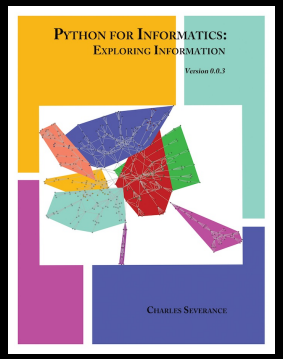
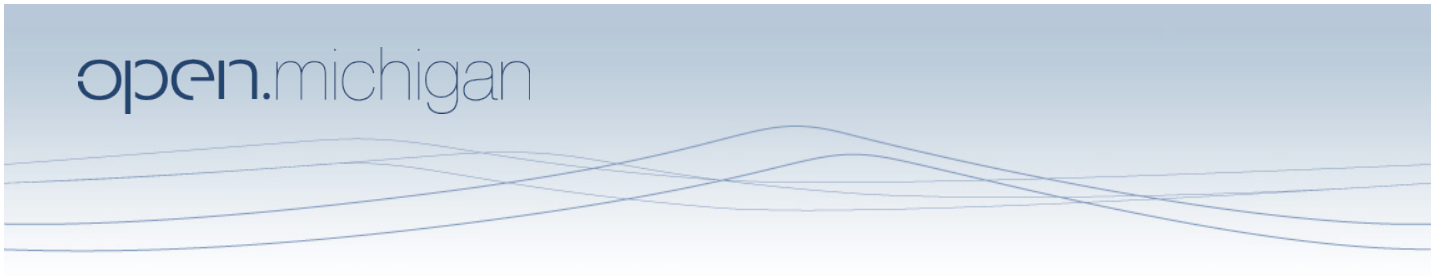
Why Program?

Chapter 1

Python for Informatics: Exploring Information
www.pythonlearn.com



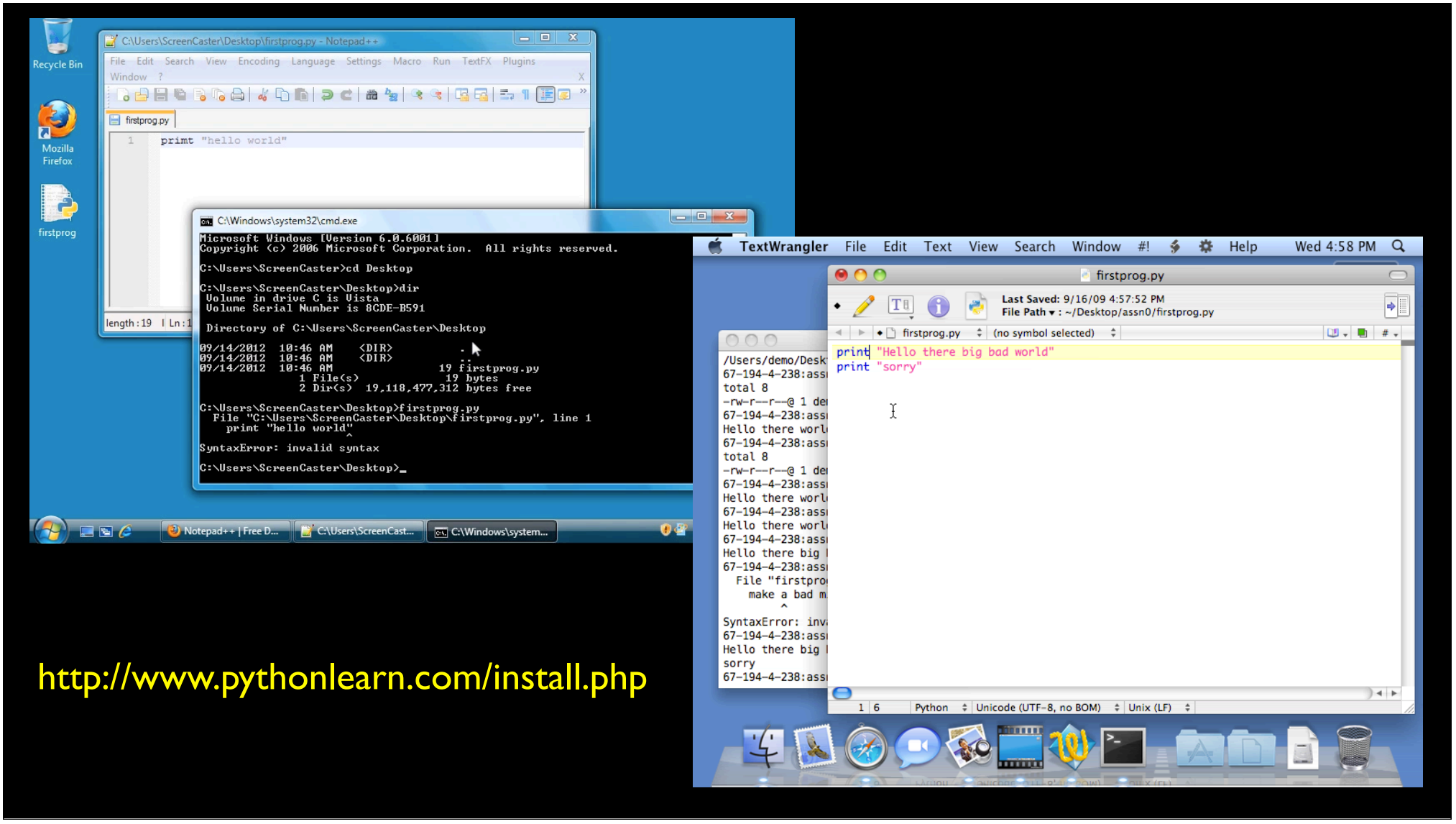


Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.
http://creativecommons.org/licenses/by/3.0/.

Copyright 2010- Charles Severance

Pre-Requisite: Please Install Python

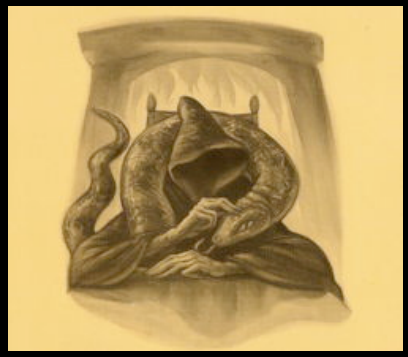

PythonLearn
Book Install Informatics Instructor Python About
Setting up your PythonLearn Devlopment Environment
We have separate pages for each of the commonly used Operating Systems:
Setting up the PythonLearn Environment in Microsoft Windows
Setting up the PythonLearn Environment on a Macintosh
Note: Make sure that you install the latest version of Python 2.x - do not install Python 3.x. There are signficant differences between Python 2 and Python 3 and this book is still Python 2.
You will need Quicktime (or iTunes) installed on your computer to view any video materials or screencasts. You should probably download the high quality copies of these files or screencasts to your computer and view/play them locally. They are rather large files and you will want to move back and forth as well as start and stop the podcasts so you can perform the steps as indicated.
http://www.pythonlearn.com/install.php



http://www.pythonlearn.com/install.php

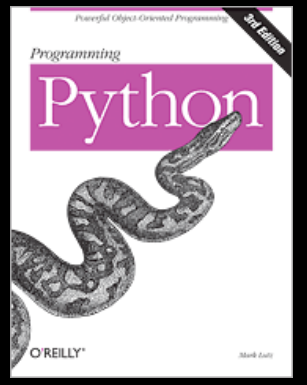
Python as a Language

Parseltongue is the language of serpents and those who can converse with them. An individual who can speak Parseltongue is known as a Parselmouth. It is a very uncommon skill, and may be hereditary. Nearly all known Parselmouths are descended from Salazar Slytherin.



http://harrypotter.wikia.com/wiki/Parseltongue

Python is the language of the Python Interpreter and those who can converse with it. An individual who can speak Python is known as a Pythonista. It is a very uncommon skill, and may be hereditary. Nearly all known Pythonistas use software inititially developed by Guido van Rossum.



Early Learner: Syntax Errors

- We need to learn the Python language so we can communicate our instructions to Python. In the beginning we will make lots of mistakes and speak gibberish like small children.
- When you make a mistake, the computer does not think you are "cute". It says "syntax error" - given that it *knows* the language and you are just learning it. It seems like Python is cruel and unfeeling.
- You must remember that *you* are intelligent and *can* learn - the computer is simple and very fast - but cannot learn - so it is easier for you to learn Python than for the computer to learn English...

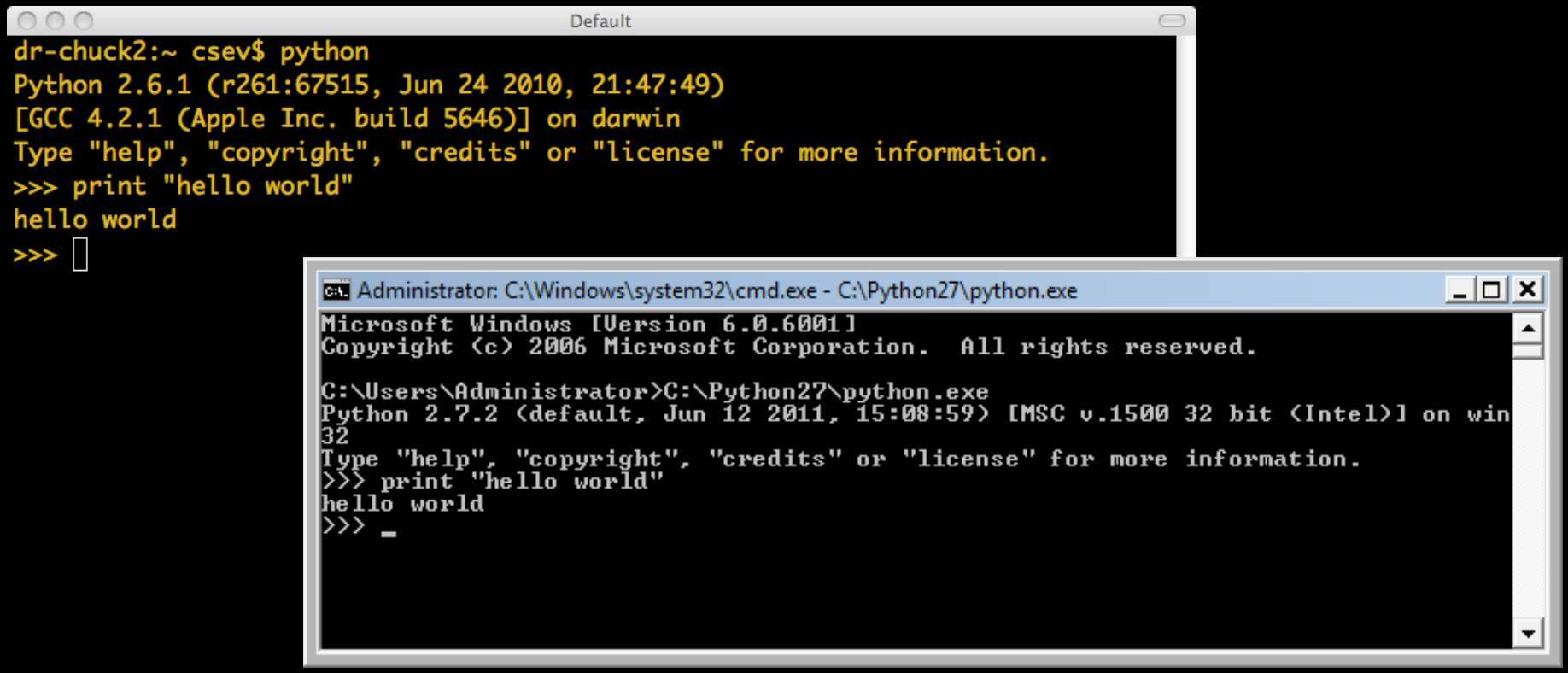
Talking to Python

```
csev$ python
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
csev$ python
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 1
>>> print x
1
>>> x = x + 1
>>> print x
2
>>> exit()
```

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

Lets Talk to Python...



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 can not 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

Sentences or Lines

```
x = 2          <-- Assignment Statement

x = x + 2      <-- Assignment with expression

print x        <-- Print statement
```

Variable Operator Constant Reserved Word

Programming Paragraphs

Python Scripts

- Interactive Python is good for experiments and programs of 3-4 lines long
- But 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 convention, we add ".py" as the suffix on the end of these files to indicate they contain Python

Writing a Simple Program

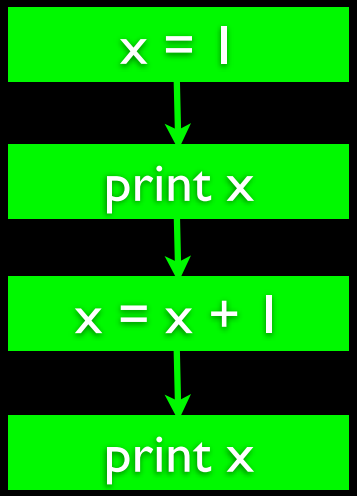
Interactive versus Script

- 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 execut 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



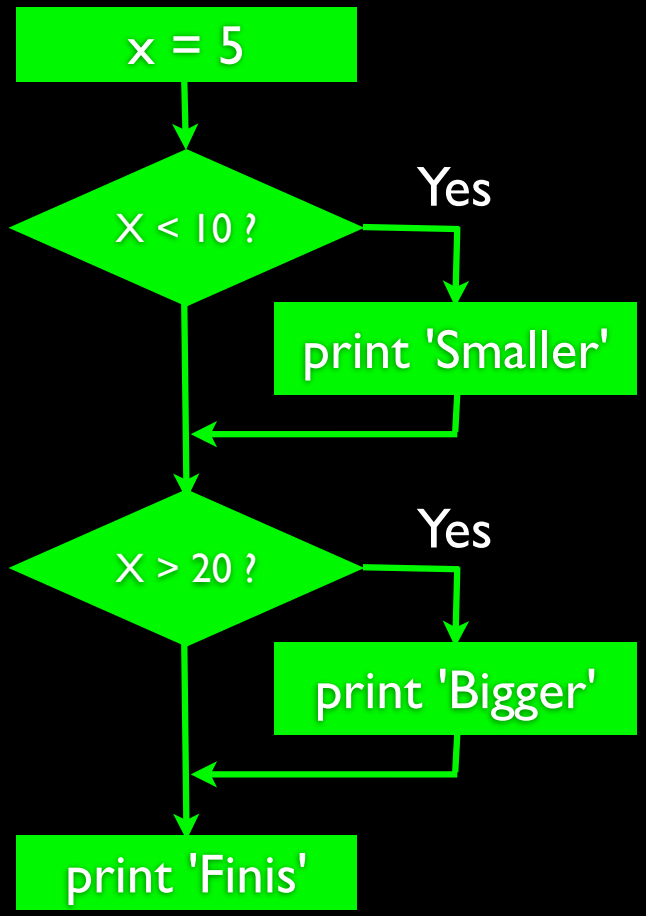
Program:

```
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.
We as programmers set up "paths" for the program to follow.



Conditional Steps

Program:

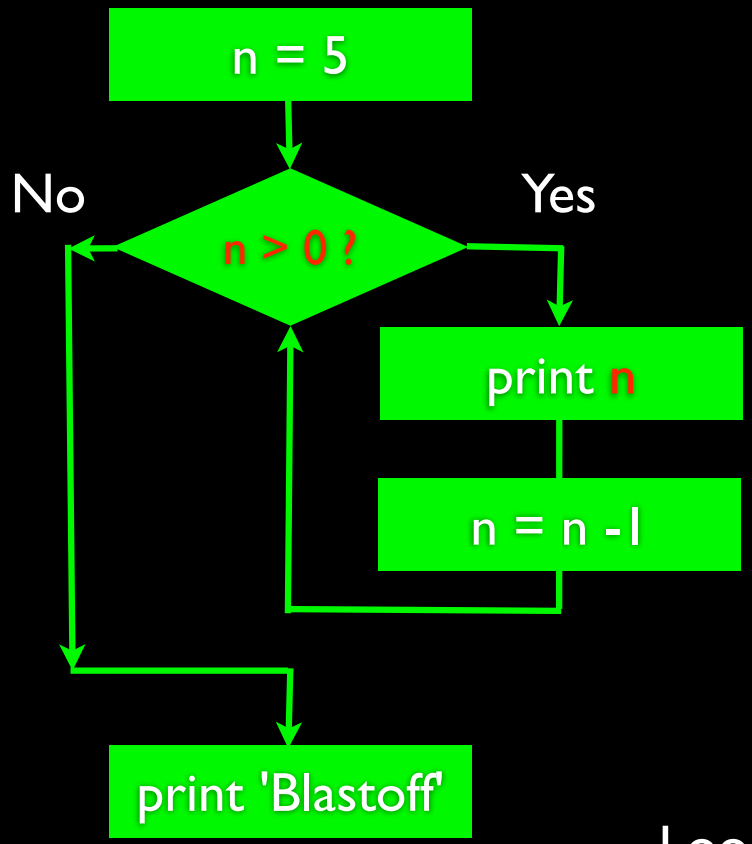
```
x = 5
if x < 10:
    print 'Smaller'

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
```

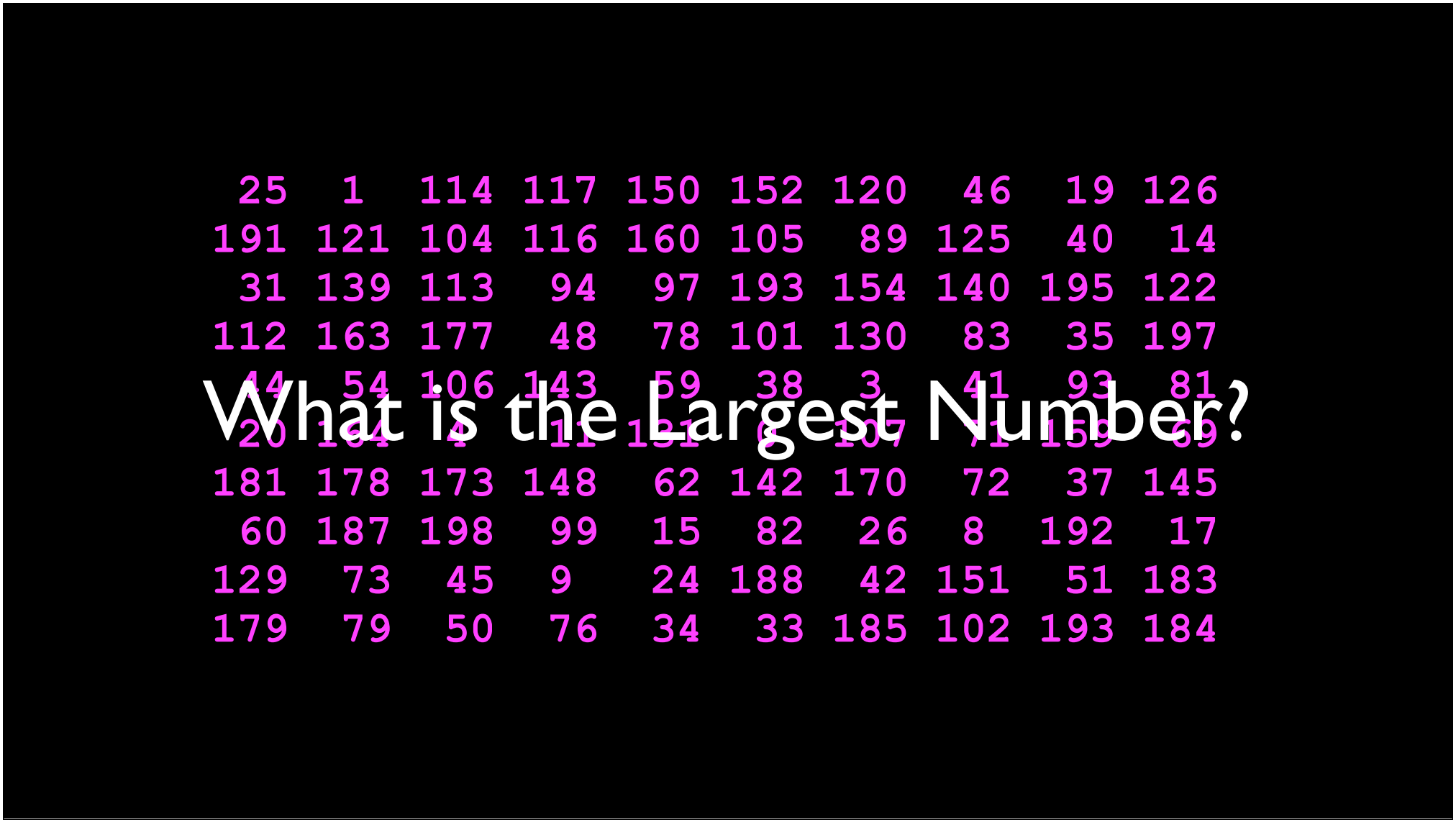
Sequential

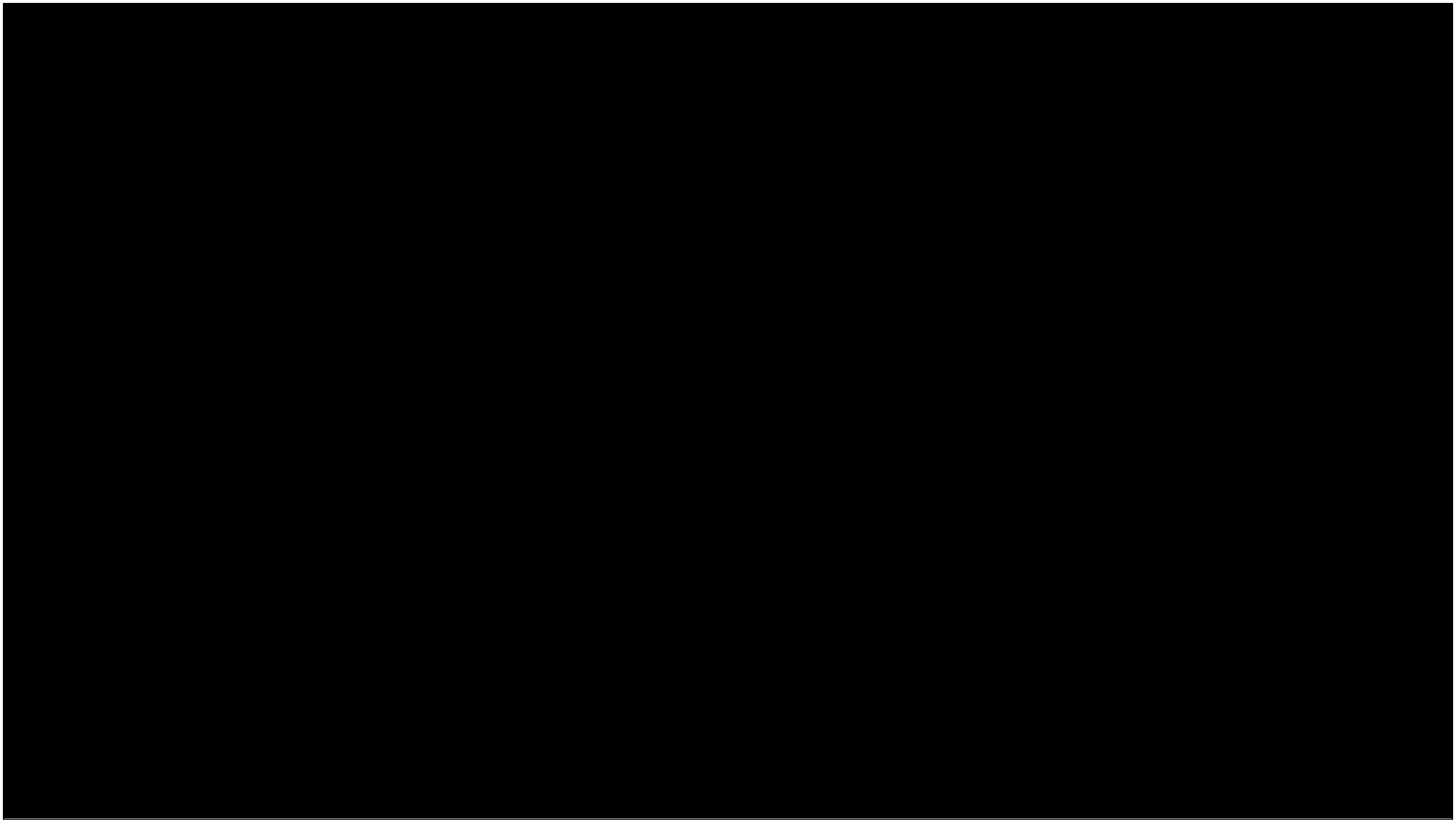
Repeated

Conditional

An Animated Short Python Story...

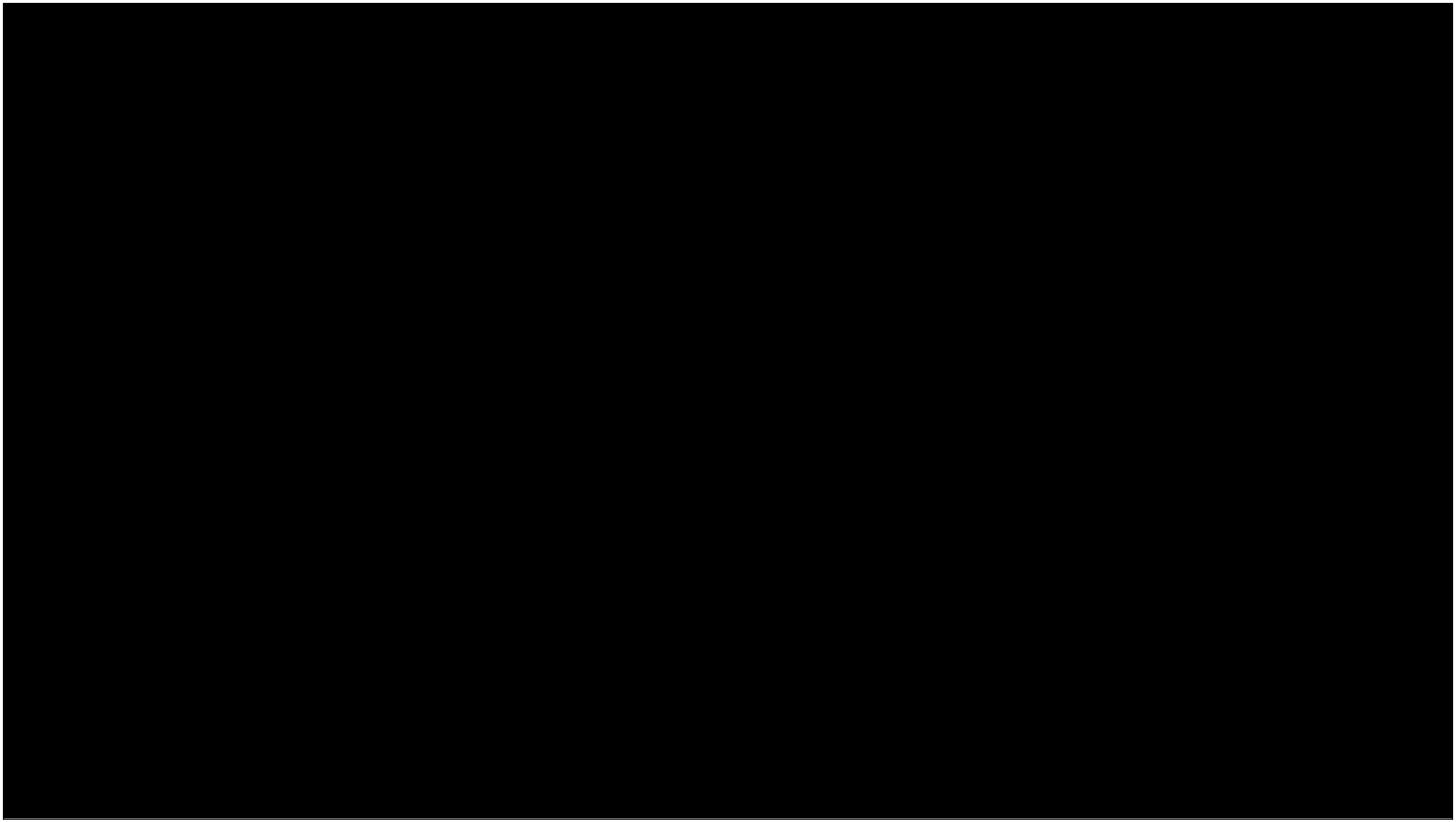
Finding the largest number in a list of numbers...


25 1 114 117 150 152 120 46 19 126
191 121 104 116 160 105 89 125 40 14
31 139 113 94 97 193 154 140 195 122
112 163 177 48 78 101 130 83 35 197
44 54 106 143 59 38 3 41 93 81
181 178 173 148 62 142 170 72 37 145
60 187 198 99 15 82 26 8 192 17
129 73 45 9 24 188 42 151 51 183
179 79 50 76 34 33 185 102 193 184
What is the Largest Number?

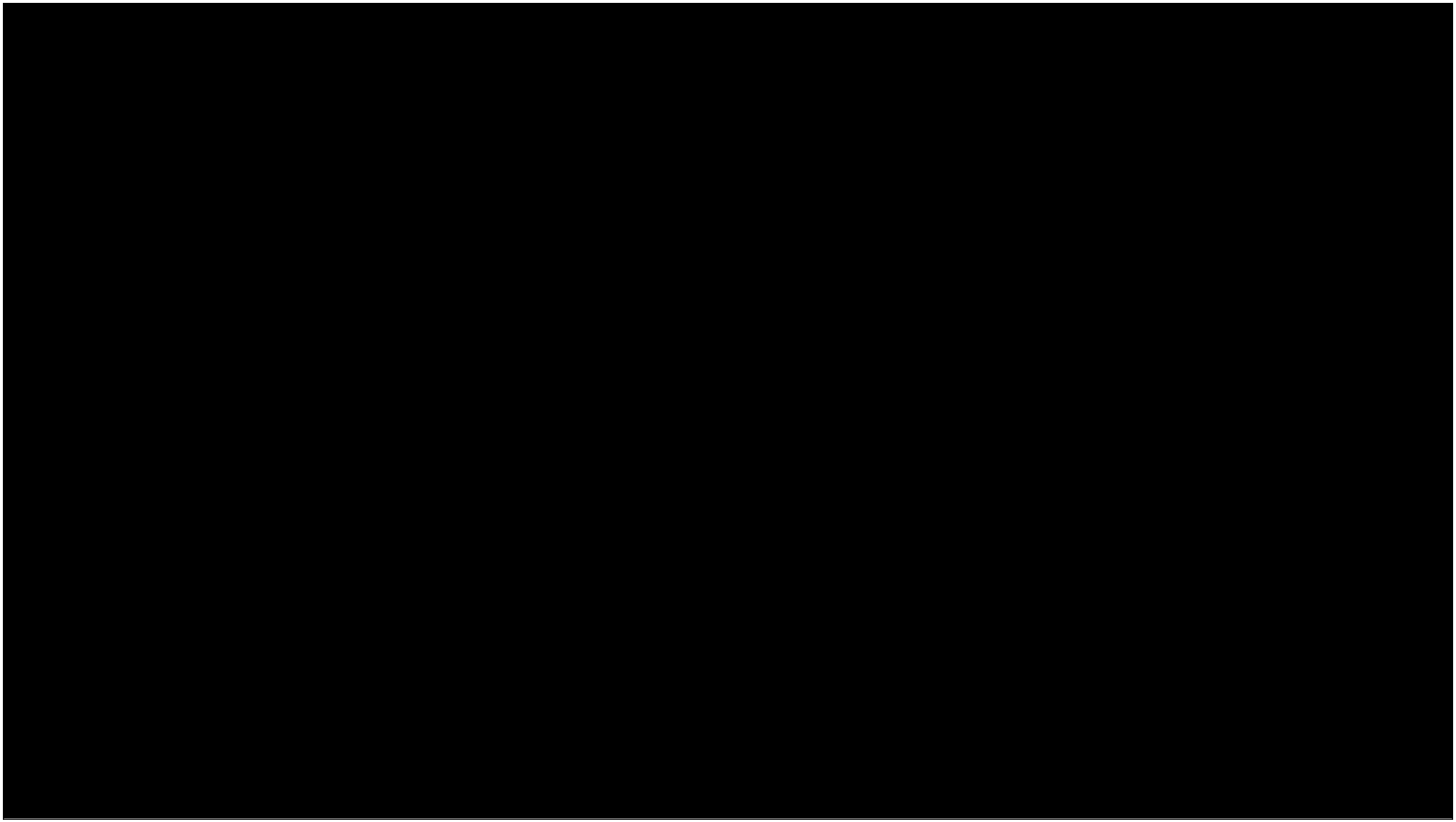


```
 25   1  114 117 150 152 120  46  19 126
191 121 104 116 160 105  89 125  40  14
 31 139 113  94  97 193 154 140 195 122
112 163 177  48  78 101 130  83  35 197
 44  54 106 143  59  38   3  41  93  81
 20 164   4  11 131   0 107  71 159  69
181 178 173 148  62 142 170  72  37 145
 60 187 198  99  15  82  26   8 192  17
129  73  45   9  24 188  42 151  51 183
179  79  50  76  34  33 185 102 193 184
```

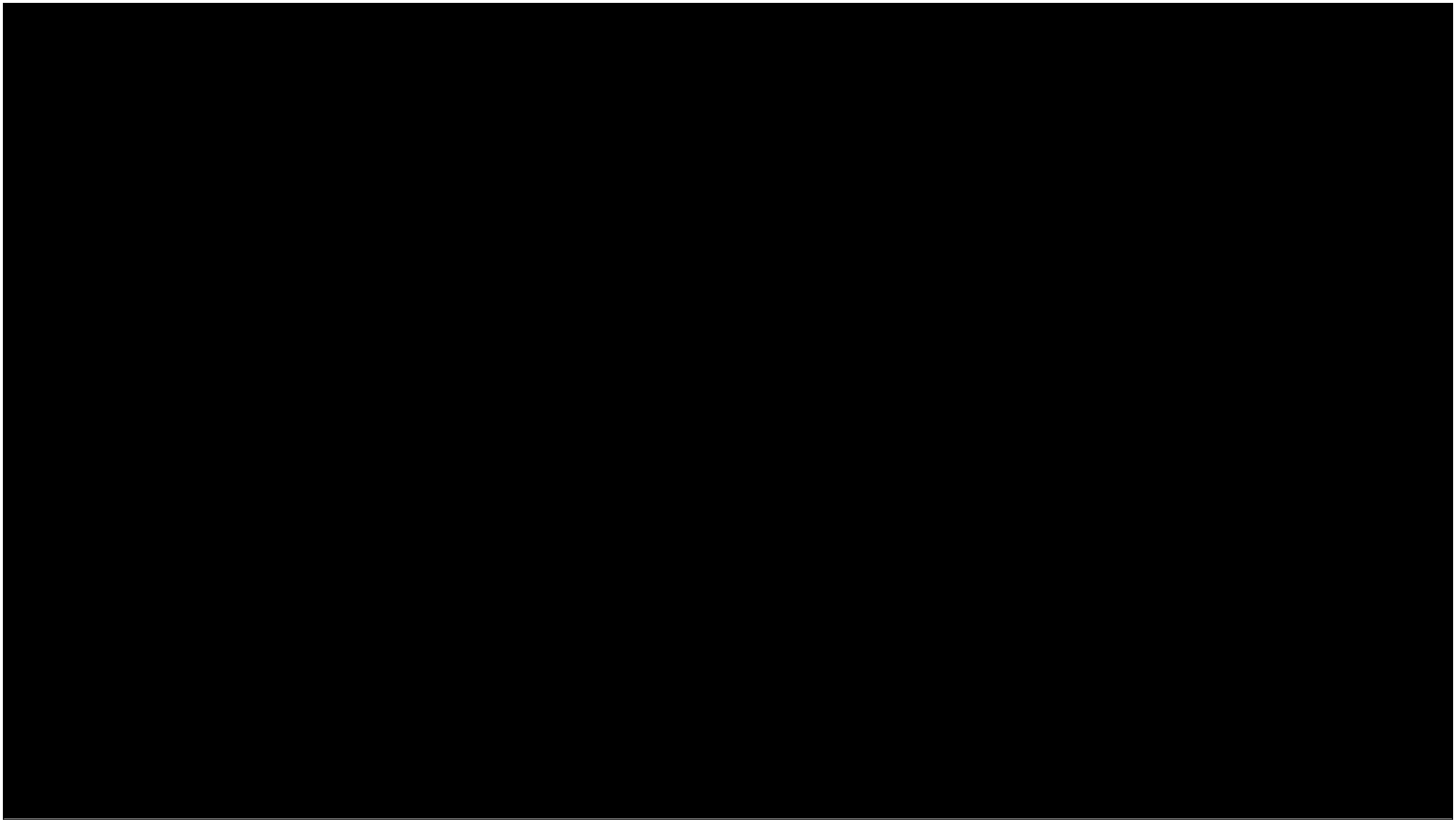
What is the Largest Number?



126	19	46	120	152	150	117	114	1	25
14	40	125	89	005	160	116	104	121	**197**
122	195	140	154	149	97	94	113	139	31
197	35	83	130	101	78	48	177	163	112
81	93	41	3	38	59	143	106	54	44
69	159	71	107	0	131	11	4	164	20
145	37	72	170	142	62	148	173	178	181
17	092	8	26	82	15	99	169	187	60
183	51	151	42	188	24	9	45	73	129
184	193	102	185	33	34	76	50	79	179



What is the Largest Number?



What is the Largest Number?

largest_so_far

-13 41 74

```
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.

A word used to read data from a user.

A sentence about updating one of many counts.

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

Summary

- This is a quick overview of Chapter 1
- We will revisit these concepts throughout the course
- Focus on the big picture