

Programa 11

Descripción del problema:

Se desea realizar un programa que guarde el nombre, la edad y lo que se les recomendaba hacer, como teníamos antes, para un número de presentes que el usuario dirá. Después de preguntar la información de cada uno de ellos, el programa debe preguntar un número y dar la información referida a la persona que se introdujo en esa posición.

Materia nueva:

Los Diccionarios son un tipo de datos que puede guardar para una sola variable un conjunto de campos. Pongamos un ejemplo: Imaginemos un alumno. Existe un montón de información que podríamos querer guardar sobre él y hacer una tabla como:

```
|Nombre|Apellido 1|Apellido 2|Altura|Nº Calzado|Color de pelo| |--|--|--|--|--| |Jorge|Bes|Tuán|2  
m|47|Negro|
```

Cuando alguien nos preguntara por este alumno, lo haría diciéndonos: Dime el primer apellido del alumno, el color de pelo... Es decir, nos diría el campo que quiere. Tras esto, nosotros iríamos a la celda y responderíamos con la información allí contenida.

Bueno, pues esto es un Diccionario para Python. Tiene unos valores que se llaman Key que en este caso sería la fila superior (Nombre, Apellido1...) y otros que se llaman Value que serían los de la fila inferior.

Una duda que suele asaltar es: ¿Por qué sólo una fila? ¿Y si queremos guardar un conjunto de 1000 alumnos? Pues muy sencillo, se hace una lista de diccionarios, que es lo que vamos a hacer en nuestro programa. Al fin y al cabo, una lista es un conjunto de cosas, nadie ha dicho que haya algo que no se pueda meter ahí.

La sintaxis para declarar un diccionario vacío es:

```
diccionario = {}
```

Y para guardar cualquier dato es:

```
diccionario["Key1"]=datoaguardar
```

Aquí hay que decir que, a diferencia de las listas, donde había que añadir un nuevo componente con una orden, aquí es tan fácil como poner una nueva "Key" para que la añada al conjunto de ellas. Es decir, es como tener una tabla dinámica donde podemos añadir columnas sobre la marcha.

Un diccionario tiene muchísimas más posibilidades en Python, como se puede ver en:

<https://docs.python.org/3/library/stdtypes.html#typesmapping>

No obstante, en este curso no se va sino a introducir este tipo de variable. Queda demostrada la potencia de Python y el nivel de abstracción de los datos y se anima a los alumnos a completar su formación conforme lo vayan necesitando para sus proyectos.

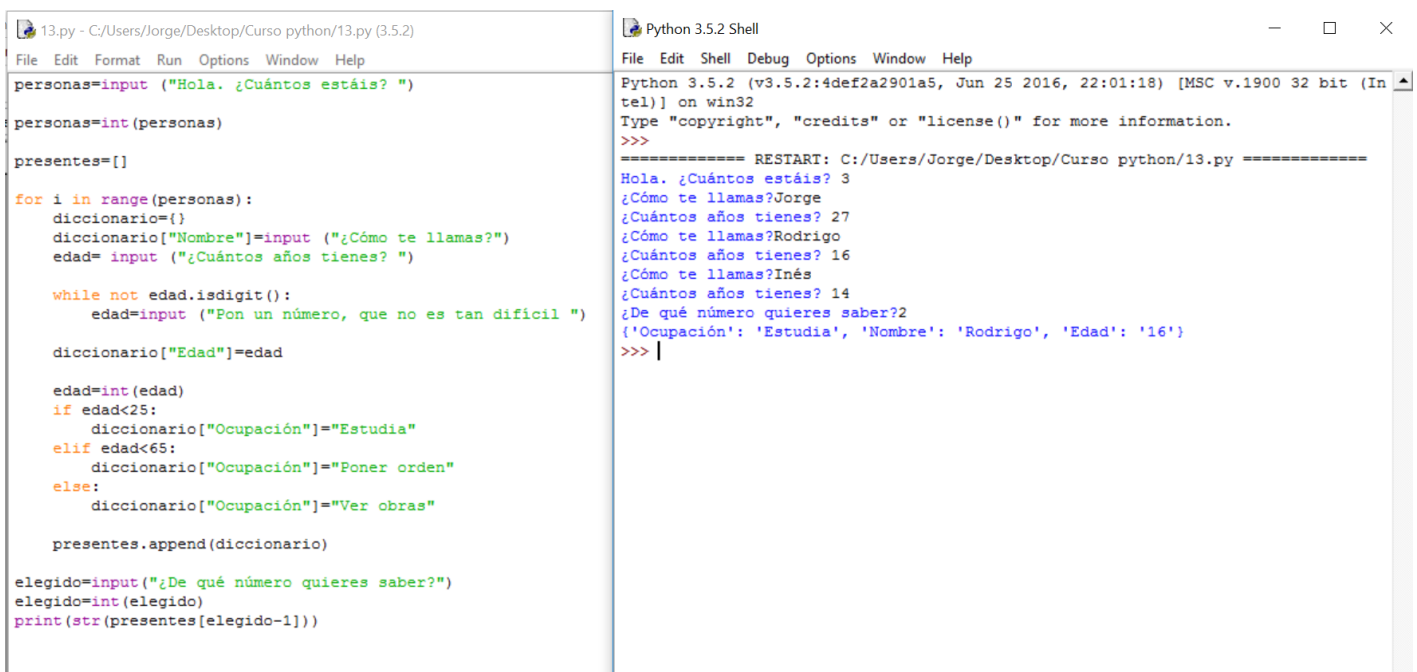
NOTA: Si vas a introducir cambios en un diccionario dentro de una estructura como **for** o **while**, que los ejecutará varias veces, declara esa variable dentro de la estructura y no fuera. Más adelante se explica el por qué.

Solución

Algoritmo:

- 1.- Preguntaremos cuánta gente hay
- 2.- A cada uno le preguntaremos su nombre y su edad. Guardaremos las dos variables anteriores y la tarea que tienen que hacer (poner orden, estudiar y mirar obras) en un diccionario.
- 3.- Pediremos que nos diga el número de persona que quiere que le mostremos.
- 4.- Lo mostraremos en pantalla.

Solución:



The image shows a side-by-side comparison of a Python script and its execution. On the left is a text editor window titled '13.py - C:/Users/Jorge/Desktop/Curso python/13.py (3.5.2)'. It contains a Python script that asks for the number of people, then for each person's name, age, and a task (study, order, or see works). On the right is a 'Python 3.5.2 Shell' window showing the script's execution. The user enters '3' for the number of people, and then provides details for three people: Jorge (27 years old, studies), Rodrigo (16 years old, orders), and Inés (14 years old, sees works). The final output is a dictionary for Rodrigo: {'Ocupación': 'Estudia', 'Nombre': 'Rodrigo', 'Edad': '16'}.

```
13.py - C:/Users/Jorge/Desktop/Curso python/13.py (3.5.2)
File Edit Format Run Options Window Help
personas=input ("Hola. ¿Cuántos estáis? ")
personas=int(personas)

presentes=[]

for i in range(personas):
    diccionario={}
    diccionario["Nombre"]=input ("¿Cómo te llamas?")
    edad= input ("¿Cuántos años tienes? ")

    while not edad.isdigit():
        edad=input ("Pon un número, que no es tan difícil ")

    diccionario["Edad"]=edad

    edad=int(edad)
    if edad<25:
        diccionario["Ocupación"]="Estudia"
    elif edad<65:
        diccionario["Ocupación"]="Poner orden"
    else:
        diccionario["Ocupación"]="Ver obras"

    presentes.append(diccionario)

elegido=input ("¿De qué número quieres saber?")
elegido=int(elegido)
print(str(presentes[elegido-1]))

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Jorge/Desktop/Curso python/13.py =====
Hola. ¿Cuántos estáis? 3
¿Cómo te llamas?Jorge
¿Cuántos años tienes? 27
¿Cómo te llamas?Rodrigo
¿Cuántos años tienes? 16
¿Cómo te llamas?Inés
¿Cuántos años tienes? 14
¿De qué número quieres saber?2
{'Ocupación': 'Estudia', 'Nombre': 'Rodrigo', 'Edad': '16'}
>>> |
```

Comentarios:

Aquí hay que comentar un error bastante curioso que tiene más que ver con cómo hace las cosas Python que con la programación en sí. Prueba a cambiar la declaración `diccionario={}` a justo antes del **for** y usa la siguiente orden al final del programa:

print (str(presentes)) Para que saque por pantalla toda la lista que hemos introducido.

Verás que ha guardado el último tantas veces como gente había. Esto es porque, cada vez que se declara, hace uno nuevo, si no, sólo se está alterando el mismo todo el rato.

No hace falta que lo entiendas, realmente es un error de Python, aunque conozco que se da en varios lenguajes más. Simplemente, quédate con que las variables, para que se puedan usar varias veces, es necesario que tengan su espacio en memoria diferente, y eso lo haces declarándolas de nuevo ya que Python les busca un nuevo hueco en memoria aunque las llame igual.

Es necesario también destacar el recurso de restar 1 a la variable que nos dice qué persona queremos para ajustar el que el ordenador empiece a contar por el 0 y nosotros por el 1.

Revision #2

Created 1 February 2022 11:42:11 by Equipo CATEDU

Updated 1 February 2022 11:42:12 by Equipo CATEDU