

Elementos básicos de programación

Usando Python

Mateo Suster
msuster@campus.ungs.edu.ar

Matemática para Economistas III
Instituto de Industria
Universidad Nacional de General Sarmiento

Tipos de Datos y Operaciones

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Las operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow$

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Las operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow 7$

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Las operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow 7$
- ▶ Se puede sumar strings: probar “yo y” + “ mi trasero”.

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Los operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow 7$
- ▶ Se puede sumar strings: probar “yo y” + “ mi trasero”.
- ▶ Producto: $2*8 \rightarrow$

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Las operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow 7$
- ▶ Se puede sumar strings: probar “yo y” + “ mi trasero”.
- ▶ Producto: $2*8 \rightarrow 16$

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Los operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow 7$
- ▶ Se puede sumar strings: probar “yo y” + “ mi trasero”.
- ▶ Producto: $2*8 \rightarrow 16$
- ▶ Resto: $5 \% 2 \rightarrow$

Tipos de Datos y Operaciones

Los programas manipulan *valores* de diferentes *tipos*. Por ejemplo:

- ▶ 1 es un valor de tipo **entero** (int).
- ▶ 2.5 es un valor de tipo **número “real”** (float).
- ▶ “hola” es un valor de tipo **caracter** (string).
- ▶ “5” es un valor de tipo **caracter** (string).
- ▶ [7.0, 420, “tira de asado”] es dato tipo **lista** (list).
- ▶ False es un valor de tipo **booleano** (bool).
 - ▶ Valores de verdad: Denotan el resultado de una evaluación lógica: los valores “verdadero” (True) y “falso” (False)
- ▶ También existe un tipo de dato `NoneType`, que indica *sin ningún valor asignado* (será de utilidad cuando veamos funciones)

Las operaciones son, por ejemplo:

- ▶ Suma/Resta: $3+4 \rightarrow 7$
- ▶ Se puede sumar strings: probar “yo y” + “ mi trasero”.
- ▶ Producto: $2*8 \rightarrow 16$
- ▶ Resto: $5 \% 2 \rightarrow 1$

Variables, expresiones y asignaciones

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor
 - ▶ `b = 3` asigna a la variable `b` el valor 3

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor
 - ▶ $b = 3$ asigna a la variable b el valor 3
- ▶ Una **expresión** es una combinación de variables, valores y operadores.

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor
 - ▶ $b = 3$ asigna a la variable b el valor 3
- ▶ Una **expresión** es una combinación de variables, valores y operadores.
 - ▶ $1+1$ es una expresión que da como resultado 2.

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor
 - ▶ $b = 3$ asigna a la variable b el valor 3
- ▶ Una **expresión** es una combinación de variables, valores y operadores.
 - ▶ $1+1$ es una expresión que da como resultado 2.
- ▶ Una **asignación** es una instrucción que guarda en una variable una expresión.

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor
 - ▶ `b = 3` asigna a la variable `b` el valor 3
- ▶ Una **expresión** es una combinación de variables, valores y operadores.
 - ▶ `1+1` es una expresión que da como resultado 2.
- ▶ Una **asignación** es una instrucción que guarda en una variable una expresión.
 - ▶ `long = len([1,3,'a'])` asigna a la variable `long` la longitud de la lista `[1,3,'a']`

Variables, expresiones y asignaciones

- ▶ Una **variable** es una dirección de memoria que almacena un valor
 - ▶ `b = 3` asigna a la variable `b` el valor 3
- ▶ Una **expresión** es una combinación de variables, valores y operadores.
 - ▶ `1+1` es una expresión que da como resultado 2.
- ▶ Una **asignación** es una instrucción que guarda en una variable una expresión.
 - ▶ `long = len([1,3,'a'])` asigna a la variable `long` la longitud de la lista `[1,3,'a']`

Asignación: `variable = expresión`

La **asignación**, nuestra primera instrucción

VARIABLE = EXPRESIÓN

La **asignación**, nuestra primera instrucción

VARIABLE = EXPRESIÓN

La asignación almacena el valor de la *expresión* en la dirección en memoria denotada por *variable*

La **asignación**, nuestra primera instrucción

VARIABLE = EXPRESIÓN

La asignación almacena el valor de la *expresión* en la dirección en memoria denotada por *variable*

- ▶ $x = 1000$
- ▶ $x = x + 2$
- ▶ $x = y$
- ▶ $x = x + y * 22 / 33$

La **asignación**, nuestra primera instrucción

VARIABLE = EXPRESIÓN

La asignación almacena el valor de la *expresión* en la dirección en memoria denotada por *variable*

- ▶ $x = 1000$
- ▶ $x = x + 2$
- ▶ $x = y$
- ▶ $x = x + y * 22 / 33$



La **asignación**, nuestra primera instrucción

VARIABLE = EXPRESIÓN

La asignación almacena el valor de la *expresión* en la dirección en memoria denotada por *variable*

- ▶ `x = 1000`
- ▶ `x = x + 2`
- ▶ `x = y`
- ▶ `x = x + y * 22 / 33`

- ▶ `1000 = x`
- ▶ `x + 2 = x`
- ▶ `len(x) = 1`



La **asignación**, nuestra primera instrucción

VARIABLE = EXPRESIÓN

La asignación almacena el valor de la *expresión* en la dirección en memoria denotada por *variable*

- ▶ `x = 1000`
- ▶ `x = x + 2`
- ▶ `x = y`
- ▶ `x = x + y * 22 / 33`

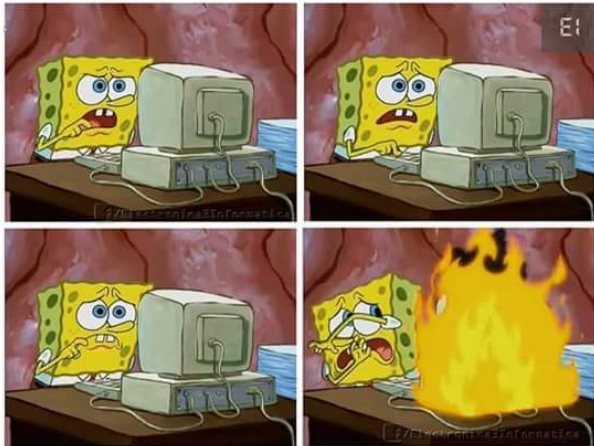
- ▶ `1000 = x`
- ▶ `x + 2 = x`
- ▶ `len(x) = 1`



Lo que se viene

Lo que se viene

EN TU PRIMERA CLASE DE PROGRAMACIÓN



A codea(t)r!

[Link](#) a nuestra página de la materia