

Variables and Data Types.

When you write code it is often nice to define variables:

Example Code

```
|| message = "Hello World"
|| print(message)
```

If we run this it still prints "Hello World"

message is a variable. It is convenient to store data in variables.

Example Code

```
|| message = "Python is so much fun"
|| print(message)
```

What is the output?

Example Code

```
|| a = 5
|| b = 2
|| c = a + b
|| print(c)
```

here a, b, and c are variables.

What does this print?

This gets more important the more complicated your code gets.

STYLE POINTS → Avoid using l | O etc things that are hard to tell apart.

usually reserve: j, i, n, m for loops.

name things so they make sense, Python should be easy to read.

Variables are labels for the data they represent
A variable references the data value.

So how does a computer store variables?

- or the data referenced by variables.

It depends on the data type BUT it always has to convert into computer memory.

BINARY

How to count:

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10

we only have 2 numbers

→ we do the same thing!
just base 10

What about letters...

They use ASCII values.

Eg. A = 65 on the ascii table
then

65 = 01000001 in binary

BUT 01000001 ≠ 65

so how does the computer know what we want?

DATA TYPES !

int()	Integer
float()	floating point numbers
bool()	boolean TRUE/FALSE
str()	strings.

Python figures out the data type (or tries to) when you define the variable.

OTHER LANGUAGES — require you to strictly define variables

Be aware of this → int mynum = 5;

Strings - words, characters, sentences, paragraphs ...

"my strings are cool"

'my strings are cool'

double or single
quotes are fine.

LOTS OF OPERATIONS ON STRINGS.

— see the practice notebook.

Numbers - integers or floats. ^{→ contain a decimal point.}

$a = 1$ $b = 2$

$a = 1.01$ $b = 0.002$ -OR- $c = 1.0$

You can add +, subtract -, multiply *, divide /

Integers are sometimes converted to floats
when you do operations.

Eg $4/2 = 2.0$, $1 + 2.0 = 3.0$

long numbers: if you want you can use underscores.

$u = 1_000_000_000$

Comments are really helpful - to explain your code