

## CLASE 01 - INTRODUCCION

### VARIABLES

Se crean automáticamente al asignarles un valor inicial. Por eso no se declaran y utilizar una variable no inicializada provoca `NameError`.

>**Nombres:** Letras, números y guión bajo. Deben tener sentido con su valor. Evitar “1” o “o”. Nunca coincidir las con palabras reservadas o iniciarlas con números.

>**Datos:** str, float o int, boole.

Asignaciones múltiples (es decir, en un sólo renglón, a) asignar a varias variables el mismo valor o b) asignar varias variables con sus respectivos valores).

>**Impresión:** Python agrega automáticamente un espacio como separador. `f(): print()`

>**Especificadores de conversión:** Se omite la coma. Si hay más de una variable, se encierran entre paréntesis al final del print.

Sintaxis: (ValorDeRelleno)AnchoTotal.CantDecimales

> *%d para números enteros*

```
print("Cantidad: %6d" %cant)           #Cantidad:      10
```

> *%f para números reales*

```
print("Precio: %5.2f" %precio)         #Precio: 12.50
```

>**Relleno con ceros:** ceros delante de un número.

```
print("%04d" %a)                       #0003
```

>**Ingreso de valores** `f(): input()` - siempre devuelve str salvo que se especifique int/float

>**Operadores Aritméticos:** + - \* / // % \*\*

>**Orden de evaluación:**

- 1) Potenciación
- 2) Menos Unario
- 3) Multiplicación
- 4) División

>**Asignación extendida:** +=<var>

## ESTRUCTURAS DE CONTROL

### >ESTRUCTURA ALTERNATIVA:

Formato 1 - `if <condición>: ...`

Formato 2 - `if <condición>: ... // else: ...`

Formato 3 - `if <condición>: ... // elif: ... // else: ...`

### >ESTRUCTURA ITERATIVA:

Formato 1: - `while <condición>: ...`

Formato 2: - `for <variable> in <secuencia>: ...`

>Operadores relacionales: `== > < >= <= !=`

>Operadores lógicos: `and or not`

>Condiciones encadenadas: `a<b>c ...equivale a... a<b and b>c`

>Instrucción *break*: abandona el último ciclo ejecutado

>Cláusula *else*: El código de esta cláusula se ejecuta si el ciclo terminó en forma normal, sin *break*

>Instrucción *continue*: Fuerza una nueva iteración si haber llegado al final de la anterior.

>*f()*: `range(<vinicial>,<vfinal>,<inc>)` - No incluye al último.

---