Programming Glossary

## Types

type: Every value in a programming language has a type that determines what sort of value it is, and what you can do with it. There are primitive types such as integers, strings, and lists, as well as more complex types, including ones we ourselves can define. Also, a function that returns the type of a given argument.

>>> type(5)

<class 'int'>

>>> type('hello')

<class 'str'>

int: Short for integer. A type equivalent to integers in math: positive and negative whole numbers with no decimal. e.g. 7, -3, 0, 65536. Also the name of a function that takes one argument, converts it to an int, and returns it. e.g. int('5') == 5.

float: A type that represents a real number: positive and negative numbers that can have decimals. e.g. 3.5, -17.89, 1.0, 0.15. Also the name of a function that takes one argument, converts it to a float, and returns it. e.g. float('8.5') == 8.5.

str: Short for string. A type that represents a sequence of characters: letters, numbers, symbols, etc. In short, the type used to work with text. Surrounded by either single or double quotation marks. e.g. 'hello', 'A', '57.8'.

bool: Short for Boolean. A type that represents True or False. Used in conditionals to determine which branch the program will follow, or how many times to run a loop, for instance.

list: A type that represents a sequence of values. Can be a list of numbers, a list of strings, a list of lists, or anything else. e.g. [5, 7, 8] or ['Adam', 'Xavier', 'Eliza'].

## Variables

variable: A name assigned to a specific value. That name then refers to that value until changed later. We use descriptive variable names, such as students for a list of students or n\_groups for the number of groups.

immutable: Types that cannot change (“mutate”) without losing their identity: e.g., adding 1 to 3 makes it 4, which is no longer the same thing as 3. A variable assigned to an immutable value must be manually updated. The types int, float, str, bool are immutable.

mutable: Types that can change (“mutate”) while still maintaining their identity: e.g. replacing the first item in a list does not change which list it is. A variable assigned to a mutable value may undergo changes

## Input/Output

input: A function that waits for the user to type something, and then returns what they typed as a string. Can be given an argument that will be shown to the user as a prompt.

print: A function that displays a given argument to the user.

## Conditionals

if:

elif:

else:

## Iterables

index:

index notation:

slice notation:

## Loops

loop:

for loop — for element in container:

range loop — for i in range(N):

while loop — while condition:

## Functions

function:

signature:

arguments:

return value:

annotations:

docstring:

body: