IIT Madras
BSc Degree

Home Lesson-3.1

Lesson-2.4

Library

A library is a collection of functions that share a common theme. This is a loose definition and will become clear when we start working with a library.

`calendar`

Consider the following problem:

> In the year 3000, 15^{th} August will fall on which day of the week?

Python to the rescue:

```
import calendar
calendar.prmonth(3000, 8)
```

When the above code is executed, the output is:

```
    August 3000
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

15th of August falls on a Friday. Isn't that lovely? It took just two lines of code! `calendar` is one among several libraries in Python's standard library. A comprehensive list can be found here. Going back to the code, `calendar` is the name of the library and `import` is the keyword used to include this library as a part of the code.

`calendar` is a collection of functions that are related to calendars. `prmonth` is one such function. It accepts `<year>` and `<month>`, as input and displays the calendar for `<month>` in the year `<year>`. If we want to use a function in `calendar`, we must first import the library. Let us see what happens if skip this step:

```
# import calendar
calendar.prmonth(3000, 8)
```

It gives the following error:

```
NameError: name 'calendar' is not defined
```

To access a function defined inside a library, we use the following syntax:

```
<calendar>.<function>(<arguments>)
```

Another way to solve the problem is to use the function `weekday`:

```
import calendar
print(calendar.weekday(3000, 8, 15))
```

The output of the above code is `4`. Days are mapped to numbers as follows:

Day	Number
Monday	0
Tuesday	1
Wednesday	2
Thursday	3
Friday	4
Saturday	5
Sunday	6

`time`

Let us now try to answer this hypothetical question:

> You are stranded on an island in the middle of the Indian Ocean. The island has a computing device that has just one application installed in it: a Python interpreter. You wish to know the current date and time.

Solution

```
from time import ctime
print('The current time is:', ctime())
```

The output is:

```
The current time is: Fri Apr  2 12:24:43 2021
```

The syntax of the import statement in line-1 looks different. `from` is a new keyword. The first line of the code is essentially doing the following: from the library called `time` import the function called `ctime`. This way of importing functions is useful when we need just one or two functions from a given library:

```
from time import ctime, sleep
print('Current time is:', ctime())
print('I am going to sleep for 10 seconds')
sleep(10)
print('Current time is:', ctime())
```

`sleep(x)` is a function in `time` that suspends the execution of the program for `x` seconds. If we would be using several functions in the library, then it is a bad idea to keep importing each of them individually. In such cases, it is good to fall back on importing the entire library.

`this`

As a fun exercise, consider the following code:

```
import this
```

This gives the following output:

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

These are some nuggets of wisdom from Tim Peters, a "major contributor to the Python programming language" [refer]. Some of the points make immediate sense, such as "readability counts".