



**UNIVERSIDAD  
DE GRANADA**

**Python: Manejo de Estructuras de Datos.  
Strings, Listas y Diccionarios  
Práctica 2 – Guión 7**

**Informática Aplicada a la Biología**

Dpto. Ciencias de la Computación e Inteligencia Artificial  
E.T.S. de Ingenierías Informática y de Telecomunicación  
Universidad de Granada



## Índice de contenido

1. Introducción .....	3
2. Ejercicios obligatorios .....	4
2.1 Cadenas de Texto (Strings).....	4
2.2 Listas de variables (list) .....	4
2.3 Diccionarios (dict) .....	5
3. Ejercicios sobre bioinformática .....	6
4. Ejercicios voluntarios .....	7

## 1. Introducción

El objetivo de esta práctica es que el alumno se familiarice con algunas de las principales estructuras de datos de python: `string`, `list` y `dictionary` (cadenas de texto, listas y diccionarios).

Es necesario insistir que las *transparencias de teoría* contienen todas las indicaciones necesarias para poder realizar la sesión de prácticas actual. Chequea el contenido de las mismas para poder trabajar cómodamente de forma autónoma.

Por último, se recuerda al estudiante a hacer uso de la ayuda integrada en el entorno como guía para investigar los elementos básicos de la interface y resolver la relación de ejercicios propuesta.

Recordamos que todos los ejercicios deben realizarse con el editor de Spyder, en el correspondiente fichero-py con nombre `P2_<num_ejercicio>.py` para no perder lo hecho (Ejemplo: `P2_1.py` `P2_2.py` `P2_3.py` para los ejercicios 1, 2 y 3 respectivamente y así en adelante).

Para el correcto desarrollo de esta sesión de prácticas, se deben realizar los siguientes pasos:

- **Debemos abrir Spyder.**
- **Elegimos como Carpeta de Trabajo el escritorio o una carpeta que hayamos creado para trabajar (**este paso es importante**).**
- **Recuerde usar `print` para que los cálculos (o resultados) sean visibles al ejecutar los scripts.**
- **Subiremos todos los archivos generados (incluso los no acabados) en la tarea creada en PRADO para cada sesión antes de irnos.**

*Muy importante recordar estos aspectos ya indicados en la primera práctica:*

- La resolución de los problemas y actividades puede hacerse en grupo, pero la entrega durante las horas de prácticas es individual.
- La nota final de cada sesión dependerá del número de ejercicios entregados. Sin embargo, es preferible entregar solamente aquéllos que se sepan resolver.
- En caso de detectar algún tipo de copia, o una implementación que no se sepa explicar con claridad, se evaluará con un 0.
- Es muy importante que la asignatura se lleve al día para poder realizar los ejercicios propuestos en estos guiones.
- Debemos tener instalado *Anaconda* en casa tanto para realizar trabajo autónomo, como para la resolución del trabajo autónomo de Python.

## 2. Ejercicios obligatorios

### 2.1 Cadenas de Texto (Strings)

1. Crear 3 variables de tipo “string” (cadena de texto) con el contenido que quieras. A continuación, probar lo siguiente:
  - a. Imprime las 3 variables con 3 sentencias “print” diferentes.
  - b. Imprime las 3 variables en una única sentencia print como: “Cadena1: <contenido1>, Cadena2: <contenido2>, Cadena3: <contenido3>” donde contenido1, contenido2, y contenido3 es el valor asignado a cada cadena.
  - c. Imprime la primera letra de la primera cadena, después la cuarta letra de la segunda cadena y luego la última letra de la tercera cadena. Pista: [ ? ]
  - d. Imprime las tres cadenas anteriores todas en mayúsculas. Pista: funciones de string.
  - e. Por último, imprime la longitud de cada una de las tres cadenas anteriores.
2. Sustituye todas las apariciones del carácter ‘s’ de una palabra por el símbolo ‘\$’. Por ejemplo, la palabra “soluciones” quedaría como “\$olucionese\$”.
3. Amplía el ejercicio anterior para que modifique todas las apariciones del primer carácter por ‘\$’. Por ejemplo, la palabra “cocotero” quedaría como “\$o\$otero”.
4. Dadas dos palabras “string1” y “string2”, crea una frase “string3” como unión de las dos anteriores. Por ejemplo, string1 = “hola”, string2 = “mundo”, string3 = “hola mundo”. ¡No olvides el espacio intermedio entre ambas!
5. Elimina el carácter “n-ésimo” de un string no vacío. Por ejemplo, si se decide eliminar el carácter 4 de la palabra “Python” quedaría como “Pyton”. ¡Cuidado! No es directo “eliminar” datos para las cadenas de texto, hay que usar un pequeño truco.

### 2.2 Listas de variables (list)

1. Imagina que gestionas un refugio de animales. Crea una lista “animales” con aquéllos que mantengas en dicho refugio, con al menos 4 tipos diferentes. Ahora haz las siguientes tareas:
  - a. Muestra en pantalla todos los animales de tu lista.
  - b. Sustituye un animal cualquiera por un “perezoso”.
  - c. Usa la función “index” para encontrar en qué lugar se ha guardado el “perezoso”.
  - d. Usa la función “insert” para insertar en esa posición una “pitón”.
2. Crea una lista de números, con al menos 4 valores (enteros o reales).
  - a. Imprime el resultado de sumar el primer valor con el tercero.
  - b. Ahora la multiplicación del segundo con el cuarto.
3. Crea una lista vacía llamada “maleta”.
  - a. Añade varios “ítems” secuencialmente (ej. gafas, calcetines...).
  - b. Imprime el número de elementos de tu maleta y a continuación imprime la variable “maleta”, es decir, su contenido completo (como hiciste con los “animales”).
  - c. Por último, imprime únicamente los dos primeros ítems, y a continuación los dos últimos. Pista: recuerda el famoso “slicing” con el uso de [X:Y]

## 2.3 Diccionarios (dict)

1. Dado el siguiente código para asociar diferentes huéspedes con su número de habitación:

```
residentes = {'Pepito':101, 'Jaimito':703, 'Juanita':502}
```

imprime cuál es la habitación de “Juanita”.

2. Crea una variable de tipo diccionario que contenga un menú (dale dicho nombre). Considera los siguientes
  - a. Las claves serán los nombres de los platos, y el valor será su precio.
  - b. Añade al menos 4 platos, imprime primero el número de elementos de tu menú y continuación imprime la variable “menu”.
  - c. Cambia el precio de uno de tus artículos.
  - d. Por último, elimina uno de los platos “menos populares” (a tu elección).
3. Chequea el siguiente código que muestra el ejemplo de un inventario para un juego de arcade:

```
inventario = {  
    'oro' : 500,  
    'bolsa' : ['yesca', 'cuerda', 'gema'], # Asigna nueva lista a la clave 'bolsa'  
    'mochila' : ['xilofono','daga', 'saco de dormir','trozo de pan']  
}
```

```
# Añade una clave 'bolsa tela' y asignar una lista a la misma  
inventario['bolsa tela'] = ['manza', 'rubi', 'perezoso']
```

```
# Ordena la lista que hay en la clave 'bolsa'  
inventario['bolsa'].sort()
```

Ahora haz los siguientes cambios:

- a. Añade una nueva clave al inventario que se llame “bolsillo” y que sea una lista que contenga “concha”, “baya extraña” y “pelusa”. Pista: los nuevos elementos se añaden “solos”, simplemente hay que “nombrarlos”.
- b. Ordena los elementos que tienes en la “mochila”. Pista: usa una función llamada “**sort**”
- c. Por último, elimina el objeto “daga” de la mochila e incrementa en 50 la cantidad de oro.

## 3. Ejercicios sobre bioinformática (extra)

1. Es bien conocido que todo nucleótido tiene su complementario, a saber, A y T son complementarios, así como C y G. Por tanto, podemos tener una cadena de ADN **AGTCGCATAGT** y su complementaria **ACTATGCGACT**.

Cuidado porque la lectura de la cadena parece incorrecta, pero hay que asumir la dirección o comienzo (5') y fin (3') de la cadena. De este modo, los hilos o cadenas de ADN que son complementarias se muestran en direcciones opuestas: 5' -> 3'.

Como podrás intuir, en este ejercicio tendrás que calcular el complementario de una cadena de ADN. Para ello, nada tan fácil como primero calcular el inverso de la cadena (“dar la vuelta”) y luego tomar el complementario de cada letra (gen) de la cadena.

- a. Calcula el “inverso” de una cadena. Si la cadena es AAAACCCGGT se obtendrá TGGCCCAAAA. Como “pista” utiliza los índices que hemos visto en clase para acceder a las posiciones del `string`.
  - b. Ahora calcula el “complementario” de una cadena de ADN. Si la cadena es AAAACCCGGT el resultado será TTTTGGGCCA. Como pista, utiliza las funciones auxiliares para el tratamiento de un `string` vistas en clase (y a lo largo de esta práctica).
  - c. Ahora tendrás que unificar ambos pasos para obtener el resultado final.
2. En este ejercicio, dada una cadena de ARN que representa el codón de inicio, debes imprimir cuál es el aminoácido que codifica. En lugar de utilizar los más de 20 aminoácidos que existen céntrate en los 9 esenciales: *Isoleucina*, *Leucina*, *Lisina*, *Metionina*, *Fenilalanina*, *Treonina*, *Triptófano*, *Valina* e *Histidina*. Utiliza las tablas siguientes para ayudarte a crear un diccionario con el que resolver este problema.

Aminoácido	Codones					
Alanina	GCA	GCC	CGC	GCU		
Arginina	AGA	AGG	CGA	CGC	CGG	CGU
Aspartato	GAC	GAU				
Asparagina	AAC	AAU				
Cisteína	UGC	UGU				
Glutamato	GAA	GAG				
Glutamina	CAA	CAG				
Glicina	GGA	GGC	GGG	GGU		
Histidina	CAC	CAU				
Isoleucina	AUA	AUC	AUU			
Leucina	UUA	UUG	CUA	CUC	CUG	CUU
Lisina	AAA	AAG				
Metionina	AUG					
Fenilalanina	UUC	UUU				
Prolina	CCA	CCC	CCG	CCU		
Serina	AGC	AGU	UCA	UCC	UCG	UCU
Treonina	ACA	ACC	ACG	ACU		
Triptófano	UGG					
Tirosina	UAC	UAU				
Valina	GUA	GUC	GUG	GUU		
Stop	UAA	UAG	UGA			

## 4. Ejercicios voluntarios

En esta última sección, se plantean diversos ejercicios que a priori NO serán puntuables para las prácticas.

El objetivo de los mismos es servir como refuerzo mediante trabajo autónomo o mediante resolución por el profesor en pizarra.

1. Escriba un script que pida al usuario un valor de temperatura C en grados celsius y muestre las equivalencias a grados Fahrenheit, Kelvin y Reamur

$$F = 9/5 (C - 32)$$

$$K = C + 273,15$$

$$R = 8/10 C$$

2. Escriba un script que pida las coordenadas de dos puntos en el plano,  $(x_1, y_1)$ ,  $(x_2, y_2)$ , y calcule los coeficientes  $(a, b, c)$  de la ecuación general de la recta  $ax+by+c=0$  que los une. El cálculo de los coeficientes se realiza mediante las expresiones:  $a=y_2-y_1$ ,  $b=x_1-x_2$ ,  $c=y_1 x_2 - y_2 x_1$ . Muestre el resultado en la forma  $ax+by+c=0$  utilizando los valores calculados.
3. Un objeto con temperatura inicial  $T_0$  que se coloca en tiempo  $t = 0$  en un refrigerador cuya temperatura constante es  $T_s$ , sufrirá un cambio de temperatura dado por la ecuación

$$T = T_s + (T_0 - T_s) e^{-kt}$$

donde T es la temperatura del objeto en el tiempo t y k es una constante. Asumiendo  $k = 0.45$ , ¿Cuál será la temperatura de una lata después de 1,2 y 3 horas si su temperatura inicial era 120° F y el refrigerador funciona a 38°F ?. El resultado debe redondearse al entero más cercano.

4. El número de combinaciones posibles  $C_{n,r}$  para tomar r objetos de un conjunto de n se define como

$$C_{n,r} = n! / (r! * (n-r)!)$$

Escriba un script que solicite los valores de n y r y calcule  $C_{n,r}$  (utilice la función “factorial”)

5. Escriba un script que pida las coordenadas de dos puntos en el plano,  $(x_1, y_1)$ ,  $(x_2, y_2)$ , e informe de la distancia euclídea entre ellos. Una vez mostrado el resultado debe pedir un tercer punto  $(x_3, y_3)$  e informar del área del triángulo formado tomando dichos puntos como vértices.

6. La distancia  $d$  desde un punto  $(x_0, y_0)$  a la línea  $ax+by+c = 0$  es

$$d = \frac{|a x_0 + b y_0 + c|}{\sqrt{a^2 + b^2}}$$

Escriba un script que solicite los valores  $(x_0, y_0)$  y determine la distancia del punto a la línea  $3x+5y-2 = 0$ . Utilice las funciones `sqrt` y `abs`. Generalícelo para que funcione con cualquier línea de la forma  $ax+by+c = 0$ .