Loops and Orbits-Week 1-Day 1- Computer Science

Computer science is far more than just programming, and the subject has been denigrated by people who equate the two.

We will mostly be learning programming in Python, but I hope to do a little more than just that. In particular, I want to teach some software engineering best practices. As one example, documentation!

Documentation in Markdown

HTML is incredibly widespread. It was designed to:

* Be easy to render on a wide variety of devices — even computer terminals with no graphics.
* Be easy to write and read in its raw form.
* Have common word processing features like titles, subtitles and bulleted lists.
* Incorporate URLs (hyperlinks to other documents)

The convenience of linking to other documents is really what caused the web to go viral. That was 1993.

HTML rapidly became bloated, especially with the addition of JavaScript in 1995.

The two features in the above list were lost.

HTML stands for Hypertext Markup Language.

Enter, Markdown, (see the pun — computer scientists just love little plays on words like that!). Markdown was introduced in 2004 to restore simplicity to making documentation.

```
## Documentation in Markdown

This entire text block is Markdown! Understand these features! Markdown:

* Is easy to render on a wide variety of devices &mdash; even computer
terminals with no graphics
* Is easy to write and read in its raw form
* Has common word processing features like titles, subtitles, and bulleted
lists
* Incorporates URLs hyperlinks to other documents
  * Example: link to a page that introduces [GitHub-flavored Markdown]
(https://guides.github.com/features/mastering-markdown)
```

Programming in Python

Expressions

The following are examples of expressions:

```
2+2
15-13
2*2
15/13
```

In Python 2, 15/13 gave 1. In Python 3, 15/13 gives 1.1538461538461537. This is a huge change in the meaning of division.

Python 2 is getting obsolete. We will only consider versions 3.6 and later (3.8 is current).

Variables and Assignments

The following define variables and/or put new values into them:

```
x=0
x=x+1
```

Does the last assignment statement make any sense? What if you subtract x from both sides of x=x+1? Does that mean 0=1?

Some Types (not all of them)

Boolean Type

Integer Type ← discuss exact representation

Floating Type ← discuss mantissa & exponent

String Type

NEWBIE MISTAKE!!
Confusing == with =.
Comparison is not assignment!

Comparisons

>	read this as "is less than"
<	read this as "is greater than"
==	read this as "is equal to"
>=	read this as "is greater or equal to"
<=	read this as "is less than or equal to"
!=	read this as "is not equal to"

Example 5 != 3 is read as:

"five is not equal to three?"

Read comparison operators as questions. Your tone should rise up a little at the end, even if you are just listening to the voice in your head.

Control flow

Now that we have comparisons, we can use them in code to have the program decide what to do next.

```
x = 27      ← position of ball in meters
v = 6       ← velocity of ball in meters/second

t = 0
while t < 10:
    x = x + v*1   ← new position of ball after moving with velocity v for 1 second
    t = t + 1

t, x
```

I hope you can see that we are already making a good start on learning enough of the rules of programming to do simulations.

Congratulate yourself on getting to this point in one day!