Chapter 3: Imperative Programming

Resource: Introduction to Computing Using Python by Ljubomir Perkovic

Topics to cover:

- First Python program
- 3.2 Execution Control Structures
    - One-Way Decisions
    - Two-Way Decisions
    - for loops
    - Range
- 3.3 User defined functions
- 3.4 Variable and Assignments
- 3.5 Parameter Passing
-

First Python Program

Create file **hello.py**

```
line1= ‘Hello Python dev’

line2=’Welcome to the world of Python!’

print(line1)

print(line2)
```

Closer look

Hello.py = module

Module = file containing python code that ends in **.py** (math.py is a module)

Built-in functions:

- **print()** - prints

Create input.py

```
first = input(‘Enter your first name: ‘)

last = input(‘Enter your last name: ‘)

line1 = ‘Hello ‘ + first + ‘ ‘ + last + ‘....’

print(line1)

print(‘Welcome to the world of Python!’)
```

-

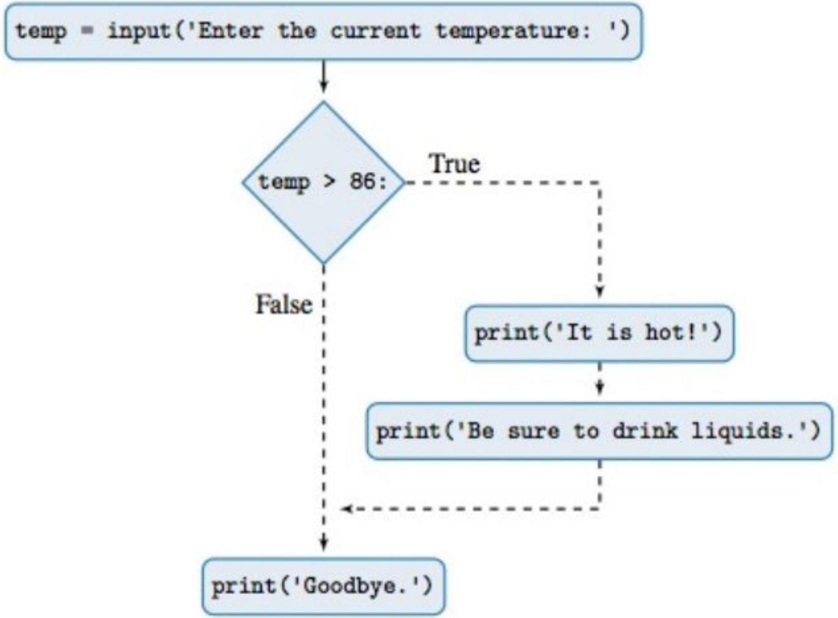
Closer look

Built-in function:

- **input()** - prints input argument and waits for user to type something and hit `Enter` or `Return`
    - Returns string **always**
- **eval()** - evaluates string as if it were a python expression
    - Can use this with input to return more than just strings

Execution Control Structures: One-Way Decisions

One-Way Decisions: control whether to execute a fragment of code based on a condition (**if** condition is True, code fragment is executed otherwise it is not)



```
temp = input('Enter the current temperature: ')
```

```
temp > 86:
```

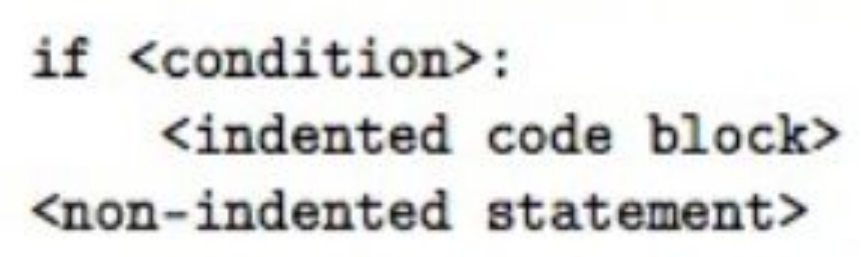
```
print('It is hot!')
```

```
print('Be sure to drink liquids.')
```

```
print('Goodbye.')
```

if statement

1. Evaluate boolean expression (**if** followed by boolean expression and `:`)
2. If True, execute indented code block (indented 4 spaces)
3. If False, skip indented code block
4. Execute non-indented code block



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