# Why Python?

Introduction

# Why Learn Python ?

5th Largest StackOverflow Community

4th Most-Used Language at GitHub

3rd Largest Meetup Community

Java programs end up being longer and more painful to write than the equivalent Python programs

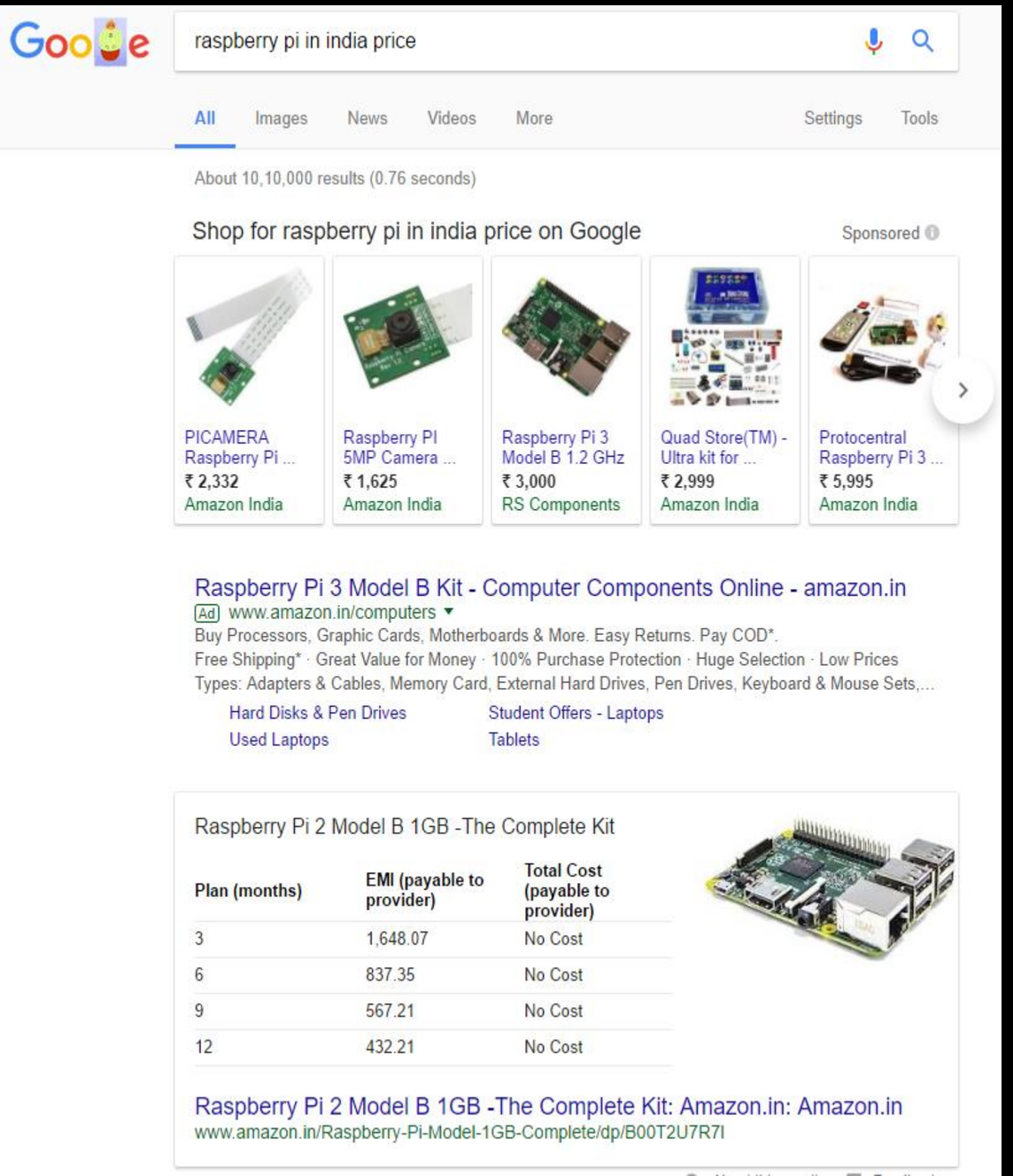
```
name = raw_input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
   counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count >
bigcount:
        bigword = word
        bigcount = count
print bigword, bigcount
```

```
python words.py
Enter file: words.txt
to 16
```

```
python words.py
Enter file: clown.txt
the 7
```




YES !

You could start coding in Python with just

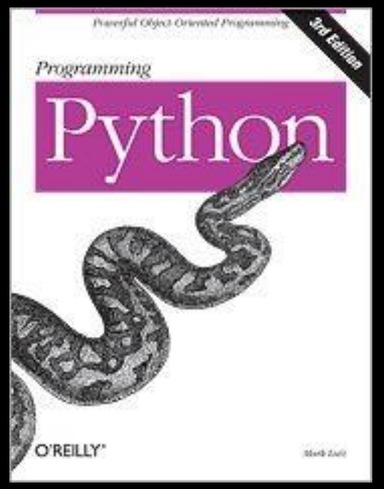
**~ Rs. 1,625**

of upfront hardware cost!

An interpreted **language**, **Python** has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in **languages** such as C++ or Java.

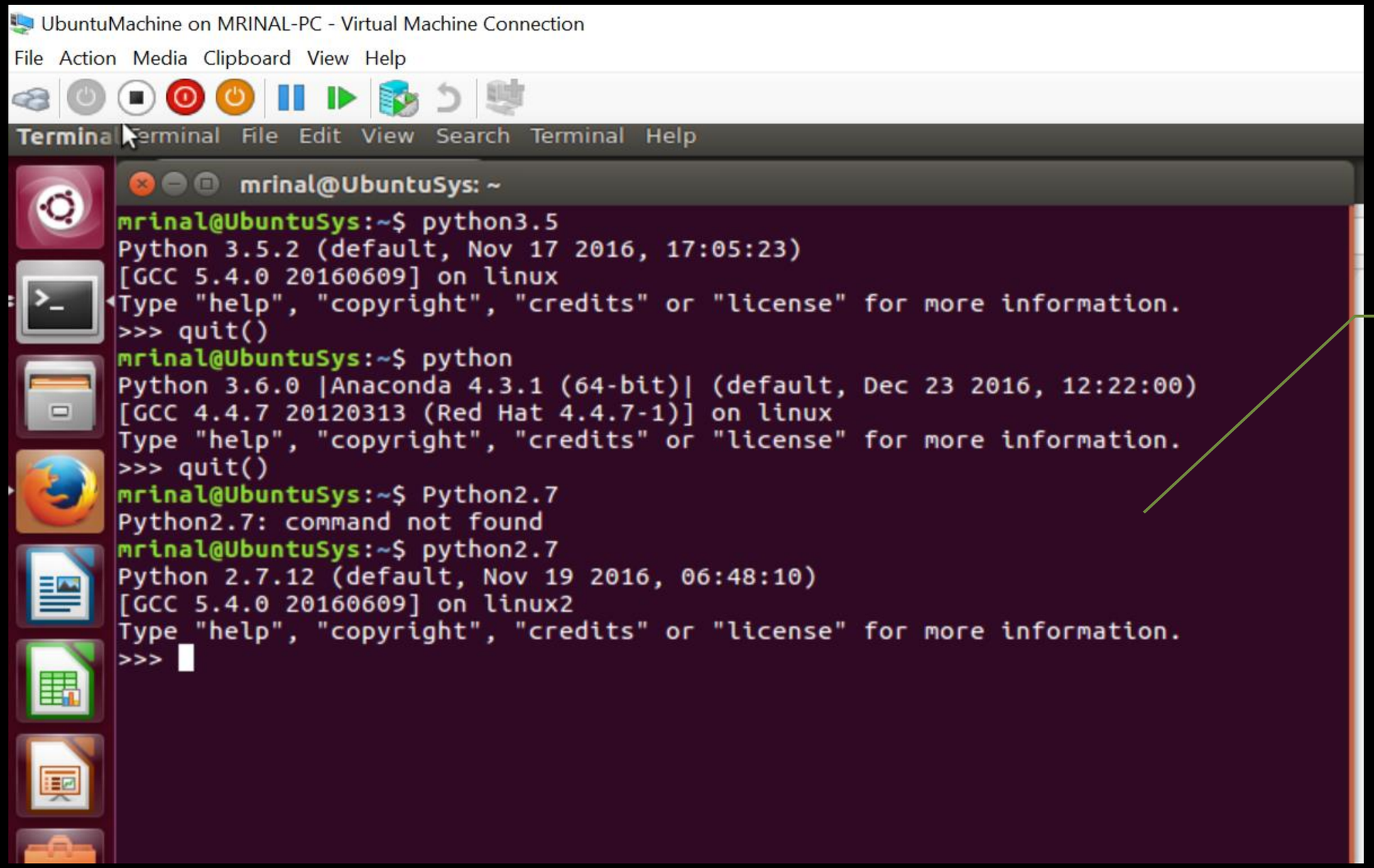
Python (programming language) - Wikipedia
https://en.wikipedia.org/wiki/Python_(programming_language)

www.python.org

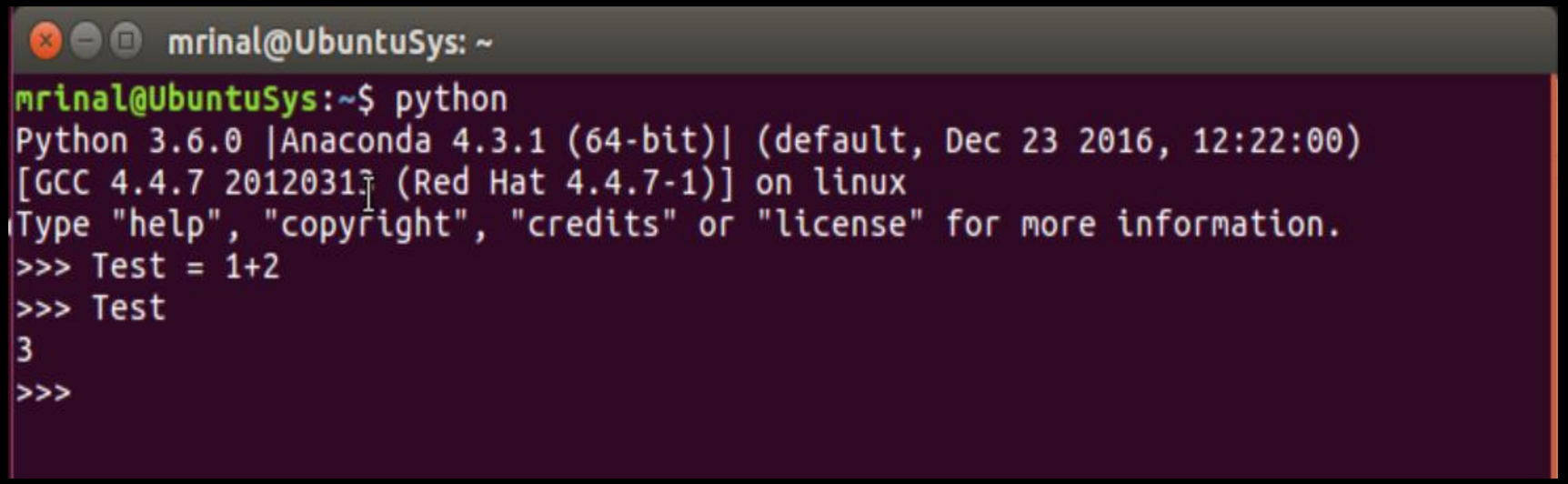



Guido van Rossum is the author of the Python programming language. In the Python community, Van Rossum is known as a "Benevolent Dictator For Life" (BDFL), meaning that he continues to oversee the Python development process, making decisions where necessary.

# Talking to Python




You can call-out different version of Python…




This is a good test to make sure that you have Python correctly installed. Note that quit() also works to end the interactive session.

# What Do We Say?

# Elements of Python

- Vocabulary / Words - Variables and Reserved words (Chapter 2)
- Sentence structure - valid syntax patterns (Chapters 3-5)
- Story structure - constructing a program for a purpose

```
name = raw_input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
   counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count >
bigcount:
        bigword = word
        bigcount = count
print bigword, bigcount
```

# A short “story” about how to count words in a file in Python

```
python words.py
Enter file: words.txt
to 16
```

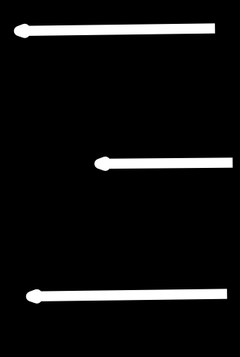
# Reserved Words

- You cannot use reserved words as variable names / identifiers

and del for is raise assert elif from
lambda return break else global
not try class except if or while
continue exec import pass yield
def finally in print as with

# Sentences or Lines

```
x = 2
x = x + 2
print x
```



Assignment statement

Assignment with expression

Print statement

Variable

Operator

Constant

Reserved Word

# Programming Paragraphs

# Python Scripts

- Interactive Python is good for experiments and programs of 3-4 lines long.
- Most programs are much longer, so we type them into a file and tell Python to run the commands in the file.
- In a sense, we are “giving Python a script”.
- As a convention, we add “.py” as the suffix on the end of these files to indicate they contain Python.

# Writing a Simple Program

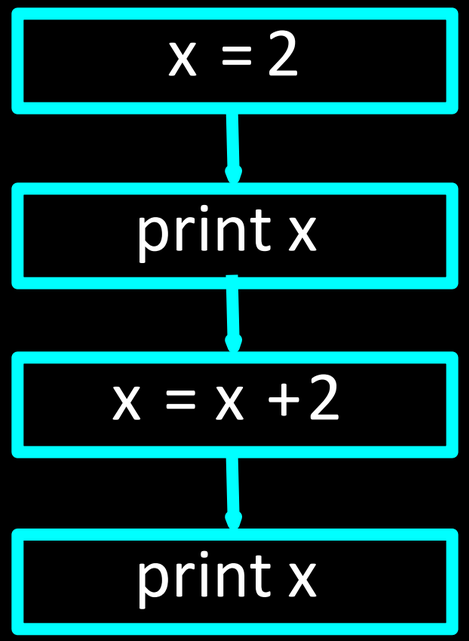
# Interactive versus Script

- Interactive
  - > You type directly to Python one line at a time and it responds
- Script
  - > You enter a sequence of statements (lines) into a file using a text editor and tell Python to execute the statements in the file

# Program Steps or Program Flow

- Like a recipe or installation instructions, a program is a sequence of steps to be done in order.
- Some steps are conditional - they may be skipped.
- Sometimes a step or group of steps are to be repeated.
- Sometimes we store a set of steps to be used over and over as needed several places throughout the program (Chapter 4).

# Sequential Steps



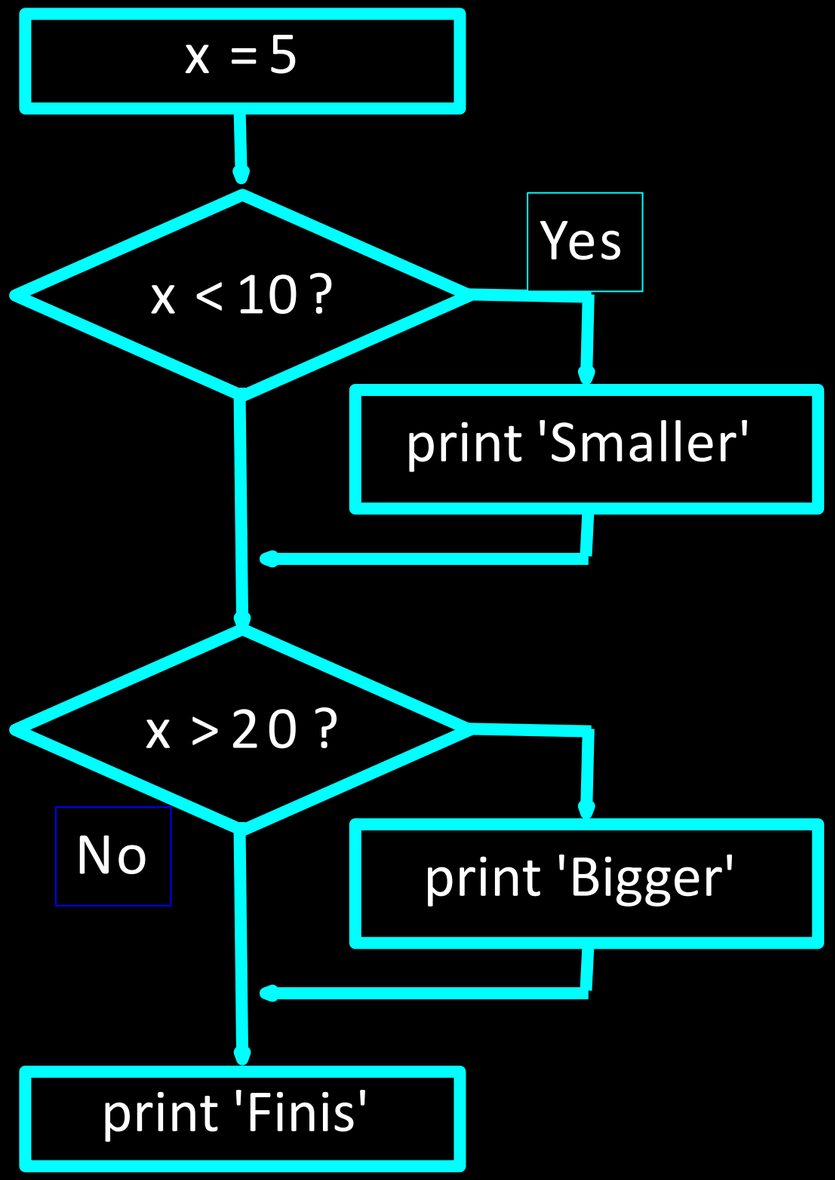

Program:

```
x = 2
print x
x = x +2
print x
```

Output:

```
2
4
```

When a program is running, it flows from one step to the next.
As programmers, we set up “paths” for the program to follow.




# Conditional Steps

Program:

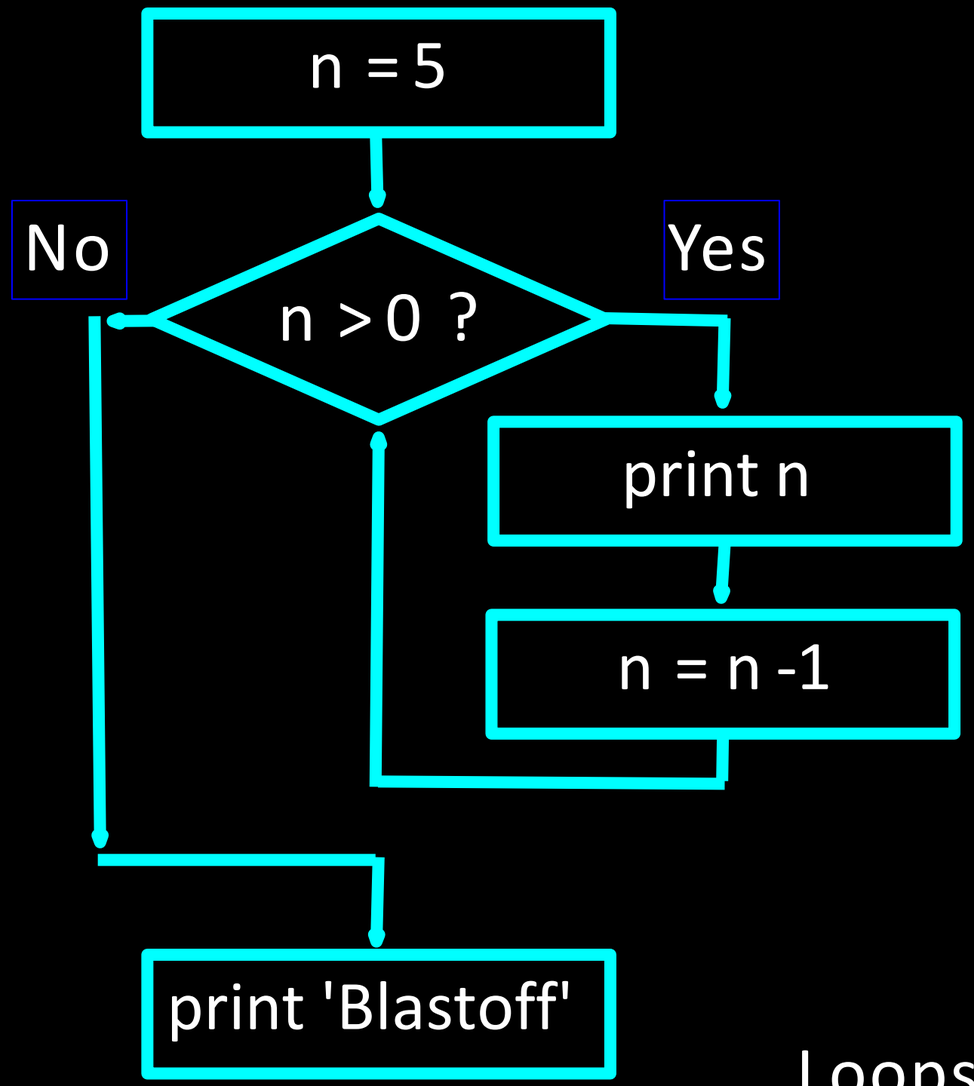
```
x = 5
if x <10:
    print 'Smaller'
if x >20:
    print 'Bigger'

print 'Finis'
```

Output:

```
Smaller
Finis
```

# Repeated Steps




Program:

```
n = 5
while n >0 :
    print n
    n = n – 1
print 'Blastoff!'
```

Output:

```
5
4
3
2
1
Blastoff!
```

Loops (repeated steps) have iteration variables that change each time through a loop. Often these iteration variables go through a sequence of numbers.

```
name = raw_input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
   counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count >
bigcount:
        bigword = word
        bigcount = count

print bigword, bigcount
```

Sequential

Repeated

Conditional

```
name = raw_input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()
counts = dict()
for word in words:
    counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count >
bigcount:
        bigword = word
        bigcount = count

print bigword, bigcount
```

A short Python “Story” about how to count words in a file

A word used to read data from a user

A sentence about updating one of the many counts

A paragraph about how to find the largest item in a list

# Summary

- This is a quick overview of Topic # 1
- We will revisit these concepts throughout the course
- After the break, lets focus on the big-picture