

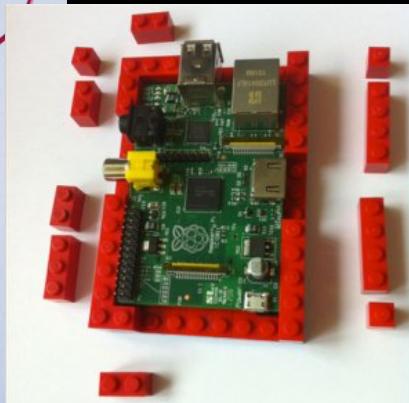
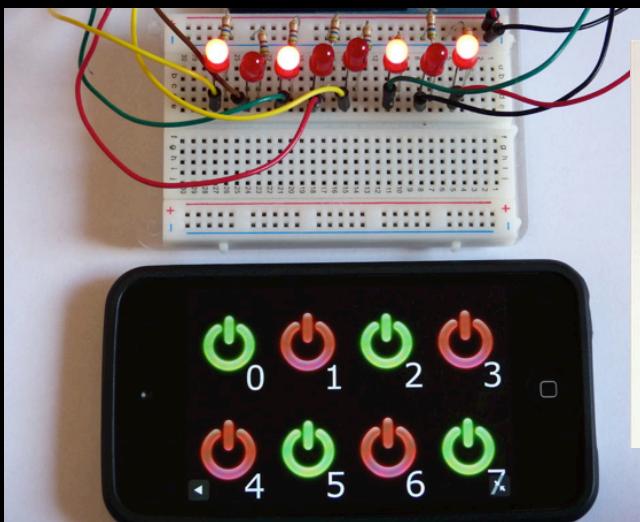
Creative Thinking through Programming

Day 1

What's in your stack yo?



Programming is awesome!



P5 is awesome too!

<http://p5js.org>

AND THEN THERE'S

THE TOOL FOR WHEN YOU GET SERIOUS

Openframeworks.org

This class uses python 2.7.10 and pycharm

Programming is problem solving

- Problem solving means the ability to formulate problems, think creatively about solutions, and express a solution clearly and accurately.
- How you solve the problem is called the algorithm.
- An algorithm is a step by step list of instructions, or a program, that if followed exactly will solve the problem under consideration.
- These programs are written in **programming languages**.

Python

- Python is an example of a **high-level language** (C++, PHP, and Java.)
- There are also low level programming languages (assembly, machine language)
- High level languages use human friendly sets of instructions to create programs. Low level languages are not as human friendly and are written for the hardware.

Machine language vs High level language

print “hello”

The computer would translate this high language into a low level language for you → 01010101 00011101 00111100

verses writing this yourself

01010101 00011101 00111100

Clearly, high level languages are faster and less prone to error than machine languages

Binary

All low level languages are in binary. Aka a series of 1's and 0's. It's the matrix all over again.

How binary works.

Binary is a series of bits that form 8byte patterns that represent numeric values. Each value represents either on or off. You can think of them like little switches.

Binary Math

Each bit represents either if it should be counted or not. So 0 = don't count me. 1 = count me.

00000001 is 1

00000010 is 2

00000011 is 3

00000100 is 4

00000101 is 5

And so on.....

128	64	32	16	8	4	2	1	
0	0	0	0	0	0	0	0	
128	64	32	16	8	4	2	1	
0	0	0	0	1	0	0	0	= 8
128	64	32	16	8	4	2	1	
0	0	1	0	0	0	0	0	= 32
128	64	32	16	8	4	2	1	
0	0	1	0	0	1	1	0	= 38
128	64	32	16	8	4	2	1	
1	0	0	0	0	1	0	1	= 133

Ascii

Ascii is a way to translate binary into letters and numbers

ASCII Code: Character to Binary

0	0011 0000	o	0100 1111	m	0110 1101
1	0011 0001	p	0101 0000	n	0110 1110
2	0011 0010	q	0101 0001	o	0110 1111
3	0011 0011	r	0101 0010	p	0111 0000
4	0011 0100	s	0101 0011	q	0111 0001
5	0011 0101	t	0101 0100	r	0111 0010
6	0011 0110	u	0101 0101	s	0111 0011
7	0011 0111	v	0101 0110	t	0111 0100
8	0011 1000	w	0101 0111	u	0111 0101
9	0011 1001	x	0101 1000	v	0111 0110
A	0100 0001	y	0101 1001	w	0111 0111
B	0100 0010	z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	i	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(0010 1000
N	0100 1110	l	0110 1100)	0010 1001
				space	0010 0000

Unicode

This is kind of the worm hole but...

There's also Unicode! Unicode is more advanced mapping that allows for far more characters, other languages and so on. It can be 8, 16 or 32 bits.

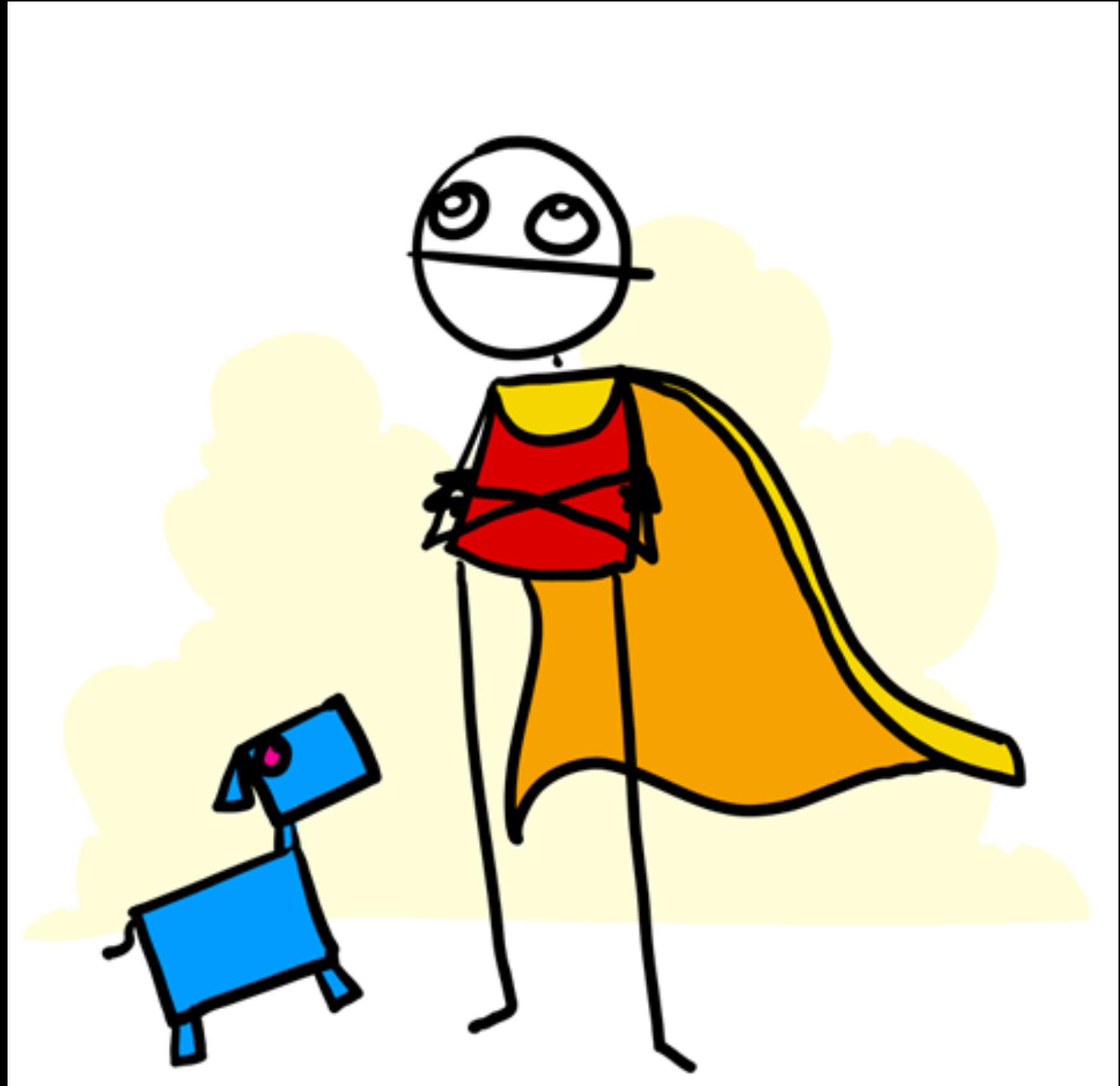
“an international encoding standard for use with different languages and scripts, by which each letter, digit, or symbol is assigned a unique numeric value that applies across different platforms and programs.”

Unicode is common online.

Python!

Python uses Unicode but because it's higher level. You will almost never, ever think about this.

Where this gets bad is data from the internet but you don't have to deal with in this class.... If you are lucky, you'll never have to think this low level.



Interpreters & Compilers

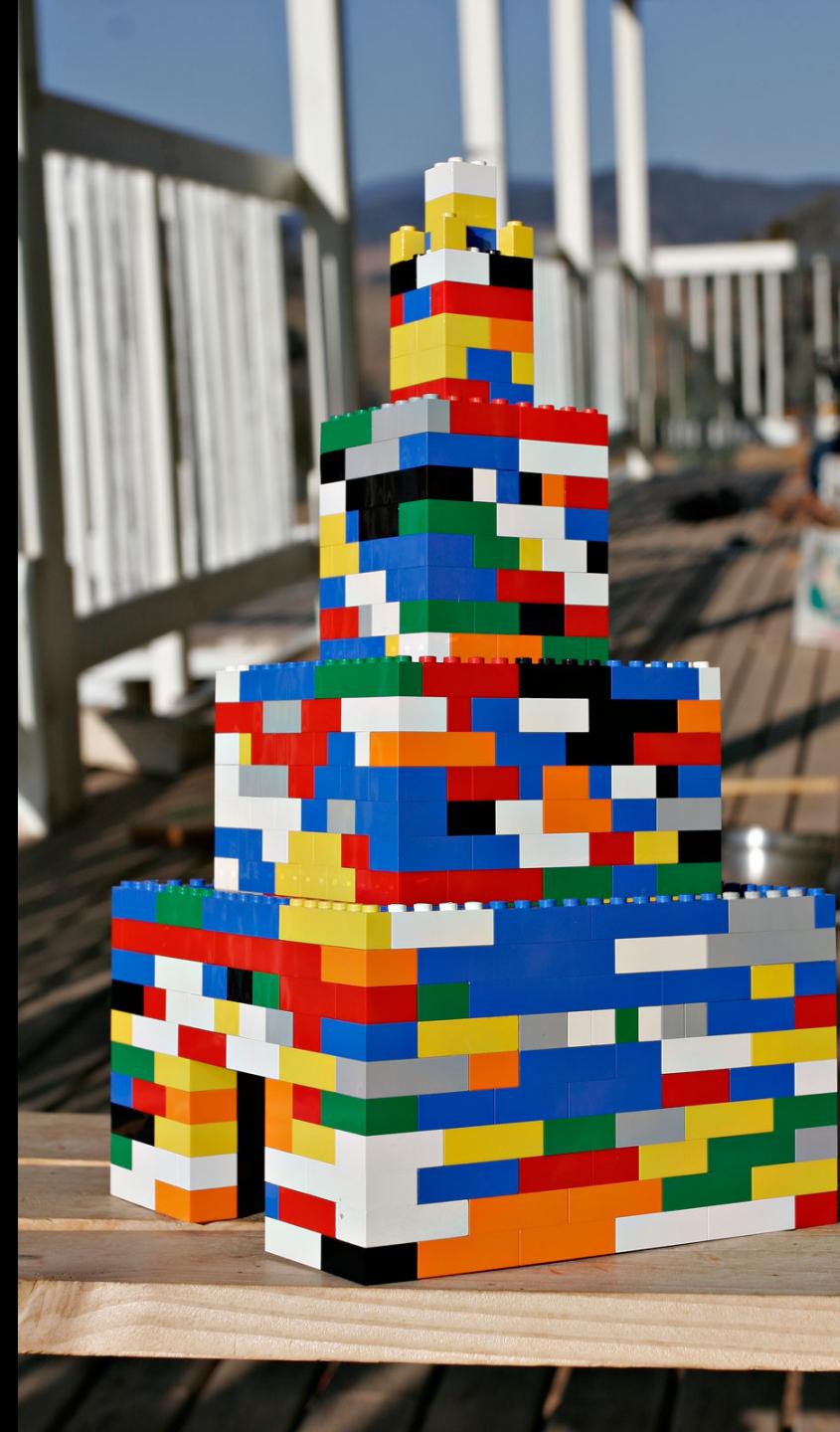
An interpreter reads a high-level program and executes it, meaning that it does what the program says. It processes the program a little at a time, alternately reading lines and performing computations.



javascript

Compilers

A compiler reads the program and translates it completely before the program starts running. In this case, the high-level program is called the **source code**, and the translated program is called the **object code** or the **executable**. Once a program is compiled, you can execute it repeatedly without further translation.



Python does both

- Many modern languages use both processes. They are first compiled into a lower level language, called **byte code**, and then interpreted by a program called a **virtual machine**. Python uses both processes, but because of the way programmers interact with it, it is usually considered an interpreted language.
- There are two ways to use the Python interpreter: *shell mode* and *program mode*. In shell mode, you type Python expressions into the **Python shell**. In program mode you type them into a file.

WHAT IS THIS VOODOO

Meet Unix on mac and the command prompt on windows

<http://mally.stanford.edu/~sr/computing/basic-unix.html>



Hardware vs software

Hardware yo!



Software

Apps by name ▾

Search icon

Acer Games	Calculator NEW	Flipboard NEW	Intel® Experience Center	newsXpresso	Scan NEW
Adobe Acrobat Distiller XI	Calendar	Food & Drink NEW	Internet Explorer	OneNote NEW	SkyDrive
Adobe Acrobat XI Pro	Camera	Games	Kindle	OverDrive Media Console for...	Skype
Adobe Creative Cloud	Content Manager Assistant for...	GDrive Pro NEW	Mail	PC App Store	Soluto
Adobe FormsCentral	Desktop	gmail calendar NEW	Maps	PC settings	Sound Recorder NEW
Adobe Illustrator CC	Dropbox	Health & Fitness NEW	Maps App V6 NEW	People	Sports
Adobe Illustrator CC (64 Bit)	Evernote Touch	Help+Tips NEW	Music	Photos	Spotify
Alarms NEW	Facebook NEW	Hulu Plus	Netflix	Reader	Store
Amazon	Finance	Intel(R) WiDi	News	Reading List NEW	Sublime Text 2

Up arrow icon

< >

How a program works

- A program lives on your hard disk
- The instructions of a program get loaded into ram. The machine code is fetched into your CPU
- These instructions get decoded and then executed on your CPU



Pycharm

For this class we will be using pycharm and python 2.7

<http://www.python.org/>

<http://www.sublimetext.com/3>

Input variables and output

Variables are types of data stored in containers. Think of them like a box that holds something in it

`x = 10`

`x` is the variable

`=` is the assignment operator and sets the value on the left of the equation to be equal to the variable on the right

10 is the type. It's an integer number. Or a hole number.



Variables can hold different type of data

- These basic or “scalar” types are
- Int = 10
- Float = 10.0
- Boolean = true or false (0 or 1)
- None = none... ☺ More on this later

Not Scalar but important

String = “a set of words or characters in quotes”

Special or reserved words (Thou shall not)

Do not put spaces in your file names or variable names

Do not use special characters &!\$*+-~,/\%^()

Rule of thumb? Anything that's not a letter number or underscore is out of the running for a variable name

All variables must start with a letter or an underscore. No numbers first.

Here's Python's complete list of reserved words:

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

Print something

Print is a function in python.
A function is a set of
instructions that does
something. You sometimes
put things into it. Sometimes
it returns values.

Functions are like little
machines.

I don't really care how a
coffee pot works. All I know is
if I put water and coffee grinds
into it, it makes coffee and I
can take coffee out of it.



Python particulars

You can add numbers using +

You can multiply with *

You can divide with /

Strings

You can also add strings together. This is called concatenation

x = "hello "

y = " world"

Print x + y

Your lab is at:

https://github.com/phoenixperry/hku_python/blob/master/labs/lab_01.md