



BAC Pro CIEL	<h2>Python – Module 7</h2> <h3>Utiliser le module random</h3>	 Année 2025/2026
		

## A. Importation de module

La fonction `random()` appartient au module `random`. Elle fait partie du module `random`, qui est une collection de fonctions conçues pour générer des nombres pseudo-aléatoires. Pour utiliser ces fonctions, il est nécessaire d'importer le module.

Un module est simplement un fichier Python (`.py`) contenant du code Python (fonctions, classes, variables). Il permet d'organiser le code de manière logique et réutilisable.

Une librairie est souvent un terme plus large qui désigne une collection de modules, ou un module de grande taille qui offre de nombreuses fonctionnalités.

Python est livré avec une vaste bibliothèque standard (Standard Library) qui contient des centaines de modules pour des tâches courantes, comme les mathématiques, les opérations de fichiers, les requêtes réseau, et bien sûr, la génération de nombres aléatoires.

Lorsque vous démarrez un script Python, seules les fonctions et types de base (`print()`, `input()`, `int()`, ...) sont disponibles. Pour utiliser des fonctionnalités plus avancées et spécifiques (comme la génération de nombres aléatoires, l'accès à la date/heure, ...) vous devez explicitement dire à Python que vous voulez charger le module contenant ces fonctionnalités.

C'est là qu'intervient l'instruction `import`.

Il y a plusieurs manières d'importer des modules ou des parties de modules.

La méthode la plus simple et la plus recommandée est d'importer tout le module. Pour accéder à une fonction du module, vous devez utiliser le nom du module suivi d'un point puis le nom de la fonction :

```
import nom_du_module
```

Exemple :

```
import random
```

```
mon_nombre = random.random()
```

```
mon_autre_nombre = random.randint(1,10)
```

Si vous n'avez besoin que d'une ou deux fonctions spécifiques d'un module, vous pouvez les importer directement. Cela vous permet d'utiliser le nom de la fonction sans le préfixe du module. La syntaxe est la suivante :

```
from nom_du_module import nom_de_la_fonction
```

Exemple :

```
from random import randint  
mon_nombre_aleatoire = randint(1,6)
```

Pour éviter les conflits de nom, ou le nom n'est pas parlant ou bien encore le nom du module est trop long, vous pouvez renommer le nom du module (un alias). La syntaxe est la suivante :

```
import nom_du_module as alias
```

Exemple :

```
import random as aleatoire  
nombre = aleatoire.randint(1,100)
```

Une autre technique existe permettant d'importer toutes les fonctions et variables d'un module directement dans l'environnement du programme. Cette technique est fortement déconseillée car il peut y avoir des conflits de noms avec d'autres modules appelés voir des variables que vous créez dans vos scripts. La syntaxe (non recommandé) est :

```
from nom_du_module import *
```

Exemple :

```
from random import *
```



## B. Module random

Dans le module random, on va retrouver plusieurs fonctions dont le détail est décrit sur la documentation de Python (<https://docs.python.org/3/library/random.html>).

Dans le cas des exemples suivants on considère que l'on a appelé la librairie random :

```
import random
```

Les fonctions couramment utilisées sont :

```
random.random()
```

Cette fonction renvoie un nombre flottant (décimal - float) aléatoire compris dans l'intervalle [0.0 ; 1.0[

Par exemple, la sortie de la fonction peut avoir comme valeur :

```
0.6778704449511866  
0.4913461834706535  
0.08183475853616906
```

```
random.randint(a,b)
```

Cette fonction renvoie un nombre entier (int) aléatoire compris entre a et b. Les deux bornes sont incluses.

Exemple :

```
random.randint(1,6)
```

On pourrait avoir en sortie :

```
3  
5  
4  
6  
2  
1  
4
```



**random.choice(sequence)**

Cette fonction permet de renvoyer un élément aléatoire d'une séquence non vide.

Exemple :

```
couleurs = ["rouge", "vert", "bleu", "orange", "jaune", "rose"]
```

```
random.choice(couleurs)
```

On pourrait avoir en sortie :

```
orange  
bleu  
rose  
jaune  
orange  
rouge  
vert  
jaune
```

**random.shuffle(list)**

Cette fonction permet de mélanger aléatoirement les éléments d'une liste.

Exemple :

```
cartes = ["As", "Roi", "Dame", "Cavalier", "Valet"]
```

```
print("Mes cartes actuelles :", cartes)
```

```
random.shuffle(cartes)
```

```
print("Mes cartes mélangées :", cartes)
```

On pourrait avoir en sortie :

```
Mes cartes actuelles : ['As', 'Roi', 'Dame', 'Cavalier', 'Valet']
```

```
Mes cartes mélangées : ['Roi', 'Cavalier', 'As', 'Valet', 'Dame']
```



**random.uniform(a,b)**

Cette fonction renvoie un nombre flottant (décimal) aléatoire compris entre a et b.

Exemple :

```
nombre = random.uniform(4.5 , 7.9)
```

On pourrait avoir en sortie :

4.865762003923171

6.527709217104741

5.925994939338326

5.5302431221667065

4.807992874636737

**random.randrange(start, stop, step)**

Cette fonction renvoie un élément sélectionné aléatoirement depuis range(start,stop,step). C'est un équivalent aléatoire à la fonction range() vu dans un module précédent.

Exemple :

```
nombre = random.randrange(0,10,2)
```

On pourrait avoir en sortie :

8

6

8

8

2

0

**! IMPORTANT !**

Les nombres générés par le module random sont des nombres pseudo-aléatoires et non des nombres réellement aléatoires. Cela signifie qu'ils sont générés par un algorithme déterministe, basé sur une « graine » (seed). Par défaut, cette graine est souvent basée sur l'heure système ce qui donne l'impression d'aléatoire.

Pour des applications nécessitant une très haute sécurité ou un véritable aléatoire (cryptographie par exemple), il faudrait utiliser des sources d'aléatoires cryptographiquement sûres, généralement fournies par le module os.urandom ou des services externes. Néanmoins, pour la plupart des cas d'utilisation courants, random est largement suffisant.



## C. Récapitulatif

<code>random.random()</code>	Nombre décimal entre 0.0 et 1.0
<code>random.randint(a,b)</code>	Entier entre a et b inclus
<code>random.choice(liste)</code>	Élément aléatoire d'une liste
<code>random.shuffle(liste)</code>	Mélange la liste
<code>random.uniform(a,b)</code>	Décimal entre a et b
<code>random.randrange(...)</code>	Entier pris dans une plage

## D. Erreurs fréquentes

Oublier l'import :

```
nombre = random.randint(1, 6)
```

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

Confondre randint() et randrange() :

```
random.randint(1, 6) peut donner 6
```

```
random.randrange(1, 6) ne peut pas donner 6
```

Utiliser shuffle() comme si elle renvoyait une nouvelle liste :

```
cartes = ["As", "Roi", "Dame"]  
melange = random.shuffle(cartes)  
print(melange)
```

Résultat :

```
None
```

Utiliser choice() sur une liste vide :

```
random.choice([])
```

```
IndexError: Cannot choose from an empty sequence
```



## E. A retenir

un module doit être importé avant d'être utilisé  
import random est la méthode la plus simple et lisible  
random.randint(a,b) inclut les deux bornes  
random.randrange(start,stop) n'inclut pas stop  
random.shuffle() modifie directement la liste  
les nombres générés sont pseudo-aléatoires  
random n'est pas adapté à la cryptographie

## F. Synthèse

