python™

A ROADMAP FOR AGRICULTURE DATA SCIENCE (PART 1/8)

NO COMPUTER SCIENCE OR MATH BACKGROUND IS REQUIRED

PART 1 - CONTENTS:

- How Computers Work?
- Why we Program?
- Installing and Using Python
- Variables and Expressions
- Conditional Code
- Functions
- Loops and Iteration

Duration: 2 Months

Classes :
Saturday & Sunday

Time:
9.00 – 10.00 AM

Requirements :
Laptop/Mobile Phone + Internet

FEE – COMPLETELY FREE

Starting Date: 17 July 2021

Platform: Zoom/Microsoft Teams

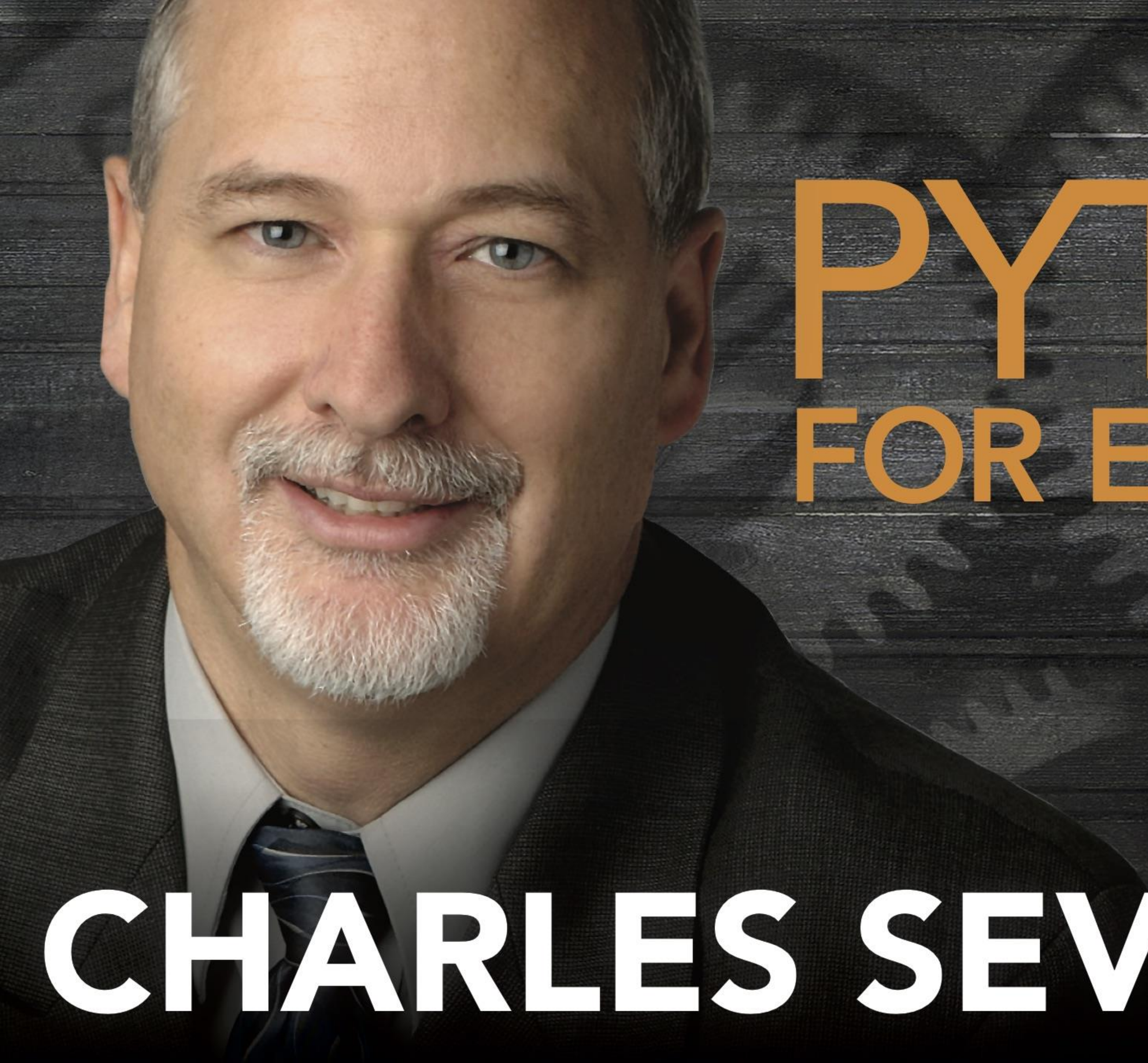
INSTRUCTOR
Dr. Saqib Ali, Department of Computer Science, UAF

CONTACT: 03001750077

Registration: https://forms.gle/YMZ433MqYn5PpAdC6

WhatsApp Group Link: https://chat.whatsapp.com/ICKD6jxqILVJ1P7KxUpCcF

QR Code



PYTHON FOR EVERYBODY

CHARLES SEVERANCE

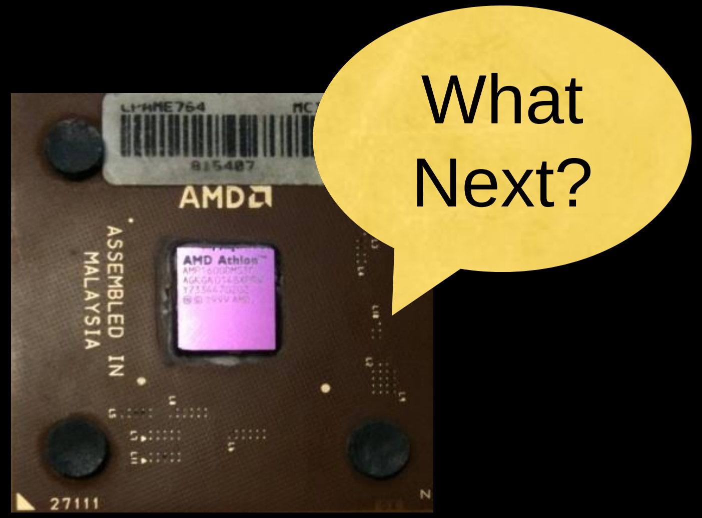
Why Program?

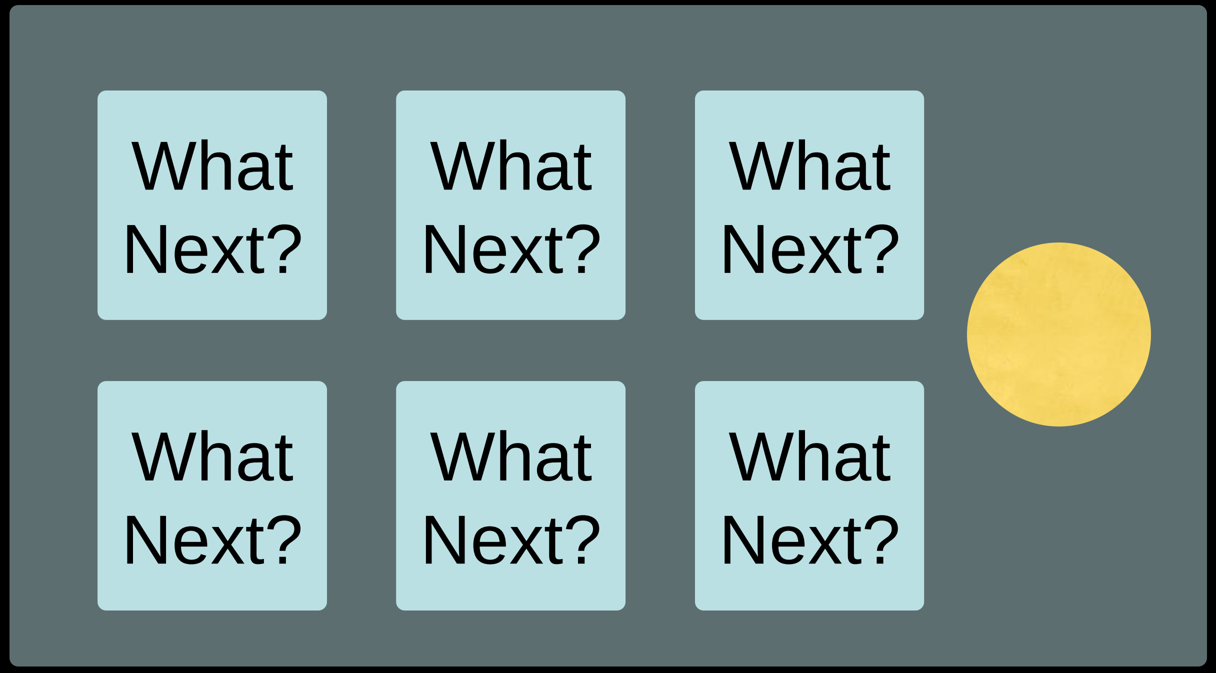
Chapter 1

Python for Everybody
www.py4e.com

Computers Want to be Helpful...

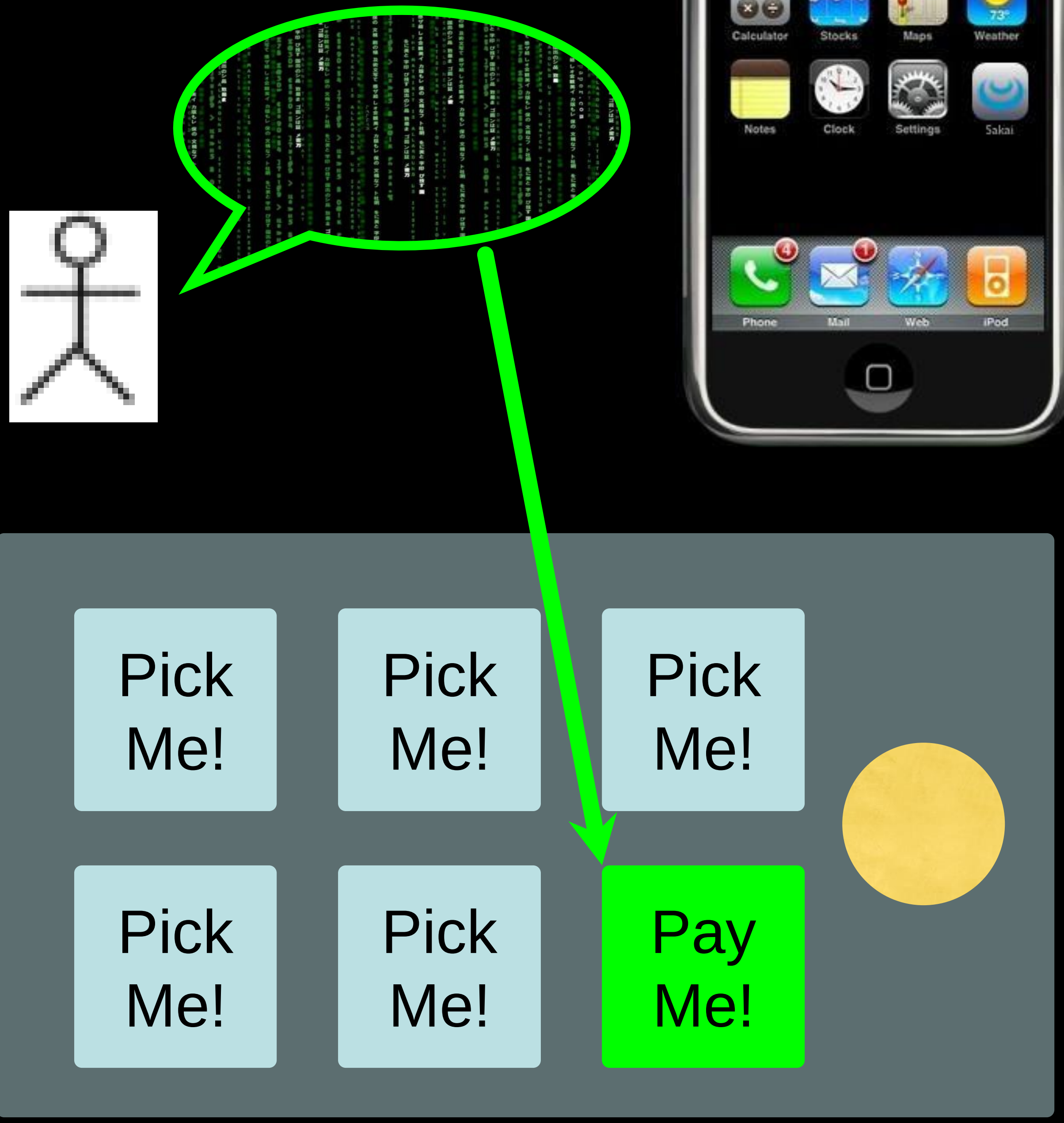
- Computers are built for one purpose - to do things for us
- But we need to speak their language to describe what we want done
- Users have it easy - someone already put many different programs (instructions) into the computer and users just pick the ones they want to use





Programmers Anticipate Needs

- iPhone applications are a market
- iPhone applications have over 3 billion downloads
- Programmers have left their jobs to be full-time iPhone developers
- Programmers know the ways of the program

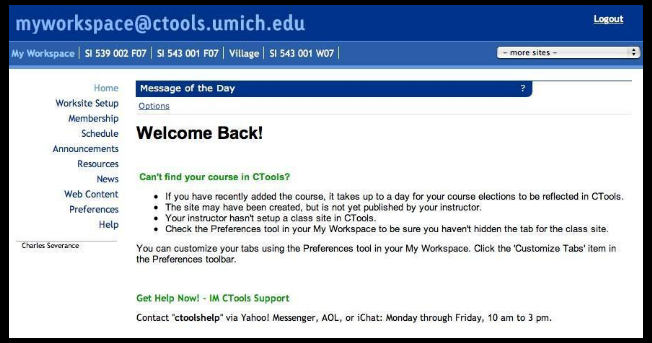


Users vs. Programmers

- Users see computers as a set of tools - word processor, spreadsheet, map, to-do list, etc.
- Programmers learn the computer “ways” and the computer language
- Programmers have some tools that allow them to build new tools
- Programmers sometimes write tools for lots of users and sometimes programmers write little “helpers” for themselves to automate a task

Why be a Programmer?

- To get some task done - we are the user and programmer
  - Clean up survey data
- To produce something for others to use - a programming job
  - Fix a performance problem in the Sakai software
  - Add a guestbook to a web site



myworkspace@ctools.umich.edu Logout

My Workspace | SI 539 002 F07 | SI 543 001 F07 | Village | SI 543 001 W07 | - more sites -

Home
Worksite Setup
Membership
Schedule
Announcements
Resources
News
Web Content
Preferences
Help

Charles Severance

Message of the Day ?

Options

Welcome Back!

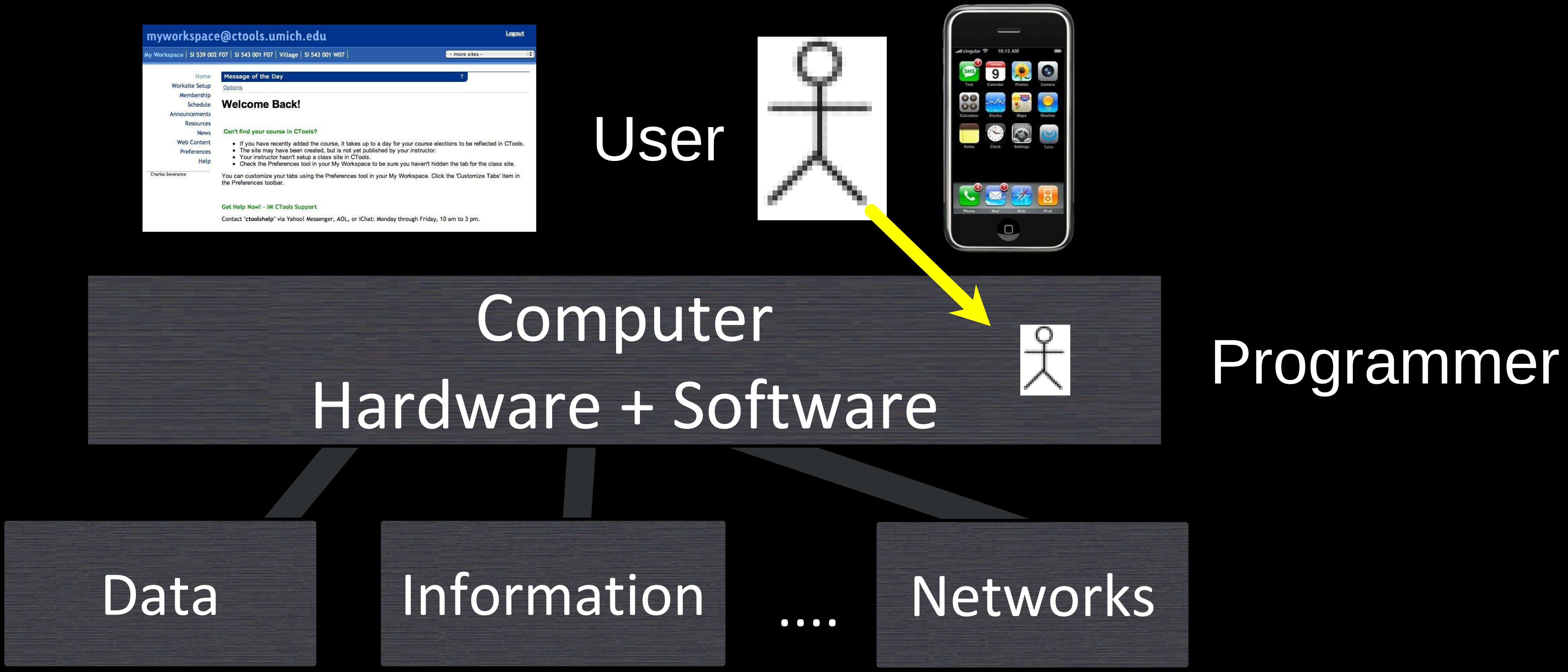
Can't find your course in CTools?

- If you have recently added the course, it takes up to a day for your course elections to be reflected in CTools.
- The site may have been created, but is not yet published by your instructor.
- Your instructor hasn't setup a class site in CTools.
- Check the Preferences tool in your My Workspace to be sure you haven't hidden the tab for the class site.

You can customize your tabs using the Preferences tool in your My Workspace. Click the 'Customize Tabs' item in the Preferences toolbar.

Get Help Now! - IM CTools Support

Contact "**ctoolshelp**" via Yahoo! Messenger, AOL, or iChat: Monday through Friday, 10 am to 3 pm.



From a software creator’s point of view, we build the software. The end users (stakeholders/actors) are our masters - who we want to please - often they pay us money when they are pleased. But the data, information, and networks are our problem to solve on their behalf. The hardware and software are our friends and allies in this quest.

What is Code? Software? A Program?

- A sequence of stored instructions

  - It is a little piece of our intelligence in the computer

  - We figure something out and then we encode it and then give it to someone else to save them the time and energy of figuring it out

- A piece of creative art - particularly when we do a good job on user experience

```python
name = input('Enter file:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    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)
```

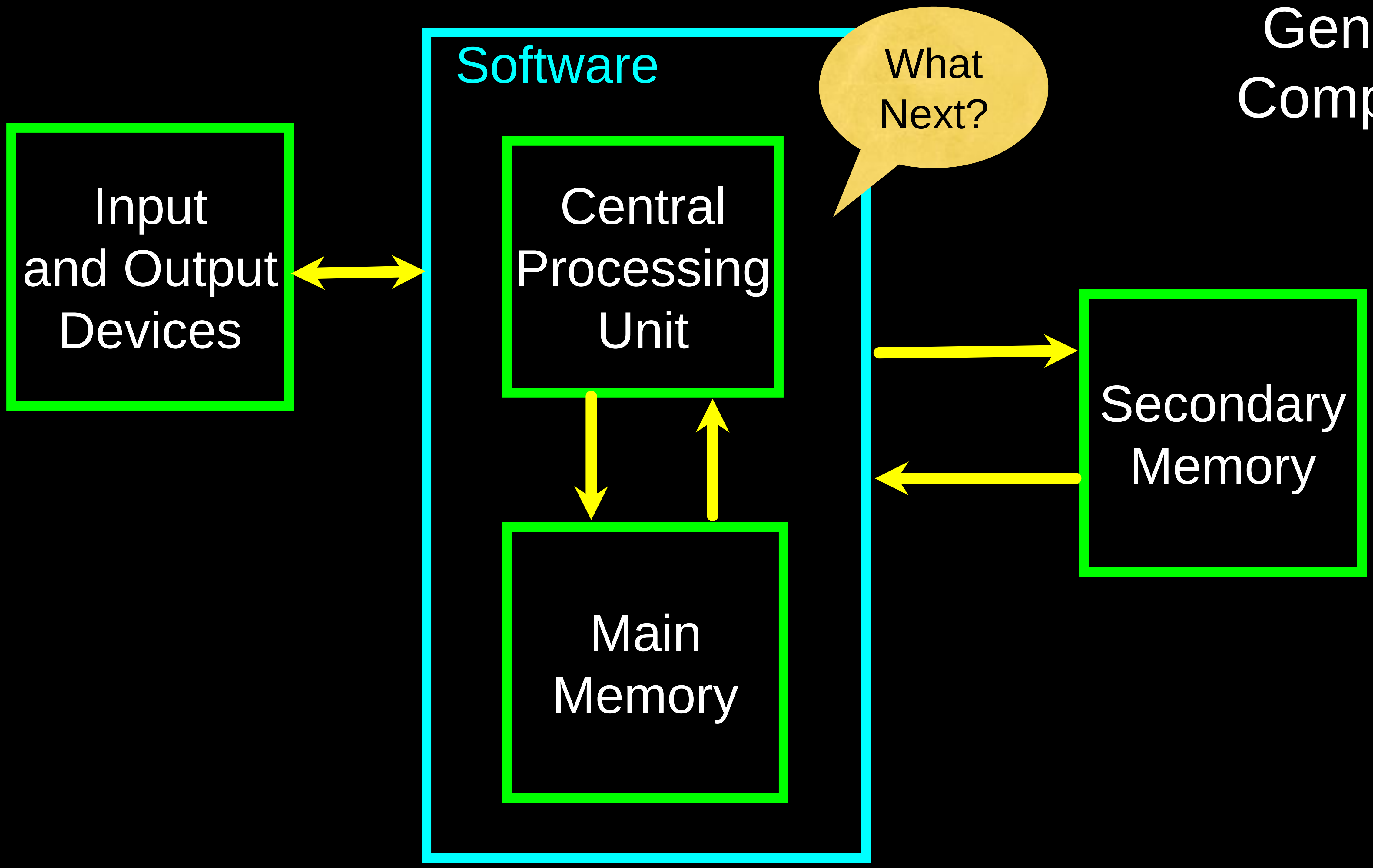
```
python words.py
Enter file: words.txt
to 16
```

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

Hardware Architecture

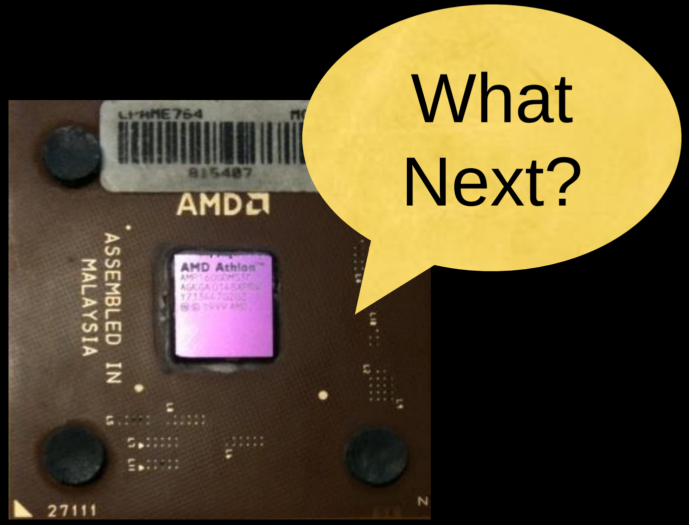
http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg

Generic Computer



Definitions

- Central Processing Unit: Runs the Program - The CPU is always wondering "what to do next". Not the brains exactly - very dumb but very very fast



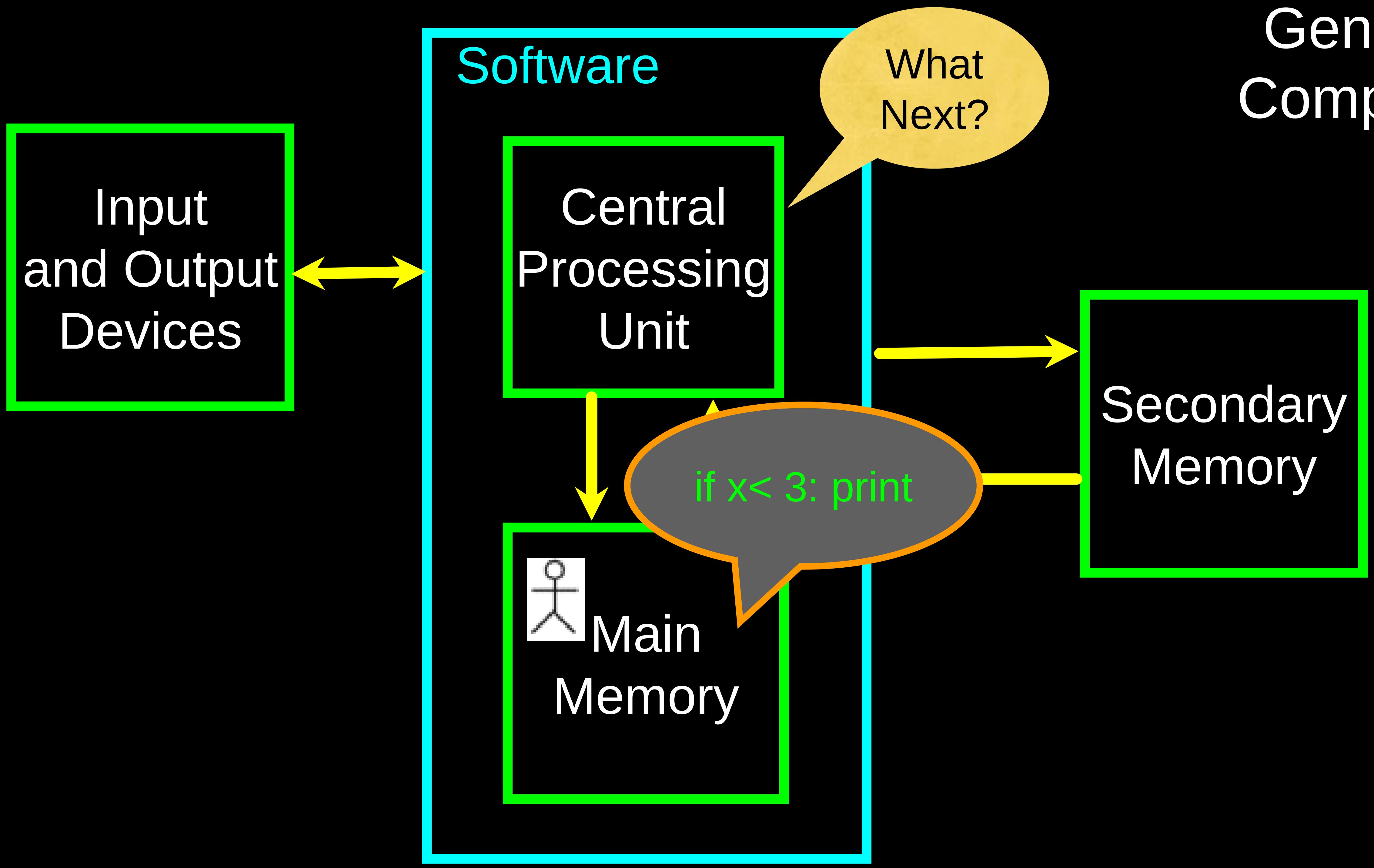
- Input Devices: Keyboard, Mouse, Touch Screen

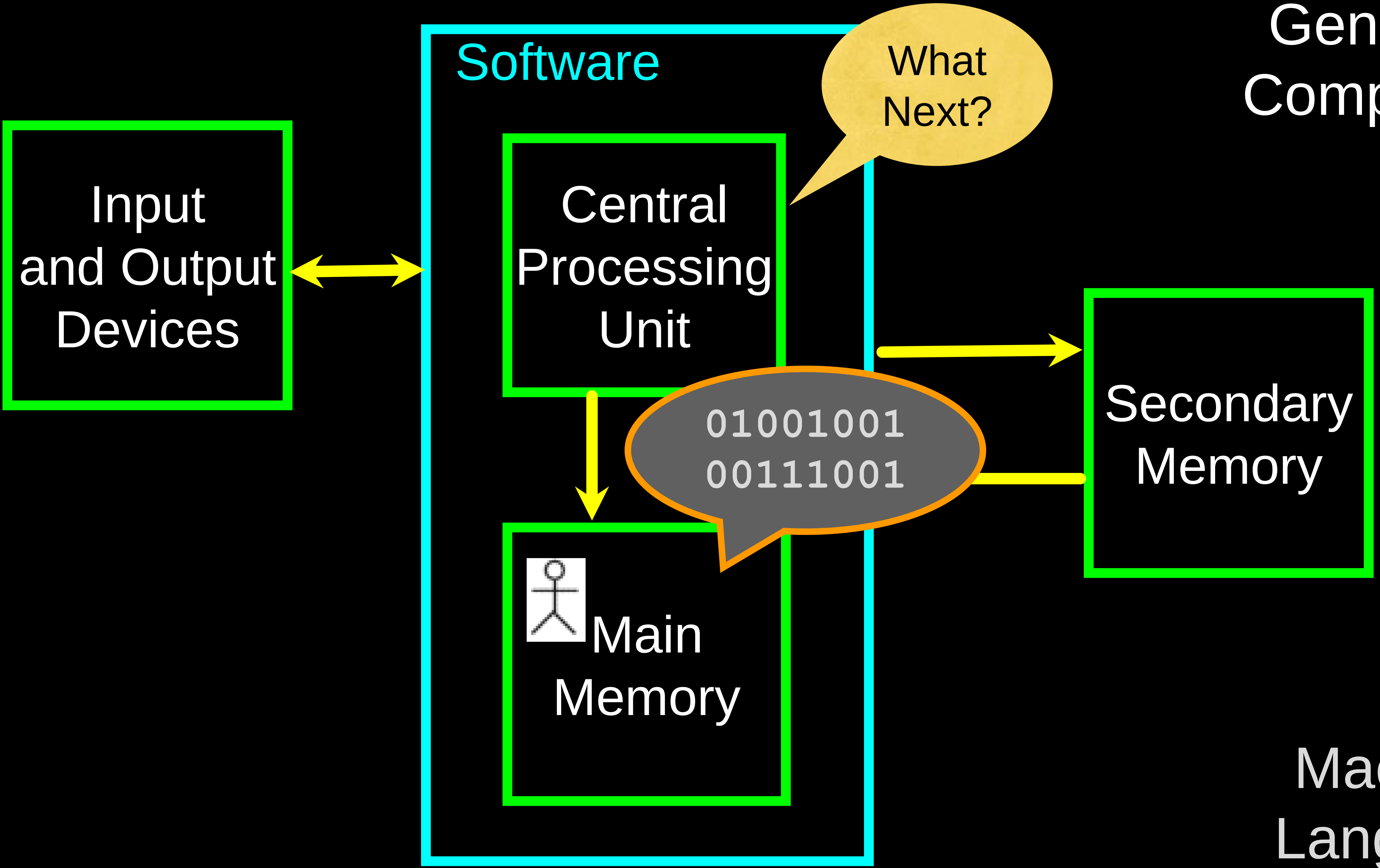
- Output Devices: Screen, Speakers, Printer, DVD Burner

- Main Memory: Fast small temporary storage - lost on reboot - aka RAM

- Secondary Memory: Slower large permanent storage - lasts until deleted - disk drive / memory stick

Generic Computer

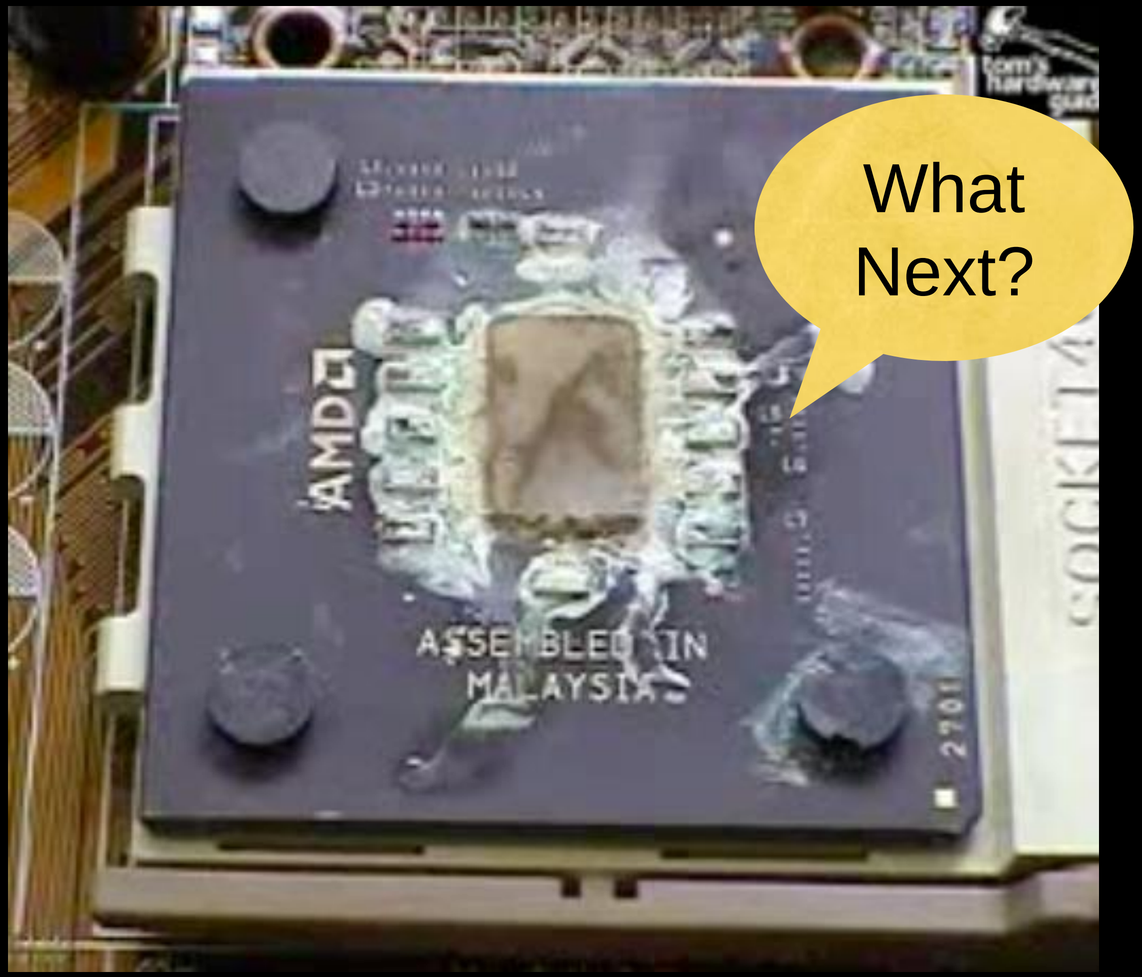




Generic Computer

Machine Language

Totally Hot CPU

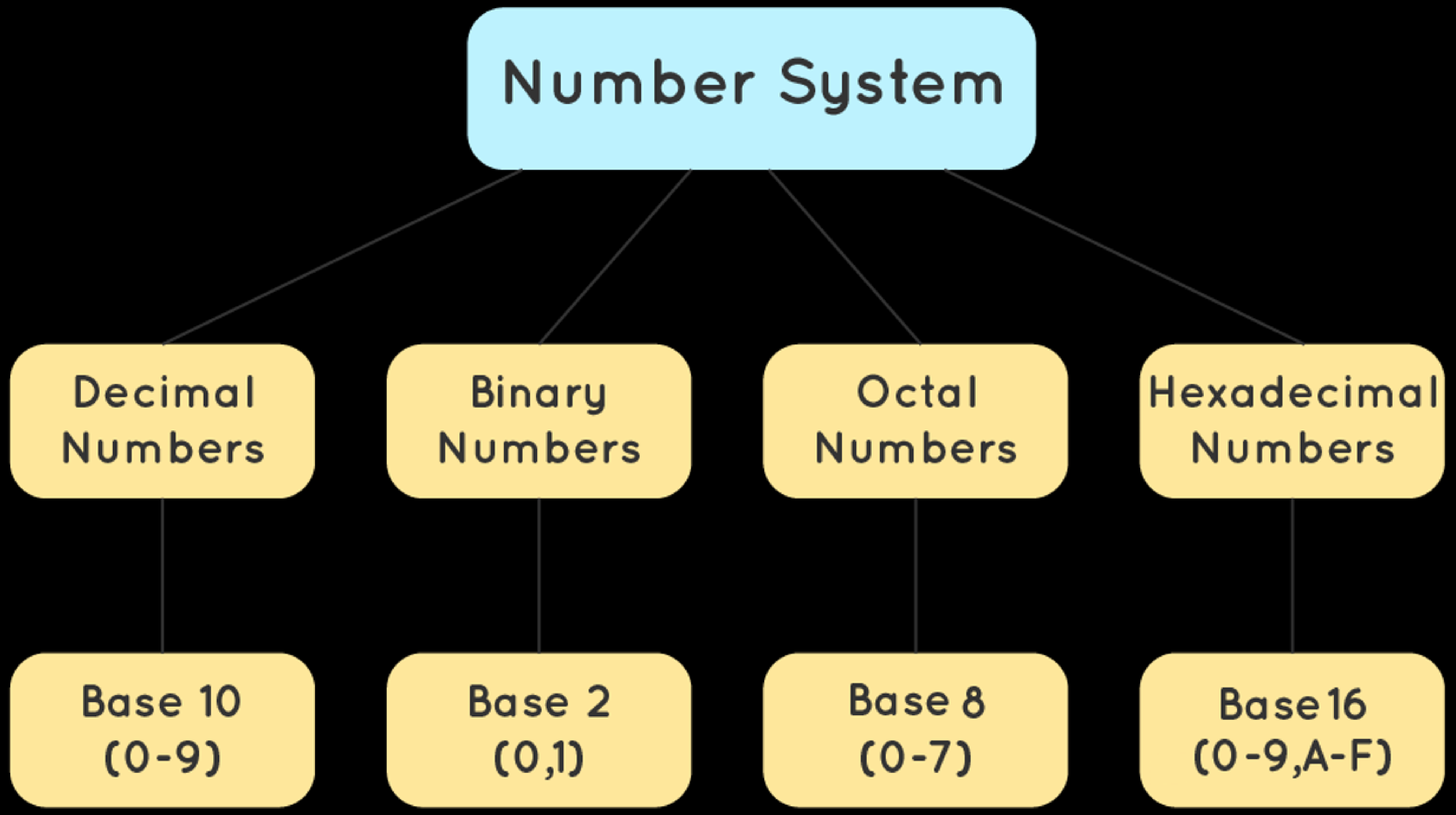


http://www.youtube.com/watch?v=y39D4529FM4

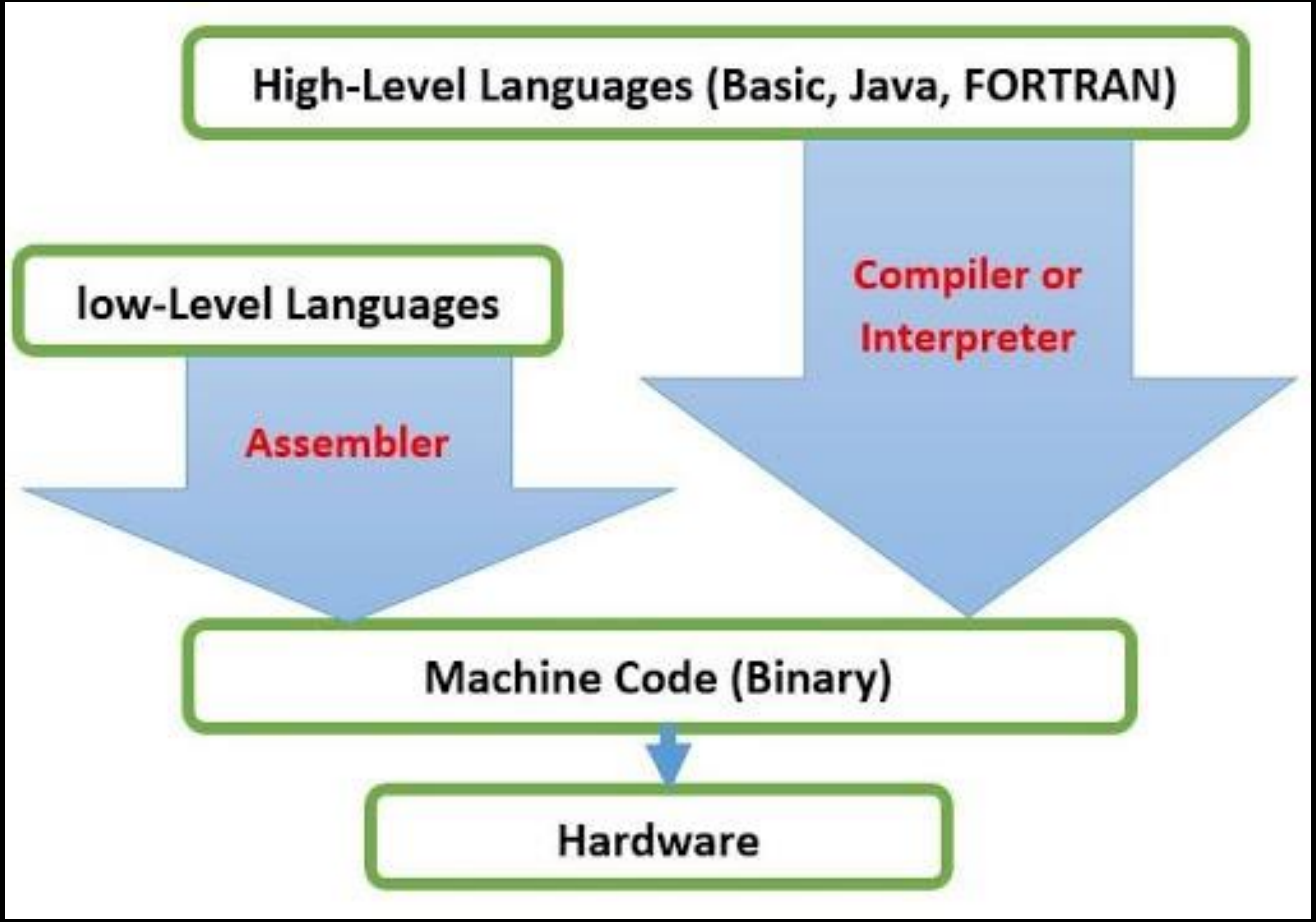
Hard Disk in Action

http://www.youtube.com/watch?v=9eMWG3fwiEU

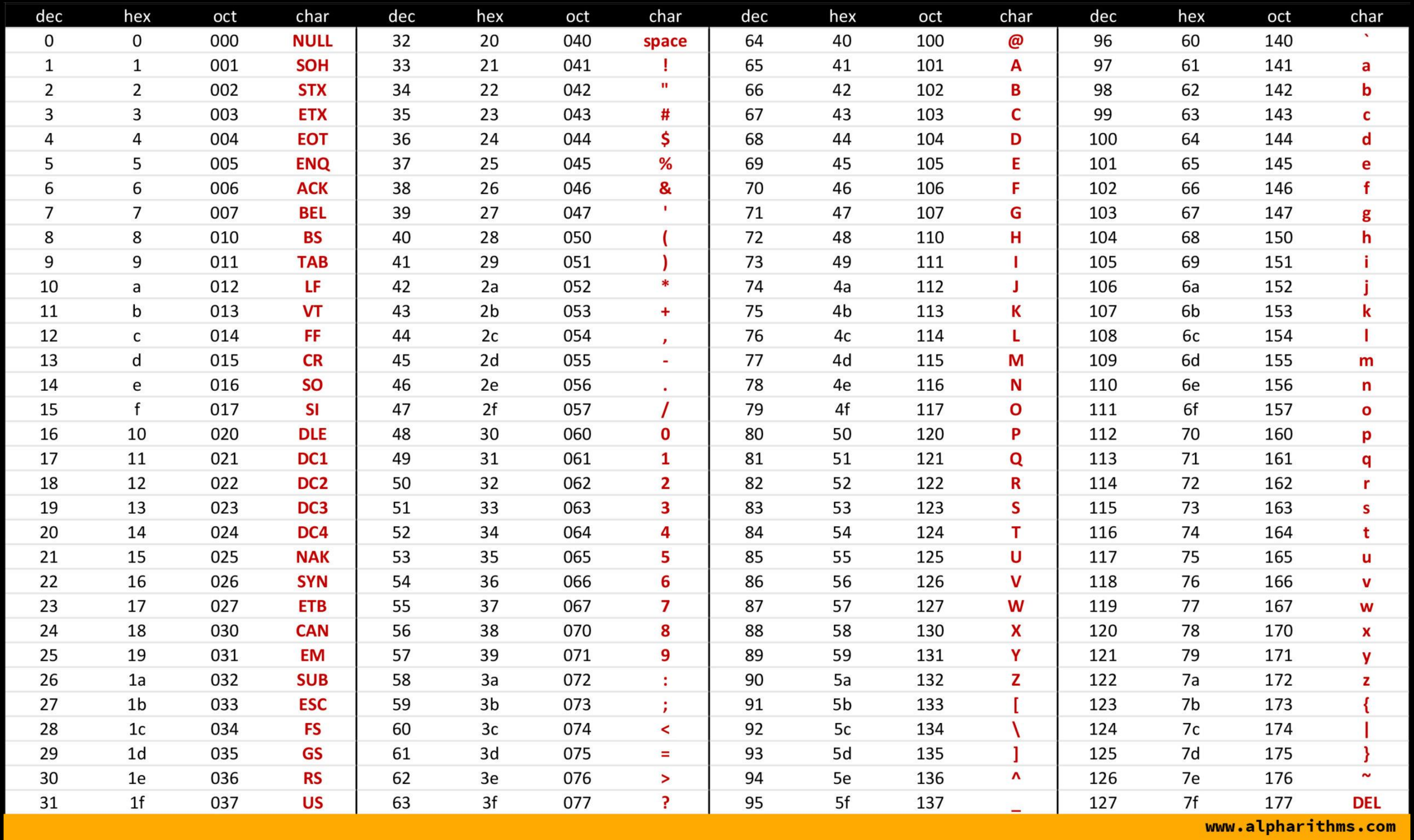
Number System



Computer – Native Language



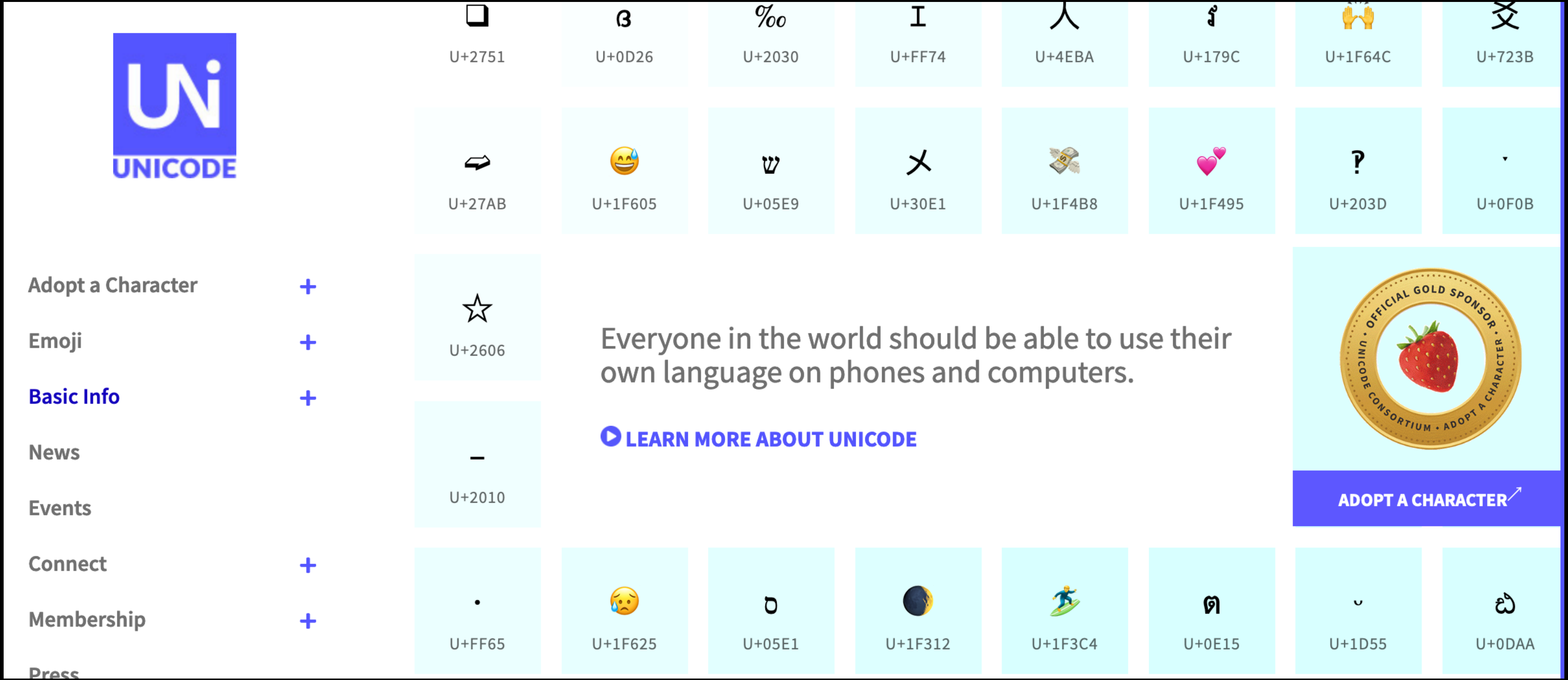
ASCII



dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	|
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

www.alpharithms.com

UNICODE


UNICODE

Adopt a Character +

Emoji +

Basic Info +

News

Events

Connect +

Membership +

U+2751 U+0D26 U+2030 U+FF74 U+4EBA U+179C U+1F64C U+723B

U+27AB U+1F605 U+05E9 U+30E1 U+1F4B8 U+1F495 U+203D U+0F0B

U+2606

U+2010

Everyone in the world should be able to use their own language on phones and computers.

LEARN MORE ABOUT UNICODE

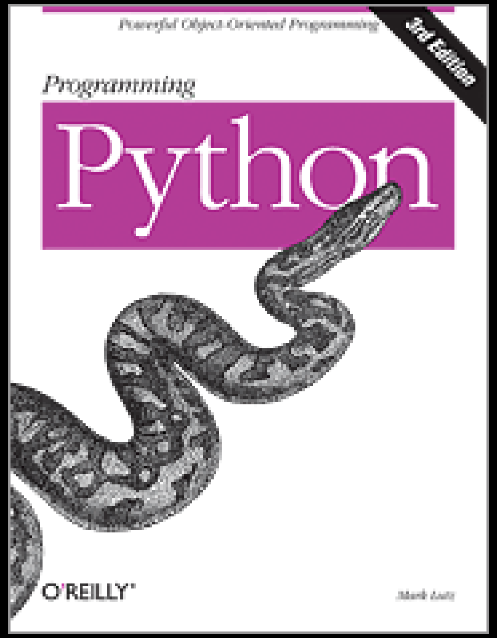
OFFICIAL GOLD SPONSOR • ADOPT A CHARACTER • UNICODE CONSORTIUM

ADOPT A CHARACTER

U+FF65 U+1F625 U+05E1 U+1F312 U+1F3C4 U+0E15 U+1D55 U+0DAA

Python as a Language

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 initially 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.

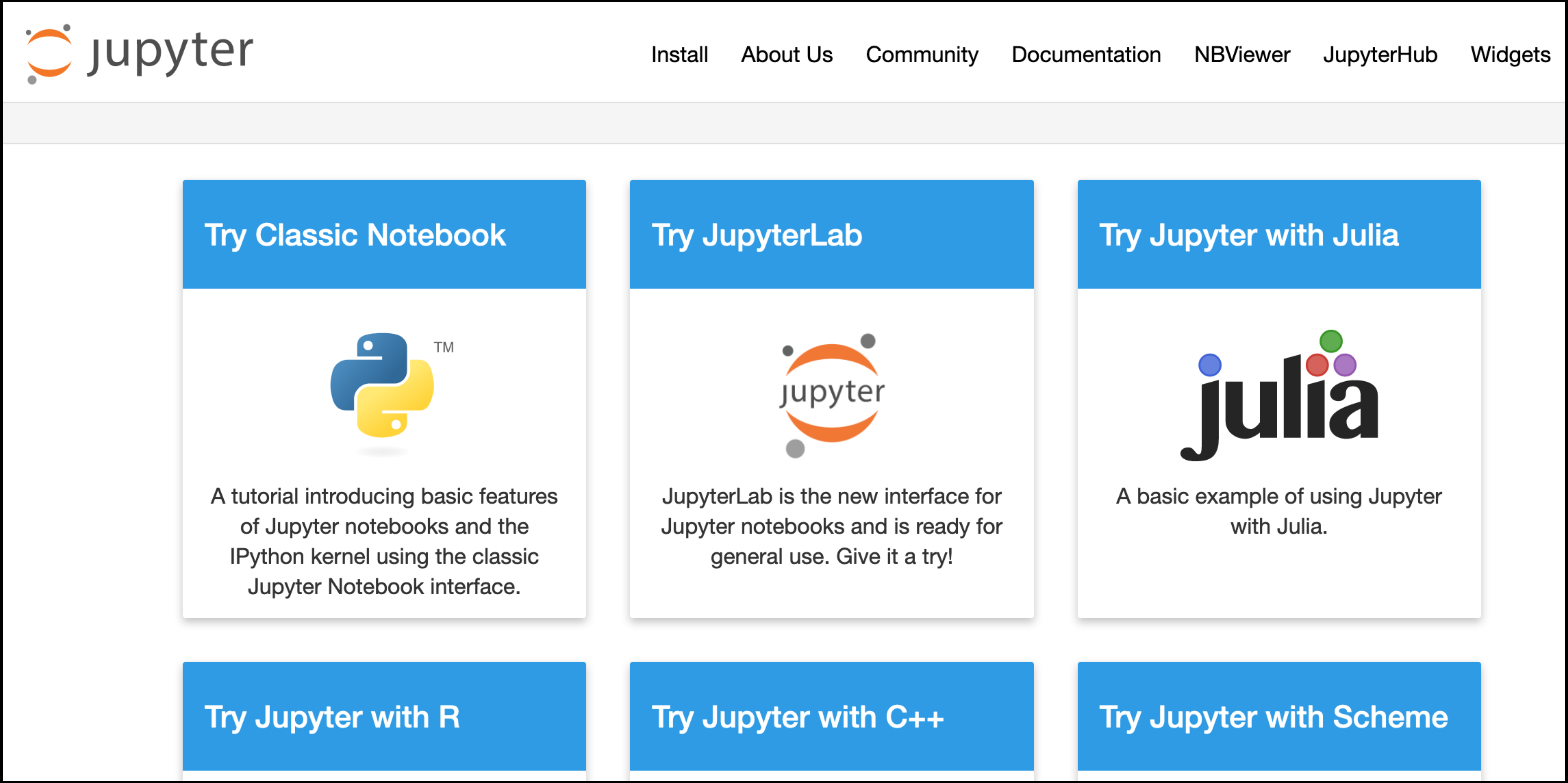
- When you make a mistake, the computer does not think you are “cute”. It says “syntax error” - given that it knows the language and you are just learning it. It seems like Python is cruel and unfeeling.

- 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

1. Jupyter – Browser Only

https://jupyter.org/try



jupyter

Install About Us Community Documentation NBViewer JupyterHub Widgets

Try Classic Notebook

A tutorial introducing basic features of Jupyter notebooks and the IPython kernel using the classic Jupyter Notebook interface.

Try JupyterLab

JupyterLab is the new interface for Jupyter notebooks and is ready for general use. Give it a try!

Try Jupyter with Julia

A basic example of using Jupyter with Julia.

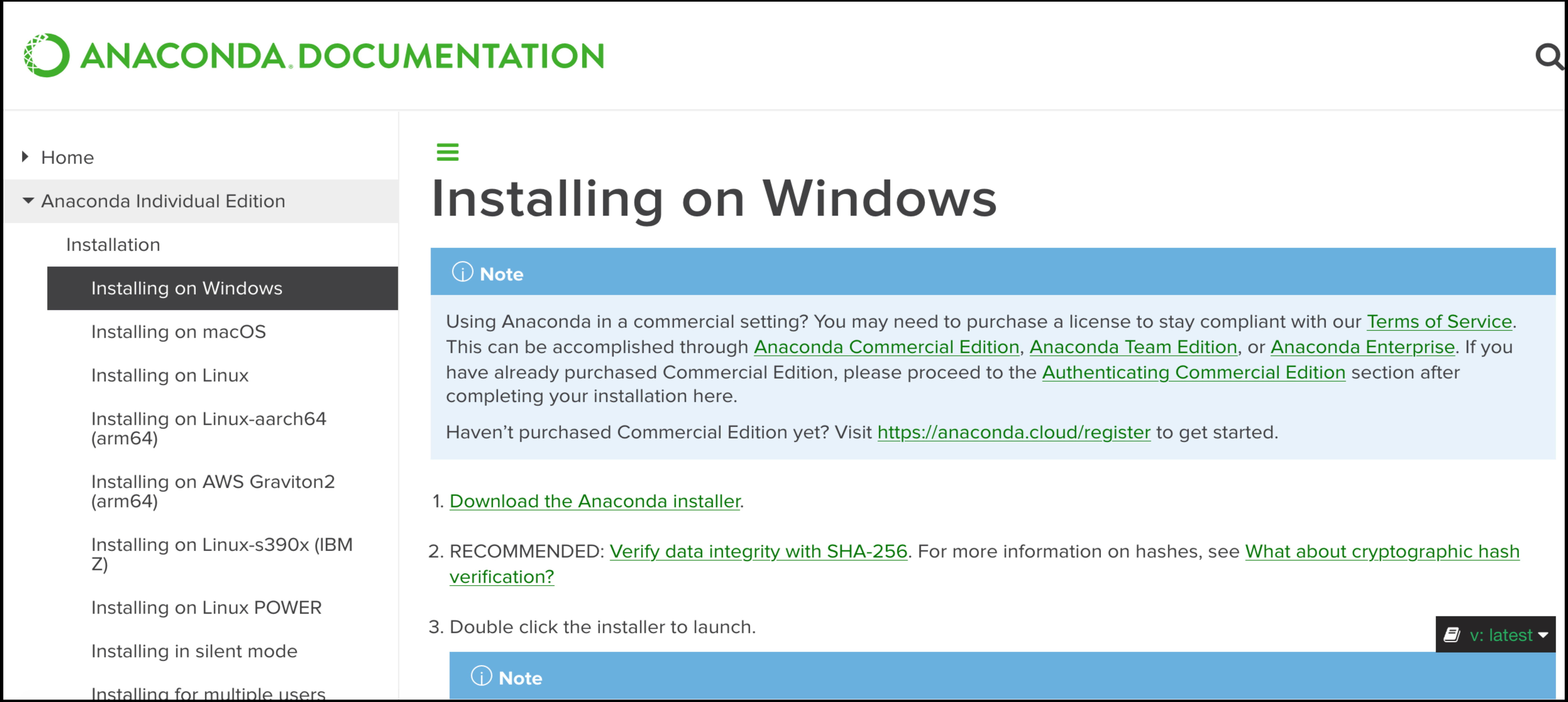
Try Jupyter with R

Try Jupyter with C++

Try Jupyter with Scheme

2. Jupyter – Installation

https://docs.anaconda.com/anaconda/install/windows/



ANACONDA. DOCUMENTATION

- Home
- Anaconda Individual Edition
  - Installation
    - Installing on Windows
    - Installing on macOS
    - Installing on Linux
    - Installing on Linux-aarch64 (arm64)
    - Installing on AWS Graviton2 (arm64)
    - Installing on Linux-s390x (IBM Z)
    - Installing on Linux POWER
    - Installing in silent mode
    - Installing for multiple users

Installing on Windows

Note

Using Anaconda in a commercial setting? You may need to purchase a license to stay compliant with our Terms of Service. This can be accomplished through Anaconda Commercial Edition, Anaconda Team Edition, or Anaconda Enterprise. If you have already purchased Commercial Edition, please proceed to the Authenticating Commercial Edition section after completing your installation here.

Haven't purchased Commercial Edition yet? Visit https://anaconda.cloud/register to get started.

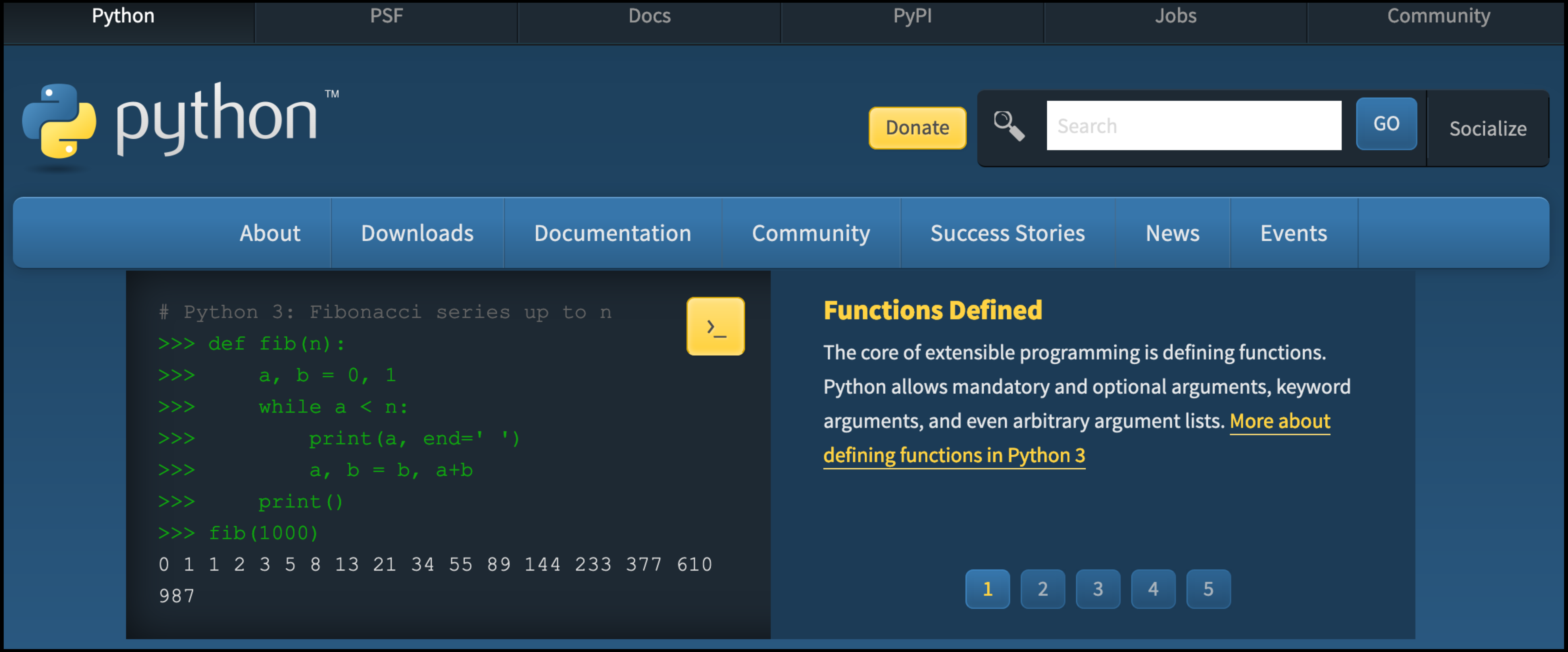
1. Download the Anaconda installer.
2. RECOMMENDED: Verify data integrity with SHA-256. For more information on hashes, see What about cryptographic hash verification?
3. Double click the installer to launch.

Note

v: latest

3. pyhton.org

https://www.python.org/

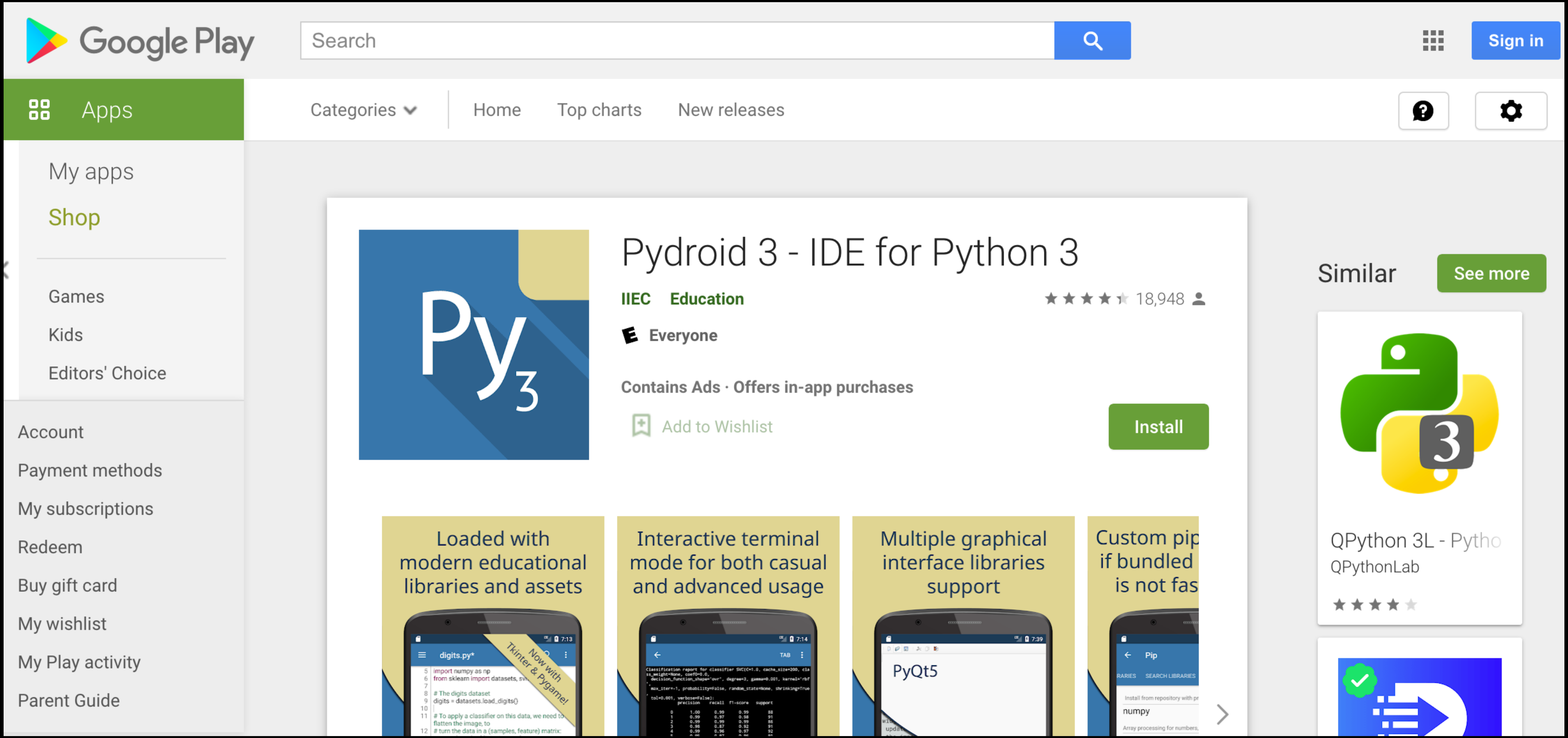


```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

4. Python on Android



Google Play

Search

Sign in

Apps

Categories · Home · Top charts · New releases

My apps

Shop

Games

Kids

Editors' Choice

Account

Payment methods

My subscriptions

Redeem

Buy gift card

My wishlist

My Play activity

Parent Guide

Pydroid 3 - IDE for Python 3

IIEC Education ★★★★½ 18,948

Everyone

Contains Ads · Offers in-app purchases

Add to Wishlist

Install

Loaded with modern educational libraries and assets

Interactive terminal mode for both casual and advanced usage

Multiple graphical interface libraries support

Custom pip if bundled is not fas

Similar

See more

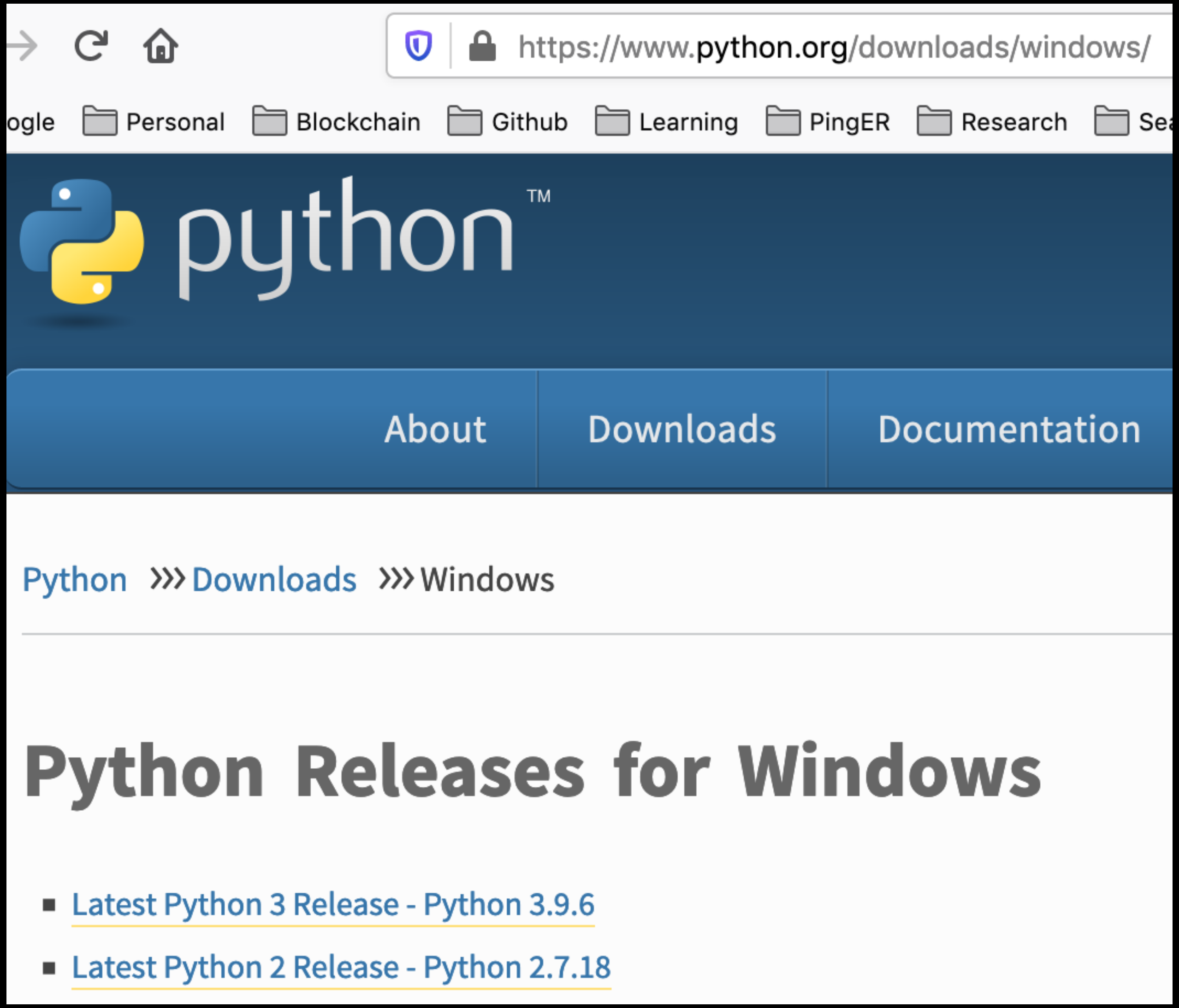
QPython 3L - Pytho

QPythonLab

5. Installing Python - Interactive

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
>>>
```

What next?

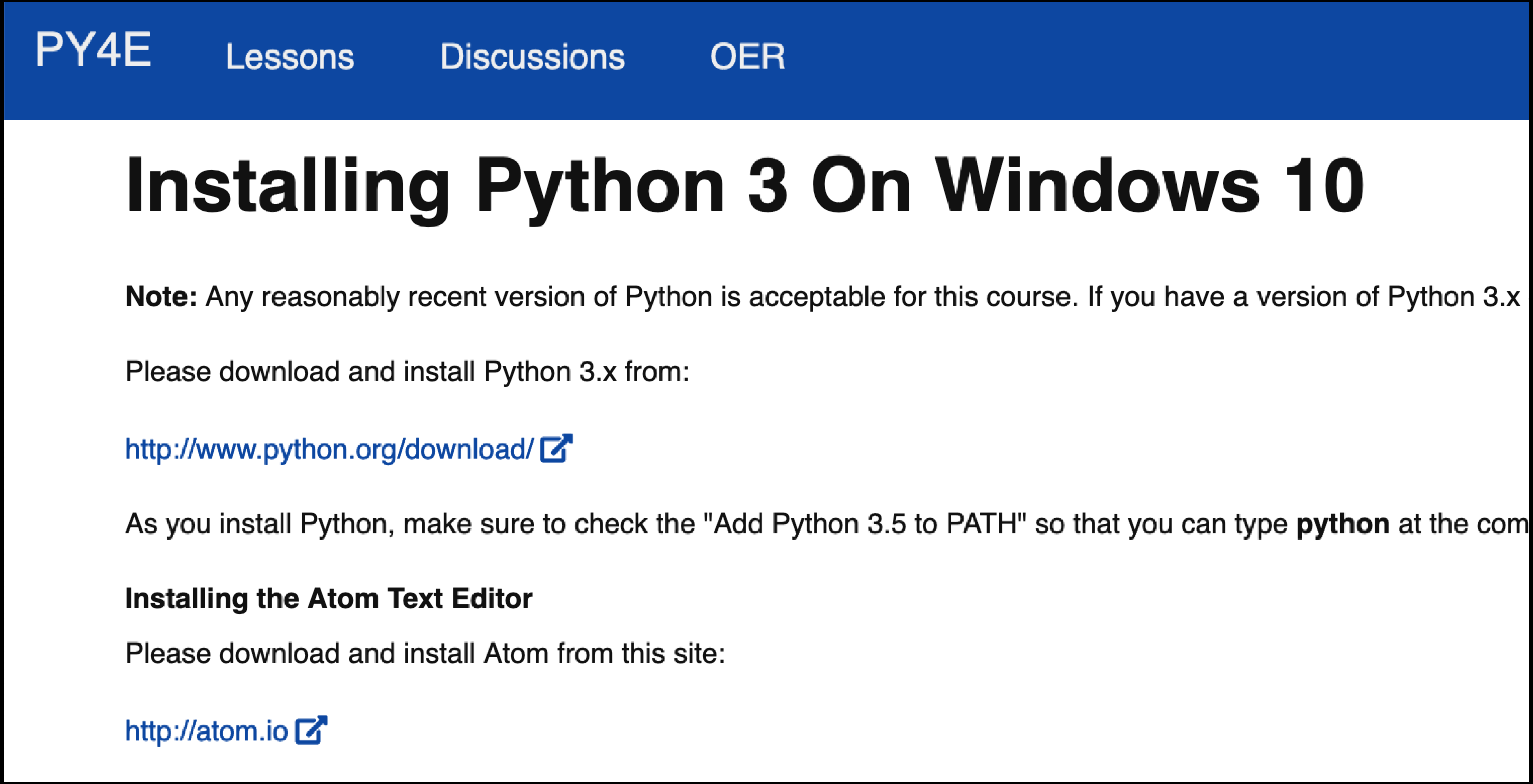


Python >>> Downloads >>> Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.9.6
- Latest Python 2 Release - Python 2.7.18

6. Installing Python - Script



PY4E Lessons Discussions OER

Installing Python 3 On Windows 10

Note: Any reasonably recent version of Python is acceptable for this course. If you have a version of Python 3.x

Please download and install Python 3.x from:

http://www.python.org/download/

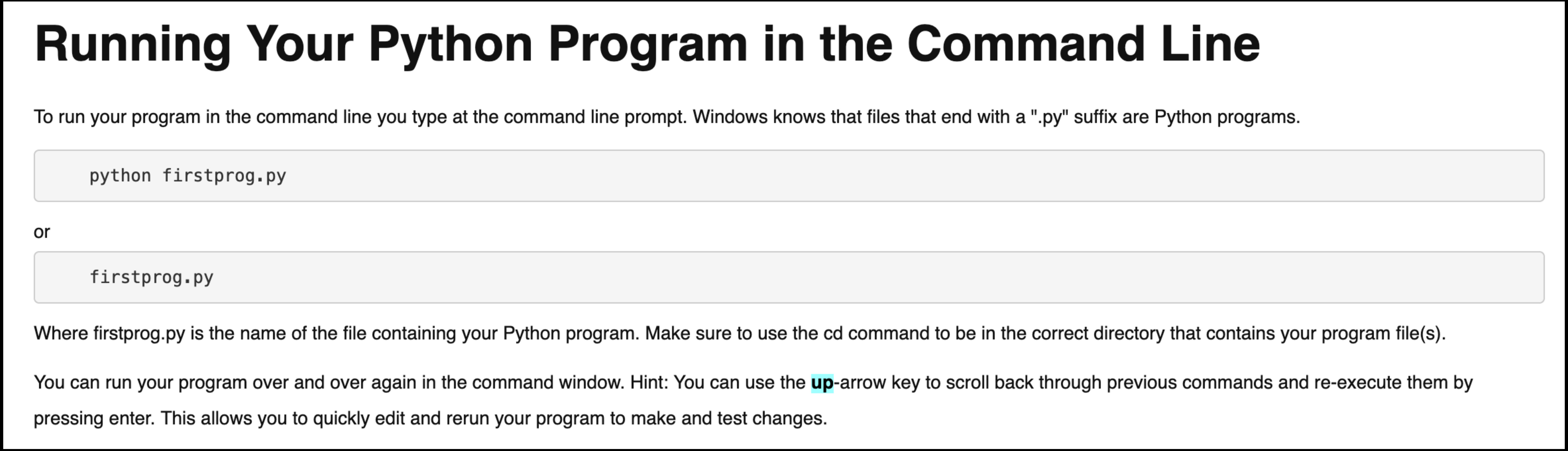
As you install Python, make sure to check the "Add Python 3.5 to PATH" so that you can type **python** at the com

Installing the Atom Text Editor

Please download and install Atom from this site:

http://atom.io

6. Installing Python - Script



Running Your Python Program in the Command Line

To run your program in the command line you type at the command line prompt. Windows knows that files that end with a ".py" suffix are Python programs.

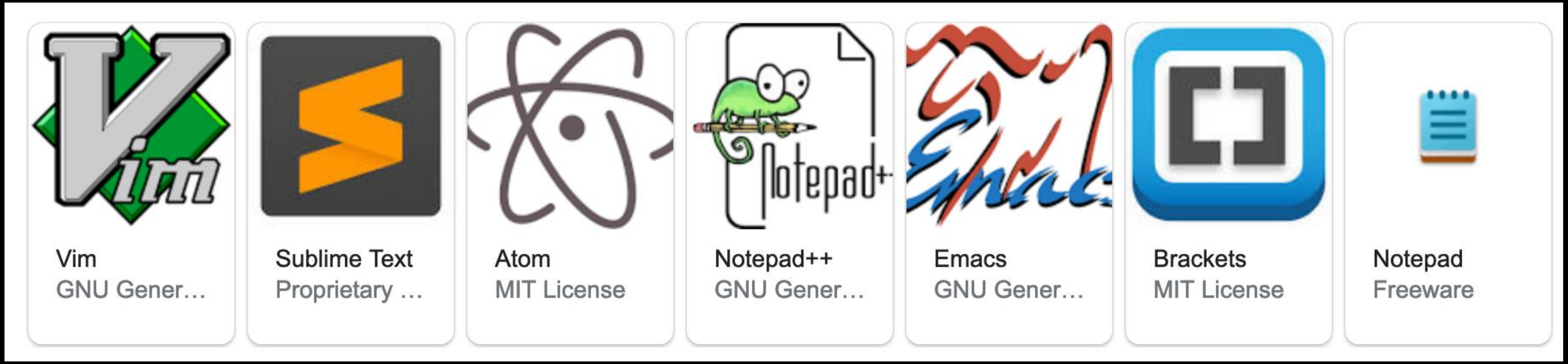
```
python firstprog.py
```

or

```
firstprog.py
```

Where firstprog.py is the name of the file containing your Python program. Make sure to use the cd command to be in the correct directory that contains your program file(s).

You can run your program over and over again in the command window. Hint: You can use the **up**-arrow key to scroll back through previous commands and re-execute them by pressing enter. This allows you to quickly edit and rerun your program to make and test changes.



```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"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.

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 = input('Enter file:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    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

```
False   class   return  is      finally
None    if      for     lambda  continue
True    def     from    while   nonlocal
and     del     global  not     with
as      elif    try     or      yield
assert  else    import  pass
break   except  in      raise
```

Sentences or Lines

`x = 2`	←	Assignment statement
`x = x + 2`	←	Assignment with expression
`print(x)`	←	Print statement

Variable Operator Constant Function

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.

Interactive versus Script

- Interactive

  - You type directly to Python one line at a time and it responds

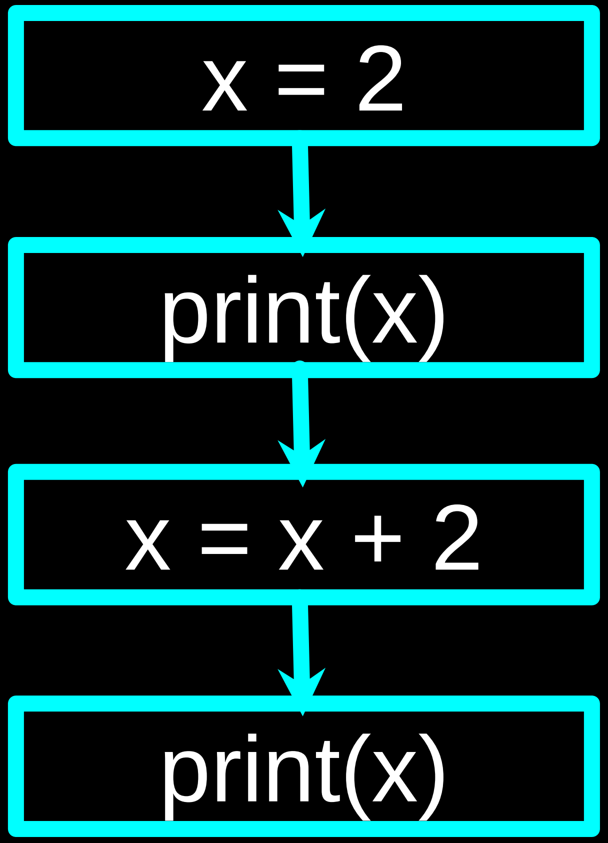
- Script

  - 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 is 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:

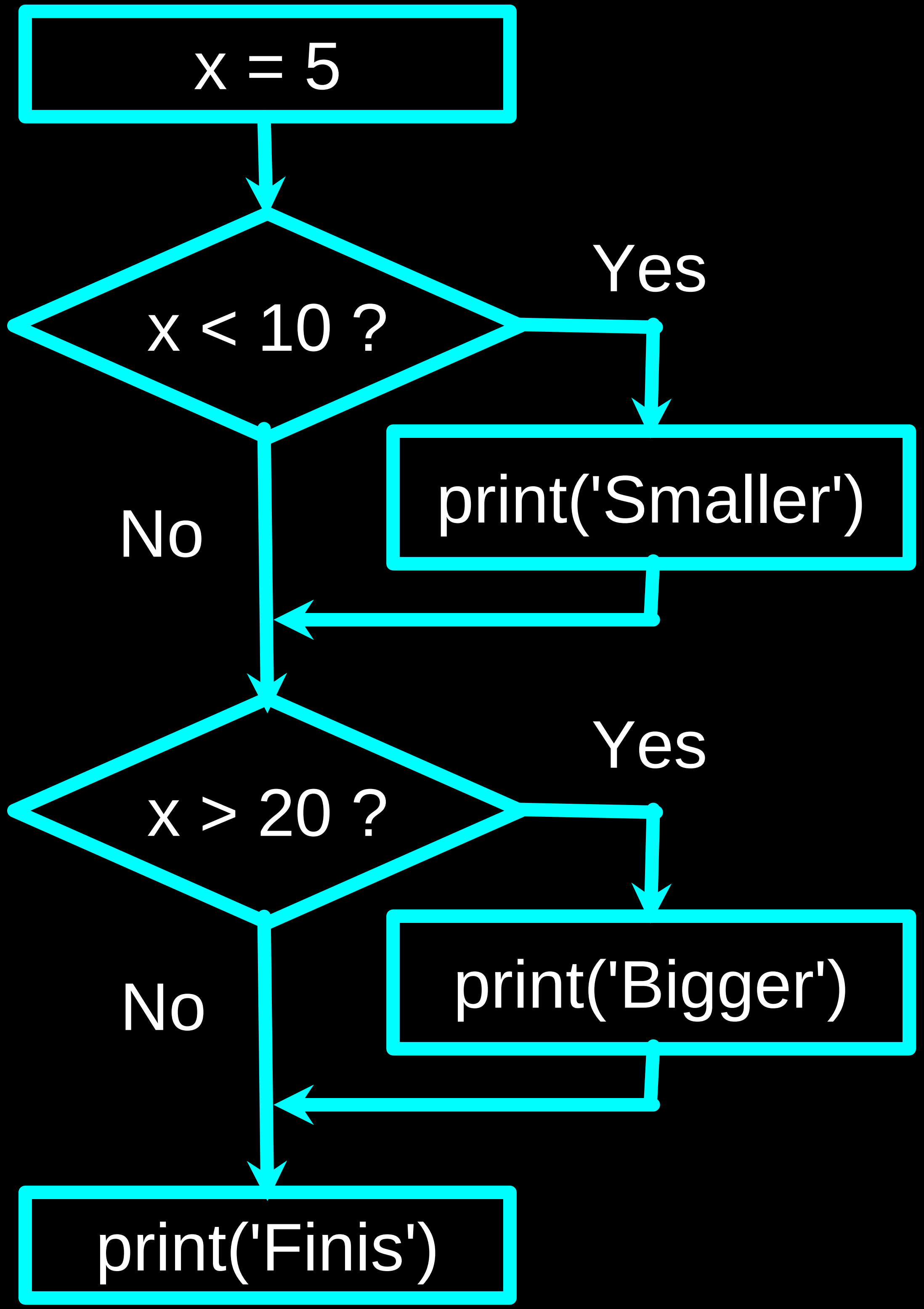
```
x = 2
print(x)
x = x + 2
print(x)
```

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:

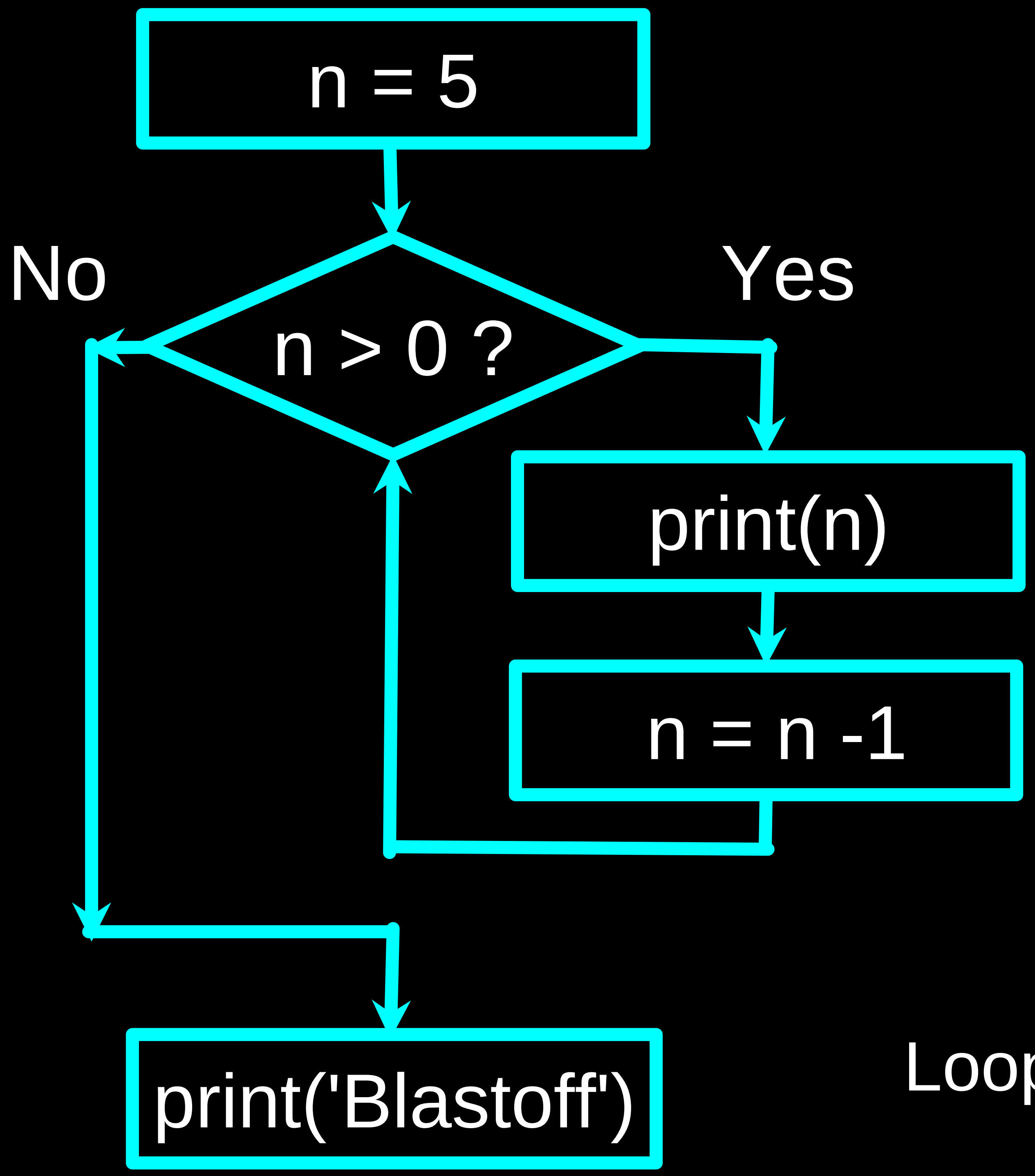
```
x = 5
if x < 10:
    print('Smaller')
if x > 20:
    print('Bigger')

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.

```python
name = input('Enter file:')
handle = open(name, 'r')

counts = dict()
for line in handle:
    words = line.split()
    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 = input('Enter file:')
handle = open(name, 'r')

counts = dict()
for line in handle:
    words = line.split()
    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 Chapter 1
- We will revisit these concepts throughout the course
- Focus on the big picture

Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

… Insert new Contributors and Translators here

Continue…