

2º SMR	Fecha	21/04/2020
FCT 2020 Prácticas de Programación en Python		

FCT 2020 Prácticas de Programación en Python

Ejercicios prácticos para probar los conocimientos adquiridos en el curso online

2º SMR	Fecha	21/04/2020
FCT 2020 Prácticas de Programación en Python		

Índice

[Índice](#)

[Práctica 1](#)

[Práctica 2](#)

[Práctica 3](#)

[Práctica 4](#)

[Práctica 5](#)

[Práctica 6](#)

[Práctica 7](#)

[Práctica 8](#)

[Práctica 9](#)

[Práctica 10](#)

[Práctica 11](#)

2º SMR	Fecha	21/04/2020
FCT 2020 Prácticas de Programación en Python		

Práctica 1

Escribe un programa que muestre el siguiente texto:

BIENVENIDO A EMPAREJANDO.COM

Necesitamos conocer algunos datos sobre ti para encontrar tu pareja ideal.

A continuación, el programa irá realizando algunas preguntas de una en una y guardando las respuestas en variables:

Tu nombre:

Año de nacimiento:

¿Te gusta Taburete?

Y al final, responderá de la siguiente manera:

Hola, “nombre”. Si no me equivoco, tienes “edad” años.

Si le gusta taburete, dirá:

OK Boomer, lo tuyo va a ser un caso difícil.

Si no le gusta, dirá:

Bueno, al menos es un comienzo. Veremos qué se puede hacer contigo.

Práctica 2

Partiendo del programa anterior, añadir un bucle while que cuente desde el 1 hasta la edad del usuario y escriba en pantalla las siguientes frases:

Que no cumple 1

Que no cumple 2

Que no cumple 3

...

¡Qué sí cumple “años”!

Práctica 3

Hacer lo mismo que en la práctica 2 pero usando un bucle For en vez de While. Ojo, en vez de usar un For con listas o tuplas, como explican en el curso que estáis haciendo, lo más sencillo es crear una progresión [como las que explican aquí](#)

2º SMR	Fecha	21/04/2020
FCT 2020 Prácticas de Programación en Python		

Práctica 4

Partiendo de la Práctica 2 o la 3, guardar la información del usuario en una variable de tipo lista. A continuación, imprimir los datos de la lista usando un bucle For.

Práctica 5

Repite la Práctica 4 pero guardando los datos del usuario en una variable de tipo diccionario en lugar de en una lista.

Práctica 6

Programa tu primer cifrador César. El programa debe pedir al usuario una frase de texto en claro y un número para cifrar. A continuación devolverá el texto cifrado.

Mejora: convierte todo el texto en minúsculas antes de cifrarlo.

Práctica 7

Modifica el programa de la práctica 6 metiendo el código de cifrado dentro de una función llamada `cifracesar(texto,key)` que devuelve el texto cifrado.

Práctica 8

Añade una función que se llame `descifracesar`. Ahora, cuando el usuario meta un texto y una clave, primero debes llamar a la función de cifrado e imprimir el texto cifrado en pantalla, y a continuación descifrarlo. así comprobamos si funciona bien.

Práctica 9

Al programa de la práctica 7, añádele un menú con tres opciones:

1. Cifrar
2. Descifrar
3. Salir

El menú se ejecutará en bucle hasta que se pulse la opción de salir. Usa la función `cifracesar` de la práctica anterior y programa una nueva función `descifracesar` que haga lo contrario.

Práctica 10

¡Práctica colaborativa!

Una de las cosas que hacíamos en la uni era coger los programas de otros compañeros y los hacíamos fallar, por ejemplo escribiendo una letra en vez de un número, o escribiendo un texto con caracteres extraños, o pulsando opciones de menú que no existen...

2º SMR	Fecha	21/04/2020
FCT 2020 Prácticas de Programación en Python		

Mejora la práctica 8 añadiendo control de errores, por ejemplo:

- Comprueba que el usuario ha introducido un número en vez de un texto cuando le pides la clave de cifrado.
- Añade soporte de excepciones por si al hacer la búsqueda en el diccionario, la letra que buscas no está.

El programa no puede colgarse nunca. Si algo falla hay que explicárselo al usuario y volver a preguntar, o al menos volver al menú inicial.

Cuando creáis que vuestros programas son robustos, compartidlos con otros compañeros y pedidles que los prueben a ver cuánto duran...

Práctica 11

Ya tienes un cifrador fantástico. Ahora busca un poco de ayuda en Internet para hacer que, en vez de tener que escribir un texto a mano, podamos ejecutar el programa indicando como parámetro el nombre de un archivo, la clave de cifrado y si queremos que el programa cifre o descifre. Por ejemplo:

cesar -c 5 archivo.txt → Para cifrar con clave 5 el archivo.txt

cesar -d 3 archivo.txt → Para descifrar con clave 5 el archivo.txt

¡Habéis programado un comando como los del sistema operativo para cifrar y descifrar! Ahora sí que sois programadores PRO.