**Use the all() function to determine if a person has a required set of skills.**

**Exercise**

Use the **all()** function to determine if a person has a required set of skills.

**Sample Function Definition**

1. def has\_required\_skills(person, skills):

The input parameters are:

1. A dictionary **person** with the following keys:
   * **name**: a string representing the person's name (e.g. "John Doe")
   * **age**: an integer representing the person's age (e.g. 30)
   * **skills**: a list of strings representing the person's skills (e.g. **['Python', 'JavaScript', 'C++']**)
2. A list of strings **skills** representing the required skills for a certain task (e.g. **['Python', 'JavaScript']**)

**Output**

The function must return a Boolean value indicating whether the person has all the required skills.

**Solution below:**

----------------

**SOLUTION:**

1. def has\_required\_skills(person, skills):
2. return all(skill in person['skills'] for skill in skills)
4. john = {
5. 'name': 'John Doe',
6. 'age': 30,
7. 'skills': ['Python', 'JavaScript', 'C++']
8. }
10. jane = {
11. 'name': 'Jane Smith',
12. 'age': 25,
13. 'skills': ['Python', 'Java']
14. }
16. required\_skills = ['Python', 'JavaScript']
18. print(has\_required\_skills(john, required\_skills)) # True
19. print(has\_required\_skills(jane, required\_skills)) # False

In this example, the **has\_required\_skills()** function is used to check if the person **john** has the required skills **['Python', 'JavaScript']**. Since **john** has both of these skills, the function returns **True**. For the person **jane**, the function returns **False** because she does not have the required skill **JavaScript**.