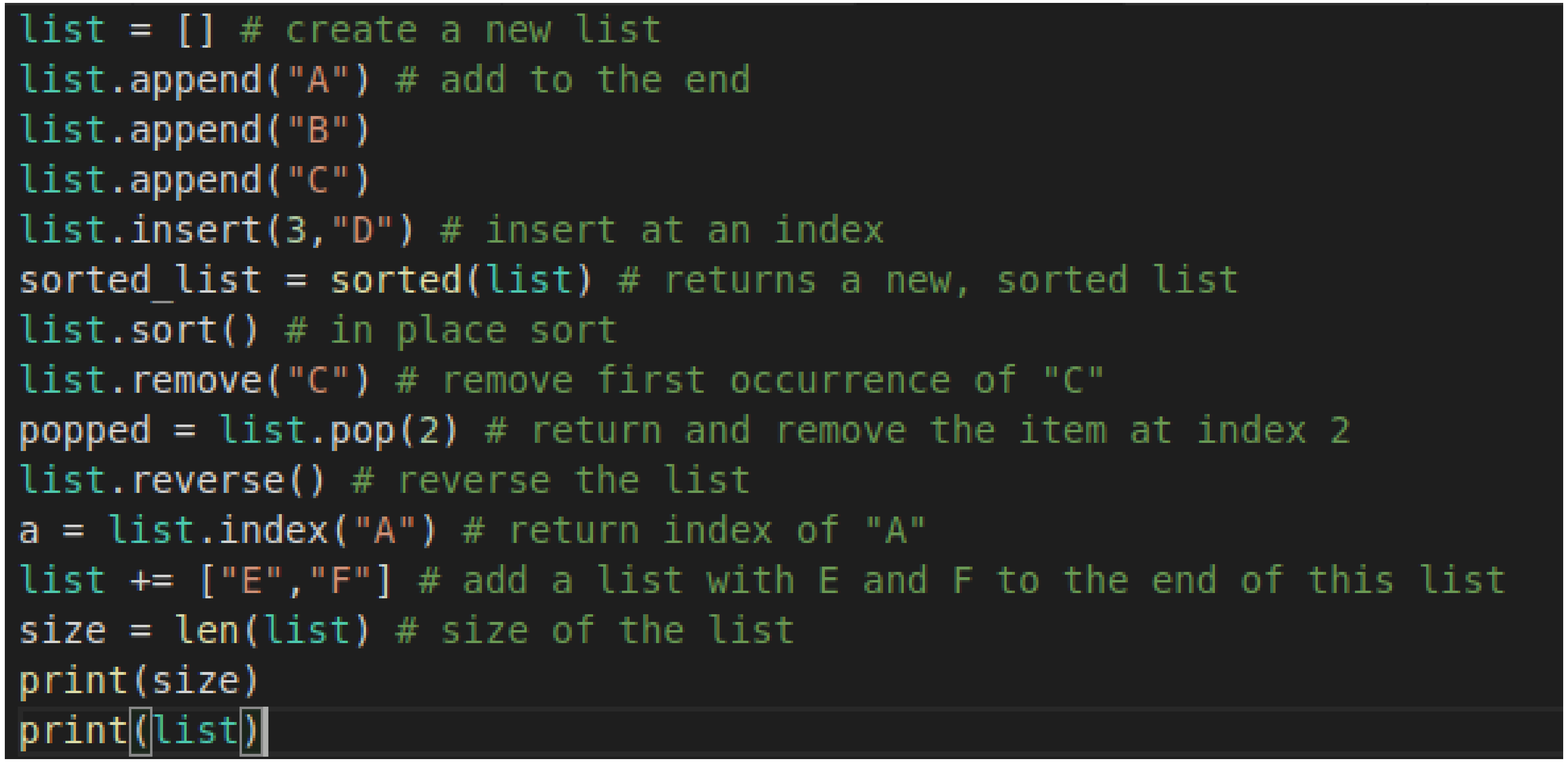


```
def add(x, y):
    return x+y # If return is skipped, the method returns a None

print(add(1,2))

# Can be called with named arguments

print(add(x=5,y=7)) # Will print 12

# Can take default arguments too
def add(x, y=2):
  return x+y

print(add(5)) # Will return 7
```

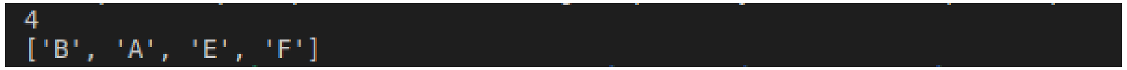


```
3
12
7
```

Lists



```
list = [] # create a new list
list.append("A") # add to the end
list.append("B")
list.append("C")
list.insert(3,"D") # insert at an index
sorted_list = sorted(list) # returns a new, sorted list
list.sort() # in place sort
list.remove("C") # remove first occurrence of "C"
popped = list.pop(2) # return and remove the item at index 2
list.reverse() # reverse the list
a = list.index("A") # return index of "A"
list += ["E","F"] # add a list with E and F to the end of this list
size = len(list) # size of the list
print(size)
print(list)
```



```
4
['B', 'A', 'E', 'F']
```