

# Informática

## Listas

Fernando Rosa Velardo

Sistemas Informáticos y Computación

Curso 2012/2013

Grado en Matemáticas, Grado en Ing. Matemática, Grado en  
Matemáticas y Estadística

# Introducción

Para resolver algunos problemas, es mejor agrupar diversos datos, en vez de considerarlos variables diferentes.

## Vectores

Podemos describir un vector  $(x, y, z) \in R^3$  utilizando 3 variables:

$x, y, z = 3, 4, 5$

¿Y si queremos utilizar más vectores? ¿O vectores de mayores dimensiones?

## Calificaciones

Queremos realizar un programa que permita gestionar las calificaciones de los alumnos de una clase. ¿Declaramos 40 variables? ¿Y si hay más alumnos?

Necesitamos una **estructura de datos** más compleja.

# Listas

Una **lista** es una estructura de datos que permite guardar una colección de datos

Utilizando **un único nombre**, podemos hacer referencia a toda la colección.

## Ejemplo

```
v = [ 3 , 4 , 5 ]
```

# Acceso a los elementos de una lista

## Ejemplo

```
v = [ 3 , 4 , 5 ]
```

Para poder acceder a un elemento particular de la lista, utilizamos el **índice**

Los valores posibles para el índice son enteros a **partir de 0**.

Elementos de la lista:  $v[0]$ ,  $v[1]$ , ...,  $v[2]$

**$v[i]$** : “variable” sobre la cual se pueden realizar todo tipo de operaciones: escribir, leer, ...

# Funciones y operaciones sobre listas

- Lista vacía: `[]`
- Longitud de una lista: `len(l)`
- Concatenación de listas: `lista1 + lista2`
- Repetición de una lista: `n*lista`
- Corte: `lista[a:b]` devuelve una lista con los elementos de `lista` desde las posiciones `a` hasta `b-1`. En particular, `lista[:]` devuelve una copia de `lista`
- Rango: `range(a:b)` devuelve una lista con los valores desde `a` hasta `b-1`

# Comparación de listas

- `lista1==lista2` devuelve `True` cuando las listas son idénticas
- También se pueden usar `!=`, `<`, `>`, `<=`, `>` y `>=`
- El orden que se considera entre listas es el lexicográfico (el de los diccionarios)

# Asignación/copia de listas

¿Qué ocurre si asignamos una lista a una variable?

```
a = [ 2 , 3 , 4 ]  
b = a  
print a  
print b  
a[1] = 77  
print a  
print b
```

¡No hemos creado una copia de la lista! Sólo hemos creado un *alias*, una nueva referencia a la misma lista.

# Para copiar una lista

```
a = [ 2 , 3 , 4 ]  
b = a [:]  
print a  
print b  
a[1] = 77  
print a  
print b
```



# Bucle for-in sobre una lista

Podemos recorrer secuencialmente una lista, mediante un **for**.

## Cálculo del máximo de una lista

```
def maximoLista(v):  
    """  
    Entrada: v lista no vacia  
    Salida: maximo de los valores en v  
    """  
    maximo=v[0]  
    for valor in v:  
        if maximo<valor:  
            maximo=valor  
    return maximo
```

# Recorrido de una lista I

**Ejemplo.** Programa que, dados un conjunto de datos, muestra su diferencia respecto de la media.

```
def media(v):  
    suma=0  
    for valor in v:  
        suma+=valor  
    return float(suma)/len(v)
```

```
def difMedia(v):  
    m=media(v)  
    print 'Media:',m  
    for valor in v:  
        print valor, valor-m
```

# Búsqueda secuencial en una lista

Podemos aplicar el patrón de búsqueda entre los elementos de una lista

**Ejemplo.** Programa que busca un dato en una lista

```
def esta(valor, v):  
    i, enc, n = 0, False, len(v)  
    while not enc and (i < n):  
        if v[i] == valor:  
            enc = True  
        else:  
            i = i + 1  
    return enc
```

# Búsqueda secuencial en una lista

**Ejemplo.** Programa que dice si los datos de una lista están ordenados de manera creciente

```
def esCreciente(v):  
    #Busqueda de i tal que v[i-1]>v[i]  
    i,enc,n=1,False,len(v)  
    while not enc and (i<n):  
        if v[i-1]>v[i]:  
            enc=True  
        else:  
            i=i+1  
    return not enc
```

# Adición de elementos a una lista

```
>>> a=[ 2 , 3 , 5 ]
>>> a[len(a)] = 7
Traceback (most recent call last):
  File "<pyshe11#55>", line 1, in <module>
    a[len(a)] = 7
IndexError: list assignment index out of range
```

¿Entonces cómo se añaden elementos a una lista?

## Adición de elementos a una lista II

### Opción 1 (la mala)

```
>>> a=a+[7]
>>> a
[2, 3, 5, 7]
```

**Pega:** se crea una nueva lista, copiando los elementos de la lista a

### Opción 2 (la buena)

```
>>> a.append(7)
>>> a
[2, 3, 5, 7]
```

**Ventaja:** se *modifica* la lista a

# Más sobre append

- append es un **método** del tipo de las listas
- Los métodos son funciones *especiales* presentes en algunos tipos
- Se invocan usando la notación . (punt)

`expresion.metodo`

donde `metodo` es un método presente en el tipo al que se evalúa la expresión

# Ejemplo

Devolver una lista que contenga sólo los valores que NO sean múltiplos de un número dado

```
def filtrarMultiplos(lista, d):  
    multiplos = []  
    for valor in lista:  
        if valor % d != 0:  
            multiplos.append(valor)  
    return multiplos
```