9913_exp3

February 26, 2024

[1]:
```
def same(str):
    return str


string = input("What is your name?")
print(f"Your name is {same(string)}")
```

```
What is your name?Mark
Your name is Mark
```

[2]:
```
def subtract_three(n):
    return(n-3)

no = int(input("Enter a number: "))
print(f"The value is {subtract_three(no)}")
```

```
Enter a number: 4
The value is 1
```

[3]:
```
def intro(str):
    return(f"Hello, My name is {str} and i love SI 106")

name = input("WHats your name?")
print(intro(name))
```

```
WHats your name?Becky
Hello, My name is Becky and i love SI 106
```

[4]:
```
def total(ls):
    sum = 0
    for i in ls:
        sum+=i
    return sum

l = [1,2,3,4,5]
print(f"The total value is {total(l)}")
```

```
The total value is 15
```

[5]:
```
eve_nums = []
count = 0
while count<16:
    if count%2==0:
        eve_nums.append(count)
    count = count +1
print(eve_nums)
```

```
[0, 2, 4, 6, 8, 10, 12, 14]
```

[7]:
```
sum1=0
first = [65,78,21,33]
for x in first:
    sum1 = sum1+x

sum2 = 0
second = [65,78,21,33]
i = 0
while i < len(second):
    sum2 = sum2 + second[i]
    i = i+1
if sum1==sum2:
    print("They are equal")
```

```
They are equal
```

[15]:
```
def addit(n1):
    n1 = n1+5
    return n1

def mult(n2):
    n2 = n2*addit(n1)
    return n2
n1 = int(input("Enter n1: "))
n2 = int(input("Enter n2: "))
print("The value is ",mult(n2))
```

```
Enter n1: 5
Enter n2: 2
The value is  20
```

[16]:
```
medal = {}

for i in range(3):
    key = input("Type of medal: ")
    value = int(input("Number of medals: "))
    medal[key] = value
```

```
print(medal)
```

```
Type of medal: gold
Number of medals: 33
Type of medal: silver
Number of medals: 17
Type of medal: bronze
Number of medals: 12
{'gold': 33, 'silver': 17, 'bronze': 12}
```

[21]:
```
lst_tups = [("Articuno", "Moltres", "Zaptos"), ("Beedrill", "Metapod",␣
 ↪"Charizard", "Venasaur",
"Squirtle"), ("Oddish", "Poliwag", "Diglett","Bellsprout"), ("Ponyta",␣
 ↪"Farfetch", "Tauros",
"Dragonite"),("Hoothoot", "Chikorita", "Lanturn", "Flaaffy", "Unown",␣
 ↪"Teddiursa", "Phanpy"),
("Loudred", "Volbeat", "Wailord", "Seviper", "Sealeo")]

third=[]
for i in lst_tups:
    third.append(i[2])
print(third)
```

```
['Zaptos', 'Charizard', 'Diglett', 'Tauros', 'Lanturn', 'Wailord']
```

[22]:
```
nums = ['1450', '33', '871', '19', '14378', '32', '1005', '44', '8907', '16']
def last_char(nums):
    return nums[-1]
nums_sorted = sorted(nums, key=last_char, reverse = True)
print(nums_sorted)
```

```
['19', '14378', '8907', '16', '1005', '44', '33', '32', '871', '1450']
```

[1]:
```
def print_board(board):
   ␣
 ↪print(f"|{board[0]}|{board[1]}|{board[2]}|\n|{board[3]}|{board[4]}|{board[5]}|\n|{board[6]}

def check_win(board, player):
    # Check rows, columns, and diagonals for a win
    return (
        (board[0] == board[1] == board[2] == player) or
        (board[3] == board[4] == board[5] == player) or
        (board[6] == board[7] == board[8] == player) or
        (board[0] == board[3] == board[6] == player) or
        (board[1] == board[4] == board[7] == player) or
        (board[2] == board[5] == board[8] == player) or
        (board[0] == board[4] == board[8] == player) or
        (board[2] == board[4] == board[6] == player)
```

```
    )

def check_tie(board):
    # Check if the board is full and no player has won
    return "-" not in board

def player_turn(board, player, position):
    # Update the board with the player's move
    if board[position - 1] == "-":
        board[position - 1] = player
        return True
    else:
        print("Invalid move. The position is already occupied. Try again.")
        return False

def play_game():
    empty_board = ["-", "-", "-", "-", "-", "-", "-", "-", "-"]
    select_board = ['1', '2', '3', '4', '5', '6', '7', '8', '9']

    print_board(select_board)

    player1 = "X"
    player2 = "O"
    current_player = player1

    while True:
        print(f"\n{current_player}'s turn.")
        position = int(input("Enter the position (1-9): "))

        if 1 <= position <= 9:
            if player_turn(empty_board, current_player, position):
                print_board(empty_board)

                if check_win(empty_board, current_player):
                    print(f"{current_player} wins!")
                    break
                elif check_tie(empty_board):
                    print("It's a tie!")
                    break

                # Switch to the next player
                current_player = player2 if current_player == player1 else␣
 ↪player1
            else:
                continue
        else:
            print("Invalid position. Please enter a number between 1 and 9.")
```

```
if __name__ == "__main__":
    play_game()
```

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

X's turn.
|X|-|-|
|-|-|-|
|-|-|-|

O's turn.
|X|-|-|
|-|O|-|
|-|-|-|

X's turn.
|X|X|-|
|-|O|-|
|-|-|-|

O's turn.
|X|X|-|
|-|O|O|
|-|-|-|

X's turn.
|X|X|X|
|-|O|O|
|-|-|-|
X wins!
```

[3]:
```
import random

def choose_word():
    words = ["python", "java", "javascript", "ruby", "html", "css"]
    return random.choice(words)

def display_word(word, guessed_letters):
    return ''.join(letter if letter in guessed_letters else '-' for letter in␣
 ↪word)

def word_guessing_game():
    word_to_guess = choose_word()
    guessed_letters = []
```

```
    attempts = 0

    print("Welcome to the Word Guessing Game!")
    print("Category: Programming Languages")

    while '-' in display_word(word_to_guess, guessed_letters):
        print("Word: " + display_word(word_to_guess, guessed_letters))
        user_guess = input("Enter a letter: ").lower()

        if user_guess.isalpha() and len(user_guess) == 1:
            if user_guess in guessed_letters:
                print("You already guessed that letter. Try again.")
            elif user_guess in word_to_guess:
                guessed_letters.append(user_guess)
                print("Good guess!")
            else:
                attempts += 1
                revealed_letter = random.choice([letter for letter in␣
 ↪word_to_guess if letter not in guessed_letters])
                guessed_letters.append(revealed_letter)
                print(f"Incorrect! Revealing a letter:␣
 ↪{display_word(word_to_guess, guessed_letters)}")
        else:
            print("Invalid input. Please enter a single letter.")

    print(f"Congratulations! You guessed the word '{word_to_guess}' in␣
 ↪{attempts} attempts.")

if __name__ == "__main__":
    word_guessing_game()
```

```
Welcome to the Word Guessing Game!
Category: Programming Languages
Word: ----

Enter a letter:  a

Incorrect! Revealing a letter: --b-
Word: --b-

Enter a letter:  r

Good guess!
Word: r-b-

Enter a letter:  u

Good guess!
Word: rub-

Enter a letter:  y
```

```
Good guess!
Congratulations! You guessed the word 'ruby' in 1 attempts.
```

[ ]: