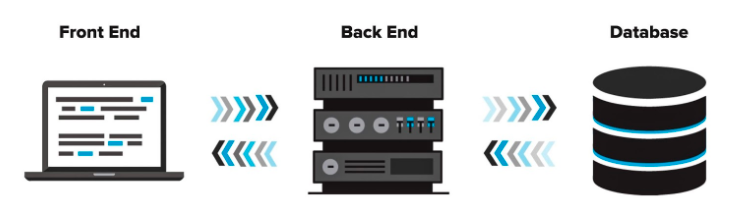
Python

Python overview



HTML python mySQL

CSS

Javascript

Here are some reasons why people in the development community think Python is great!

**Readability** - Python's syntax closely resembles English with punctuation rules that promote consistent format.

**Libraries** - Python has been around for more than 20 years. There are tons and tons of resources and libraries that you can take advantage of. If you can think of a task for which you'd like to use Python there is probably a library out there that a skilled developer made just for that purpose.

**Community** - It helps to ask others about things you're not sure of or problems that you just cannot get around. There is a massive community built around Python development built by people who are enthusiastic about Python and always happy to lend a hand!

**Scope** - Python is effective across a broad range of project types - scientific computing, data analytics, machine learning, game creation, web development, and more!

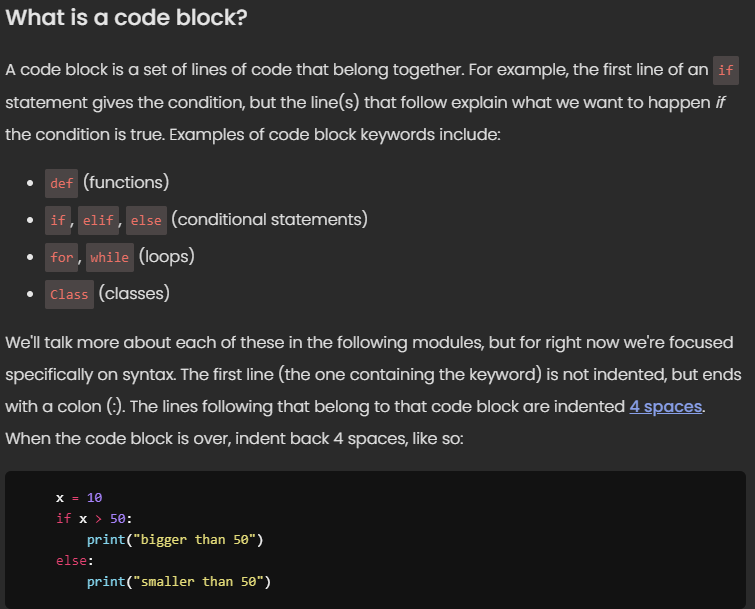
**Ease** - Python has earned a reputation for being easy to learn due to the simplicity of its syntax. It is now the most popular language taught at top universities for instruction on fundamental and advanced computing concepts

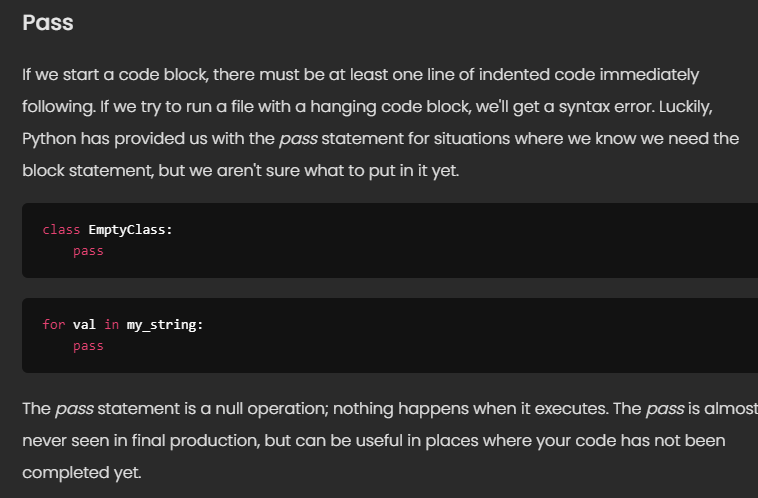
Python Syntax

INDENTING

Unlike what you may have seen in other languages, like JavaScript, Python has *no brackets or braces*. Instead, the most important aspect of Python is **indentation** to indicate which lines belong to which code blocks.

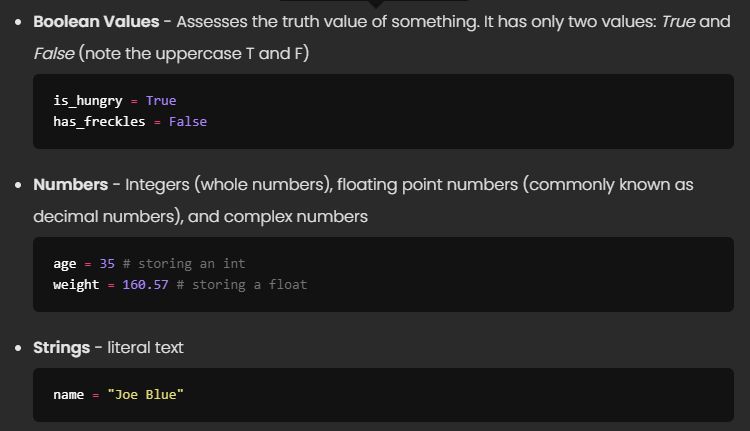
<https://peps.python.org/pep-0008/#tabs-or-spaces>





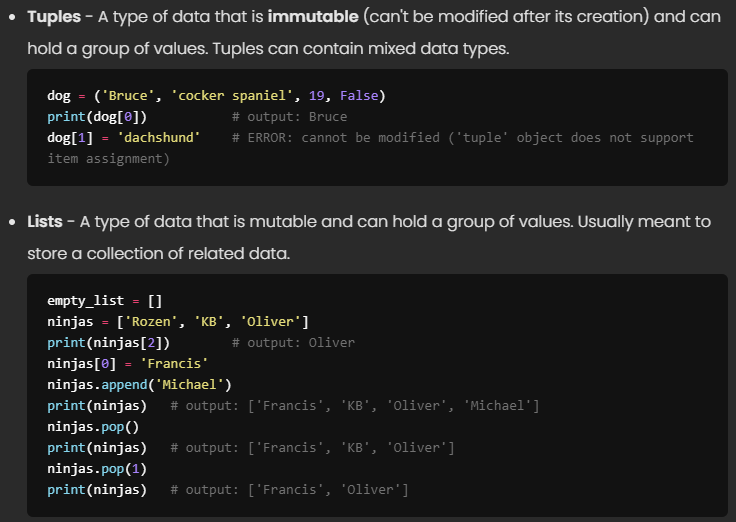
PRIMATIVE DATA TYPES

These are the basic building blocks of a language. Most languages have these in common:

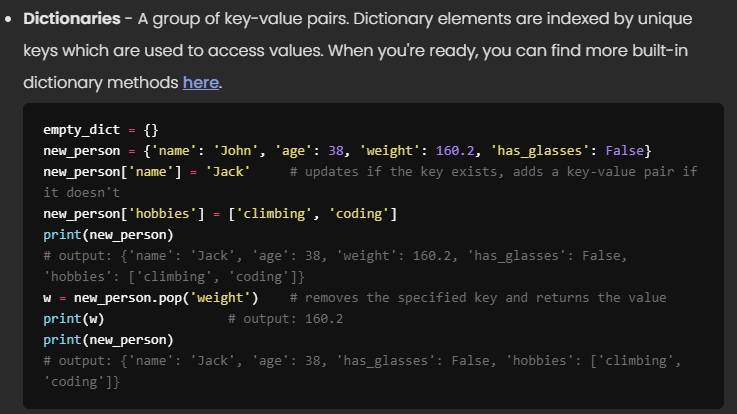


COMPOSITE TYPES

These are collections composed of the above primitive types.

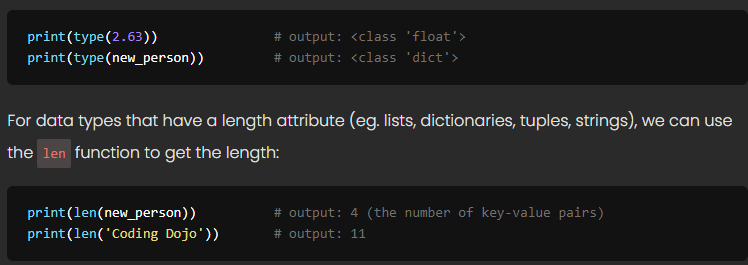


**Dictionaries** - A group of key-value pairs. Dictionary elements are indexed by unique keys which are used to access values. When you're ready, you can find more built-in dictionary methods [here](https://www.w3schools.com/python/python_ref_dictionary.asp).

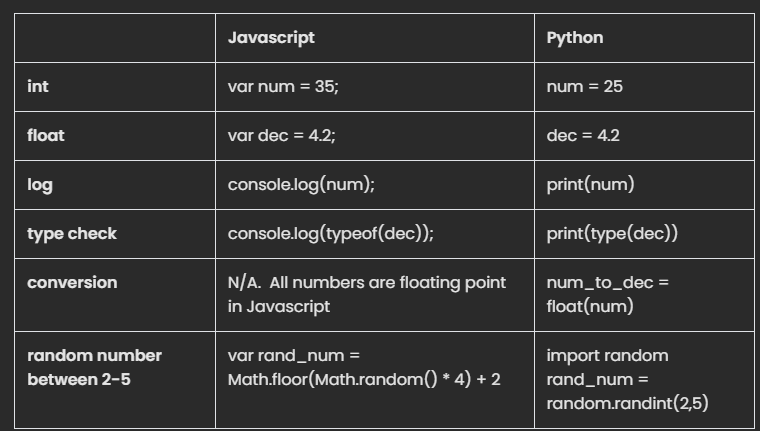


Common Functions

If we're ever unsure of a value or variable's data type, we can use the type function to find out. For example:



Numbers



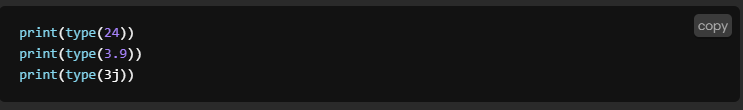
**Numbers --**There are 3 basics types of numbers in Python.

int - whole numbers, positive or negative.  ex. 35

float - decimal numbers, positive or negative.  ex. 4.2

complex - are a part of the real number system and are often referenced with the letter j.  ex. 1 + 3j.  \*\*Note\*\* If you're not sure if you need to use them, it's safe to say you can ignore this data type.

**Type -** If you are unsure on which type a number is, you can use the type() to view the object type of any object.



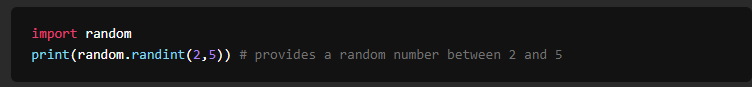
**Conversion**

All Python objects have data type methods we can use to convert number types from one to another



Random Number

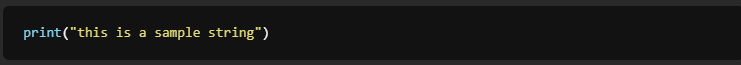
Python does not have a built in random number generator, use the [random module](https://docs.python.org/3/library/random.html) instead.



Strings

**String Literals**

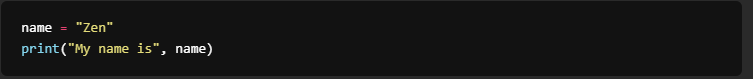
Strings are any sequence of characters (letters, numerals, ~($/}\#, etc.) enclosed in single or double quotes. We can display a string like this:



**Concatenating Strings and Variables with the print function**

There are multiple ways that we can print a string containing data from variables.

The first is by adding a comma after the string, followed by the variable. Note that the comma is outside the closing quotation mark of the string. The print() function inserts a space between elements separated by a comma.



The second is by concatenating the contents into a new string, with the help of **+**.

