


## 2 Expresiones y declaraciones.

**Objetivo.** Explicar el concepto de variable, etiqueta, objetos y como se usan mediante algunos ejemplos.

**Funciones de Python:** - `print()`, `type()`, `id()`, `chr()`, `ord()`, `del()`

[MACTI-Algebra\\_Lineal\\_01](#) by [Luis M. de la Cruz](#) is licensed under [Attribution-ShareAlike 4.0 International](#) 

### 2.1 Expresiones

En matemáticas se define una expresión como una colección de símbolos que juntos expresan una cantidad, por ejemplo, el perímetro de una circunferencia es  $2\pi r$ .

En Python una **expresión** está compuesta de una combinación válida de valores, variables, operadores, funciones y métodos, que se puede evaluar y **da como resultado al menos un valor**.

En esencia, una **expresión es cualquier cosa que pueda ser evaluada y producir un resultado**.

Las expresiones pueden ser simples o complejas, pero en general, representan un valor único, por ejemplo:

```
a = 2**32
```

Véase más en [The Python language reference: Expressions](#) y [Python expressions](#).

Veamos algunos ejemplos:

#### Expresiones simples

```
23
```

23

```
5 + 3
```

8

```
a = 5  
a ** 2
```

25

#### Expresión que ejecuta una función

```
len('Hola mundo')
```

10

### Expresiones usando operadores

```
# Otros ejemplos
x = 1
y = x + 2
z = y ** 3

print(x)
print(y)
print(z)

# Operación Booleana
7 == 2 * 2 * 2
```

1  
3  
27

False

### Expresiones más complicadas

```
# Se combinan varias operaciones matemáticas
b = 2.14
c = 0.1 + 4j

(3.141592 * c + b) / a
```

(0.4908318400000001+2.5132736j)

Observa que en todos los ejemplos anteriores se produce al menos un valor como resultado de la ejecución de cada expresión.

## 2.2 Declaraciones

Una **declaración** (*statement*) se puede pensar como el elemento autónomo más corto de un lenguaje de programación. Un programa se forma de una secuencia que contiene una o más declaraciones. Una declaración contiene componentes internos, que pueden ser otras declaraciones y varias expresiones.

En términos simples, una **declaración** es una **instrucción que realiza una acción**.

Puede ser una asignación de valor a una variable, una llamada a una función, una estructura de control de flujo (como un ciclo o una condición), una definición de función, etc.

Véase más en [Simple statements](#), [Compound statements](#) y [Python statements \(wikipedia\)](#).

Veamos algunos ejemplos:

### Declaración que hace una asignación

```
x = 0
```

### Declaración usando un condicional

```
if x < 0:  
    pass
```

### Declaración que realiza un ciclo

```
for i in range(0,5):  
    pass
```

### Declaración de una función

```
def mult(a, b):  
    return a * b
```

Here is a note