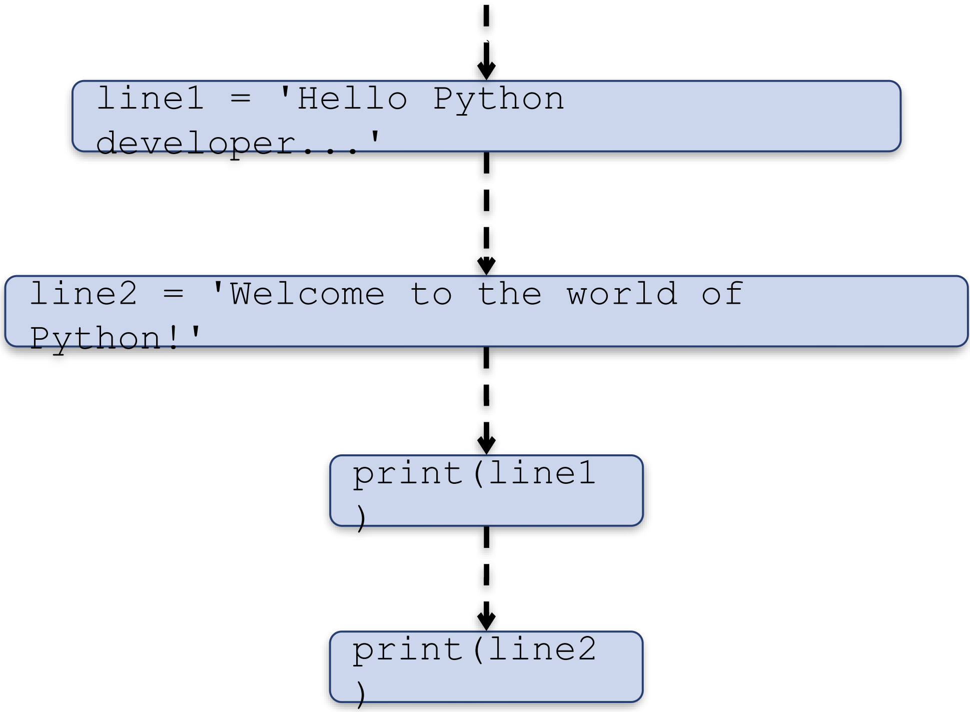
Imperative Programming

- What is a Python program?
- print() statement
- input() statement
- Type conversion statements str(), int(), float()
- `if/elif/else` conditional statement

Python program

A Python program is a sequence of Python statements

- Stored in a text file with the extension .py
- Executed using an integrated development environment (IDE) or “from the command line”



```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

```
line1 = 'Hello Python developer...'
line2 = 'Welcome to the world of Python!'
print(line1)
print(line2)
```

hello.py

```
$ python hello.py
Hello Python developer…
```

Built-in function `print()`

The builtin function `print()` echoes its parameter to the output window

- The argument can be any object: an integer, a float, a string, a list …
- The "string representation" of the object is printed
- When you output a string using `print`, the quote marks around the string are omitted
- Be default, a print statement ends its output with a newline (return). But you can change this to any string you like, including the empty string. Try `print('hello', end='!!!')`

```
>>> print(0)
0
>>> print(0.0)
0.0
>>> print('zero')
zero
>>> print([0, 1, 'two'])
[0, 1, 'two']
```

Built-in function `input()`

The builtin function `input()` requests and reads input from the user interactively

- Its (optional) input argument is the request message
- Typically used on the right side of an assignment statement

When executed:

1. `input` prints the request message
2. `input` accepts typed input from the user
3. The user input may then be assigned to the variable on the left side of the assignment statement

```
>>> first = input('Your first name: ')
Your first name: Michael
>>> last = input('Your last name: ')
Your last name: Rojas
>>> first + ' ' + last
'Michael Rojas'
```

Change string input to another type

Function `input()` treats anything the user enters as a string

What if we want the user to interactively enter a number?

Use a type conversion function

- int() changes a string, Boolean or float type to an int
- float() changes a string, Boolean or int type to a float
- str() changes an int, float or Boolean type to a string

```
>>> age = input('Enter
your age: ')
Enter your age: 18
>>> age
'18'
>>> int(age)
18
```

Exercise

Write a program that:

1. Requests the user's name
2. Requests the user's age
3. Computes the user's age one year from now and prints the message shown

```
>>> Enter your name: Marie
Enter your age: 17
Marie, you will be 18 next year!
```

```
name = input('Enter your name: ')
age = int(input('Enter your age: '))
line = name + ', you will be ' + str(age+1) + ' next year!'
print(line)
```

Exercise

Write a program that:

1. Requests the user’s name
2. Requests the user’s age
3. Prints a message saying whether the user is eligible to vote or not

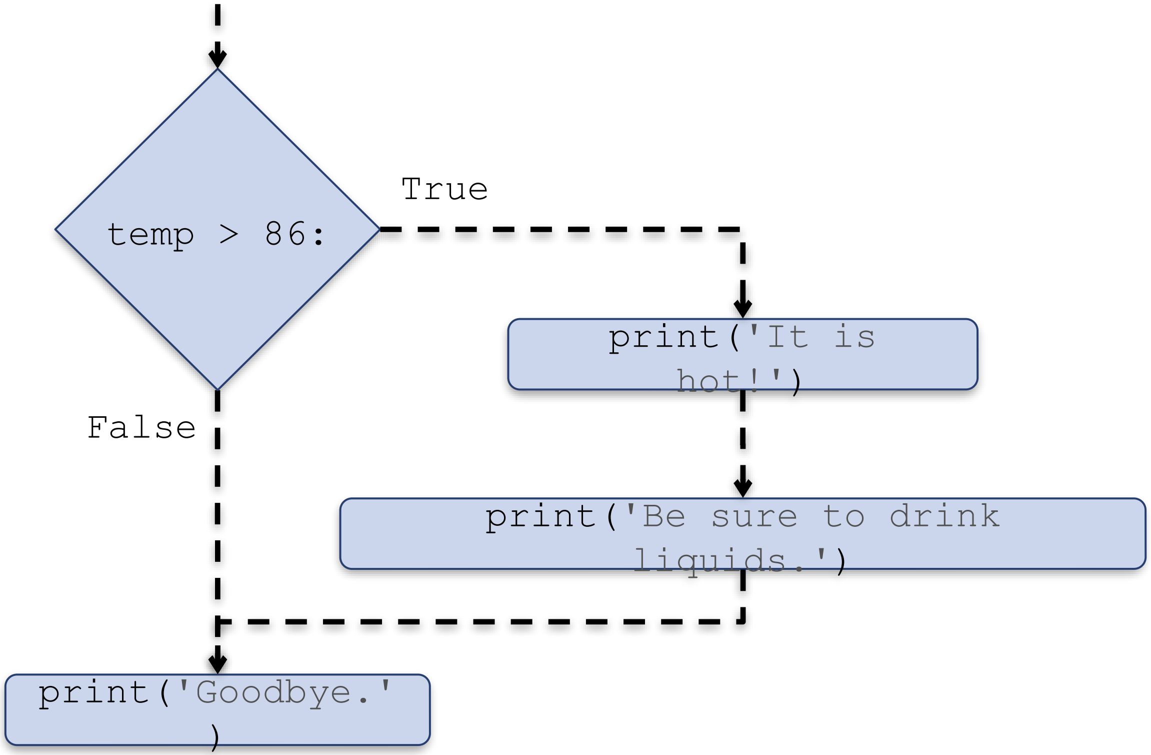
To do this, we need a way to execute a Python statement if a condition is true

if statement

```
if <condition>:
    <indented code block>
<non-indented statement>
```

```
if temp > 86:
    print('It is hot!')
    print('Be sure to drink liquids.')
print('Goodbye.')
```

The value of `temp` is 90.



temp > 86:

True

False

print('It is hot!')

print('Be sure to drink liquids.')

print('Goodbye.')

Exercises

Write corresponding if statements:

a) If `age` is greater than 62, print `'You can get Social Security benefits’`

b) If `'large bonuses'` appears in the string `report` print `'Vacation time!'`

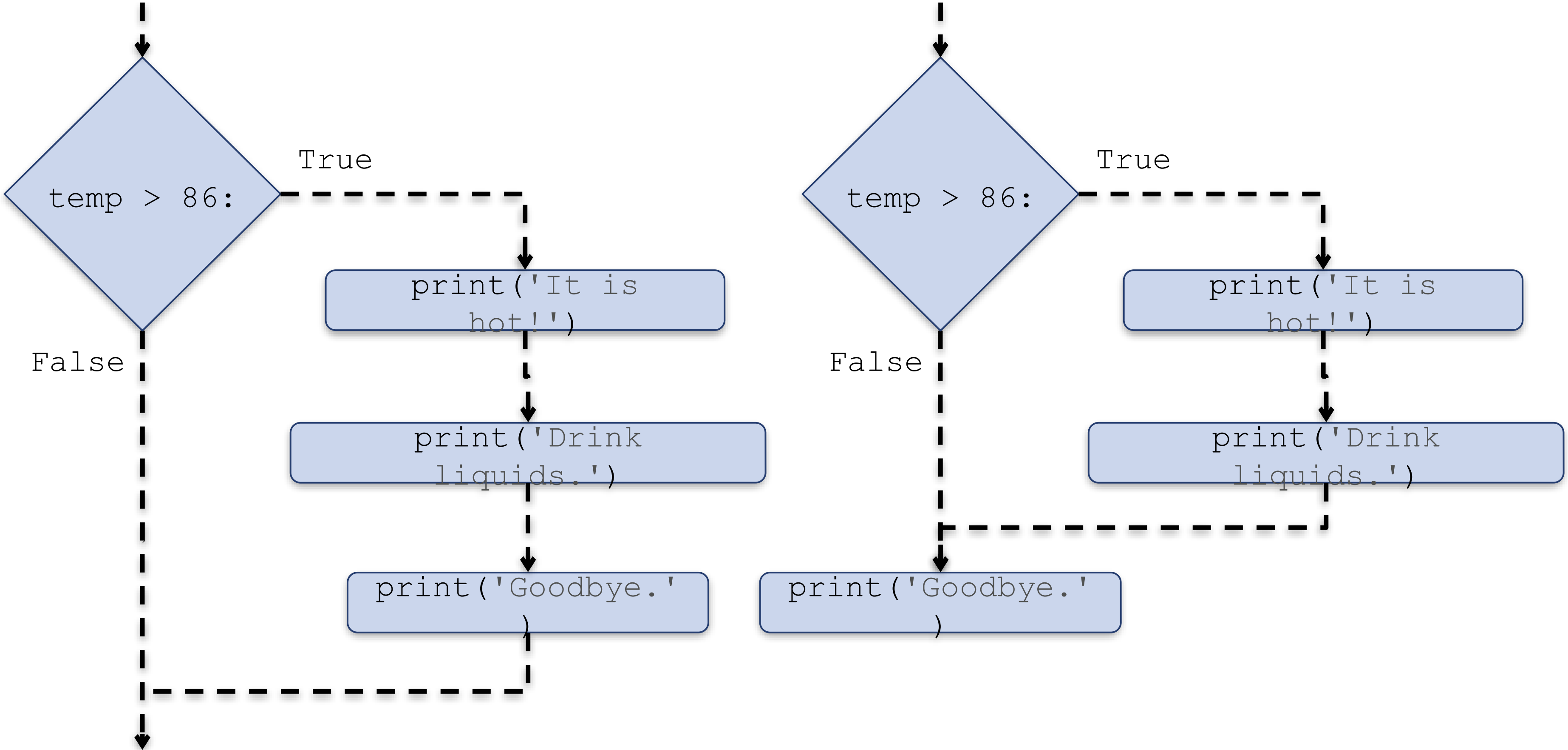
c) If `hits` is greater than 10 and `shield` equals 0, print `"You're dead..."`

```
>>> hits = 12
>>> shield = 0
>>> if hits > 10 and shield == 0:
        print("You're dead...")


You're dead...
>>> hits, shield = 12, 2
>>> if hits > 10 and shield == 0:
        print("You're dead...")


>>>
```

Indentation is critical

```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
    print('Goodbye.')
```

```
if temp > 86:
    print('It is hot!')
    print('Drink liquids.')
print('Goodbye.')
```

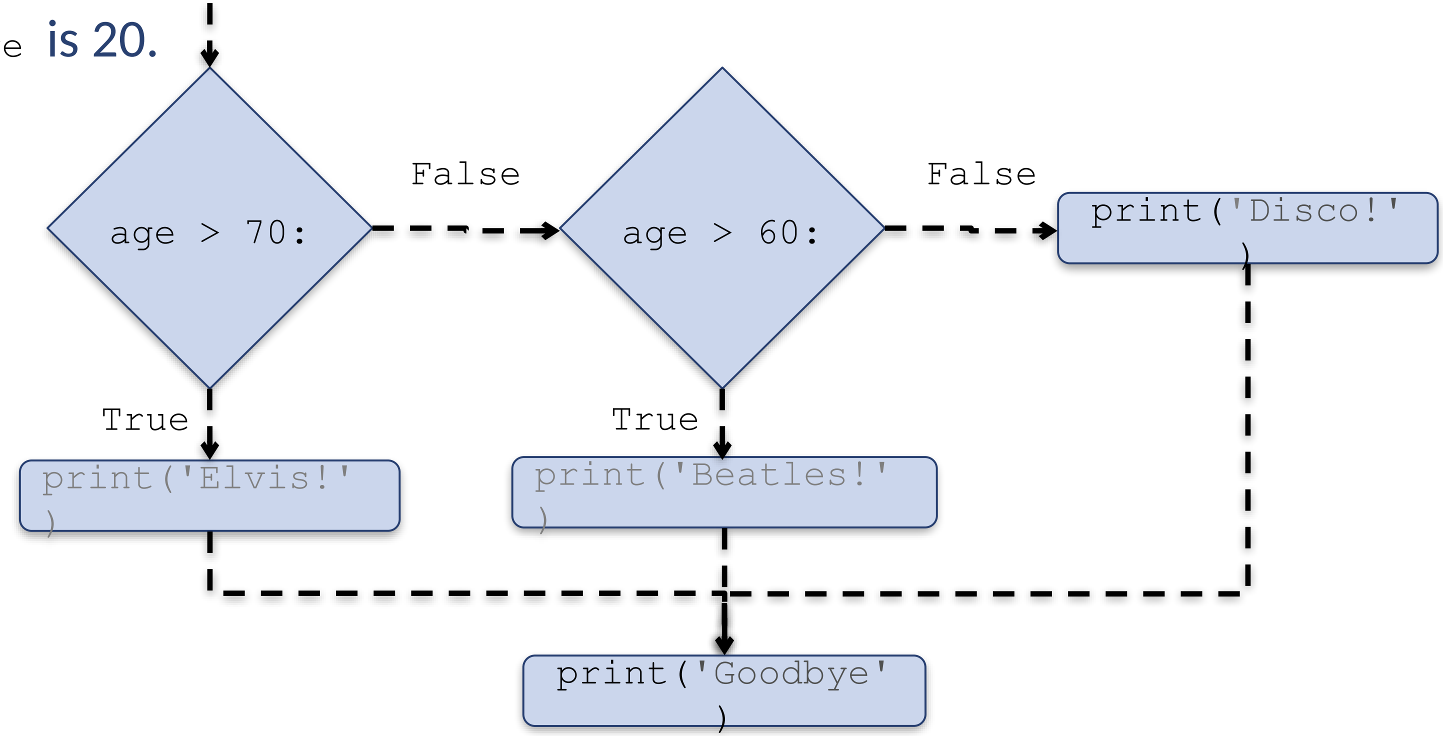


if/elif/else statement

```
if <condition>:
    <indented code block 1>
elif:
    <indented code block 2>
else:
    <indented code block 3>
<non-indented statement>
```

```
if age > 70:
    print('Elvis!')
elif age > 60:
    print('Beatles!')
else:
    print('Disco!')
print('Goodbye')
```

The value of `age` is 20.



Exercise

Extend this music preference program by adding more `elif` clauses so that:

1) It requests the user's name
2) It requests the user's age
3) It prints a message with the user's musical preference for all ages, by decade

```
>>>
Enter your name: Marie
Enter your age: 82
Marie, you like Elvis.
>>>
============RESTART==========
======
>>>
Enter your name: Marie
Enter your age: 40
Marie, you like Sinead.
>>>
```

```python
name = input('Enter your name: ')
age = int(input('Enter your age: '))
if age > 70:
    print(name + ", you like Elvis.")
else:
    print(name + ", you like Sinead.")
```

Exercise

Remember in lecture 1 we noted that you must be careful with arithmetic operations on floats.

```
f = int(input(‘Enter first number: ’))
g = int(input(‘Enter second number: ’)
if f == g:
    print("Equal")
else:
    print("Not Equal")
```

```
>>> >>> 8.0 == 0.1 + 0.1 +
0.1 + 0.1 + 0.1 + 0.1 + 0.1 +
0.1
False
>>>
============RESTART==========
======
>>> f = 0.1
>>> g = 0.1 + 0.1 + 0.1 + 0.1
+ 0.1 + 0.1 + 0.1 + 0.1 + 0.1
+ 0.1
>>> if f == g:
        print("Equal")
else:
        print("Not Equal")


Not Equal
```

Instead, you should do something like:

```
f = int(input(‘Enter first number: ’))
g = int(input(‘Enter second number: ’)
diff = f - g
theta = 0.001
if diff < theta:
    print("Equal")
else:
    print("Not Equal")
```