

Python Interview Questions

Want to become an expert in cracking Python interview questions?

Start with practicing the questions below. Whether a question involves multiple choice or live coding, we will give you hints as you go and tell you if your answers are correct or incorrect.

After that, take our timed public Python Interview Questions Test (/tests/start-challenge?generatorUrl=python-online-test&difficultyHard=False&backUrlSkill=9).

To use our service for testing candidates, buy a pack of candidates (/pricing).

1. File Owners

PYTHON ()

LANGUAGE ()

PUBLIC ()

Implement a `group_by_owners` function that:

- Accepts a dictionary containing the file owner name for each file name.
- Returns a dictionary containing a list of file names for each owner name, in any order.

For example, for dictionary `{'Input.txt': 'Randy', 'Code.py': 'Stan', 'Output.txt': 'Randy'}` the `group_by_owners` function should return `{'Randy': ['Input.txt', 'Output.txt'], 'Stan': ['Code.py']}`.

Difficulty

Easy  ()

Time

10min

Python 3.6.5 ?

[Copy to IDE](#)

[Show starting code \(/questions/original/11846\)](/questions/original/11846)

```
1 class FileOwners:
2
3     @staticmethod
4     def group_by_owners(files):
5         # vinay26k.github.io
6         owner_ = {}
7         for file, owner in files.items():
8             if owner in owner_:
9                 owner_[owner].append(file)
10            else:
11                owner_[owner]=[file]
12        return owner_
13
14 files = {
15     'Input.txt': 'Randy',
16     'Code.py': 'Stan',
17     'Output.txt': 'Randy'
18 }
19 print(FileOwners.group_by_owners(files))
```

Run

Show Hint

Hints

Output

Tests: 3 pass / 0 fail

Run OK

```
{'Randy': ['Input.txt', 'Output.txt'], 'Stan': ['Code.py']}
```

Report an issue (<mailto:support@testdome.com?subject=Report question: https://www.testdome.com/d/python-interview-questions/9, QuestionId: 11846>)
Your score is 100%, perfect!

2. Palindrome

PYTHON ()

STRINGS ()

PUBLIC ()

A palindrome is a word that reads the same backward or forward.

Write a function that checks if a given word is a palindrome. Character case should be ignored.

For example, `is_palindrome("Develed")` should return `True` as character case should be ignored, resulting in "develed", which is a palindrome since it reads the same backward and forward.

DifficultyEasy  ()**Time**

10min

Python 3.6.5 ?

[Copy to IDE](#)[Show starting code \(/questions/original/7962\)](/questions/original/7962)

```
1 class Palindrome:
2
3     @staticmethod
4     def is_palindrome(word):
5         # vinay26k.github.io
6         return word.lower() == word.lower()[::-1]
7
8 print(Palindrome.is_palindrome('Develed'))
```

[Run](#)[Show Hint](#)[Hints](#)[Output](#)[Tests: 3 pass / 0 fail](#)

Run OK

True

Report an issue (<mailto:support@testdome.com?subject=Report question: https://www.testdome.com/d/python-interview-questions/9, QuestionId: 7962>)
Your score is 100%, perfect!

3. Playlist

[PYTHON](#) ()[ALGORITHMIC THINKING](#) ()[SEARCHING](#) ()[PUBLIC](#) ()[NEW](#) ()

A playlist is considered a repeating playlist if any of the songs contain a reference to a previous song in the playlist. Otherwise, the playlist will end with the last song which points to *None*.

Implement a function *is_repeating_playlist* that returns true if a playlist is repeating or false if it is not.

For example, the following code prints "True" as both songs point to each other.

```
first = Song("Hello")
second = Song("Eye of the tiger")

first.next_song(second);
second.next_song(first);


print(first.is_repeating_playlist())
```

Difficulty

Easy  ()

Time

15min

Python 3.6.5 

[Copy to IDE](#)

[Show starting code \(/questions/original/17253\)](/questions/original/17253)

```
9  def is_repeating_playlist(self):
10     """
11     :returns: (bool) True if the playlist is repeating, False if
12     """
13     # vinav26k.github.io
```

```

14     songs_in_playlist = set()
15     current_song = self
16     while(current_song):
17         if current_song.name in songs_in_playlist: # if we already have this song
18             return True
19         songs_in_playlist.add(current_song.name)
20         current_song = current_song.next
21     return False
22
23 first = Song("Hello")
24 second = Song("Eye of the tiger")
25
26 first.next_song(second);
27 second.next_song(first);
28

```

Run

Hints

Output

Tests: 4 pass / 0 fail

Hint 1: A data structure can be used to identify if a song appears twice in a playlist.

Report an issue (<mailto:support@testdome.com?subject=Report question: https://www.testdome.com/d/python-interview-questions/9, QuestionId: 17253>)
Your score is 100%, perfect!

4. Binary Search Tree

PYTHON ()

ALGORITHMIC THINKING ()

DATA STRUCTURES ()

PUBLIC ()

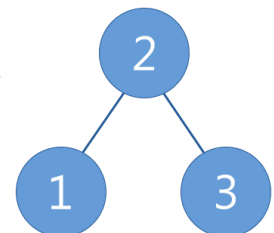
NEW ()

Binary search tree (BST) is a binary tree where the value of each node is larger or equal to the values in all the nodes in that node's left subtree and is smaller than the values in all the nodes in that node's right subtree.

Write a function that checks if a given binary search tree contains a given value.

For example, for the following tree:

- n1 (Value: 1, Left: null, Right: null)
- n2 (Value: 2, Left: n1, Right: n3)



- n3 (Value: 3, Left: null, Right: null)


Call to `contains(n2, 3)` should return `True` since a tree with root at n2 contains number 3.

Difficulty

Easy  ()

Time

20min

Python 3.6.5 

[Copy to IDE](#)

[Show starting code \(/questions/original/14288\)](/questions/original/14288)

```
0
7
8   @staticmethod
9   def contains(root, value):
10       # vinay26k.github.io
11       if root is None:
12           return False
13       if root.value == value:
14           return True
15       elif root.value < value:
16           if root.right:
17               return BinarySearchTree.contains(root.right, value)
18       elif root.value > value:
19           if root.left:
20               return BinarySearchTree.contains(root.left, value)
21       return False
22
23 n1 = BinarySearchTree.Node(value=1, left=None, right=None)
24 n3 = BinarySearchTree.Node(value=3, left=None, right=None)
25 n2 = BinarySearchTree.Node(value=2, left=n1, right=n3)
26
```

Run

Show Hint

Hints

Output

Tests: 3 pass / 0 fail

Run OK

True

Report an issue (<mailto:support@testdome.com?subject=Report question>:
<https://www.testdome.com/d/python-interview-questions/9>, QuestionId: 14288)
Your score is 100%, perfect!

5. Two Sum

PYTHON ()

ALGORITHMIC THINKING ()

SEARCHING ()

PUBLIC ()

NEW ()

Write a function that, when passed a list and a target sum, returns two distinct zero-based indices of any two of the numbers, whose sum is equal to the target sum. If there are no two numbers, the function should return *None*.

For example, `find_two_sum([3, 1, 5, 7, 5, 9], 10)` should return a single *tuple* containing any of the following pairs of indices:


- 0 and 3 (or 3 and 0) as $3 + 7 = 10$
- 1 and 5 (or 5 and 1) as $1 + 9 = 10$
- 2 and 4 (or 4 and 2) as $5 + 5 = 10$

Difficulty

Easy  ()

Time

30min

Python 3.6.5 

[Copy to IDE](#)

[Show starting code \(/questions/original/16305\)](#)

```
1 class TwoSum:
2
3     @staticmethod
4     def find_two_sum(numbers, target_sum):
5         """
6         :param numbers: (list of ints) The list of numbers.
7         :param target_sum: (int) The required target sum.
8         :returns: (a tuple of 2 ints) The indices of the two element
9         """
10        # vinay26k.github.io
11        if len(numbers) <= 1:
12            return False
13        buff_dict = {}
14        for i in range(len(numbers)):
15            if numbers[i] in buff_dict:
16                return (buff_dict[numbers[i]], i)
17            else:
18                buff_dict[target_sum - numbers[i]] = i
19
20    print(TwoSum.find_two_sum([3, 1, 5, 7, 5, 9], 10))
```

Run

Show Hint

Hints

Output

Tests: 4 pass / 0 fail

Run OK

(0, 3)

Report an issue (<mailto:support@testdome.com?subject=Report question: https://www.testdome.com/d/python-interview-questions/9, QuestionId: 16305>)

Your score is 100%, perfect!

6. League Table

PYTHON ()

COLLECTIONS ()

SORTING ()

PUBLIC ()

NEW ()

The *LeagueTable* class tracks the score of each player in a league. After each game, the player records their score with the *record_result* function.

The player's rank in the league is calculated using the following logic:

1. The player with the highest score is ranked first (rank 1). The player with the lowest score is ranked last.
2. If two players are tied on score, then the player who has played the fewest games is ranked higher.
3. If two players are tied on score and number of games played, then the player who was first in the list of players is ranked higher.


Implement the *player_rank* function that returns the player at the given rank.

For example:

```
table = LeagueTable(['Mike', 'Chris', 'Arnold'])
table.record_result('Mike', 2)
table.record_result('Mike', 3)
table.record_result('Arnold', 5)
table.record_result('Chris', 5)
print(table.player_rank(1))
```

All players have the same score. However, Arnold and Chris have played fewer games than Mike, and as Chris is before Arnold in the list of players, he is ranked first. Therefore, the code above should display "Chris".

Difficulty

Hard  ()

Time

20min

Python 3.6.5 ?

[Copy to IDE](#)

[Show starting code \(/questions/original/11195\)](/questions/original/11195)

```
4 class LeagueTable:
5     def __init__(self, players):
6         self standings = OrderedDict([(player, Counter()) for player in players])
7
8     def record_result(self, player, score):
9         self standings[player]['games_played'] += 1
10        self standings[player]['score'] += score
11
12    def player_rank(self, rank):
13        # vinay26k.github.io
14        ranks = []
15        for player in self standings:
16            ranks.append((player, self standings[player]['games_played'], self standings[player]['score']))
```

```

17         return sorted(ranks, key=lambda x: (-x[2], x[1]))[rank-1][0]
18
19 table = LeagueTable(['Mike', 'Chris', 'Arnold'])
20 table.record_result('Mike', 2)
21 table.record_result('Mike', 3)
22 table.record_result('Arnold', 5)
23 table.record_result('Chris', 5)

```

Run

Show Hint

Hints

Output

Tests: 4 pass / 0 fail

Run OK

Chris

Report an issue (<mailto:support@testdome.com?subject=Report question: https://www.testdome.com/d/python-interview-questions/9, QuestionId: 11195>)
Your score is 100%, perfect!

7. Path

PYTHON ()

DATA STRUCTURES ()

STRINGS ()

PUBLIC ()

Write a function that provides change directory (cd) function for an abstract file system.

Notes:

- Root path is '/'.
- Path separator is '/'.
- Parent directory is addressable as '..'.
- Directory names consist only of English alphabet letters (A-Z and a-z).
- The function should support both relative and absolute paths.
- The function will not be passed any invalid paths.
- Do not use built-in path-related functions.

For example:

```

path = Path('/a/b/c/d')
path.cd('../x')
print(path.current_path)

```



should display '/a/b/c/x'.

Difficulty

Hard ()

Time

30min

Python 3.6.5 

[Copy to IDE](#)

[Show starting code \(/questions/original/12282\)](/questions/original/12282)

```
1 ▼ class Path:
2 ▼     def __init__(self, path):
3         self.current_path = path
4
5 ▼     def cd(self, new_path):
6         # vinay26k.github.io
7         go_back_count = new_path.split('/').count('..')
8 ▼         if go_back_count:
9             self.current_path = '/'.join(self.current_path.split('/')
10 ▼         else:
11             self.current_path += '/' + new_path
12
13
14 path = Path('/a/b/c/d')
15 path.cd('../x')
16 print(path.current_path)
```

Run

Show Hint

Hints

Output

Tests: 4 pass / 0 fail

Run OK

/a/b/c/x

Report an issue (<mailto:support@testdome.com?subject=Report question: https://www.testdome.com/d/python-interview-questions/9, QuestionId: 12282>)
Your score is 100%, perfect!

If you feel ready, take one of our timed public Python Interview Questions tests:

- [HTML/CSS, JavaScript, Python Algorithms, and SQL Online Test \(Easy \(/tests/start-challenge?generatorUrl=html-css-javascript-sql-python-algorithms-online-test&difficultyHard=False&backUrlSkill=9\)\)](/tests/start-challenge?generatorUrl=html-css-javascript-sql-python-algorithms-online-test&difficultyHard=False&backUrlSkill=9)))
- [HTML/CSS, JavaScript, Python, and SQL Online Test \(Easy \(/tests/start-challenge?generatorUrl=html-css-javascript-python-sql-online-test&difficultyHard=False&backUrlSkill=9\) / Hard \(/tests/start-challenge?generatorUrl=html-css-javascript-python-sql-online-test&difficultyHard=True&backUrlSkill=9\)\)](/tests/start-challenge?generatorUrl=html-css-javascript-python-sql-online-test&difficultyHard=False&backUrlSkill=9) / Hard (/tests/start-challenge?generatorUrl=html-css-javascript-python-sql-online-test&difficultyHard=True&backUrlSkill=9)))
- [Python Online Test \(Easy \(/tests/start-challenge?generatorUrl=python-online-test&difficultyHard=False&backUrlSkill=9\)\)](/tests/start-challenge?generatorUrl=python-online-test&difficultyHard=False&backUrlSkill=9)))
- [Python Algorithms Online Test \(Easy \(/tests/start-challenge?generatorUrl=python-algorithms-online-test&difficultyHard=False&backUrlSkill=9\)\)](/tests/start-challenge?generatorUrl=python-algorithms-online-test&difficultyHard=False&backUrlSkill=9)))
- [Python Algorithms and SQL Online Test \(Easy \(/tests/start-challenge?generatorUrl=sql-python-algorithms-online-test&difficultyHard=False&backUrlSkill=9\)\)](/tests/start-challenge?generatorUrl=sql-python-algorithms-online-test&difficultyHard=False&backUrlSkill=9)))
- [Python and SQL Online Test \(Easy \(/tests/start-challenge?generatorUrl=python-sql-online-test&difficultyHard=False&backUrlSkill=9\) / Hard \(/tests/start-challenge?generatorUrl=python-sql-online-test&difficultyHard=True&backUrlSkill=9\)\)](/tests/start-challenge?generatorUrl=python-sql-online-test&difficultyHard=False&backUrlSkill=9) / Hard (/tests/start-challenge?generatorUrl=python-sql-online-test&difficultyHard=True&backUrlSkill=9)))

Not exactly what you are looking for? Go to our [For Jobseekers \(/for-jobseekers\)](/for-jobseekers) section.

EXPLORE

[Home](#)
 [\(https://www.testdome.com\)](https://www.testdome.com)
[Tour \(/tour\)](/tour)
[Tests \(/tests\)](/tests)
[Questions \(/questions\)](/questions)
[Pricing \(/pricing\)](/pricing)
[For Jobseekers ▾](#)
[Sign in \(/sign-in\)](/sign-in)

CONTACT

[@ support@testdome.com \(mailto:support@testdome.com\)](mailto:support@testdome.com)
[@TestDome \(https://twitter.com/testdome\)](https://twitter.com/testdome)
[TestDome+](https://plus.google.com/109021510536856899050)
 [\(https://plus.google.com/109021510536856899050\)](https://plus.google.com/109021510536856899050)
[☎ +44-20-8938-3088](tel:+442089383088)
[✉ Mailing address:](#)
 Gemmeus Ltd
 Kemp House
 152 City Road
 London EC1V 2NX
 United Kingdom

LEGAL

[Terms of Use \(/legal/terms-of-use\)](/legal/terms-of-use)
[Privacy Policy \(/legal/privacy-policy\)](/legal/privacy-policy)
 © TestDome

[Back to Top](#)

