



# ENGIN 604 INTRODUCCIÓN A PYTHON PARA LAS FINANZAS

## VARIABLES, CONVENCIONES & TIPOS DE DATOS - PAUTA

**Profesor:** *Gabriel E. Cabrera*  
**Ayudante:** *Alex Den Braber*



### 1. Variables

Una Variable en Python es un espacio en la memoria RAM de tu computadora donde puedes guardar resultados y objetos. Para crear una variable ocupamos una “asignación”, es decir, una expresión con una igualdad. Lo que está a la izquierda del signo igual será el nombre de la variable donde quedará guardado el valor, y lo que está a la derecha del igual es el valor a guardar (*Aplicaciones de Manejo de Datos, 2020*).

Por ejemplo, creemos la variable `ncel` y guardemos el valor 12345678:

```
ncel = 12345678 # el simbolo = es el de asignación
ncel
```

```
## 12345678
```

En python no es necesario terminar la linea de código con ; (semicolons), pero si cumple un rol:

```
a = 1; b = 2; c = 3
```

También se puede asignar valores a varias variables en una sola linea de código:

```
a, b, c = 1, 2, 3
```

### 2. Convenciones

1. Existen nombre de variables reservadas que no se deben usar debido que están reservados por Python:

```
# librería
import keyword

# nombres reservados
keyword.kwlist
```

2. Los comentarios simples se realizan con #.

```
# esto es un comentario simple
```

3. Los comentarios multi-line comienzan con ''' y terminan con '''.

```
"""
esto es
un
comentario
multiple
"""
```

### 3. Tipos de Datos (escalares)

Existen cuatro tipos de datos o escalares: numéricos (**integers** y **floats**), **strings** (cadenas de texto) y **booleanos** (True & False).

```
'''Tipos básicos de datos'''
# integers
integer = 2
type(integer) # identificación
isinstance(integer, int) # identificación

# float
floating = 3.14
type(floating)
isinstance(floating, float)

# string
string = "hola mundo"
type(string)
isinstance(string, str)

# boolean
boolean = True
type(boolean)
isinstance(boolean, bool)

'''casting'''
# integers
int('1')
int(True)
int(False)

# float
float('1')
float(True)
float(False)

# string
str(1)
str(True)
str(False)

# boolean
bool(1)
bool(0)
```

1. Cuales son los tipos de datos en Python? ¿Cómo puedo identificarlos y convertirlos en otros tipos de datos?

#### 3.1. Expresiones numéricas

1. ¿Qué operaciones numericas puedo aplicar?

```
a = 4
b = 3

a + b # suma

## 7
```

```

a - b # resta

## 1

a * b # multiplicación

## 12

a / b # división

## 1.3333333333333333

a % b # modulo

## 1

a // b # parte entera

## 1

a ** b # potencia

## 64

```

2. Cual es el resultado de:

$$1 + 2^3/4 \times 5$$

```

1 + 2 ** 3 / 4 * 5

## 11.0

```

3. Calcule el valor presente de un flujo de efectivo de \$1000 por un total de 5 períodos. Asuma una tasa de descuento de 10 %. El valor presente (VP) es igual a:

$$VP = \sum_{t=0}^T \frac{F_t}{(1+r)^t}$$

```

valor_presente = 1000 / (1 + 0.1) ** 1 + 1000 / (1 + 0.1) ** 2 + 1000 / (1 + 0.1) ** 3
valor_presente += 1000 / (1 + 0.1) ** 4 + 1000 / (1 + 0.1) ** 5
print('El valor presente es: ' + str(valor_presente))

## El valor presente es: 3790.7867694084475

```

### 3.2. Cadenas de Texto

1. Considerando el siguiente texto: “*Hay ciertas cosas que el dinero no puede comprar, para todo lo demás existe mastercard*”. Responda:

- a. ¿Cuántos caracteres contiene el texto?

```

texto = 'Hay ciertas cosas que el dinero no puede comprar, para todo lo demás existe mastercard'
len(texto)

## 86

texto[0] # primer caracter

## 'H'

texto[0:5] # primeros 5 caracteres

## 'Hay c'

```

```
texto[-1] # último caracter
```

```
## 'd'
```

- b. ¿Cuántas veces se encuentra la letra r? ¿Cuántas veces se encuentra la letra r luego de la “,”?

```
texto.count('r')
```

```
## 7
```

```
texto.count('r', texto.find(','))
```

```
## 3
```

- c. ¿En que posición está “mastercard”?

```
texto.find('mastercard')
```

```
## 76
```

- d. Reemplace “mastercard” por “Mastercard”.

```
texto.replace('mastercard', 'Mastercard')
```

```
## 'Hay ciertas cosas que el dinero no puede comprar, para todo lo demás existe Mastercard'
```

2. Escriba su nombre completo en minúsculas respetando los espacios.

- a. Imprima en la pantalla su nombre completo en mayúscula. (e.g GUIDO VON ROSSUM)

```
lower_name = "guido von rossum"  
lower_name.upper()
```

```
## 'GUIDO VON ROSSUM'
```

- b. Imprima en la pantalla su nombre completo con solo las primeras letras en mayúsculas. (e.g Guido Von Rossum)

```
lower_name.title()
```

```
## 'Guido Von Rossum'
```

3. Añada el siguiente formato<sup>1</sup> al texto: ‘2.45535 dolares equivale a \$CLP1808.12’

- a. De ‘2.45535’ a 2.45 (dos decimales)

- b. ‘dolares’ a texto

- c. De ‘1808.12’ a 1808 (número entero)

```
template = '{0:.2f} {1:s} equivale a $CLP{2:d}'  
template.format(2.45535, 'dolares', 1808)
```

```
## '2.46 dolares equivale a $CLP1808'
```

4. ¿Cuanto es ‘2’ + ‘25’?

```
'2' + '25' # concatenación, deben ser str()
```

```
## '225'
```

---

<sup>1</sup><https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting>

### 3.3. Lógica Booleana

1. ¿Qué operaciones lógicas puedo aplicar?

Operadores de comparación:

```
a = 4
```

```
b = 3
```

```
a > b # mayor que
```

```
## True
```

```
a < b # menor que
```

```
## False
```

```
a == b # igual a
```

```
## False
```

```
a != b # distinto a
```

```
## True
```

```
a >= b # mayor o igual a
```

```
## True
```

```
a <= b # menor o igual a
```

```
## False
```

operadores lógicos:

```
x = True
```

```
y = False
```

```
x and y # True y False = False
```

```
## False
```

```
x or y # True o False = True
```

```
## True
```

```
not x # no True = False
```

```
## False
```

2. Considerando el siguiente texto: “Hay ciertas cosas que el dinero no puede comprar, para todo lo demás existe mastercard”. Responda:

- a. ¿Se encuentra la letra ‘z’?

```
texto = 'Hay ciertas cosas que el dinero no puede comprar, para todo lo demás existe mastercard'  
'z' in texto
```

```
## False
```

- b. ¿Se encuentra la letra ‘m’?

```
'm' in texto
```

```
## True
```

- c. ¿Se encuentra la palabra ‘visa’?

```
'visa' in texto
```

```
## False
```

3. Muestre que se cumple la ley de De Morgan.

```
A = True
B = False

not(A and B) == (not(A) or not(B))
```

```
## True
```

```
A = False
B = True

not(A and B) == (not(A) or not(B))
```

```
## True
```

```
A = True
B = True

not(A and B) == (not(A) or not(B))
```

```
## True
```

```
A = False
B = False

not(A and B) == (not(A) or not(B))
```

```
## True
```

## 4. Apéndice

Cuadro 1: Ley de De Morgan

A	B	A and B	not(A and B)	not A or not B
True	False	False	not(False) = True	not(True) or not(False) = True
False	True	False	not(False) = True	not(False) or not(True) = True
True	True	True	not(True) = False	not(True) or not(True) = False
False	False	False	not(False) = True	not(False) or not(False) = True