import random

**# random()**  
print(random.random())  
# 0.851771074527212

# **uniform()** - random floating point number between 1 and 10  
print(random.uniform(1, 10))

# **randint()** - random integers between 1 and 10  
print(random.randint(1, 6))

# **choice()** - pick a random value from list of values  
names = ["John", "Klary", "Hyphy"]  
print("Hello "+random.choice(names))

# **choices()** - print a list of 5 random results from colors list, **not** **unique**  
colors = ["Red", "Black", "Green"]  
print(random.choices(colors, k=5))  
# ['Green', 'Black', 'Black', 'Black', 'Red']

# adding **weights** for items of your list  
print(random.choices(colors, weights=[18, 18, 2], k=5))

# **shuffle()** - Shuffle a range of 52 cards  
deck = list(range(1, 53))  
random.shuffle(deck)

# **sample(arg1, arg2)** - Get 5 random **unique** decks from that deck

# **sample() vs choices()** - didn’t use choices() because **choices()** can randomly select same card multiple times, but **sample()** method randomly select decks uniquely.  
hand = random.sample(deck, k=5)  
print(hand)

**# e.g.**  
first\_names = ['John', 'Jane', 'Corey', 'Travis', 'Dave', 'Kurt']  
last\_names = ['Smith', 'Doe', 'Jenkins', 'Robinson', 'Davis']  
street\_names = ['Main', 'High', 'Pearl', 'Maple', 'Park', 'Oak', 'Pine']  
fake\_cities = ['Metropolis', 'Eerie', "King's Landing"]  
states = ['AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL']  
  
for num in range(100):  
 first = random.choice(first\_names)  
 last = random.choice(last\_names)  
  
 phone = f'{random.randint(100, 999)}-555-{random.randint(1000,9999)}'  
  
 street\_num = random.randint(100, 999)  
 street = random.choice(street\_names)  
 city = random.choice(fake\_cities)  
 state = random.choice(states)  
 zip\_code = random.randint(10000, 99999)  
 address = f'{street\_num} {street} St., {city} {state} {zip\_code}'  
  
 email = first.lower() + last.lower() + '@bogusemail.com'  
  
 print(f'{first} {last}\n{phone}\n{address}\n{email}\n')

# **Socratica**

# **TASK -** Generate integer random numbers from from interval [3, 7)

***# THE HARD WAY***

import random, math

def my\_random():

return math.floor(10 \* random.random())

for n in range(10):

print(my\_random())

***# THE EASY WAY***

for n in range(10):*- random.randint()*

print(random.randint(3, 7))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **TASK -** Generate integer random numbers from from interval [3, 7)

***# THE HARD WAY***

import random, math

def my\_random():

return 10 \* random.random()

for n in range(10):

print(my\_random())

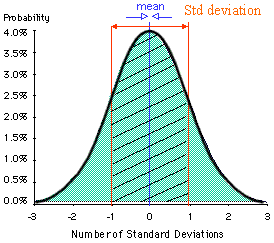
***# THE EASY WAY*** *- random.uniform()*

for n in range(10):

print(random.uniform(3, 7))

# **normalvariation(mean, standardDeviation)** - to generate random numbers from a normal distribution.

# **standard deviation** - defines how wide you can spread out the numbers around the **mean**. If **standard deviation value** is large, the numbers around **mean value** would spread out, if the **standard deviation value** is small, the numbers around **mean value** would be tightly grouped.



# eg.01

for n in range(5):

print(random.normalvariate(0, 1)) # mean is zero

# 0.8820232173753841

# 1.9017970168352376

# 0.6664332760186771

# 1.5824931668758215

# -1.3433966771887738

# Notice how random numbers are bunched around the mean 0

# eg.02

for n in range(5):

print(random.normalvariate(5, 1)) # mean is five, standardDeviation is 1

# 5.047405200266482

# 4.947938890286223

# 5.055621864912361

# 3.4092553787726754

# 4.654417283393494

# Notice how random numbers are bunch around the mean 5