**Python Standard Library**

**Overview**

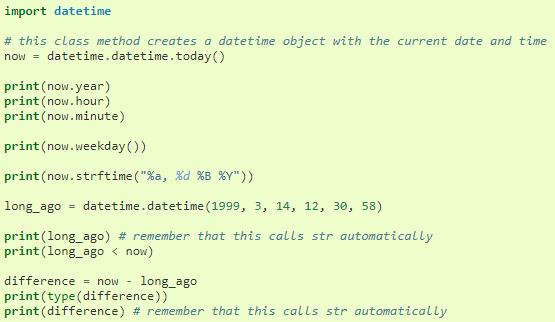
|  |
| --- |
| Python comes with a built-in selection of modules which provide commonly used functionality. We have encountered a few such as **print**(), **len**(), **max**(), to name a few.  We will look at a couple of more examples in this section. This is only a brief overview of a small subset of the available modules - you can see the full list, and find out more details about each one, reading the Python Standard Library documentation (<https://docs.python.org/3.3/library/index.html>). |

**Date and time: datetime**

The datetime module provides use with objects which we can use to store information about dates and times:

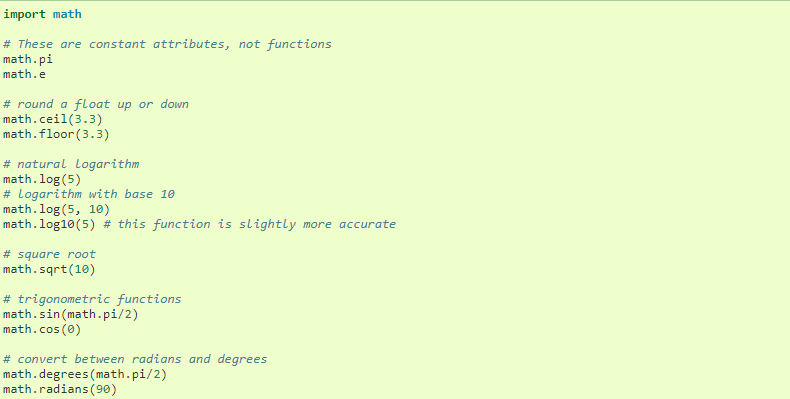
* datetime.date - is used to create dates which are not associated with a time.
* datetime.time - is used for times which are independent of a date.
* datetime.datetime - is used for objects which have both a date and a time.
* datetime.timedelta - objects store differences between dates - if we subtract one datetime from another, the result will be a timedelta (difference between two dates)

We can query these objects for a particular component (like the year, month, hour or minutes), perform arithmetic on them, and extract printable versions from them if we need to display them. Here are a few examples:



**Mathematical functions: math**

The **math** module is a collection of mathematical functions. They can be used on floats or integers, but are mostly intended to be used on floats, and usual return floats. Here are a few examples:



If you need mathematical functions to use on complex numbers, you should use the cmath module instead.

**Pseudo-random numbers: random**

In Python we can use the random module to generate random numbers, and do a few more things which depend on randomness. The core function of this module is to generate random float between 0 and 1, and most of the other functions are derived from it. Here are a few examples.



**Task 1**

Write a program which randomly picks an integer from 1 to 100. Your program should prompt the user for guesses - if the user guesses incorrectly, it should print whether the guess is too high or too low. If the user guesses correctly, the program should print how many guess the user took to guess the right answer. You can assume that the user will enter valid input.

**Task 2**

Write a program to enter a student name and surname, three test marks for a student. A function is called to calculate the average mark. Another function is used to indicate if the average mark is a pass mark (>=50%) or a fail mark (<50%). The average as well as the result must be displayed as out as well.

Add another function called ten\_dates toprint ten dates, each two week apart, starting from today, in the form YYYY-MM-DD.

**Extra Material**

Parsing CSV files: csv

CSV stands for comma-separated values - it’s a very simple file format for storing tabular data. Most spreadsheets can easily be converted to and from CSV format.

In a typical CSV file, each line represents a row of values in the table, with the columns separated by commas. Field values are often enclosed in double quotes, so that any literal commas or newlines inside them can be escaped.



Python’s csv module takes care of all this in the background, and allows us to manipulate the data in a CSV file in a simple way, using the **reader** class:



Similarly we can write to a CSV file using the write **class**:



For comprehensive details regarding the python 3 standard library, you can read the documentation which is available [here](https://docs.python.org/3.3/library/index.html).

For useful modules for daily development, you can have a look [here](https://wiki.python.org/moin/UsefulModules).