

## Bucles / Listas - Lista de ejercicios - Soluciones en el curso

1.- Realizar un programa que **imprima en pantalla los números del 1 al 100 (incluidos los dos)**.

Lo haremos con las diferentes opciones

- while
- for (usando "range")

2.- Realizar un programa que **imprima en pantalla los números impares del 1 al 100 (incluidos los dos)**..

Lo haremos con las diferentes opciones

- while
- for (usando "range") y sin usar un condicional, solo con el uso del rango

3.- **Lo mismo que en el 2**, pero en vez de números impares, los **valores que son múltiplos de 3 (incluidos los dos)**..

4.- Realizar un programa que **imprima en pantalla los números del 100 al 1 (incluidos los dos)**..

Lo haremos con las diferentes opciones y al terminar, escribimos "Descontado finalizado"

- while
- for (usando "range")

5.- Realizar un programa que **imprima en pantalla los números impares del 100 al 1 (incluidos los dos)**..

Lo haremos con las diferentes opciones y al terminar, escribimos "Descontado finalizado"

- while
- for (usando "range") y sin usar un condicional, solo con el uso del rango

6.- Lo mismo que en el 5, pero en vez de números impares, los valores que son múltiplos de 3 **(incluidos los dos)**.

**Bucles / Listas - Lista de ejercicios - Soluciones en el curso**

7.- Basándonos en lo que hemos aprendido antes, vamos a añadir algunos casos como los siguientes.

- a) Imprimir los números del 1 al 75 (incluido).
- b) Imprimir los números del 50 al 100 (incluido).
- c) Imprimir los números del -50 al 0 (incluido).

Especificamos el valor inicial y final. Si el inicial es menor que el final, cuenta de manera ascendente y si no, de manera descendente. No hay rango de números a introducir, vamos a procurar que entre el menor y el final no haya una diferencia mayor de 100 números.

8.- Escribe un programa que haga una **suma de todos los números que tenemos dentro de una lista**.

**Solo** se permitirán los números que sean de tipo **float / int**.

Si alguno de ellos no lo es, lo ignoramos sin hacer ninguna suma.

Casos de uso / Resultado:

[1, 2, 3, 4, 5, 6, 7]	28
[1, 'Anartz', 4]	5
[1, 'Anartz', True, 90]	91

**Bucles / Listas - Lista de ejercicios - Soluciones en el curso**

9.- Realizamos un programa similar al anterior, pero en este caso NO se ignora, que vamos a procurar introducir solo número enteros desde teclado. Decimos cuantos números y luego pedimos esa cantidad de números. Mínimo un número y hacemos la suma.

Casos de uso / Resultado:

¿Cuantos números introduciremos? 4 (Introducimos los números desde teclado: 1, 3, 4, 5	13
¿Cuantos números introduciremos? 9 (Introducimos los números desde teclado: 1, 3, 4, 5, 4, 5, 6, 6, 6	40

10.- Escribe un programa que haga una **multiplicación de todos los números que tenemos dentro de una lista**.

**Solo** se permitirán los números que sean de tipo **float / int**.

Si alguno de ellos no lo es, lo ignoramos sin hacer ninguna **multiplicación**.

Casos de uso / Resultado:

[1, 2, 3]	6
[1, 'Anartz', 4]	4
[1, 'Anartz', True, 90]	90

**Bucles / Listas - Lista de ejercicios - Soluciones en el curso**

11.- Similar a lo del 9, donde con esos números introducidos haremos el paso **para hacer la multiplicación de todos los elementos**.

Casos de uso / Resultado:

¿Cuántos números introduciremos? 4 (Introducimos los números desde teclado: 1, 3, 4, 5)	60
¿Cuántos números introduciremos? 2 (Introducimos los números desde teclado: 1, 20)	20

12.- Escribe un programa que haga nos diga cual **es el número superior de una lista dada**.

**Solo** introducir números.

Casos de uso / Resultado:

[1, 2, 300, 4, 5, 6, -10, 100]	300
[167, -1345, 34, 555, 5666, 55677]	55677
[178293, 93902, 2819, -120293939]	178293

Resultado en consola:

```
El mayor de la lista: [1, 2, 300, 4, 5, 6, -10, 100]  
=====> 300
```

**Bucles / Listas - Lista de ejercicios - Soluciones en el curso**

13.-Escribe un programa que haga nos diga cual **es el número inferior de una lista dada**.

**Solo** introducir números.

Casos de uso / Resultado:

[1, 2, 300, 4, 5, 6, -10, 100]	-10
[167, -1345, 34, 555, 5666, 55677]	-1345
[178293, 93902, 2819, -120293939]	-120293939

Resultado en la consola:

```
El menor de la lista: [178293, 93902, 2819, -120293939]
=====> -120293939
```

14.- Obtener una **lista sin duplicados**. Eliminar los que ya existan, para tener una lista vacia de valores repetidos (**no se puede hacer uso de set** que lo veremos más tarde).

Casos de uso / Resultado:

[1, 2, 3, 4, 5, 6, 7, 1, 3, 45]	[1, 2, 3, 4, 5, 6, 7, 45]
[167, 'Anartz', 4, 4, 4, 5]	[167, 'Anartz', 4, 5]
[1, 'Anartz', True, 90, True, False]	[1, 'Anartz', True, 90, False]

Resultado en la consola:

```
Lista original: [1, 2, 3, 4, 5, 6, 7, 1, 3, 45]
Lista sin duplicados: [1, 2, 3, 4, 5, 6, 7, 45]
```

## Bucles / Listas - Lista de ejercicios - Soluciones en el curso

15.- Filtrar los valores de una lista que contiene una longitud de n caracteres ó más. Tendremos que añadir información de tipo str

n: Valor que especificamos introduciendo desde teclado

['Java', 'Javascript', 'Go', 'Python', 'Rust']
Introduce nº caracteres: 5
Resultado: ['Javascript', 'Python']

```
Introduce nº caracteres: 5
Lista original: ['Java', 'Javascript', 'Go', 'Python', 'Rust']
Resultado: ['Javascript', 'Python']
```

16.- Escribir un programa, donde añadamos una lista con valores enteros, hagamos un backup de ella (clonando o copiando su contenido) en una nueva lista llamada “backup”.

Una vez realizado el backup, modificamos los valores de la lista original duplicando los valores de cada elemento.

Comprobamos que las listas no son iguales

```
Original: [2, 3, 6, 8, 10]
Copia: [2, 3, 6, 8, 10]
¿Iguales?: True
Original: [4, 6, 12, 16, 20]
Copia: [2, 3, 6, 8, 10]
¿Iguales?: False
```

## Bucles / Listas - Lista de ejercicios - Soluciones en el curso

17.- Especificar mediante teclado, cuantos números vamos a tener en una **lista que debe de ser mínimo 8 elementos**, usamos como base una lista como la que tenemos aquí:

[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]

Luego generaremos una lista de números de 1 hasta el número límite especificado por teclado donde tiene que tener mínimo 8 elementos y en el resultado vamos a obtener los primeros 4 elementos y los últimos dos elementos.

Ejemplo con 9 elementos:

```
Introduce nº elementos: 3
Debes de especificar 8 ó más. 3 no es correcto.
Introduce nº elementos: 9
Lista seleccionada: [1, 2, 3, 4, 5, 6, 7, 8, 9]
Primeros 4 elementos: [1, 2, 3, 4]
Últimos 2 elementos: [8, 9]
```

## Bucles / Listas - Lista de ejercicios - Soluciones en el curso

18.- Crear un acrónimo mediante el uso de un bucle y lo aprendido anteriormente en los strings.

¿Qué es un acrónimo? Los acrónimos son palabras formadas por siglas o abreviaturas. Cada sigla o abreviación representa una palabra, es decir, agrega un significado. Por ejemplo: FIFA, NASA.

- FiFA: Fédération Internationale de Football Association
- NASA: National Aeronautics Space Administration
- NBA: National Basketball Association

Casos de uso y su resultado:

anartz mugika ledo	AML
bootcamp python 3	BP3
graphql angular desde 0	GAD0

En consola:

```
Introduce la palabra a convertir en acrónimo: anartz mugika ledo
Palabra introducida: anartz mugika ledo => AML
```

```
Introduce la palabra a convertir en acrónimo: graphql angular desde 0
Palabra introducida: graphql angular desde 0 => GAD0
```

```
Introduce la palabra a convertir en acrónimo: bootcamp python 3
Palabra introducida: bootcamp python 3 => BP3
```



## Bucles / Listas - Lista de ejercicios - Soluciones en el curso

19.- La sucesión “Mira y di” (en inglés, [Look-and-Say](#)) es una sucesión de números naturales en donde cada término se obtiene agrupando las cifras iguales del anterior y recitándolas. Por ejemplo, si  $x(0) = 1$  se lee como “un uno” y por tanto  $x(1) = 11$ . Análogamente,

```
x(1) = 11    ("dos unos"                --> "21")
x(2) = 21    ("un dos un uno"           --> "1211")
x(3) = 1211  ("un uno un dos dos unos"  --> "111221")
x(4) = 111221 ("tres unos dos doses un uno --> "312211")
x(5) = 312211 ("un tres un uno dos doses dos unos --> "13112221")
```

Lo que se pide es que creamos un programa en el que especificando el número entero inicial entre 0 y 9 (incluidos) y cuantas sucesiones demos un resultado. En el ejemplo anterior, empezamos con 1 y pedimos 5 sucesiones, la idea es crear el programa para ello.