Modules

A module is a file containing a set of codes or a set of functions which can be included to an application. A module could be a file containing a single variable, a function or a big code base.

Creating a Module

To create a module we write our codes in a python script and we save it as a .py file. Create a file named mymodule.py inside your project folder. Let us write some code in this file.

Built in functions

In Python we have lots of built-in functions. Built-in functions are globally available for your use that mean you can make use of the built-in functions without importing or configuring. Some of the most commonly used Python built-in functions are the following:

- print(), len(), type(), int(), float(), str(), input(), list(), dict(), min(), max(), sum(), sorted(), open(), file(), help(), and dir().
- In the following table you will see an exhaustive list of Python built-in functions taken from python documentation.

Built-in Functions

abs ( ), delattr (), hash (), memoryview(), set (), a11(), dict(), help (), min(), setattr (), any(), dir(), hex(), next (), slice(), ascii(), divmod(), id(), object (), sorted (), bin(), enumerate(), input (), oct ( ), staticmethod (), bool(), eval(), nt(), open (), str(), breakpoint(), exec(), isinstance(), ord(), sum (), bytearray (), filter (), issubclass (), pow (), super (), bytes (), float(), iter (), print(), tuple(), callable(), format (), len(), property (), type (), chr (), frozenset), list(), range (), vars (), classmethod(), getattr (), locals (), repr (), zip (), compile(), globals (), map ( ), reversed, *import*(), complex(), hasattr(, max (), round ()

Import Built-in Modules

Like other programming languages we can also import modules by importing the file/function using the key word import. Let's import the common module we will use most of the time. Some of the common built-in modules: math, datetime, os,sys, random, statistics, collections, json,re

OS Module

Using python os module it is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating, changing current working directory, and removing a directory (folder), fetching its contents, changing and identifying the current directory.

Unsupported Cell Type. Double-Click to inspect/edit the content.

Statistics Module

The statistics module provides functions for mathematical statistics of numeric data. The popular statistical functions which are defined in this module: mean, median, mode, stdev etc.

```python
from statistics import * # importing all the statistics modules
ages = [20, 20, 4, 24, 25, 22, 26, 20, 23, 22, 26]
print(mean(ages))       # ~22.9
print(median(ages))     # 23
print(mode(ages))       # 20
print(stdev(ages))      # ~2.3
```

```
21.09090909090909
22
20
6.106628291529549
```

Math Module

Module containing many mathematical operations and constants.

```
import math
print(math.pi)           # 3.141592653589793, pi constant
print(math.sqrt(2))      # 1.4142135623730951, square root
print(math.pow(2, 3))    # 8.0, exponential function
print(math.floor(9.81))  # 9, rounding to the lowest
print(math.ceil(9.81))   # 10, rounding to the highest
print(math.log10(100))   # 2, logarithm with 10 as base
```

```
3.141592653589793
1.4142135623730951
8.0
9
10
2.0
```

Now, we have imported the math module which contains lots of function which can help us to perform mathematical calculations. To check what functions the module has got, we can use help(math), or dir(math). This will display the available functions in the module. If we want to import only a specific function from the module we import it as follows:

```
from math import pi
print(pi)
```

```
3.141592653589793
```

It is also possible to import multiple functions at once

```
from math import pi, sqrt, pow, floor, ceil, log10
print(pi)                 # 3.141592653589793
print(sqrt(2))            # 1.4142135623730951
print(pow(2, 3))          # 8.0
print(floor(9.81))        # 9
print(ceil(9.81))         # 10
print(math.log10(100))    # 2
```

```
3.141592653589793
1.4142135623730951
8.0
9
10
2.0
```

But if we want to import all the function in math module we can use * .

```
from math import *
print(pi)                  # 3.141592653589793, pi constant
print(sqrt(2))             # 1.4142135623730951, square root
print(pow(2, 3))           # 8.0, exponential
print(floor(9.81))         # 9, rounding to the lowest
print(ceil(9.81))          # 10, rounding to the highest
print(math.log10(100))     # 2
```

```
3.141592653589793
1.4142135623730951
8.0
9
10
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-2296638340.py in <cell line: 0>()
      5 print(floor(9.81))          # 9, rounding to the lowest
      6 print(ceil(9.81))           # 10, rounding to the highest
```

When we import we can also rename the name of the function.

```
from math import pi as  PI
print(PI) # 3.141592653589793
```

```
3.141592653589793
```

Next steps: Explain error

String Module

A string module is a useful module for many purposes. The example below shows some use of the string module.

```
import string
print(string.ascii_letters) # abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
print(string.digits)        # 0123456789
print(string.punctuation)   # !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

Random Module

By now you are familiar with importing modules. Let us do one more import to get very familiar with it. Let us import random module which gives us a random number between 0 and 0.9999.... The random module has lots of functions but in this section we will only use random and randint.

```
from random import random, randint
print(random())   # it doesn't take any arguments; it returns a value between 0 and 0.9999
print(randint(5, 20)) # it returns a random integer number between [5, 20] inclusive
```

```
0.5190572310181213
6
```

Exercises: Level 1

Writ a function which generates a six digit/character random_user_id.

Write a function which returns an array of seven random numbers in a range of 0-9. All the numbers must be unique.

```
import random

def unique_random_seven():
    # random.sample picks unique elements from a range
    return random.sample(range(10), 7)

# Example usage
numbers = unique_random_seven()
print(numbers)
```

```
[5, 9, 3, 4, 6, 8, 2]
```

```
Start coding or generate with AI.
```