2. Entrando en materia

Entrando en materia

Objetivos

- Conocer las diferentes estructuras cíclicas de Python
- Profundizar en los tipos de datos conocidos.

Programa 7

Descripción del problema:

Se va a modificar el programa 6 para saludar a más gente, primero preguntaremos cuánta gente hay e iremos saludando uno a uno comentando su edad como hasta ahora.

Materia nueva:

Cuando sabemos el número de veces que tenemos que hacer algo, aunque podríamos usar **while**, se usa otra estructura:

For significa en inglés “por”. Lo que quiere decir es “por cada uno”. Tiene muchas sintaxis aunque sólo daremos una de momento.

```
for nombredevariableentera in range(númeroinicial, númerofinal):

    Órdenes
```

Hay que decir que, si númeroinicial =0 no hace falta ponerlo y quedaría:

```
for nombredevariableentera in range (númerofinal):
```

¡Atención! Python empieza en el número inicial, imaginemos **range (1,5).** Cuando a cualquier persona le resulta normal contar de uno a cinco de la siguiente manera:

1, 2, 3, 4, 5

A Python no, resulta que él cuenta los números que ha puesto y dice: "No puede ser: 5-1=4. Debo hacerlo cuatro veces" y cuenta de esta manera:

1, 2, 3, 4

He empezado en el 1 por ser range(1,5) . Si hubiera sido **range (**4**)**:

0, 1, 2, 3

Como se ve, el último número no lo va a tocar. **Cuidado con esto ya que ocasiona la mayor parte de problemas en los programas**.

Solución

Algoritmo:

1.- Saludo y pregunto cuánta gente hay

2.- Para cada uno de ellos:

2.1.- Pido el nombre

2.2.- Pido la edad

2.3.- Compruebo que se puede convertir en número y, si no puede, repito la pregunta hasta que pueda. Voy guardando en una variable las veces que hago esto.

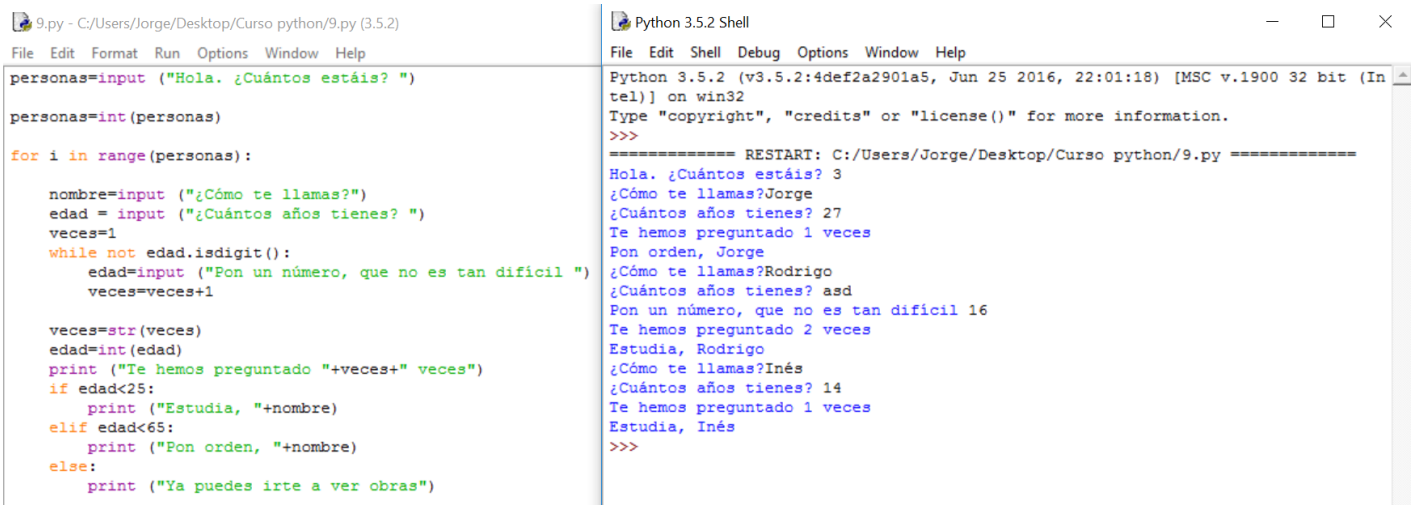
2.4.- Le digo las veces que se lo he preguntado.

2.5.- Comparo la edad y, si es menor de 25 le digo que estudie

Si es menor de 65 le digo que ponga orden

Si es mayor o igual a 65 le digo que se vaya

Solución:



Explicación:

Es necesario darse cuenta de que se ha identado el programa anterior para que forme parte de la estructura for, pues tenía que repetir todo el programa anterior. Menos mal que tenemos la identación para aclarar un poco las diferentes partes que componen el programa.

Si quieres identar una gran cantidad de texto, el editor tiene una opción para identar un montón de líneas a la vez y que no sea necesario hacerlo a mano. Está en **format: indent region**.

Programa 8

Descripción del problema:

Queremos realizar un programa que modifique el número 7. Esta vez sólo queremos saludar pero, en vez de hacerlo uno a uno, lo haremos a todos a la vez, es decir, es necesario que recuerde los nombres y se dirija a ellos con un “Hola, persona1, persona2, persona3... y personaúltima”.

Materia nueva:

Existe otro tipo de datos que se llama List (lista). Una lista es, tal y como se puede pensar, lo mismo que en la vida real salvo que, en Python, el primer elemento siempre es el 0. Imaginemos que queremos preparar una cena. Todo lo que se nos ocurra podemos hacerlo en Python, por ejemplo:

|Vida real|Python| |--|--| |Contar el número de comensales:|len(lista) len es una abreviatura de length| |Decir qué comensal ocupa el número 3, por ejemplo|lista[3] Siempre entre corchetes| |Escoger del 7 al 14|lista[7:14] Tened en cuenta el 0!!| |Añadir uno al final|lista.append(otrocomensal)|

Hay muchas más posibilidades, como se puede ver en:

https://docs.python.org/3/tutorial/datastructures.html

Este tipo de datos hay que declararlos, bien para decir que están vacíos, bien porque los queremos con datos iniciales.

Se declara nombrando la lista y poniendo:

listanueva **[]** Si la queremos vacía para ir llenándola a lo largo del programa.

```
listanueva= ["Lunes","Martes","Miércoles","Jueves","Viernes"]
```

Si queremos los días laborables. Es decir, se introducen los elementos separados por comas.

De momento, nos conformaremos con lo expuesto y resolveremos el problema.

Ahora retomaremos a una vieja conocida: **str(**algoqueconvertiratexto**)**

Esta función también puede convertir una lista a texto. Usadlo en este programa: **str (** lista **)**

Solución

Algoritmo:

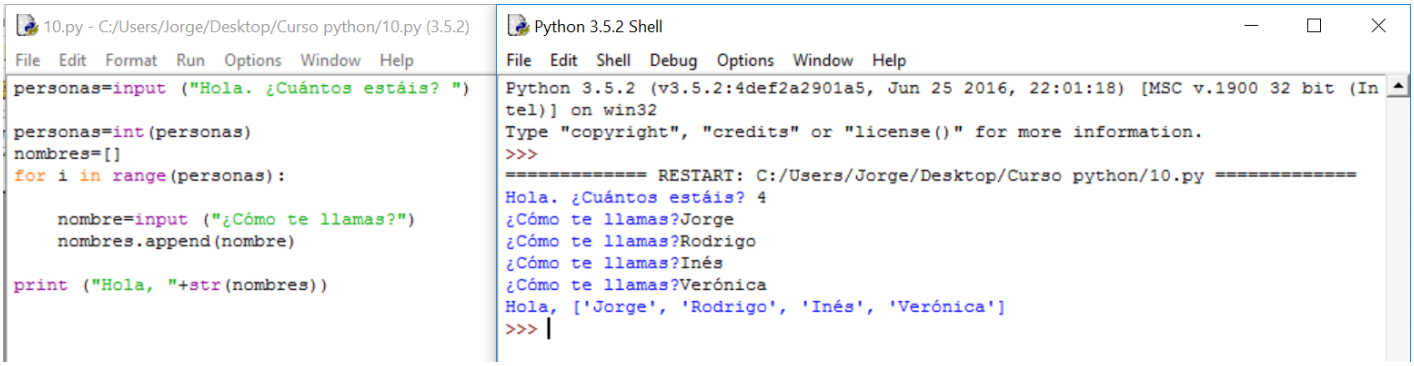
1.- Saludo y pregunto cuánta gente hay

2.- Para cada uno de ellos:

2.1.- Pido el nombre y lo voy almacenando en algún sitio: Una lista.

3.- Saludo a todos poniendo la lista después de "Hola, ".

Solución:



Comentarios:

Como se ve, la transformación a texto de la lista es bastante mala para presentarla en pantalla. A lo largo de los dos ejercicios siguientes vamos a ver cómo solucionarlo.

Programa 9

Descripción del problema:

Este programa va a solucionar el problema de visualización del programa 8 cambiando la cadena de texto progresivamente. Es decir, una variable de texto va a incluir el saludo y los nombres de cada uno de los presentes. Para hacerlo, aunque hay muchas posibilidades, se va a modificar el programa 8. Esto implica que se van a seguir guardando los nombres en una lista.

Materia nueva:

Aquí hay varias apreciaciones respecto a la estructura **for**.

Si nos paramos a pensar, el conjunto de los números enteros es como una lista. Los días de la semana forman otra lista, y un largo etcétera. Tenemos un elemento inicial y otros siguientes ordenados; eso define una lista.

Pues bien, **for** puede, no sólo recorrer una lista de enteros sino también una de cualquier otra cosa.

Su sintaxis es muy parecida a la que vimos en el programa 7 pero usando la lista dentro de **range**:

```
python
for i in range(lista):
```

Si sólo fuera cuestión de números, podríamos haber simulado una estructura for por medio de un while:

```
i=0

while i<numerofinal:
órdenes
i=i+1
```

Si **for** se ha ganado un puesto en Python es por la potencia que otorga esta posibilidad.

En este programa vamos a hacer referencia a dos situaciones más que van indisolublemente unidas:

- El cambio recurrente en las variables.
- La lectura de programas.

¿Por qué van unidas? Pues muy sencillo, porque para entender bien qué estamos realizando si cambiamos una variable cada vez que se ejecuta un paso en un **for** es necesario tener las ideas muy claras, y eso lo da la lectura de programas; debemos ser capaces de seguir la traza de las diferentes variables a lo largo del programa para saber cómo van a acabar.

¿Qué es un cambio recurrente? Es un cambio que se repite, generalmente dentro de una estructura tipo **while** y **for**.

No es algo del todo nuevo, hemos realizado cambios como veces=veces+1 donde hemos cambiado veces por su siguiente. La cuestión es que ahora lo meteremos en una estructura que va a realizar ese cambio un montón de veces. No hay ningún cambio de sintaxis, esta explicación no va a parar a ninguna orden o estructura nueva, es simplemente para que tomes conciencia del paso tan grande que se va a realizar.

A continuación te presento un método para hacer una lectura de programa. No es la única forma y cada uno puede hacerlo según le parezca, eso sí, el orden es imprescindible.

Se pone: “Situación inicial” y se escriben las variables tal y lo que valen en su inicio.

Cuando hay un cambio, se escribe debajo de la variable el nuevo valor o la salida por pantalla. Así se va realizando hasta el final.

Es necesario poner la cadena de texto entre comillas para señalar que es texto.

Se representa a continuación el del programa 7:

|Situación inicial|||||Pantalla| |--|--|--|--|--|--|--| |Personas (entero)|i|nombre|edad|veces|Hola, ¿Cuántos estáis?| |"3"|||||| |3|||||| ||0||||| ||||||¿Cómo te llamas?| |||"Jorge"|||| ||||||¿Cuántos años tienes?| ||||27||| |||||1|| |||||"1"|| ||||27||| ||||||Te hemos preguntado 1 veces| ||||||Pon orden, Jorge| ||1||||| ||||||¿Cómo te llamas?| |||Rodrigo|||| ||||||¿Cuántos años tienes?| ||||"asd"||| |||||1|| ||||||Pon un número, que no es tan difícil| ||||"16"||| |||||2|| |||||"2"|| ||||16||| ||||||Te hemos preguntado 2 veces| ||||||Estudia, Rodrigo| ||2||||| ||||||¿Cómo te llamas?| |||"Inés"|||| ||||||¿Cuántos años tienes?| ||||"14"||| ||||||Te hemos preguntado 1 veces| ||||||Estudia, Inés| |3|2|"Inés"|14|1|||

Situación final. En el caso de que quisiéramos ver otra situación, nos moveríamos a la línea en cuestión y miraríamos el valor de cada una de ellas. Las celdas en blanco no implican que las variables no tengan valor, su valor es el del último cambio: el primero que esté subiendo por esa columna desde la fila donde se quiere saber el valor. Simplemente se ha hecho así por limpieza e identificar fácilmente cuál es la variable que está cambiando en cada momento.

Se deja como ejercicio hacerlo para el presente programa.

Solución

Algoritmo:

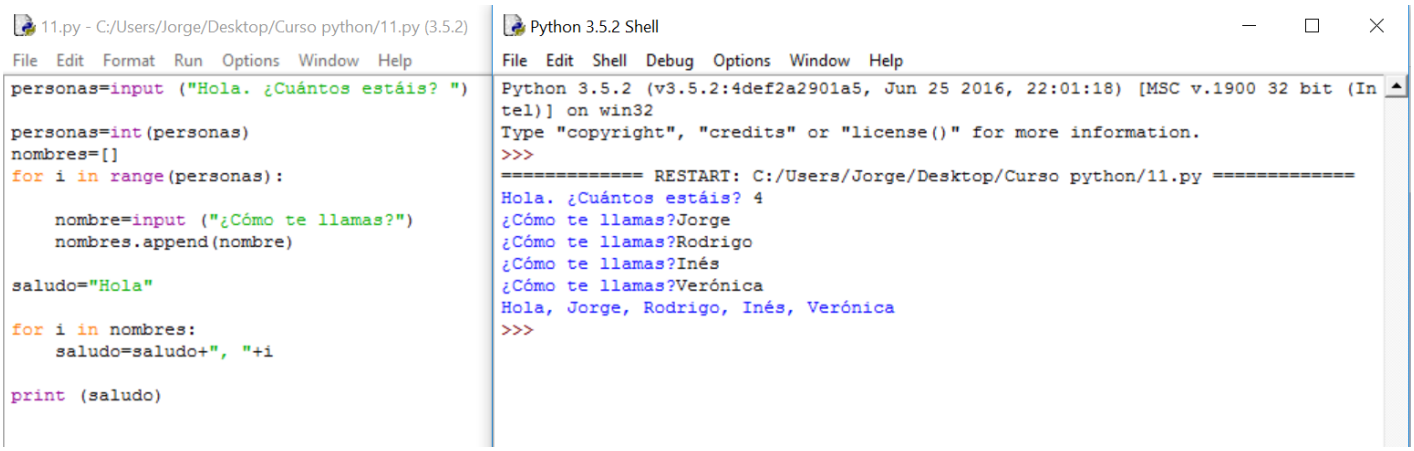
1.- Saludo y pregunto cuánta gente hay

2.- Para cada uno de ellos:

2.1.- Pido el nombre y lo voy almacenando en algún sitio: Una lista.

3.- Modifico la variable donde haya puesto "Hola, " para ir añadiéndole nombres según recorro la lista.

Solución:



Comentarios:

Es necesario hacer la tabla indicada en el apartado de **Materia nueva** para entender qué está pasando con la variable **saludo**. Queda como ejercicio.

Queda como ejercicio conseguir que entre **Inés** y **Verónica** haya una **y.**

Programa 10

Descripción del problema:

Se va a modificar el problema 8 pero usando una lista y recorriéndola sólo una vez. Es decir, deberemos encontrar una forma de pasar de la lista a una Cadena de caracteres.

Materia nueva:

Una Cadena de caracteres es, sin lugar a dudas, una lista: Tiene un carácter inicial y el resto van ordenados. Es como una lista de caracteres.

Python, debido a lo mucho que se utilizan las Cadenas de caracteres, les ha reservado un tipo de variable especial, como hemos visto a lo largo del curso, y no nos hace trabajar todo el rato con listas para escribir algo, por suerte. Sin embargo, no las ha separado del todo y mantienen varias órdenes en común con las listas. Este programa está hecho sólo para sensibilizar sobre las Cadenas de caracteres y abrir una ventana en algo que se había dado como acotado.

A continuación, se va a presentar una orden referida a las Cadenas de caracteres, necesaria para solucionar el problema que tenemos entre manos, y aprovecharemos para practicar la lectura de la documentación de Python. Tal y como se puede leer

https://docs.python.org/3.1/library/stdtypes.html#string-methods:

str.join (**iterable**)

Return a string which is the concatenation of the strings in the **iterable**. A Type error will be raised if there are any non-string values in **seq**, including bytes objects. The separator between elements is the string providing this method.

Si se ha abierto el enlace, vemos que algunas de las palabras son enlaces que nos llevan a la página donde se explica lo que es. De momento, lo que con este curso se puede leer es poco pero haremos el intento y te pido que lo hagas con **.isdigit()** que también está en ese enlace.

Si traducimos casi literalmente: Devuelve una Cadena que es la concatenación de cadenas presentes en el "iterable" (quiere decir: A container object capable of returning its members one at a time. Un objeto contenedor capaz de devolver sus miembros de uno en uno). Se lanzará un error de tipo (ya vimos uno en los primeros programas antes de introducir **str()**) si no hay valores tipo Cadena en la secuencia, incluyendo Cadenas de bytes (fuera del alcance de este curso y novedad en Python 3.0). El separador entre elementos es la Cadena sobre la que se le aplica el método (lo

primero que se pone antes del punto: **str**).

Lo que, traducido, viene a decir:

separador **.join(**cadenasaañadir**)** Esta orden junta las cadenas que tenga la variable **cadenasaañadir** separándolas con la cadena **separador**.

¿Qué puede ser **cadenasaañadir**? Como se ha visto, cualquier cosa que tenga cadenas, en nuestro caso será una lista pero hay muchas más posibilidades.

De momento, no hay que preocuparse porque la orden la he traducido y no se te va a pedir que uses ninguna otra sin explicarla. De todas formas, era necesario abrir el campo a un aprendizaje posterior, aunque quede sólo para los más osados ya que hay más posibilidades para aprender sin lidiar con semejante redacción, lo normal es que sólo aquellos que tengan un conocimiento avanzado se atrevan.

En los ejercicios de autoevaluación se seguirá trabajando con Cadenas de caracteres. Es un tema muy extenso y es muy necesario trabajarlo.

Solución

Algoritmo:

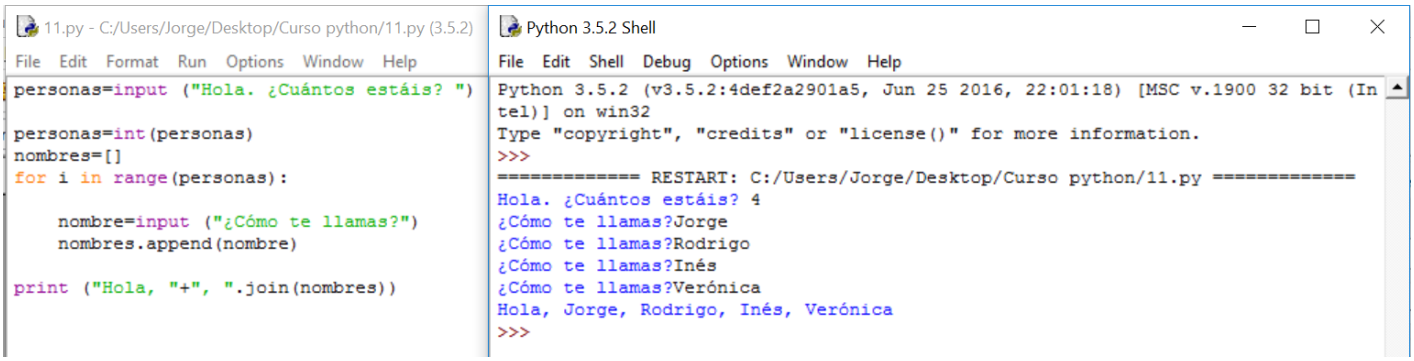
1.- Saludo y pregunto cuánta gente hay

2.- Para cada uno de ellos:

2.1.- Pido el nombre y lo voy almacenando en algún sitio: Una lista.

3.- Convierto la lista en una Cadena de caracteres que poder sumarle al saludo "Hola, ".

Solución:



Ejercicios de autoevaluación

Felicidades por haber finalizado el Módulo 2. A continuación se muestran los programas que servirán de repaso a lo aprendido:

1.- Realiza un programa que "pinte" rectángulos, para ello debe pedir un carácter, la anchura y la altura. El resultado debe ser un rectángulo en la pantalla.

Este programa es complejo. Es necesario diseñar muy bien el algoritmo y pensar muy detenidamente qué es un rectángulo.

Para finalizar cada línea y que salte a la siguiente, debemos introducir un carácter de final de línea. Son caracteres especiales que indican el final de línea, de archivo... El que os hace falta es **"\n"** Ponedlo entrecomillado, al fin y al cabo es un carácter.

2.- Realiza un programa que calcule los divisores de un número que ponga el usuario. Al final, debe preguntar si continúa o no. Si continúa, debe pedir otro número. Debe admitir, para salir, una "n" o una "N".

En este programa, sabemos que basta con dividir por todos los números menores que él hasta su raíz cuadrada pero no nos complicaremos, es mejor dividir por todos.

La parte de la opción de salida "n" o "N" va a alterar la condición de salida. Para incluirle otra condición más en la comparativa de cualquier **if** o **while**:

Si es una "o" lógica: Condición1 **or** Condición2

Si es una "y" lógica: Condición1 **and** Condición2

3.- Realiza un programa en el que el usuario va a introducir un texto y el ordenador va a decirle el número de vocales que tiene.

En este caso, se podrían usar las funciones de **string** de la página que se indicó en el programa 10. Una sería **str.lower(s)**y otra **str.count(** sub, start, end **)**

Se recomienda intentarlo con esas funciones también, pero se pide sin ellas. Si hubiera que buscar conjuntos de letras mayores sí sería necesario.

4.- Realiza un programa que guarde los números que el usuario vaya introduciendo; el criterio de finalización es cuando introduzca algo que no sea un número. Una vez que haya terminado, el usuario debe indicar un número que haya introducido y el ordenador deberá decirle el puesto en el que lo metió.

Es necesario realizarlo con una lista. Se podría realizar con una orden de las que aparecen en la página que se indicó en el programa 8: lista**.index(**elemento**)**

Soluciones a los ejercicios de autoevaluación

autoeval2.1.py - C:\Users\Jorge\Desktop\Curso python\autoeval2.1.py (3.5.2)

File Edit Format Run Options Window Help

```
caracter=input ("Elige el caracter del que quieres un
ancho=int(input("Elige la anchura: "))
altura=int (input("Elige la altura: "))
linea=""
rect=""
for i in range(ancho):
    linea=linea+caracter

for i in range(altura):
    rect=rect+linea+"\n"

print (rect)
```

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 (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Jorge\Desktop\Curso python\autoeval2.1.py ========
Elige el caracter del que quieres un rectángulo: *
Elige la anchura: 4
Elige la altura: 7
****
****
****
****
****
****
****

>>>
```

autoeval 2.2.py - C:\Users\Jorge\Desktop\Curso python\autoeval 2.2.py (3.5.2)

File Edit Format Run Options Window Help

```
continua=True
while continua:

    numero=int(input("Introduce un número: "))
    salida=""
    for i in range(1,numero+1):
        if numero%i==0:
            salida=salida+str(i)+"\n"
    print (salida)
    respuesta=input("¿Quieres continuar? (S/N): ")
    if respuesta=="N" or respuesta=="n":
        continua=False
```

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 (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Jorge\Desktop\Curso python\autoeval 2.2.py ========
Introduce un número: 90
1
2
3
5
6
9
10
15
18
30
45
90

¿Quieres continuar? (S/N): s
Introduce un número: 45
1
3
5
9
15
45

¿Quieres continuar? (S/N): n
>>>
```

autoeval 2.3.py - C:/Users/Jorge/Desktop/Curso python/autoeval 2.3.py (3.5.2)

File Edit Format Run Options Window Help

```
vocales=["a","e","i","o","u","À","E","I","O","U","á","é
cuenta=0
texto=input("Introduce un pequeño texto")
for i in vocales:
    for z in texto:
        if i==z:
            cuenta=cuenta+1
print ("El número de vocales es "+str(cuenta))
```

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 (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/Jorge/Desktop/Curso python/autoeval 2.3.py ========
Introduce un pequeño textoJueqoejaczéü
El número de vocales es 7
>>>
```

autoeval 2.4.py - C:/Users/Jorge/Desktop/Curso python/autoeval 2.4.py (3.5.2)

File Edit Format Run Options Window Help

```
continua=True
numeros=[]
while continua:
    numero=input("Introduce un número: ")
    if numero.isdigit():
        numeros.append(numero)
    else:
        continua=False
busca=input("Introduce el número del que quieres saber algo: ")
posicion=0
for i in numeros:
    posicion=posicion+1
    if i==busca:
        lugar=posicion
print("Lo metiste en la posicion: "+str(lugar))

"""Esto es un comentario, se debe introducir con
tres comillas al inicio y al final. Van muy bien
para explicar lo que se hace, ya no para el resto
sino para uno mismo. Estas líneas ni se ejecutan
ni salen por pantalla"""

"""Si se hubiera hecho con las funciones propias
de las listas, nos habriamos ahorrado:
posicion=0
for ...

Dejo la orden para que se vea: """

print("Lo metiste en la posición: "+str(numeros.index(busca)+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 (
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/Jorge/Desktop/Curso python/autoeval 2.4.py ========
Introduce un número: 8
Introduce un número: 9
Introduce un número: 4
Introduce un número: 7
Introduce un número: 12
Introduce un número: 45
Introduce un número: 2
Introduce un número: t
Introduce el número del que quieres saber algo: 7
Lo metiste en la posicion: 4
Lo metiste en la posición: 4
>>>
```