2.2 Variables, expresiones, asignación

- 2.1 Abstracción de problemas para su programación. Conceptos fundamentales
- 2.2 Variables, expresiones, asignación
- 2.3 Uso de entrada/salida por consola
- 2.4 Manejo de estructuras básicas de control de flujo: secuencial, alternativa y repetitiva
- 2.5 Definición y uso de subprogramas y funciones. Ámbito de variables
- 2.6 Entrada/salida a ficheros
- 2.7 Tipos y estructuras de datos básicas: arrays

Valores

Los programas sirven para manipular datos (información):

1. reciben unos datos de entrada,
2. hacen cálculos con ellos, y
3. producen datos de salida

¿Qué datos son los más habituales?

- números: 7, 3.1416
- textos: “Juan”, “Alonso”, “Universidad de Oviedo”
- valores lógicos: True, False

Tipos

Cada dato o valor que se maneja es de un **tipo**:

- 7 tipo **int** (*integer*, entero)
- 3.1416 tipo **float** (número en coma flotante)
- “Juan” tipo **str** (*string*, cadena de caracteres)
- True tipo **bool** (valor booleano)

type devuelve el tipo de cualquier valor o expresión

```
>>>> type(7)
<class 'int'>
>>> type(3.1416)
<class 'float'>
>>> type("Juan")
<class 'str'>
>>> type(True)
<class 'bool'>
```

Conversiones

Un valor de un tipo se puede convertir en un valor de otro tipo:

- Se indica primero el nombre del tipo al que se quiere convertir, y
- entre paréntesis el valor
- Uso más habitual: convertir una cadena en un valor de otro tipo y al revés

```
>>> int("7")
7
>>> str(7)
'7'
>>> float("3.1416")
3.1416
```

```
>>> str(3.1416)
'3.1416'
>>> int(3.1416)
3
>>> float(7)
7.0
```

Operadores

- Los programas hacen cálculos y operaciones con los datos usando **operadores**
- Con cada tipo de valor se pueden hacer operaciones diferentes

Sintaxis (operadores binarios): operando **operador** operando

```
>>> 7 + 4
11
>>> 3.1416 * 2
6.2832
>>> "Juan" + " Alonso"
'Juan Alonso'
>>> True and False
False
```

Operadores más habituales

Grupo	Operadores
Aritméticos	**+** **–** ***** **/** **//** **%** (resto) ****** (potencia)
Relacionales	**<** **<=** **>** **>=** **==** (igual) **!=** (distinto)
Lógicos	**and** **or** **not** (unario, sólo un operando)

- ***/***: división real (coma flotante)
- ***//***: división entera
- Cadenas: **+** (concatena 2 cadenas), ***** (repite varias veces la misma)

Toda operación produce un valor de un cierto tipo

Ejemplos de operaciones con valores

```
>>> 2 // 3
0
>>> 2 % 3
2
>>> 2 / 3
0.6666666666666663
>>> 2.0 // 3
0.0
>>> 2 ** 3
8
>>> not True
False
```

```
>>> 2 > 3
False
>>> 3 <= 3.1416
True
>>> 2 == 3
False
>>> 2 != 3
True
>>> "Antonio" < "Juan"
True
```

¿De qué tipo es cada resultado?

Variables

- Ejemplo: hacer un programa que sirva para calcular el área de un rectángulo dada la base y la altura

```
area = base * altura
```

  - no sabemos los valores que van a tomar base y altura, cambiarán en distintas ejecuciones del programa
  - no podemos usar valores concretos: `>>> 5 * 4`
  - para generalizar el programa usaremos nombres para referirnos a esos valores

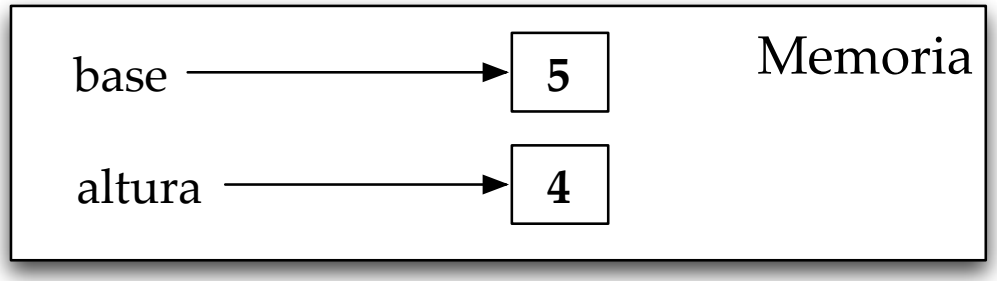
```
>>> base * altura
```

el área es lo que valga la base por la altura

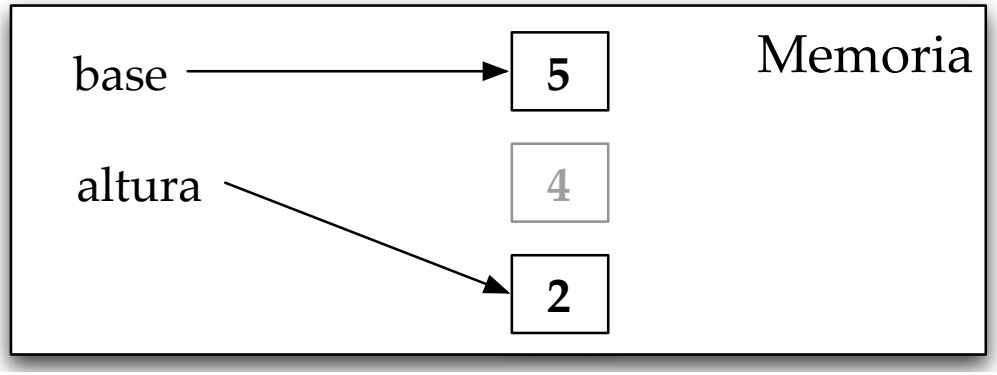
Variables

- Una variable es el nombre que se utiliza en un programa para representar un **dato o valor** que **puede cambiar**

```
>>> base = 5
>>> altura = 4
>>> base * altura
20
```




```
>>> altura = 2
>>> base * altura
10
```



Variables

- Las variables representan:
  - un **valor**,
  - que será de un cierto **tipo**,
  - y que estará en una **posición de la memoria**
- Recuerda que para acceder a…
  - el valor: se usa el nombre de la variable
  - el tipo: con la función type()

Asignación

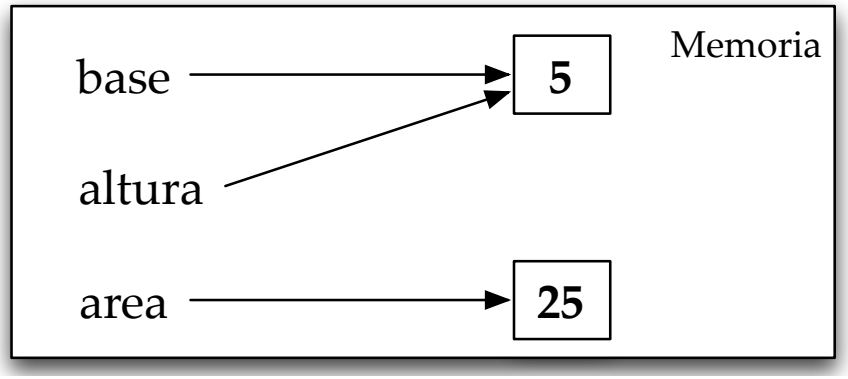
- La asignación es la instrucción que permite cambiar el valor que representa una variable

variable = expresión

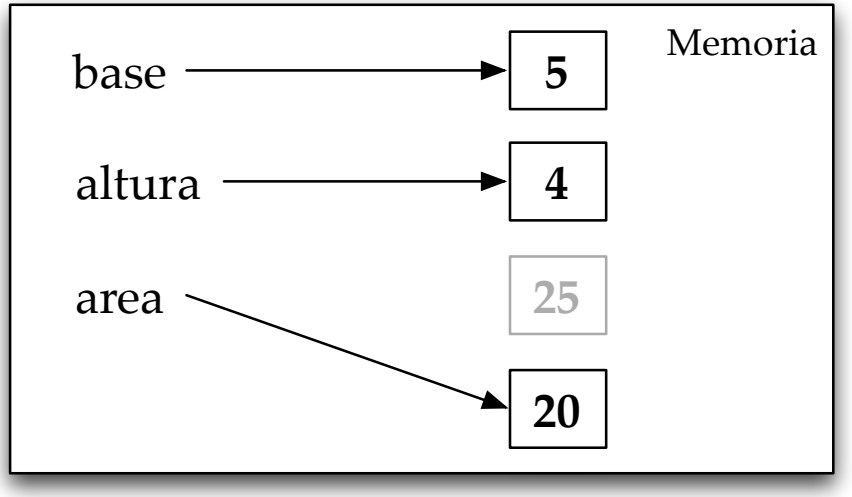
- La expresión que se asigna puede ser:
  - **un valor** (en general, el resultado de una operación): se reserva un nuevo espacio en la memoria para contenerlo
  - **otra variable**: las dos variables se referirán al mismo valor

Ejemplos de asignaciones

```
>>> base = 5
>>> altura = base
>>> area = base * altura
>>> area
25
```




```
>>> altura = 4
>>> area = base * altura
>>> area
20
```



Nombres de variables

- El nombre de una variable puede estar formado por:
  - letras
  - dígitos
  - guión bajo o subrayado "_"
- Restricciones:
  - no puede empezar por un dígito
  - no puede llamarse como una palabra reservada del lenguaje
- Las letras mayúsculas y minúsculas son diferentes
- Ejemplos: *nombre, velocidad_final, cuota, codigo_postal*

El nombre de una variable debe indicar qué dato representa dentro del programa

¿Cuándo hay que usar una variable?

- Las variables se usan para representar:
  - los datos de entrada al programa
  - los datos de salida (resultados) del programa
  - y en general: siempre que necesitemos una zona de la memoria para guardar un dato y usarlo posteriormente

```
>>> area = base * altura
>>> area
20
```

Las variables son las herramientas que usan los programas para almacenar información en la memoria del ordenador

Expresiones

- **Expresión**: combinación de operadores y operandos que produce un resultado (valor) de un cierto tipo
- El resultado y su tipo dependerá de los operandos y de los operadores que se usen
- Al poder mezclarse operadores diferentes, existen reglas para determinar el orden en que se aplican (precedencia y asociatividad)

Evaluación de Expresiones - Precedencia

Sirve para decidir qué operador debe aplicarse primero cuando en una expresión aparecen varios operadores de distintos grupos

Reglas:

1. primero se hace lo que se está entre paréntesis
2. aritméticos > relacionales > lógicos
3. operadores unarios > binarios
4. potencia > multiplicativos (∗, /,//, %) > aditivos (+, −) (también con los lógicos and > or)