RANDOM+

MATH+

DATETIME+

TIME+

OS+

RE+

import mymodule

## Re-naming a Module

import mymodule as mx

## Using the dir() Function

There is a built-in function to list all the function names (or variable names) in a module. The dir() function:

import module  
x = dir(module)  
print(x)

****Note:**** The dir() function can be used on all modules, also the ones you create yourself.

## Import From Module

You can choose to import only parts from a module, by using the from keyword.

from mymodule import func

****Note:**** When importing using the from keyword, do not use the module name when referring to elements in the module. Example: func(), ****not**** ~~mymodule.func()~~

## Python Dates

A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects.

import datetime  
  
x = datetime.datetime.now()  
print(x)

## Date Output

When we execute the code from the example above the result will be:

2025-08-05 12:03:06.098196

The date contains year, month, day, hour, minute, second, and microsecond.

### Example

Return the year and name of weekday:

import datetime  
  
x = datetime.datetime.now()  
  
print(x.year)  
print(x.strftime("%A"))

## Creating Date Objects

To create a date, we can use the datetime() class (constructor) of the datetime module.

The datetime() class requires three parameters to create a date: year, month, day.

### Example:

import datetime  
  
x = datetime.datetime(2020, 5, 17)  
  
print(x)

The datetime() class also takes parameters for time and timezone (hour, minute, second, microsecond, tzone), but they are optional, and has a default value of 0, (None for timezone).

## The strftime() Method

The datetime object has a method for formatting date objects into readable strings.

The method is called strftime(), and takes one parameter, format, to specify the format of the returned string:

### Example: Display the name of the month

import datetime  
  
x = datetime.datetime(2018, 6, 1)  
  
print(x.strftime("%B"))

A reference of all the legal format codes:

|  |  |  |
| --- | --- | --- |
| **Directive** | **Description** | **Example** |
| %a | Weekday, short version | Wed |
| %A | Weekday, full version | Wednesday |
| %w | Weekday as a number 0-6, 0 is Sunday | 3 |
| %d | Day of month 01-31 | 31 |
| %b | Month name, short version | Dec |
| %B | Month name, full version | December |
| %m | Month as a number 01-12 | 12 |
| %y | Year, short version, without century | 18 |
| %Y | Year, full version | 2018 |
| %H | Hour 00-23 | 17 |
| %I | Hour 00-12 | 05 |
| %p | AM/PM | PM |
| %M | Minute 00-59 | 41 |
| %S | Second 00-59 | 08 |
| %f | Microsecond 000000-999999 | 548513 |
| %z | UTC offset | +0100 |
| %Z | Timezone | CST |
| %j | Day number of year 001-366 | 365 |
| %U | Week number of year, Sunday as the first day of week, 00-53 | 52 |
| %W | Week number of year, Monday as the first day of week, 00-53 | 52 |
| %c | Local version of date and time | Mon Dec 31 17:41:00 2018 |
| %C | Century | 20 |
| %x | Local version of date | 12/31/18 |
| %X | Local version of time | 17:41:00 |
| %% | A % character | % |
| %G | ISO 8601 year | 2018 |
| %u | ISO 8601 weekday (1-7) | 1 |
| %V | ISO 8601 weeknumber (01-53) | 01 |

# Python Math

## Built-in Math Functions

The min() and max() functions can be used to find the lowest or highest value in an iterable

The abs() function returns the absolute (positive) value of the specified number

The pow(x, y) function returns the value of x to the power of y (xy).

## The Math Module

Python has also a built-in module called math, which extends the list of mathematical functions.

import math

The math.sqrt() method for example, returns the square root of a number

The math.ceil() method rounds a number upwards to its nearest integer, and the math.floor() method rounds a number downwards to its nearest integer, and returns the result

The math.pi constant, returns the value of PI (3.14...):

# Python RegEx

## RegEx Module

Python has a built-in package called re, which can be used to work with Regular Expressions.

import re

## RegEx Functions

The re module offers a set of functions that allows us to search a string for a match:

|  |  |
| --- | --- |
| **Function** | **Description** |
| [findall](https://www.w3schools.com/python/python_regex.asp" \l "findall) | Returns a list containing all matches |
| [search](https://www.w3schools.com/python/python_regex.asp" \l "search) | Returns a [Match object](https://www.w3schools.com/python/python_regex.asp" \l "matchobject) if there is a match anywhere in the string |
| [split](https://www.w3schools.com/python/python_regex.asp" \l "split) | Returns a list where the string has been split at each match |
| [sub](https://www.w3schools.com/python/python_regex.asp" \l "sub) | Replaces one or many matches with a string |

### Example

Search the string to see if it starts with "The" and ends with "Spain":

import re  
  
txt = "The rain in Spain"  
x = re.search("^The.\*Spain$", txt)

## Metacharacters

Metacharacters are characters with a special meaning:

|  |  |  |
| --- | --- | --- |
| **Character** | **Description** | **Example** |
| [] | A set of characters | "[a-m]" |
| \ | Signals a special sequence (can also be used to escape special characters) | "\d" |
| . | Any character (except newline character) | "he..o" |
| ^ | Starts with | "^hello" |
| $ | Ends with | "planet$" |
| \* | Zero or more occurrences | "he.\*o" |
| + | One or more occurrences | "he.+o" |
| ? | Zero or one occurrences | "he.?o" |
| {} | Exactly the specified number of occurrences | "he.{2}o" |
| | | Either or | "falls|stays" |
| () | Capture and group |  |

## Flags

You can add flags to the pattern when using regular expressions.

|  |  |  |
| --- | --- | --- |
| **Flag** | **Shorthand** | **Description** |
| re.ASCII | re.A | Returns only ASCII matches |
| re.DEBUG |  | Returns debug information |
| re.DOTALL | re.S | Makes the . character match all characters (including newline character) |
| re.IGNORECASE | re.I | Case-insensitive matching |
| re.MULTILINE | re.M | Returns only matches at the beginning of each line |
| re.NOFLAG |  | Specifies that no flag is set for this pattern |
| re.UNICODE | re.U | Returns Unicode matches. This is default from Python 3. For Python 2: use this flag to return only Unicode matches |
| re.VERBOSE | re.X | Allows whitespaces and comments inside patterns. Makes the pattern more readable |

## Special Sequences

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

|  |  |  |
| --- | --- | --- |
| **Character** | **Description** | **Example** |
| \A | Returns a match if the specified characters are at the beginning of the string | "\AThe" |
| \b | Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string") | r"\bain"  r"ain\b" |
| \B | Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string") | r"\Bain"  r"ain\B" |
| \d | Returns a match where the string contains digits (numbers from 0-9) | "\d" |
| \D | Returns a match where the string DOES NOT contain digits | "\D" |
| \s | Returns a match where the string contains a white space character | "\s" |
| \S | Returns a match where the string DOES NOT contain a white space character | "\S" |
| \w | Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore \_ character) | "\w" |
| \W | Returns a match where the string DOES NOT contain any word characters | "\W" |
| \Z | Returns a match if the specified characters are at the end of the string | "Spain\Z" |

## Sets

A set is a set of characters inside a pair of square brackets [] with a special meaning:

|  |  |
| --- | --- |
| **Set** | **Description** |
| [arn] | Returns a match where one of the specified characters (a, r, or n) is present |
| [a-n] | Returns a match for any lower case character, alphabetically between a and n |
| [^arn] | Returns a match for any character EXCEPT a, r, and n |
| [0123] | Returns a match where any of the specified digits (0, 1, 2, or 3) are present |
| [0-9] | Returns a match for any digit between 0 and 9 |
| [0-5][0-9] | Returns a match for any two-digit numbers from 00 and 59 |
| [a-zA-Z] | Returns a match for any character alphabetically between a and z, lower case OR upper case |
| [+] | In sets, +, \*, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + |

## The findall() Function

The findall() function returns a list containing all matches. The list contains the matches in the order they are found. If no matches are found, an empty list is returned.

## The search() Function

The search() function searches the string for a match, and returns a [Match object](https://www.w3schools.com/python/python_regex.asp" \l "matchobject) if there is a match. If there is more than one match, only the first occurrence of the match will be returned. If no matches are found, the value None is returned.

## The split() Function

The split() function returns a list where the string has been split at each match. You can control the number of occurrences by specifying the maxsplit parameter.

## The sub() Function

The sub() function replaces the matches with the text of your choice. You can control the number of replacements by specifying the count parameter.

## Match Object

A Match Object is an object containing information about the search and the result.

****Note:**** If there is no match, the value None will be returned, instead of the Match Object.

The Match object has properties and methods used to retrieve information about the search, and the result:

.span() returns a tuple containing the start-, and end positions of the match.  
.string returns the string passed into the function  
.group() returns the part of the string where there was a match

****Note:**** If there is no match, the value None will be returned, instead of the Match Object.