

# Guide 1 Introduction

March 13, 2019

Guía 1: Introducción a Python

Ingeniería en Estadística, Universidad de Valparaíso

Profesor: Eduardo Jorquera - eduardo.jorquera@postgrado.uv.cl

Python es un lenguaje de programación muy utilizado en estadística, *data science*, entre otras materias relacionadas y de desarrollo de *software*.

En python existen las operaciones básicas, tales como la suma (+), multiplicación (\*), potencia (\*\*), división (/), módulo(%), entre otras.

Por ejemplo, podemos imprimir (print) el resultado de una suma:

```
In [1]: print(2+1)
```

6

## 1 Actividad

Haga uso de estas cinco operaciones básicas, usando el comando print.

```
In [ ]:
```

En Python, todo es un objeto, y todo objeto tiene tres propiedades: identidad, tipo, y valor. La identidad es una dirección dada por Python que no cambia. El tipo puede ser cualquiera de los siguientes.

**None:** representa la ausencia de un valor. **Números:** Python tiene tres tipos de números: \* Entero \* Punto flotante (puede almacenar números con decimales) \* Complejo (puede almacenar la parte real e imaginaria de un número)

**Secuencias:** Son colecciones ordenadas de elementos. Hay tres tipos de secuencias en Python: \* Cadenas (strings) \* Tuplas \* Listas

Las cadenas de caracteres son muy utilizadas en python, por ejemplo, creamos una variable con algún nombre. Para esto, al lado izquierdo del signo igual ponemos el nombre de la variable, y al lado derecho del signo igual, el valor que tomará la variable:

```
In [2]: nombre="Pedro"
```

Luego, si quiero conocer el contenido de la variable nombre, simplemente la escribo:

```
In [3]: nombre
```

```
Out [3]: 'Pedro'
```

Como se trata de una cadena de caracteres, cada caracter tiene una posición en la cadena (suena redundante, no?)

Veamos qué tiene la posición 0 (en Python, siempre la primera posición es la 0)

```
In [4]: nombre[0]
```

```
Out[4]: 'P'
```

Así también con la segunda posición:

```
In [5]: nombre[2]
```

```
Out[5]: 'd'
```

Por si no lo has notado, sólo hemos visto valores enteros de las posiciones. Si queremos acceder a una posición enumerada de derecha a izquierda, podemos usar una numeración negativa:

```
In [8]: nombre[-2]
```

```
Out[8]: 'r'
```

Si eres sigiloso, habrás notado que al contar de derecha a izquierda el -0 no es tomada como la última posición en la cadena de caracteres! con la posición -1 puedes acceder a la última letra de la cadena. ¿Por qué crees que pasa esto?

Responde aquí o\_o

También se puede conocer el largo de una cadena de caracteres con el comando len:

```
In [9]: len(nombre)
```

```
Out[9]: 5
```

## 1.1 Actividad

Si no conozco el largo de una cadena de caracteres, cómo puedo saber cuál es el último caracter? use la variable nombre.

```
In [ ]:
```

También podemos juntar varias cadenas de caracteres en una sola, por ejemplo, inventemos un apellido para Pedro:

```
In [15]: apellido="Pérez"
```

Ahora, al tratarse de cadenas de caracteres, podemos usar la suma para unir (concatenar) varias cadenas:

```
In [18]: print(nombre+apellido)
```

```
PedroPérez
```

Genial...! Pero hay algo poco estético ahí, no? podemos agregar un espacio entre el nombre y el apellido, usando el mismo comando print de dos formas distintas, pon atención a su uso!

```
In [22]: print(nombre+" "+apellido)
```

Pedro Pérez

```
In [23]: print(nombre, apellido)
```

Pedro Pérez

También con el comando print podemos agregar cadenas de caracteres, de la forma que más te acomode:

```
In [28]: print(nombre+" "+apellido+" García")
```

Pedro Pérez García

```
In [27]: print(nombre, apellido, "García")
```

Pedro Pérez García

Seguramente ya habrás notado que la coma (,) en el print, agrega un espacio al imprimir.

También podemos imprimir lo que está a la izquierda o a la derecha de una cadena de caracteres:

```
In [50]: nom_comp=nombre+" "+apellido+" García" #nombre completo
         print("Ahora veremos lo que hay después del quinto caracter")
         print(nom_comp[5:])
         print("Y ahora lo que hay antes del sexto caracter (incluyendo el quinto)")
         print(nom_comp[:5])
```

Ahora veremos lo que hay después del quinto caracter

Pérez García

Y ahora lo que hay antes del sexto caracter (incluyendo el quinto)

Pedro

## 2 Actividad

Usando la notación anterior, muestre el último caracter de nom\_comp

```
In [ ]:
```

### 3 TAREA:

- Dado una cadena de caracteres de largo mayor a dos, escriba un código que contenga a los primeros dos caracteres concatenados con los dos últimos.
- Dado una cadena de caracteres de largo mayor a cuatro, escriba un código que quite los primeros dos y los últimos dos caracteres.

In [ ]: