

Solucions Exercicis Strings i Llistes

Exercise 1: Basic String Concatenation

Write a Python program that takes two strings as input from the user and prints their concatenation.

Solution

```
str1 = input("Enter the first string: ")
str2 = input("Enter the second string: ")
result = str1 + str2
print("Concatenated String:", result)
```

Exercise 2: String Length

Write a program that takes a string as input and prints its length.

Solution

```
user_string = input("Enter a string: ")
print("Length of the string:", len(user_string))
```

Exercise 3: String Slicing

Given a string input by the user, print the first three and the last three characters.

Solution

```
user_string = input("Enter a string: ")
if len(user_string) >= 3:
    print("First three characters:", user_string[:3])
```

```
    print("Last three characters:", user_string[-3:])  
  
else:  
  
    print("String is too short!")
```

Exercise 4: Count Vowels

Write a Python program to count the number of vowels in a given string.

Solution

```
user_string = input("Enter a string: ")  
  
vowels = "aeiouAEIOU"  
  
count = sum(1 for char in user_string if char in vowels)  
  
print("Number of vowels:", count)
```

Exercise 5: Palindrome Checker

Write a program to check if the input string is a palindrome.

Solution

```
user_string = input("Enter a string: ")  
  
if user_string == user_string[::-1]:  
  
    print("The string is a palindrome.")  
  
else:  
  
    print("The string is not a palindrome.")
```

Exercise 6: Remove Duplicates

Write a program that removes duplicate characters from a string while maintaining the original order.

Solution

```
user_string = input("Enter a string: ")

result = ''.join(dict.fromkeys(user_string))

print("String without duplicates:", result)
```

Exercise 7: Find Substring

Write a program to check if a substring exists within a string. If it does, print its starting index; otherwise, print "Substring not found."

Solution

```
user_string = input("Enter the main string: ")

substring = input("Enter the substring: ")

index = user_string.find(substring)

if index != -1:

    print("Substring found at index:", index)

else:

    print("Substring not found.")
```

Exercise 8: Count Words

Write a Python program to count the number of words in a string.

Solution

```
user_string = input("Enter a string: ")

word_count = len(user_string.split())

print("Number of words:", word_count)
```

Exercise 9: Reverse Words

Write a program that takes a sentence as input and prints the sentence with its words reversed.

Solution

```
user_string = input("Enter a sentence: ")

reversed_words = ' '.join(user_string.split()[::-1])

print("Sentence with reversed words:", reversed_words)
```

Exercise 10: Longest Word

Write a program to find the longest word in a given string and its length.

Solution

```
user_string = input("Enter a sentence: ")

words = user_string.split()

longest_word = max(words, key=len)

print("Longest word:", longest_word)

print("Length of the longest word:", len(longest_word))
```

Solucions d'exercicis sobre llistes

1. Escriu un programa que creï una llista amb 5 números donats per l'usuari i imprimeixi la llista completa.

Solució

```
numbers = []

for _ in range(5):

    number = int(input("Introdueix un número: "))

    numbers.append(number)

print("Llista:", numbers)
```

2. Escriu un programa que trobi el valor més gran (**max**) i el més petit (**min**) en una llista d'enters.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de
números separats per espais: ").split()]

max_value = max(numbers)

min_value = min(numbers)

print("Valor màxim:", max_value)

print("Valor mínim:", min_value)
```

3. Escriu un programa que compti el nombre d'elements parells (**even**) i senars (**odd**) en una llista d'enters donada.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de  
números separats per espais: ").split()]\n\neven_count = sum(1 for x in numbers if x % 2 == 0)\n\nodd_count = sum(1 for x in numbers if x % 2 != 0)\n\nprint("Parells:", even_count)\n\nprint("Senars:", odd_count)
```

4. Escriu un programa per invertir (**reverse**) l'ordre dels elements d'una llista sense utilitzar el mètode integrat **reverse()**.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de  
números separats per espais: ").split()]\n\nreversed_list = numbers[::-1]\n\nprint("Llista invertida:", reversed_list)
```

5. Escriu un programa que elimini els elements duplicats d'una llista i en mantingui l'ordre original.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de  
números separats per espais: ").split()]\n\nunique_numbers = list(dict.fromkeys(numbers))\n\nprint("Llista sense duplicats:", unique_numbers)
```

6. Escriu un programa que combini dues llistes de la mateixa mida en una sola llista de tuples, on cada tuple conté elements corresponents de les dues llistes.

Solució

```
list1 = [int(x) for x in input("Introdueix la primera llista de  
números separats per espais: ").split()]\n\nlist2 = [int(x) for x in input("Introdueix la segona llista de números  
separats per espais: ").split()]\n\ncombined = list(zip(list1, list2))\n\nprint("Llista combinada:", combined)
```

7. Escriu un programa que ordeni una llista d'enters de menor a major sense utilitzar el mètode integrat `sort()`.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]\n\nfor i in range(len(numbers)):\n\n    for j in range(len(numbers) - i - 1):\n\n        if numbers[j] > numbers[j + 1]:\n\n            numbers[j], numbers[j + 1] = numbers[j + 1], numbers[j]\n\nprint("Llista ordenada:", numbers)
```

8. Escriu un programa per trobar tots els elements únics (`unique`) que es repeteixen en una llista.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]
```

```
unique_repeated = set(x for x in numbers if numbers.count(x) > 1)

print("Elements repetits únics:", list(unique_repeated))
```

9. Escriu un programa que trobi la suma (**sum**) de tots els elements d'una llista, però només dels nombres positius.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de
números separats per espais: ").split()]

positive_sum = sum(x for x in numbers if x > 0)

print("Suma de positius:", positive_sum)
```

10. Escriu un programa que creï una nova llista amb els quadrats (**squares**) de tots els números d'una llista donada.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de
números separats per espais: ").split()]

squared_numbers = [x**2 for x in numbers]

print("Llista de quadrats:", squared_numbers)
```


Solucions d'exercicis avançats sobre llistes

1. Escriu un programa que rotació una llista d'enters cap a l'esquerra un nombre determinat de vegades.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]
```

```
rotations = int(input("Introdueix el nombre de rotacions: "))
```

```
rotated_list = numbers[rotations % len(numbers):] + numbers[:rotations  
% len(numbers)]
```

```
print("Llista rotada:", rotated_list)
```

2. Escriu un programa per eliminar tots els elements d'una llista que apareixen un nombre imparell de vegades.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]
```

```
filtered_list = [x for x in numbers if numbers.count(x) % 2 == 0]
```

```
print("Llista filtrada:", filtered_list)
```

3. Escriu un programa que trobi els dos nombres més grans d'una llista sense utilitzar funcions integrades com `max()` o `sort()`.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]\n\nlargest = second_largest = float('-inf')\n\nfor num in numbers:\n\n    if num > largest:\n\n        second_largest = largest\n\n        largest = num\n\n    elif num > second_largest and num != largest:\n\n        second_largest = num\n\nprint("Dos nombres més grans:", largest, second_largest)
```

4. Escriu un programa que creï una llista de llistes (matriu) de dimensions donades per l'usuari i la inicialitzi amb valors seqüencials.

Solució

```
rows = int(input("Introdueix el nombre de files: "))\n\ncols = int(input("Introdueix el nombre de columnes: "))\n\nmatrix = [[cols * i + j + 1 for j in range(cols)] for i in  
range(rows)]\n\nprint("Matriu:")\n\nfor row in matrix:\n\n    print(row)
```

5. Escriu un programa que identifiqui totes les subseqüències consecutives de números en una llista.

Solució

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]

subsequences = []

current_subsequence = [numbers[0]]

for i in range(1, len(numbers)):

    if numbers[i] == numbers[i-1] + 1:

        current_subsequence.append(numbers[i])

    else:

        subsequences.append(current_subsequence)

        current_subsequence = [numbers[i]]

subsequences.append(current_subsequence)

print("Subseqüències consecutives:", subsequences)
```

6. Escriu un programa que trobi el producte de tots els elements en una llista sense utilitzar un bucle explícit.

Solució

```
from functools import reduce

numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]

product = reduce(lambda x, y: x * y, numbers)
```

```
print("Producte de tots els elements:", product)
```

7. Escriu un programa que trobi la intersecció de dues llistes, mantenint l'ordre original dels elements de la primera llista.

Solució

```
list1 = [int(x) for x in input("Introdueix la primera llista de  
números separats per espais: ").split()]
```

```
list2 = [int(x) for x in input("Introdueix la segona llista de números  
separats per espais: ").split()]
```

```
intersection = [x for x in list1 if x in list2]
```

```
print("Intersecció:", intersection)
```

8. Escriu un programa per transformar una llista de tuples en un diccionari.

Solució

```
tuples = [tuple(input("Introdueix un tuple (clau, valor) separat per  
coma: ").split(',')) for _ in range(int(input("Quants tuples? ")))]
```

```
dictionary = {key: value for key, value in tuples}
```

```
print("Diccionari:", dictionary)
```

9. Escriu un programa que ordeni una llista de tuples basant-se en el segon element de cada tuple.

Solució

```
tuples = [tuple(input("Introdueix un tuple (element1, element2)  
separat per coma: ").split(',')) for _ in range(int(input("Quants  
tuples? ")))]
```

```
sorted_tuples = sorted(tuples, key=lambda x: x[1])
```

```
print("Llista de tuples ordenada:", sorted_tuples)
```

10. Escriu un programa per generar totes les combinacions possibles de longitud **k** d'una llista donada.

Solució

```
from itertools import combinations
```

```
numbers = [int(x) for x in input("Introdueix una llista de números  
separats per espais: ").split()]
```

```
k = int(input("Introdueix la longitud de les combinacions: "))
```

```
combinations_list = list(combinations(numbers, k))
```

```
print("Combinacions possibles:", combinations_list)
```