

# Clase 04 - Python: variables, y entrada/salida

IIC1103-07 - Introducción a la Programación

---

Cristian Ruz – `cruz@ing.puc.cl`

Martes 20-Agosto-2019

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

## Laboratorios: desde el Lunes 27-Agosto

- Laboratorios evaluados automáticamente (*hackerrank*)
- Pueden ir a cualquier laboratorio (Lunes a Jueves)
  - SIN LAPTOP
    - Lunes a Jueves, mód 5 y 6: Lab San Agustín (piso 2)
  - CON LAPTOP
    - Martes, mód 5 y 6: K200, A5, B12
    - Miércoles, mód 5 y 6: **B23**, CS203, C203
    - Jueves, mód 5 y 6: B13, K204, CS101
- ¡Este es un curso práctico! Aprender haciendo

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

# Elementos de un algoritmo

**TODOS** los programas que vamos a escribir (y que existen) se pueden construir con:

- Datos de *entrada (input)* que se *leen*
- Datos de *salida (output)* que se *escriben*
- **Variables** que *recuerdan* datos.
- **Operaciones matemáticas** que *calculan* datos.
- Instrucciones **condicionales** que se ejecutan dependiendo de una condición.
- Instrucciones **cíclicas** que se ejecutan múltiples veces dependiendo de una condición.

## Esto entra para la I1:

- Lectura (**input**) y escritura (**print**) de datos
- Uso de **variables** para recordar datos
- Cómputo de **expresiones** usando operaciones matemáticas
- Instrucciones **condicionales**: **if**, **elif**, **else**
- Instrucciones **cíclicas**: **while**, **for**
- ...y funciones ...

### Tipos de datos

```
1 a = 3
2 type(a)          #<class 'int'>
3 a = "3.55"
4 type(a)          #<class 'str'>
5 a = 3.55
6 type(a)          #<class 'float'>
```

### Operaciones v expresiones

```
1 (40 <= 2019 - 1978) and (2019 - 2015 <= 5)
```

# Operaciones sobre texto (str)

+: Concatenación de *strings*

```
1 "Yo soy" + "tu padre"      # "Yo soytu padre"
```

\*: Repetición de *strings* (siempre debe haber un entero)

```
1 "Ja"*5                     # "JaJaJaJaJa"
```

Y podemos mezclarlos

```
1 "N"+"o"*9                  # "Noooooooooo"  
2 ("N"+"o")*9               # "NoNoNoNoNoNoNoNoNo"
```



# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

# Al final del curso sabremos ...

1. Variables y expresiones
2. Control de flujo
3. Funciones y recursión  $\Leftarrow$  I1: 24-Sept, 18:30
4. Strings
5. Listas
6. Tipos de datos personalizados (objetos)
7. Ordenación y búsqueda
8. Archivos

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

# Variables

Si queremos operar sobre resultados previos necesitamos **recordar** valores.

## Variables

Una **variable** es un *nombre simbólico* utilizado para acceder a un valor almacenado en la memoria del programa.



# Variables

## Sintaxis

```
1 nombre_variable = valor
```

## Asignar

Dícese del acto de dar un valor a una variable.

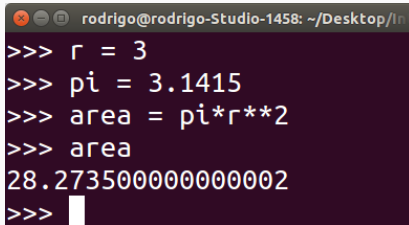
```
1 a = 3-2+9      # Ahora "a" tiene valor 10  
2 b = 4          # Ahora "b" tiene valor 4  
3 pi = 3.1415    # Ahora "pi" tiene valor 3.1415
```

¿Cuál es la gracia?

# Variables

Ahora podemos operar sobre las variables y ver sus valores.

```
1 r = 3                # Asigno valor de r (radio)
2 pi = 3.1415          # Asigno valor de pi
3 area = pi*r**2       # Calculo área
4 print(area)          # Muestro área
```

A screenshot of a terminal window with a dark background. The window title is "rodrigo@rodrigo-Studio-1458: ~/Desktop/In". The terminal shows the execution of the same Python code as the previous block. The output of the print statement is displayed on a new line.

```
>>> r = 3
>>> pi = 3.1415
>>> area = pi*r**2
>>> area
28.273500000000002
>>>
```

Otro ejemplo...

```
1 # Python como calculadora
2 res = 5          # Asigno valor inicial a res
3 res = res*2      # Multiplico res por 2
4 res = res**2     # res elevado a 2
5 res = res%6      # Módulo 5
6 print(res)       # Muestro resultado en consola (4)
```

Aprovechando algunos *atajos* del lenguaje:

```
1 # Esto ...  
2 res = 5  
3 res = res*2  
4 res = res**2  
5 res = res%6  
6 print(res)
```

```
1 # ... es igual a esto  
2 res = 5  
3 res *= 2  
4 res **= 2  
5 res %= 6  
6 print(res)
```



# Variables

**Pregunta:** ¿puede dar cualquier nombre a una variable?

**Reglas:**

1. Debe comenzar con '\_' o una letra
2. El resto pueden ser `a, b, ..., z, A, B, ..., Z, 0, 1, ..., 9`
3. **Excepto** algunas *palabras reservadas* (*keywords*)<sup>1</sup>.

**Ejemplos válidos:** `cristian`, `pesos_por_hora`, `_CLaRK`,  
`j0rg3`

**Ejemplos NO válidos:** `12deLaNoche`, `Bruce Wayne`,

**Min+Seg**\_\_\_\_\_

<sup>1</sup>`and, asset, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, field`

# Variables

**Pregunta:** ¿de qué tipo es la variable `x`?

Depende de la última *asignación* de `x`.

Podemos saber el tipo de una variable con:

`type(nombre_variable)`

```
1 x = 3          # x es int
2 x = 3.0        # x ahora es float
3 x = 3 + 0j     # x ahora es complex
4 x = "3"        # x ahora es str
5 x = True       # x ahora es bool
```

¿Esto importa? → ¡Sí!

¿Cuánto es  $3 \cdot x$ ?

# Variables

**Pregunta:** ¿Cómo cambio el tipo de la variable?

Podemos *convertir*<sup>2</sup> una variable `x`, anteponiendo un tipo de datos.

```
1 x = 3                # x es int de valor 3
2 x = float(x)         # x ahora es float de valor 3.0
3 x = complex(x)       # x ahora es complex de valor 3 + 0j
4 x = str(x)           # x ahora es str de valor "3"
5 x = bool(x)          # x ahora es bool de valor True
6 x = int(x)           # x vuelve a ser int de valor 1
```

**Obs:** `bool(x)` es `True`  $\Leftrightarrow x \neq 0$ .

<sup>2</sup>En realidad los variables no se convierten, sino que se crea una copia con otro valor

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

- 1) Evalúe polinomio  $x^4 + \frac{1}{2}x^3 + 2x^2 - x$  para un  $x$  cualquiera.
- 2) Obtenga la unidad de una variable  $x$ . (ej: si  $x = 123$ , debe obtener 3).
- 3) Obtenga la decena de una variable  $x$ . (ej: si  $x = 123$ , debe obtener 2).

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: **print**

Input: **input**

Actividad Clase 4

# Escribiendo en pantalla: print

`print` permite imprimir en consola.

## Sintaxis print

```
1 print(variable1, variable2, variable3, ...)
```

```
1 cobreloa = 0           # goles del local
2 antofagasta = 0        # goles de la visita
3 cobreloa += 1          # gol local
4 antofagasta += 1       # gol visita
5
6 print("Local", cobreloa, "-", antofagasta, "Visita")
7 # >>> "Local 1 - 1 Visita"
```

# Escribiendo en pantalla: print

`print` también puede recibir un *string* concatenado

```
2 print(string1 + string2 + string3 + ...)
```

```
1 cobreloa = 0           # goles del local
2 arica = 0              # goles de la visita
3 cobreloa += 1          # gol local
4 arica += 1             # gol visita
5
6 print("Local "+cobreloa+"-"+arica+" Visita")
```

¿Qué pasó? No podemos sumar *string* y número



# Escribiendo en pantalla: print

`print` también puede recibir un *string* concatenado

```
2 print(string1 + string2 + string3 + ...)
```

```
1 cobreloa = 0      # goles del local
2 arica = 0         # goles de la visita
3 cobreloa += 1     # gol local
4 arica += 1        # gol visita
5
6 print("Local "+str(cobreloa)+"-"+str(arica)+" Visita")
7 # >>> "Local 1-1 Visita"
```

Si vamos a concatenar, debemos *convertir* el número a *string*

# Recibiendo entrada del usuario: input

`input` entrega un *str* con un valor ingresado por el usuario.

## Sintaxis input

```
1 valorStr = input(mensaje_para_usuario)
```

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un texto: ")
3 print(a)
```

¿Cómo pido un `int`, `float` o `bool` al usuario?

no se puede... pero se pueden *convertir*.

# Input

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un str: ")
3 # pido un int al usuario
4 b = int(input("Ingrese un int: "))
5 # pido un float al usuario
6 c = float(input("Ingrese un float: "))
7 # pido un bool al usuario
8 d = bool(input("Ingrese un bool: "))
9 # imprimo valores obtenidos
10 print(a,b,c,d)
```

Para un marcador...

```
1 local = 0          # goles del local
2 visita = 0         # goles de la visita
3
4 # Guardo el equipo que anotó en variable "gol"
5 # [1: local, 2: visita, otro: fin partido]
6 gol = int(input("?¿Qué equipo anotó el gol?"))
```

# Input

¿Y después?

```
1 local = 0          # goles del local
2 visita = 0         # goles de la visita
3
4 # Guardo el equipo que anotó en variable "gol"
5 # [1: local, 2: visita, otro: fin partido]
6 gol = int(input("? ¿Qué equipo anotó el gol?"))
7
8 print("Gol del equipo: " + str(gol))
9 local += 1         # gol local
10 visita += 1       # gol visita
11
12 print("Local", local, "-", visita, "Visita")
```

# Contenidos

Laboratorios

Recapitulación

Contenidos del curso

Asignaciones

Variables

Ejercicios

Entrada/Salida (Input/Output)

Output: `print`

Input: `input`

Actividad Clase 4

Tipos de datos: `int`, `float`, `str`, `bool`

## Sintaxis `print`

```
1 print(variable1, variable2, variable3, ...)
```

## Sintaxis `input`

```
1 valorStr = input(mensaje_para_usuario)
```