



Universidad Autónoma
de Santo Domingo
PRIMADA DE AMÉRICA / Fundada el 28 de octubre de 1538



Facultad de Ciencias
Escuela de Informática

Programación Python



Fundamentos de Python: Ejercicios Entregables

Descripción

Los siguientes ejercicios son ideales para que practiquen los conceptos que cubrimos en el módulo 1: Fundamentos de Python.

Para realizar la entrega:

- 1) Crear una carpeta con su primer nombre y apellido: *juan_perez*
- 2) Enumerar los ficheros *.py: *ejercicio_01.py, ejercicio_02.py, ejercicio_nn.py*
- 3) Adjuntar una captura de la corrida del código de cada ejercicio.
- 4) Crear un fichero de texto con sus datos: *nombres y apellidos*.
- 5) Tanto los ficheros, como las capturas deben estar dentro de la carpeta del paso #1.
- 6) Subir la carpeta ya sea a github o en el cuerpo de la actividad del aula virtual.

Ejercicio 01

Crear una función que devuelva el número de palabras de una cadena de caracteres que le sea pasada como argumento.

Ejemplo de respuesta:

```
>> cuenta_palabras('hola mundo')  
>> 2
```

Ejercicio 02

Crear el código Python necesario para abstraer los elementos de su nombre y mostrarlos por pantalla.

Ejemplo de respuesta:

```
>> nombre_completo = "Juan Rodolfo Perez Gonzales"
>> Juan
>> Rodolfo
>> Perez
>> Gonzales
```

Ejercicio 03

Crear el código Python necesario para obtener el reverso de una frase.

Ejemplo de respuesta:

```
>> frase = "hola mundo"  
>> 'odnum aloh'
```

Ejercicio 04

Crear el código Python necesario para validar el palíndromo de una palabra.

Ejemplo de respuesta:

```
>> es_palíndromo("amor", "roma")  
>> True
```

Ejercicio 05

Crear el código Python necesario para cambiar de minúsculas a mayúsculas y viceversa una frase.

Ejemplo de respuesta:

```
>> cambiar_caso("hola MUNdo")  
>> "HOLA munDO"
```


Ejercicio 06

Crear el código Python necesario para obtener el hostname de una URL.

Ejemplo de respuesta:

```
>> url = "uasdvirtual.uasd.edu.do/diplomados/my"  
>> obtener_hostname(url)  
>> "uasd.edu.do"
```

Ejercicio 07

Crear el código Python necesario para guardar una lista sencilla de compras. La lista debe ser interactiva, recibir por pantalla los ítem de la lista de compra hasta que el usuario entre el str “parar”. Al final de la corrida, el script debe mostrar el total de ítems agregados a la lista.

Para agregar un ítem a la lista solo basta agregar su nombre.

Al final solo se requiere que se muestre el número total de ítems agregados a la lista de compras.

Ejercicio 08

Modificar el ejercicio #7 para obtener el número de elementos únicos en la lista de compras. Por ejemplo: Si agrego dos veces “arroz”, solo me debería contar uno.

Para agregar un ítem a la lista solo basta agregar su nombre.

Al final solo se requiere que se muestre:

- El número total de ítems agregados a la lista de compras.
- El número de ítems únicos agregados a la lista de compras.

Ejercicio 09

Modificar el ejercicio #8 para agregar más datos a la lista de compras.

Para agregar un ítem a la lista se deberá agregar:

- Nombre del ítem.
- Cantidad.
- Precio.

Al final solo se requiere que se muestre:

- El número total de ítems agregados a la lista de compras.
- El número de ítems únicos agregados a la lista de compras.

Ejercicio 10

Modificar el ejercicio #9 para obtener el **monto a pagar** por cada ítem de la lista de compras.

Para agregar un ítem a la lista se deberá agregar:

- Nombre del ítem.
- Cantidad.
- Precio.
- El **monto a pagar** será igual a la **[cantidad] * [precio]** de cada ítem.

Al final solo se requiere que se muestre:

- El número total de ítems agregados a la lista de compras.
- El número de ítems únicos agregados a la lista de compras.

Ejercicio 11

Modificar el ejercicio #10 para evitar que se pueda agregar un ítem con el mismo **nombre** a la lista de compras.

Ejercicio 12

Modificar el ejercicio #11 para obtener una representación tabular de todos los datos de los ítems de la lista de compras.

Se debe mostrar en pantalla cada uno de estos datos:

- Nombre del ítem.
- Cantidad.
- Precio.
- El **monto a pagar**; que será igual a la **[cantidad] * [precio]** de cada ítem.

Ejercicio 13

Modificar el ejercicio #12 para obtener el ítem con mayor costo (costo a pagar).

Se debe mostrar en pantalla la salida del ejercicio #12 más:

- *El producto más costoso es: "producto x"*

Ejercicio 14

Modificar el ejercicio #13 para obtener el ítem con menor costo (costo a pagar).

Se debe mostrar en pantalla la salida del ejercicio #13 más:

- *El producto más económico es: “producto x”*

Ejercicio 15

En base al ejercicio #14 crear el código Python necesario para guardar la salida de ese ejercicio en un fichero de texto.

Se debe mostrar en pantalla la salida del ejercicio #14 más:

- Se ha guardado la lista en el fichero: nombre.txt

Si necesitas ayuda con el manejo de ficheros recuerda repasar el material de la clase extra #2, al igual que este [repositorio](#) donde hace unos años realice un ejemplo del manejo de ficheros con una clase bien sencilla de implementar.

Enlace directo: <https://github.com/cjgarcia/archpy>

Ejercicio 16

Leer el contenido del fichero generado en el ejercicio #15, con la lista de compras y realizar las siguientes tareas:

- **Crear una función para buscar un ítem por su nombre.** Esta función debe retornar todos los datos del ítem en formato de diccionario.
- **Crear una función para agregar un nuevo ítem a la lista de compras.** Esta función debe tener como argumento, un diccionario con los datos del ítem y validar el nombre del ítem.
- **Crear una función para eliminar un ítem de la lista.** Esta función debe eliminar el ítem por su nombre.
- **Crear una función para guardar los cambios en el fichero txt.**

Distribución de Puntos

Ejercicio	Puntuación
1, 2, 5, 8, 10, 11, 12, 13, 14, 15	10 x 0.5
3, 4, 6, 7, 9	5 x 1
16	1 x 5
Totales	15