

That's So Random

An Introduction to the **Random** and **Calendar** Modules

What the Modules Do:

random — Generate pseudo-random numbers

Source code: Lib/random.py (<https://docs.python.org/2/library/random.html>)

The **random** module implements pseudo-random number generators for various distributions.

random.random()

Return the next random floating point number in the range [0.0, 1.0).

random.randint(a, b)

Return a random integer N such that $a \leq N \leq b$.

random.sample(population, k)

Return a k length list of unique elements chosen from the population sequence. Used for random sampling without replacement.

calendar — General calendar-related functions

Source code: Lib/calendar.py (<https://docs.python.org/2/library/calendar.html>)

The **calendar** module allows you to output calendars like the Unix `cal` program, and provides additional useful functions related to the calendar. By default, these calendars have Monday as the first day of the week, and Sunday as the last (the European convention).

calendar.weekday(year, month, day)

Returns the day of the week (0 is Monday) for year (1970–...), month (1–12), day (1–31).

calendar.month(theyear, themonth[, w[, l]])

Returns a month's calendar in a multi-line string using the `formatmonth()` of the `TextCalendar` class.

About gencal()

The `gencal()` function has no arguments. It uses `random.randint` to generate a random date (month, day, year) and then display the calendar for that random date's month.

Sample Output:

```
>>>
A random floating point number between 0.0 and 1.0 is:  0.311178935597
And here's a random integer between 298 and 7365:  2668
Plus a random sampling of three items in a pre-defined list
    [9, 23, 39485, -132, 0, -55] : [9, -55, 23]

- - - - - Generating A Random Month's Calendar - - - - -

1 18 2000 falls on Day 1 of the week.

(Monday = Day 0, Tuesday = Day 1, Wednesday = Day 2, . . . )

And here is the calendar for  1 2000 :

    January 2000
Su Mo Tu We Th Fr Sa
                   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

>>>
```