

## Práctica sobre Interacción entre Funciones 1

Crea una función (`lanzar_dados`) que arroje **dos dados al azar y devuelva sus resultados**:

- La función **debe retornar dos valores resultado**, que se encuentren *entre 1 y 6*.
- Dicha función no debe requerir argumentos para funcionar, sino que debe generar internamente los valores aleatorios.

Proporciona el resultado de estos dos dados a una función que se llame `evaluar_jugada` (es decir, esta **segunda función debe recibir dos argumentos**) y que **retorne -sin imprimirlo- un mensaje** según la suma de estos valores:

Si la **suma es menor o igual a 6**:

- `"La suma de tus dados es {suma_dados}. Lamentable"`

Si la suma es **mayor a 6 y menor a 10**:

- `"La suma de tus dados es {suma_dados}. Tienes buenas chances"`

Si la suma es **mayor o igual a 10**:

- `"La suma de tus dados es {suma_dados}. Parece una jugada ganadora"`

**Pistas:** utiliza el método `choice` o `randint` de la biblioteca `random` para elegir un valor al azar entre 1 y 6.

## Práctica sobre Interacción entre Funciones 2

Crea una función llamada `reducir_lista()` que tome una lista como argumento (crea también la variable `lista_numeros`), y **devuelva la misma lista**, pero **eliminando duplicados (dejando uno solo de los números si hay repetidos)** y **eliminando el valor más alto**. *El orden de los elementos puede modificarse*.

Por ejemplo, si se le proporciona la lista `[1,2,15,7,2]` debe devolver `[1,2,7]`.

Crea una función llamada `promedio()` que pueda recibir como argumento la lista devuelta por la anterior función, y que calcule el promedio de los valores de la misma. Debe devolver el resultado, sin imprimirlo.

### Práctica sobre Argumentos Indefinidos (\*args) 1

Crea una función llamada `suma_cuadrados` que tome una cantidad indeterminada de argumentos numéricos, y que retorne la suma de sus valores al cuadrado.

Por ejemplo para los argumentos `suma_cuadrados(1,2,3)` deberá retornar 14 (1+4+9).

### Práctica sobre Argumentos Indefinidos (\*args) 2

Crea una función llamada `suma_absolutos`, que tome un conjunto de argumentos de cualquier extensión, y retorne la suma de sus valores absolutos (es decir, que tome los valores sin signo y los sume, o lo que es lo mismo, los considere a todos *-negativos y positivos-* como positivos).

### Práctica sobre Argumentos Indefinidos (\*args) 3

Crea una función llamada `numeros_persona` que reciba, como primer argumento, un nombre, y a continuación, una cantidad indefinida de números.

La función debe **devolver** el siguiente mensaje:

```
"{nombre}, la suma de tus números es {suma_numeros}"
```

## Práctica sobre Argumentos Indefinidos (\*\*kwargs) 1

Crea una función llamada `cantidad_atributos` que cuente la cantidad de parámetros que se entregan, y devuelva esa cantidad como resultado.

## Práctica sobre Argumentos Indefinidos (\*\*kwargs) 2

Crea una función llamada `lista_atributos` que devuelva en forma de **lista** los **valores** de los atributos entregados en forma de palabras clave (*keywords*). La función debe preveer recibir cualquier cantidad de argumentos de este tipo.

## Práctica sobre Argumentos Indefinidos (\*\*kwargs) 3

Crea una función llamada `describir_persona`, que tome como parámetros su nombre y luego una cantidad indeterminada de argumentos. Esta función deberá mostrar en pantalla:

```
1 | Características de {nombre}:
2 | {nombre_argumento}: {valor_argumento}
3 | {nombre_argumento}: {valor_argumento}
4 | etc...
```

Por ejemplo:

```
describir_persona("María", color_ojos="azules", color_pelo="rubio")
```

Mostrará en pantalla:

```
1 | Características de María:
2 | color_ojos: azules
3 | color_pelo: rubio
```