

Exercicis avançats de tractament de cadenes en Python

1. Mesures de les cadenes (len, max, min):

Escriu un programa que demani a l'usuari diverses paraules separades per espais. El programa ha de calcular la paraula més llarga, la més curta i mostrar el caràcter amb el valor més alt i més baix ASCII dins de la cadena completa.

Entrada: 'Python és increïble'

Sortida esperada:

La paraula més llarga és: increïble

La paraula més curta és: és

El caràcter amb el valor ASCII més alt és: y

El caràcter amb el valor ASCII més baix és: (espai)

2. Operacions amb cadenes (+, *, \n):

Escriu un programa que demani a l'usuari una frase i un número. El programa ha de concatenar la frase amb una versió repetida de la mateixa frase el número de vegades indicat, amb cada repetició en una línia nova.

Entrada: 'Hola, món!', 3

Sortida esperada:

Hola, món!

Hola, món!

Hola, món!

3. Cerca de caràcters (count, index, find):

Escriu un programa que demani una cadena i un caràcter. El programa ha de mostrar quantes vegades apareix el caràcter en la cadena, la seva primera aparició i la seva última aparició.

Entrada: 'supercalifragilistico', 'i'

Sortida esperada:

El caràcter 'i' apareix 3 vegades.

La seva primera aparició és a la posició 7.

La seva última aparició és a la posició 17.

4. Majúscules i minúscules (upper, lower, capitalize):

Escriu un programa que demani una frase a l'usuari. El programa ha de mostrar la frase en tres versions: totes les paraules en majúscula, en minúscules, i amb només la primera lletra en majúscula.

Entrada: 'el gat menja peix'

Sortida esperada:

Frase amb majúscules: EL GAT MENJA PEIX

Frase en minúscules: el gat menja peix

Frase capitalitzada: El gat menja peix

La frase estava originalment en minúscules.

5. Concatenació de cadenes (".join()"):

Escriu un programa que demani a l'usuari una llista de paraules separades per comes i un símbol escollit per l'usuari per unir-les.

Entrada: 'poma,plàtan,taronja', '@'

Sortida esperada:

poma@plàtan@taronja

6. Divisió de cadenes (split, partition):

Escriu un programa que demani a l'usuari una frase llarga i una paraula de divisió. El programa ha de dividir la frase utilitzant 'partition' i després mostrar la llista de paraules utilitzant 'split'.

Entrada: 'A mi m'agrada la pizza, i també la pasta', 'pizza'

Sortida esperada:

Partició de la frase: ('A mi m'agrada la ', 'pizza', ', i també la pasta')

Llista de paraules: ['A', 'mi', 'm'agrada', 'la', 'pizza', 'i', 'també', 'la', 'pasta']

7. Ordenació de cadenes (sorted):

Escriu un programa que demani a l'usuari una frase i mostri els caràcters ordenats de manera ascendent i descendent.

Entrada: 'zebra'

Sortida esperada:

Caràcters ordenats (ascendent): aebzr

Caràcters ordenats (descendent): zrbca

8. Modificació de cadenes (replace, strip, lstrip, rstrip):

Escriu un programa que demani una frase amb espais al principi i al final, i un caràcter o paraula a substituir. Utilitza 'strip', 'replace', 'lstrip' i 'rstrip' per modificar la cadena i mostrar les versions.

Entrada: ' Hola, Python és genial! ', 'Python'

Sortida esperada:

Frase sense espais: 'Hola, Python és genial!'

Frase sense espais inicials: 'Hola, Python és genial! '

Frase sense espais finals: ' Hola, Python és genial!'

Frase modificada: 'Hola, [substituït] és genial!'

9. Comparació de cadenes (<, >, ==, !=):

Escriu un programa que demani dues cadenes a l'usuari i compari quina és més gran segons l'ordre alfabètic. Mostra si són iguals o diferents.

Entrada: 'gat', 'gos'

Sortida esperada:

La cadena 'gos' és major que 'gat'.

10. Comprovacions True/False (isdigit, isalpha, islower, startswith, endswith):

Escriu un programa que demani una cadena a l'usuari i comprovi si només conté dígit, lletres, espais, i si comença o acaba amb un caràcter específic.

Entrada: 'Python123'

Sortida esperada:

La cadena només conté dígit: False

La cadena només conté lletres: False

La cadena només conté espais: False

La cadena comença amb 'P': True

La cadena acaba amb '3': True