random module

January 25, 2023

1 random module

- Used to generate random values

1.1 random.randint()

- Generates a random integer in the given range a and b (inclusive)

[73]:
```
import random
print(random.randint(1, 10))
```

```
10
```

[49]:
```
import random
print(random.randint(-476, 748))
```

```
305
```

[86]:
```
# Generating a list of random integers
for i in range(10):
    print(random.randint(1, 1000), end = ' ')
```

```
71 722 729 862 233 836 340 931 395 503
```

[99]:
```
# Generating a list of random integers
lst = []
for i in range(10):
    lst.append(random.randint(1, 1000))
print(lst)
```

```
[868, 6, 259, 773, 251, 389, 914, 270, 225, 657]
```

[106]:
```
# Generating a list of random integers
lst = []
for i in range(random.randint(1, 20)):
    lst.append(random.randint(1, 1000))
print(lst)
```

```
[850, 139, 385, 120, 302]
```

[112]:

```
# Generating a list of random integers
lst = [random.randint(1, 1000) for i in range(10)]
print(lst)
```

```
[721, 97, 862, 573, 86, 174, 713, 653, 938, 454]
```

[139]:

```
# Generating a list of lists using random
marks = []
for i in range(5):
    lst = []
    for j in range(5):
        lst.append(random.randint(0, 100))
    marks.append(lst)
marks
```

[139]:

```
[[46, 51, 34, 87, 21],
 [3, 86, 90, 27, 95],
 [44, 73, 27, 15, 0],
 [77, 69, 73, 84, 64],
 [9, 83, 79, 71, 15]]
```

[148]:

```
# Generating a list of lists using random
x = [[random.randint(1, 100) for i in range(5)] for j in range(5)]
x
```

[148]:

```
[[75, 89, 37, 91, 32],
 [65, 72, 31, 27, 62],
 [65, 45, 39, 82, 97],
 [65, 12, 81, 90, 98],
 [15, 25, 62, 17, 56]]
```

1.2 random.choice()

- To get a single random value from an iterable

[166]:

```
x = [24, 21, 1, 42, 59, 78, 92, 20, 45, 9, 9, 7, 44, 11, 94]
print(random.choice(x))
```

```
20
```

[173]:

```
print(random.choice(range(5, 100, 5)))
```

```
15
```

[185]:

```
print(random.choice('hello world'))
```

```
l
```

1.2.1 random.randrange()

- Produces a random integer in the given range

- You can specify the step for range

[237]:
```
print(random.randrange(7, 100, 7))
```

```
56
```

1.3 random.sample()

- Generates a list of k values from the given iterable
- Takes two arguments
  - an iterable
  - an integer

[238]:
```
x = [24, 21, 1, 42, 59, 78, 92, 20, 45, 9, 9, 7, 44, 11, 94]
print(random.sample(x, 4))
```

```
[42, 94, 21, 7]
```

[239]:
```
z = [24, 21, 1, 42, 59]
print(random.sample(z, 20))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [239], in <cell line: 2>()
      1 z = [24, 21, 1, 42, 59]
----> 2 print(random.sample(z, 20))

File C:\Program Files\Python310\lib\random.py:482, in Random.sample(self,␣
 ↪population, k, counts)
    480 randbelow = self._randbelow
    481 if not 0 <= k <= n:
--> 482     raise ValueError("Sample larger than population or is negative")
    483 result = [None] * k
    484 setsize = 21        # size of a small set minus size of an empty list

ValueError: Sample larger than population or is negative
```

[250]:
```
lst = random.sample('this is python', 5)
print(lst)
```

```
['s', 't', 't', ' ', 'i']
```

1.4 random.shuffle()

- Used to shuffle a list of values in-place

[264]:
```
x = list(range(1, 20))
print(x)
random.shuffle(x)
```

```
print(x)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
[18, 11, 3, 16, 7, 1, 19, 8, 9, 2, 15, 10, 5, 13, 4, 14, 12, 17, 6]
```

1.5 Password Generator

[435]:

```
import string
import random
def get_password(lst, length=8):
    pwd = random.sample(lst, length)
    return "".join(pwd)

letters = string.ascii_letters + string.digits + '!@#$%^&*'
x = list(letters)
random.shuffle(x)
print(get_password(x, 10))
```

```
vCoN2sahx@
```

2 Hello