Python Bootcamp {2019}

Kirirom Institute of Technology

PROJECTS WEEK 2

PROJECT 04 : Top Words

Description :

For this project, you will write a function, that take a string (text) as parameter and will return the top 3 most occuring words in a list. Some words like (aren’t) contains characters and must be handle by your function. The punctuations mark and special characters must not be taking in consideration, you must remove them (example: Hello! will become hello). Finally, if there are not enough unique words to generate top3 you will generate top2 or top1. If the text is empty you will return an empty list.

Requirements :

- Program must be named : **04_top_words.py** and saved into **week02/projects**
- Your function must be called **top_words(text)**

- Your function must be case-insensitive and your result have to be lowercase (“Hello, hello, Hello!”) ⇒ “hello” (3 times)

- If you have more than 3 words that occurs equally, you will have to return a list that respect the order of the first occurrences.
  Example: “hi hello hi wow wow hello good good ” will return:
  ==> [“hi”, “hello”, “wow”] because ‘hi’ ‘hello’ and ‘wow’ are the first 3 words to appears in the text (even if ‘good’ appears also 2 times)

Hint :

- ❖ Review your week 01 and week 02 exercises, they might help you :)
- ❖ **Generate your own texts and make a LOT of test before submission!**
- ❖ **Take time to read the description / requirements and output AGAIN**

Kirirom Institute of Technology

Output :

```
top_words("Welcome to Kirirom: Kirirom Institute of Technology and Kirirom
National Parc. To contact us, send a message to us!"

>> ["to", "kirirom", "us"]

top_words("//can't cant Cant can't")

>> ["can't", 'cant']

top_words("hello hello Hello")

>> ["hello"]

top_words(". ??? // ####### // // ???")

>> []

top_words("")

>> []
```

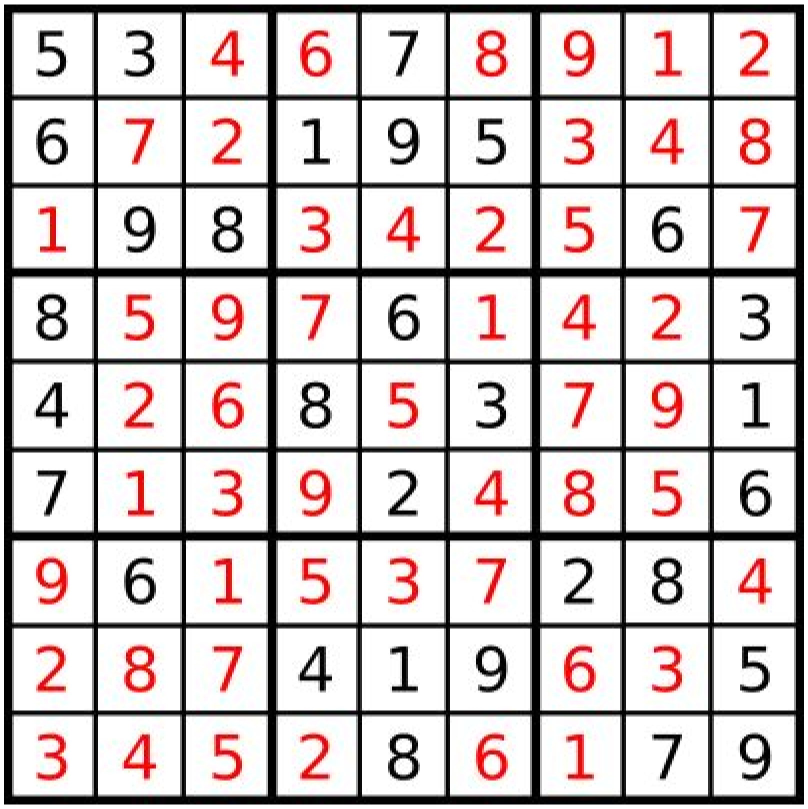
Python Bootcamp {2019}

Kirirom Institute of Technology

PROJECT 05 : Sudoku Validator

Description :

Sudoku (数独 sūdoku) (sometimes spelled as Su Doku, but also called Number Place or Nanpure) is a puzzle that is very popular in Japan. It was created in Indianapolis in 1979 by Howard Garns and it appeared in Dell Magazines afterwards.



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Python Bootcamp {2019}

You will write a function that take a two-dimensional lists (array) as parameter that represent a Sudoku. You function can return three different possibilities:

- **“Invalid Format”** : If the sudoku format is not valid (to be valid, a Sudoku board have to be sized 9x9 and contain number between 1 and 9 only)

- **“Not finished”** : If the sudoku format is valid BUT not finished (for example all the cases contains values and the size of the board is 9x9 but a line, columns or region is incorrect (for example 2 identical numbers on the same line/columns/region).

- **“Finished”** : If the sudoku format is valid AND finished without any mistakes.

More about sudoku : https://simple.wikipedia.org/wiki/Sudoku
Sudoku generator : https://www.sudokuweb.org/

Requirements :

- Program must be named : **05_sudoku.py** and saved into **week02/projects**
- Your function must be called **sudoku(board)**

Hint :

- ❖ Two-dimensional list
- ❖ Make sure you know how to play Sudoku, if not, take some time to try, it’s à good game for your brain!
- ❖ Try to write your program with ‘human logic’. How would you do to check if the sudoku is valid first, and then if it’s finished or not?
- ❖ This project doesn’t require any maths skills and formulas
- ❖ Take your time to create your function and have fun doing it.

Output :

```
sudoku([     [5, 3, 4, 6, 7, 8, 9, 1, 2],
             [6, 7, 2, 1, 9, 5, 3, 4, 8],
             [1, 9, 8, 3, 4, 2, 5, 6, 7],
             [8, 5, 9, 7, 6, 1, 4, 2, 3],
             [4, 2, 6, 8, 5, 3, 7, 9, 1],
             [7, 1, 3, 9, 2, 4, 8, 5, 6],
             [9, 6, 1, 5, 3, 7, 2, 8, 4],
             [2, 8, 7, 4, 1, 9, 6, 3, 5],
             [3, 4, 5, 2, 8, 6, 1, 7, 9]])

>> Finished

sudoku([     [1, 1, 3, 5, 7, 9, 4, 6, 8],
             [4, 9, 1, 2, 6, 1, 3, 7, 5],
             [7, 5, 6, 1, 8, 4, 2, 1, 9],
             [6, 4, 4, 1, 5, 8, 7, 9, 2],
             [5, 2, 1, 7, 1, 1, 1, 1, 1],
             [9, 8, 7, 4, 2, 6, 5, 3, 1],
             [2, 1, 4, 9, 3, 5, 2, 2, 7],
             [3, 6, 5, 8, 1, 7, 9, 2, 4],
             [8, 7, 9, 6, 4, 2, 1, 5, 3]])

>> Not Finished

sudoku([     [0, 12, 3, 5, 7, 9, 4, 6, 8],
             [4, 9, 1, 2, 3, 7, 5],
             [7, 5, 6, 1, 8, 4, 2, 1, 9],
             [6, 4, 4, 1, 5, 8, 7, 9, 2],
             [5, 2, 1, 7, 1, 1, 1, 1, 1],
             [9, 8, 7, 4, 2, 6, 5, 3, 1],
             [2, 1, 4, 9, 3, 5, 2, 2, 7],
             [3, 6, 5, 8, 1, 7, 9, 2, 4],
             [8, 7, 9, 6, 4, 2, 1, 5, 3]])

>> Invalid Format

sudoku([])

>> Invalid Format
```

Python Bootcamp {2019}

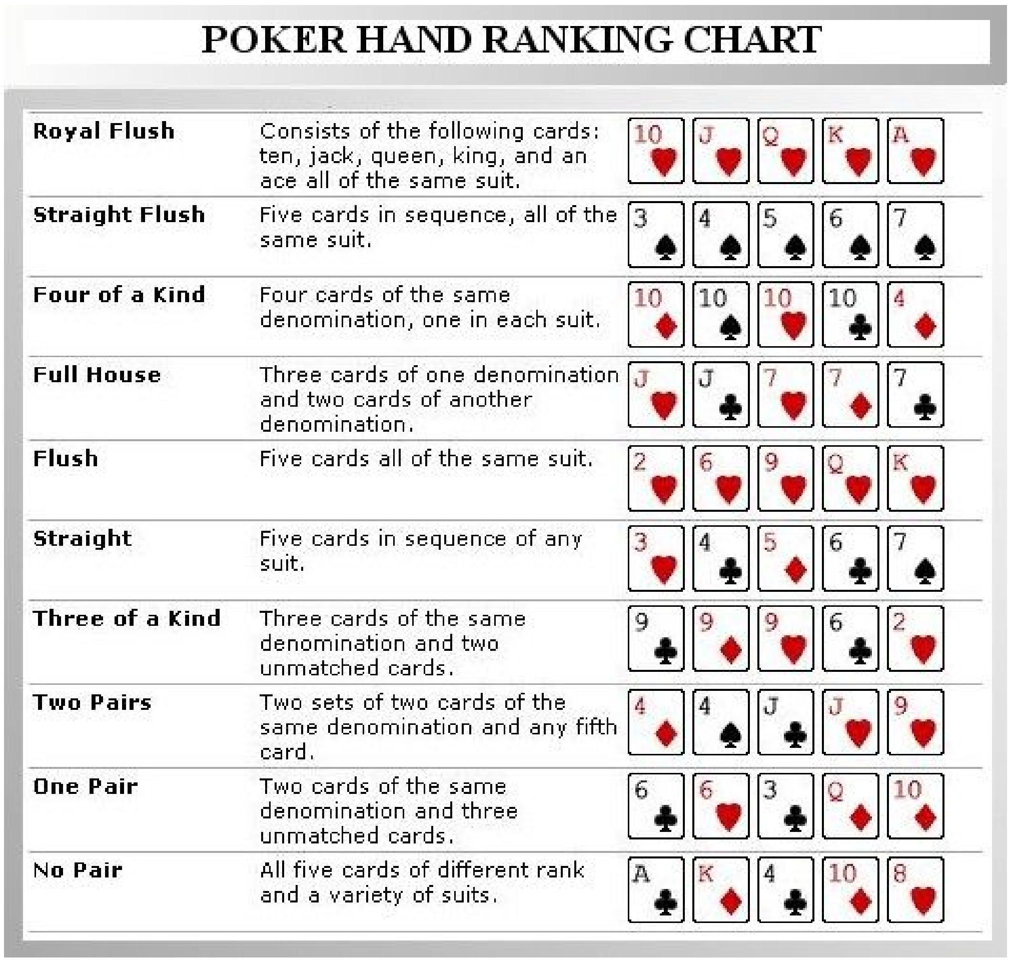
Kirirom Institute of Technology

PROJECT 06 : Best Poker Hand

Description :

In this program, you will write a function that take two strings (player_one and player_two) in parameter that will represent two poker player hands and then return the result as a string:

- **Player One WIN** (if the player_one win)
- **Player Two WIN** (if the player_two win)
- **Tie** (if the result is equal)



POKER HAND RANKING CHART

Royal Flush	Consists of the following cards: ten, jack, queen, king, and an ace all of the same suit.	10♥ J♥ Q♥ K♥ A♥
Straight Flush	Five cards in sequence, all of the same suit.	3♠ 4♠ 5♠ 6♠ 7♠
Four of a Kind	Four cards of the same denomination, one in each suit.	10♦ 10♠ 10♥ 10♣ 4♦
Full House	Three cards of one denomination and two cards of another denomination.	J♥ J♣ 7♥ 7♦ 7♣
Flush	Five cards all of the same suit.	2♥ 6♥ 9♥ Q♥ K♥
Straight	Five cards in sequence of any suit.	3♥ 4♣ 5♦ 6♣ 7♠
Three of a Kind	Three cards of the same denomination and two unmatched cards.	9♣ 9♦ 9♥ 6♣ 2♥
Two Pairs	Two sets of two cards of the same denomination and any fifth card.	4♦ 4♠ J♣ J♥ 9♥
One Pair	Two cards of the same denomination and three unmatched cards.	6♣ 6♥ 3♣ Q♦ 10♦
No Pair	All five cards of different rank and a variety of suits.	A♣ K♦ 4♣ 10♦ 8♥

The first character is the value of the card:
2, 3, 4, 5, 6, 7, 8, 9, T(en), J(ack), Q(ueen), K(ing), A(ce)

The second character represents the suit:
S(pades), H(earts), D(iamonds), C(lubs)

Each card will be separated with a space
The order of the card in the hands does not matter

Example of hand: **“AH KH QH JH TH”**

This hand correspond to Ace, King, Queen, Jack and Ten of HEART.
This is a royal flush. (The best hand in Poker)

If the two players got the same rank, you will return Tie.

In the ‘real Poker’ if two player got same rank (for example: two pair of Aces VS two pair of Kings: the Aces will win)

If you are able to do as the ‘real Poker’ ranking, **you will get bonus points for this project**. Make sure that you do it for all the hands and you are sure that you understand the rules! **If you are not sure, please ask first else your program will not be valid.**

Also you will have to name your project 06_poker_real.py if you do the BONUS. Else you will name it 06_poker_hand.py

More information about Poker Ranking:

https://en.wikipedia.org/wiki/List_of_poker_hands

Requirements :

- Program must be named : **06_poker_hand.py** and saved into **week02/projects**
- Your function must be called **poker_hand(player_one, player_two)**

Python Bootcamp {2019}

Kirirom Institute of Technology

Hint :

- string split
- loop
- array/list
- make sure you understand the rules before starting your algorithms.

Output :

Example 01:

```
poker_hand("AH KH QH JH TH", "AS KS TS 2H 3H")

>> Player 1 WIN
```

Example 02:

```
poker_hand("AH KS QS JS TH", "8S 8H 8D 8C 2H")

>> Player 2 WIN
```

Example 03:

```
poker_hand("AH KS QH JH TH", "AS KH QC JC TC")

>> Tie
```

Kirirom Institute of Technology

Explanation: (don’t print this part in your program)

- Example 01 : Player 1 WIN
  Royal Flush [Player One] > No Pair [Player Two]

- Example 02 : Player 2 WIN
  Straight [Player One] < Four of a Kind [Player Two]

- Example 03 : Tie
  Straight [Player One] == Straight [Player Two]

⚠ Warning ⚠

MAKE SURE THAT ALL YOUR PROJECTS AND EXERCISES ARE WELL TESTED AND YOU DID NOT FORGET EVEN A SINGLE CHARACTER. MAKE SURE THAT YOUR FILENAMES ARE CORRECT. MAKE SURE THAT YOUR PROJECTS AND EXERCISES RESPECTS ALL THE REQUIREMENTS.