

Parte 1. Fundamentos de la programación

Unai Pérez-Goya

Área de Lenguajes y Sistemas Informáticos

Curso 2024/2025

Parte 1.

Fundamentos de la programación

Sesión II: Gestión de Variables y sus tipos

Tabla de contenidos

Resumen de sesiones anteriores

Variables

Tipado en python

Resumen

Resumen de sesiones anteriores



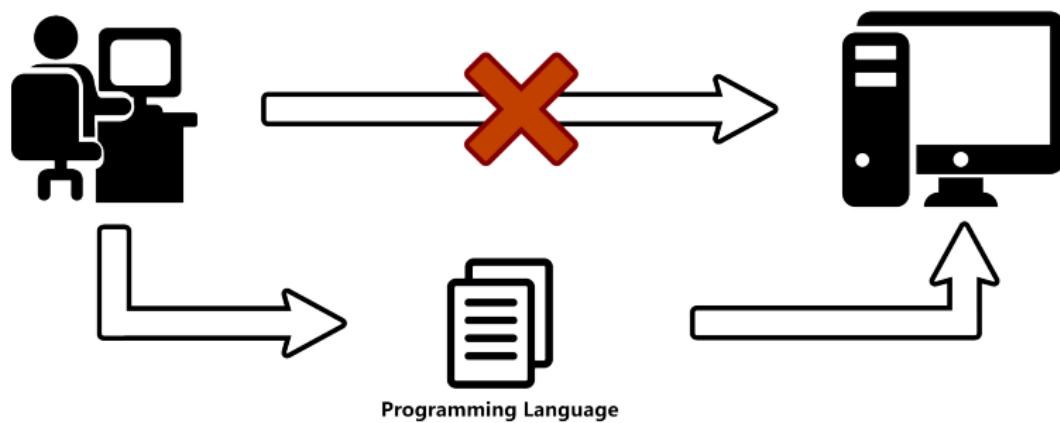
En sesiones anteriores

Lista de contenidos

1. Introducción a Python. Historia y características;
2. Lenguajes de programación. Concepto y uso;
3. Lenguajes interpretados vs. compilados. Programas ejecutables. Máquina intérprete;
4. Entorno de desarrollo integrado (IDE)

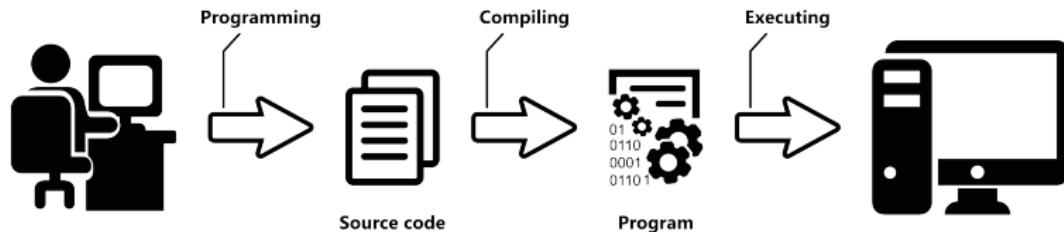
En sesiones anteriores

Lenguajes de programación



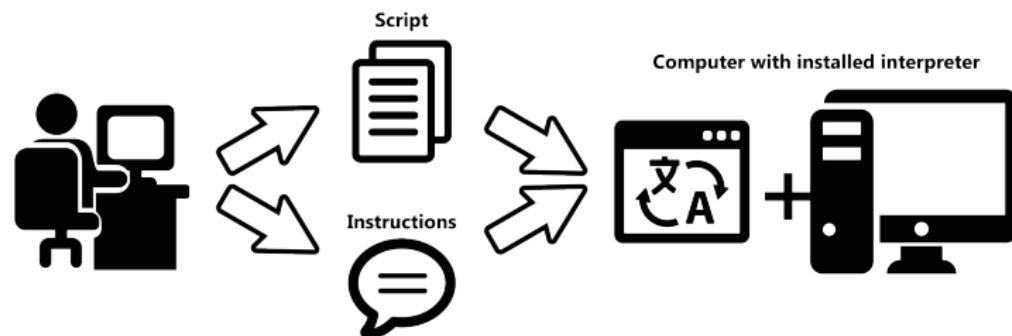
En sesiones anteriores

Proceso de compilación



En sesiones anteriores

Compilado vs. Interpretado



Integrated Development Environment (IDE)

- ▶ Spyder es un entorno de desarrollo integrado (IDE) para python.
- ▶ Basado en python, está dedicado a la programación científica.
- ▶ Es de código abierto y multiplataforma
- ▶ Se puede descargar e instalar desde <https://www.spyder-ide.org/>.



Integrated Development Environment (IDE)

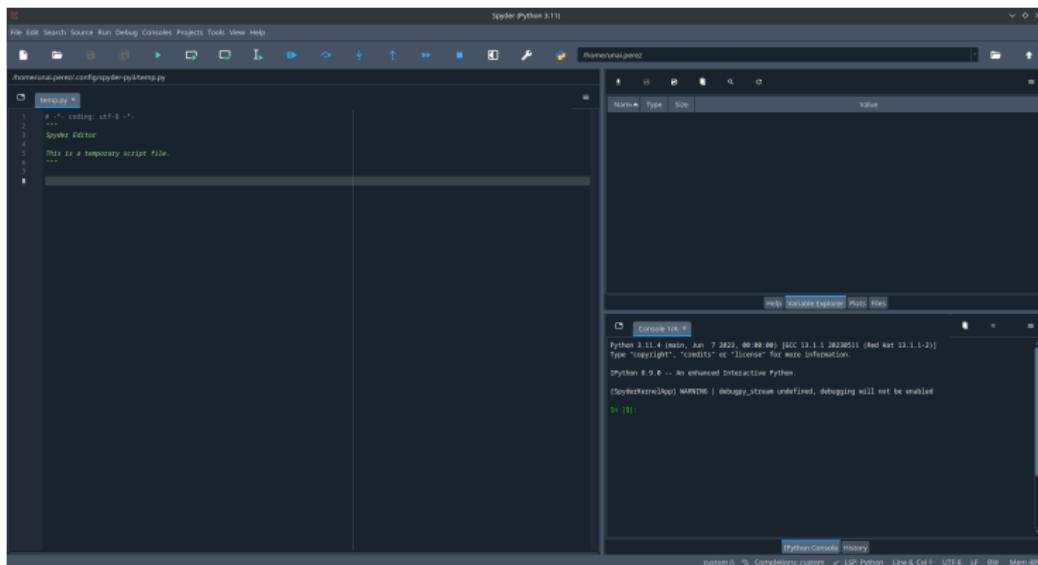


Figura: Interfaz gráfica de Spyder IDE.

Sesión de hoy

- ▶ Contenidos:
 - ▶ Variables;
 - ▶ Numéricas: `int` y `float`
 - ▶ Cadenas de caracteres: `str`
 - ▶ Booleanos: `bool`
 - ▶ Tipado en python.

Variables

Concepto de variable

- ▶ Los programas tienen como objetivo almacenar y gestionar la información.
- ▶ Una **variable** es la unidad de información más pequeña que existe dentro de un programa.
- ▶ La analogía básica, podría ser una caja.
 - ▶ Donde se puede guardar un objeto específico;
 - ▶ Cada objeto debe identificarse con un nombre;
 - ▶ Cada caja sólo puede guardar un objeto. Al introducir el nuevo objeto se pierde el anterior.

Creación de una variable: declaración

- ▶ Al proceso de asignación del nombre de variable, se le denomina declaración.
 - ▶ Indica la existencia de esta variable en el programa.
 - ▶ En la mayoría de los lenguajes de programación, se le añade el tipo de dato de la variable.

```
1 int numEntero;
```

Código: Proceso de declaración de una variable de tipo entero en C.

```
1 var numEntero : Integer;
```

Código: Proceso de declaración de una variable de tipo entero en java.

Creación de una variable: Asignación

- ▶ La asignación consiste en el proceso de asignar un valor a una variable previamente declarada.
- ▶ En python se utiliza el símbolo= para asignar un valor a una variable;

```
1 numEntero = 5;
```

Código: Proceso de asignación de una variable de tipo entero en C.

```
1 begin
2 numEntero := 5;
3 end
```

Código: Proceso de asignación de una variable de tipo entero en java.

Creación de variables

- ▶ La mayoría de veces, el proceso de declaración y asignación se realiza conjuntamente y en la misma línea;

```
1 int zenbakiOsoa = 5;
```

Código: Declaración y asignación simultanea de una variable en C.

- ▶ Python no hace esto exactamente.
- ▶ La máquina intérprete es la encargada de esta gestión, define automáticamente el tipo de variable.
 - ▶ Declarar la variable si ésta no existe.
 - ▶ Dar un valor a la variable (asignación).
 - ▶ El tipo de la variable se deduce del valor asignado.

Creación de variables

- ★ **Ejercicio** Escribe el programa que declare una sola variable y le asigne un número entero. A continuación deberá duplicar el valor de la variable y mostrarlo en pantalla.

Creación de variables: resultado

- ▶ ¿Qué errores comete el siguiente código?

```
1 # python script
2 # Descripcion: programa que declara un numero y lo
   duplica
3
4 numero_entero = 5;
5 numero_entero * 2
6 print(numero_entero)
```

Código: Asignar una variable y duplicar su valor.

Utilización de variables

- ▶ ¿Qué se puede hacer con el símbolo '='?

- ▶ Asignar un valor;
- ▶ Modificar un valor;
- ▶ Guardar el resultado de una operación;

```
1 numero_entero = 5
2 otro_numero_entero = 3
3 numero_entero = 7
4 resultado = numero_entero + otro_numero_entero
5 print(resultado)
```

Código: Diferentes opciones del operador de asignación del programa Python.

- ▶ Las variables son los elementos más básicos del programa y sirven para almacenar información simple o compleja.

Utilización de variables

- ▶ Para crear una variable basta con dar valor a un nombre;
 - ▶ Mejor si la asignáis una sola vez.
 - ▶ El valor de una variable puede cambiar, con asignar un nuevo valor es suficiente.
 - ▶ Se puede utilizar el valor `None` para anular un valor.
 - ▶ Por ejemplo, `a = None` dejará inservible la variable `a`, pero seguirá estando declarada.
- !!** Todo lo no es técnicamente correcto pero conceptualmente lo entenderemos así.

Utilización de variables: designación

- ▶ Técnicamente las variables pueden tener cualquier nombre: admite letras, números y el carácter ‘_’
- ▶ Pero, para facilitar la lectura de código, es recomendable **utilizar la convección de nombres**, y en esta asignatura **será obligatorio**.
- ▶ Convención de nombres `snake_case`:
 - ▶ Deberá dar información del valor que contiene la variable.
 - ▶ Por ejemplo, las variables utilizadas en una división deberían ser: `dividendo`, `divisor`, etc.
 - ▶ Se definirán en minúsculas todos los términos existentes en una variable
 - ▶ Ejemplo: `numero`
 - ▶ Nombres derivados de la unión de dos o más palabras
 - ▶ Por ejemplo `mi_nombre`, `mi_primer_apellido`, etc.

Uso de variables

- ▶ Algunos conceptos válidos para una variable:
 - ▶ Leer/utilizar el valor de una variable no altera su contenido.
 - ▶ El valor de una variable puede utilizarse en las operaciones (`print (a * 4)`);
 - ▶ El valor de una variable se puede guardar en otra variable. Cuando esto ocurre el valor de la variable se *duplica*.
- ▶ Los ejemplos, la prueba y error en prácticas facilitarán la comprensión de estos conceptos.

Algunas preguntas

P: ¿Una variable puede tener cualquier nombre?

R: Cualquier palabra formada por caracteres (letras y números), pero necesariamente hay que empezar por una letra. Recuerda seguir la convención de nombre `snake_case`

P: ¿Cuántas variables pueden existir en un *programa*?

R: Dependerá de la cantidad de memoria de tu ordenador.
Millones.

P: ¿Una variable puede almacenar más de un valor?

R: No, no puede;

R: Al dar un nuevo valor a una variable pierde el anterior;

R: Más adelante aprenderemos a establecer más de un valor en una variable (vectores, matrices), pero son estructuras complejas y la misma estructura se la considera un único valor.

Operaciones entre variables

- ★ **Ejercicio** Escribe el programa que declare dos números y visualice la suma, resta, división y multiplicación de los dos valores.

Operaciones entre variables: resultado

- ▶ ¿Es correcto el siguiente código? Si no lo es... ¿Qué deberías cambiar?

```
1 # python script
2 # Deskribapena: eragiketak
3 uno = 6
4 dos = 3
5 print('Suma', bat+bi)
6 print('Resta', bat-bi)
7 print('Division', bat/bi)
8 print('Multiplicacion', bat*bi)
```

Código: Operaciones entre números.

Operaciones entre variables

- ▶ Atendiendo al código de la diapositiva anterior...
 - ▶ ¿Qué ha pasado con el orden de ejecución en la función `print`?
 - ▶ ¿Puede haber una operación o una función como argumento en una función?
 - ▶ ¿Cuántos argumentos tiene la función `print`?
 - ▶ ¿Qué ha pasado con los valores que se visualizan? ¿El resultado de la división es del mismo tipo?

Tipado en python

Tipado

- ▶ El tipado, indica cómo se realiza la gestión de los tipos de datos en los lenguajes de programación;
- ▶ Cada variable tendrá un determinado tipo de valor en cada momento .
- ▶ Se suele decir que Python no tiene ningún tipo de variable. Esta expresión es errónea.
- ▶ El tipo de una variable no varía mágicamente. Alguna vez. Cambia por orden. Lo cambia el programador.
- ▶ Python tiene tipado **implícito, fuerte y dinámico** (que cambia).

Tipos de variables

- ▶ Python tiene muchos tipos de variables, nosotros aprenderemos los más básicos
- ▶ Tipo de número: Enteros, reales, ...
- ▶ Tipo textual: cadenas de caracteres y de tipo string
- ▶ Booleanos: `True` y `False`.
- ▶ Colecciones: `list`, `set`, `dic`, ...

Tipos numéricos

Tipos numéricos

- ▶ Los valores más comunes en un programa son numéricos.
- ▶ Existen diferentes tipos de números, y Python tiene diferentes tipos de variables para cada uno de ellos:
 - ▶ Números enteros (*integer*);
 - ▶ Números reales (*floats*);
 - ▶ Números complejos (*complex*);
 - ▶ ...
- ▶ Es recomendable que estudieís y recordeís los tipos, sobre todo *floats* y *integer*.
- ▶ La variable se puede almacenar con un tipo inesperado y de forma accidental según el valor dado.

Función type

- ▶ Entonces, ¿cómo puedo identificar el tipo de dato que contiene una variable?
- ▶ La función `type (o: object)` muestra en pantalla el tipo de dato del dato `o: object` en la consola.
- ▶ ¿Qué puede ser `o: object`?
 - ▶ Cualquier tipo numérico;
 - ▶ Cualquier carácter
 - ▶ ...
- ▶ Esta función no tiene argumento

Números enteros (int)

- ▶ Los del conjunto de números enteros.
- ▶ Aceptan dos tipos de operadores, los que utilizaremos son los siguientes:
 - ▶ De tipo entero: `+`, `-`, `*`, `**`, `%`, `//`, etc.
 - ▶ De tipo real: `/`, e.t.c.
- ▶ También aceptan las funciones creadas para los números.
- ▶ Como son números, admiten operaciones entre números reales, pero el resultado de las operaciones mixtas será de tipo real (`float`).

Operaciones entre variables

- ★ **Ejercicio** Escribe un programa que indique el resultado de la división entre dos números en formato entero.

Operaciones entre variables: resultado

```
1 # python script
2 # Descripcion: operaciones
3 dividendo = 6
4 divisor = 3
5 cociente = divisor// divisor
6 resto = division%divisor
7 print ('El cociente de la division es', cociente , ' y
       el resto es', resto)
```

Código: Resultado del ejercicio.

Números reales (`float`)

- ▶ Del conjunto de números reales.
- ▶ Aceptan dos tipos de operadores, los que utilizaremos son los siguientes:
 - ▶ De tipo entero: `//`, `%` etc.
 - ▶ De tipo real: `+`, `-`, `*`, `**`, `/`, etc.
- ▶ También aceptan las funciones creadas para números.
- ▶ Como son números, admiten operaciones entre números reales, pero el resultado de las operaciones mixtas será de tipo real (`float`).

Operaciones entre variables

- ★ **Ejercicio** Escribe un programa que indique el resultado de la división entre dos números en formato real.

Operaciones entre variables: resultado

```
1 # python script
2 # Descripcion: operaciones
3 dividendo = 6
4 divisor = 3
5 resultado = dividendo / divisor
6 print ('El resultado de la division es',resultado)
```

Código: Resultado del ejercicio

- ▶ ¿Si ejecutamos este resultado se entiende? ¿Qué cambios podríamos hacer para que el resultado se represente mejor?

Variables de tipo texto

Variables de tipo texto (str)

- ▶ El segundo tipo de valor genérico más común es *texto*.
 - ▶ Técnicamente, el tipo `str` (del inglés *string*).
- ▶ Utilizando la función `print` se pueden visualizar los strings directamente.
- ▶ En este aspecto no hay ninguna diferencia con las variables numéricas.
- ▶ Los textos tienen funcionalidades interesantes, normalmente asociadas a funciones.
- ▶ Se tratarán más adelante.

Variables de tipo texto (str)

- ▶ Declaración/Asignación: `a = 'Frase'`
- ▶ Por orden de caracteres, tienen operaciones similares a la lista
 - ▶ Tienen longitud `len(a)`
 - ▶ Primer carácter `a[0]`
 - ▶ Último carácter `a[len(a)-1]`
- ▶ ¡La longitud 0 es posible!
 - ▶ `a=''` → `len(a)==0.`

Operaciones en tipo texto (str)

- ▶ Se pueden realizar operaciones matemáticas sobre un string, algo muy útil en algunas aplicaciones.
 - ▶ Son utilizados para operaciones distintas de las habituales.
 - ▶ Ejemplos:
 - ▶ Multiplicación ($*$) repite n veces el string;
 - ▶ Suma ($+$) concatena los string;
 - ▶ ...
- !!** ¡Cuidado! Las sumas de tipo mixto no están permitidas, por ejemplo 'Hola' + 5

Operaciones entre variables

- ★ **Ejercicio** Escribe un programa que ejecute las operaciones aritméticas ($+, -, *, /$) entre un número y una cadena de caracteres. ¿Es posible hacer esto?

Operaciones entre variables: resultado

```
1 # python script
2 # Deskribapena: eragiketak
3 numero = 6
4 texto = 'hola'
5 multiplicacion = texto*numero
6 division = numero/texto
7 suma = texto+numero
8 resta = texto-numero
9 print('El cociente de la division es',cociente)
```

Código: Resultado del ejercicio.

La operación in

- ▶ `in` es una palabra reservada que actua como operando
- ▶ indica si una cadena está incluida en otra cadena o no;

```
1 # python script
2 # Deskribapena: Utilizando in
3 print('a' in 'abc')
4 print('d' in 'abc')
```

Código: Utilizando in.

¿Qué tipo de variable tendrá la palabra clave `in`?

```
1 # python script
2 # Deskribapena: Utilizando in
3 print(type('a' in 'abc'))
4 print(type('d' in 'abc'))
```

Código: Ver el tipo de un booleano

Operaciones entre variables

- ★ **Ariketa** Crea un programa que declare una variable de 4 caracteres e indique si todos los caracteres son dígitos. El programa deberá visualizar `True` o `False`.

Operaciones entre variables

```
1 # python script
2 # Deskribapena: in funtzioa
3 texto = '1234'
4 zero = texto[0] in '0123456789'
5 uno = texto[1] in '0123456789'
6 dos = texto[2] in '0123456789'
7 tres = texto[3] in '0123456789'
8 print(zero and uno and dos and tres)
```

Código: Resultado del ejercicio.

Booleanos

Booleanos (Boolean)

- ▶ Tipo de variable para realizar la operación básica de álgebra de Boole creada por George Bool;
- ▶ El valor basado en expresiones lógicas sólo puede tener dos valores;
 - ▶ `True`;
 - ▶ `False`;
- ▶ Pueden surgir de la comparación entre valores;
 - ▶ Operadores: `>`, `<`, `==`, `>=`, `<=`
- ▶ Los valores derivados de las expresiones lógicas se pueden relacionar con las siguientes operadores o palabras clave:: `and`, `or` y `not`.

Operaciones entre variables

★ **Ejercicio** Crear una tabla con todos los posibles resultados de todas las operaciones entre booleanos.

Constantes en Python

- ▶ Si habéis programado previamente, echaréis de menos las **constantes**.
- ▶ Python no tiene constantes.
 - ▶ Declarar variables y no realizar modificaciones.
- ▶ Eso sí, como recomendación, escribir estas variables con mayúsculas puede ayudar.
- ▶ En esta asignatura, declarar las constantes con mayusculas será obligatorio

Resumen

Contenido de la sesión

- ▶ Variables y uso básico en python
- ▶ Declaración y asignación;
- ▶ Tipos
 - ▶ Numéricos
 - ▶ Texto
 - ▶ Booleanos
- ▶ Constantes