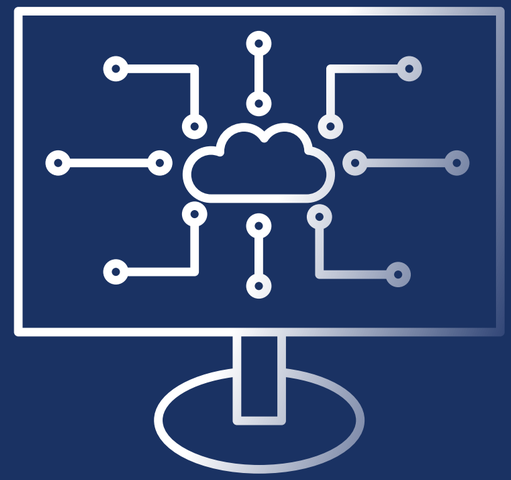
Introducción a la algoritmia

UADE

Lic. Julia Monasterio



UADE

Lic. Julia Monasterio

Clase N°7

TEMAS

- Arreglos: Listas
- Subíndices
- Listas en Python
- Ejemplos

UADE

Lic. Julia Monasterio

Arreglos: listas

Con los conocimientos que tenemos hasta el momento, como resolverían la siguiente situación:

Leer tres números e imprimirlos en el orden inverso

3 2 1

Posible solución

```
numero1= int(input("Ingrese un número\n"))
numero2=int(input("Ingrese un numero\n"))
numero3=int(input("Ingrese un numero\n"))

print(numero3,numero2,numero1)
```

```
Ingrese un número
90
Ingrese un numero
101
Ingrese un numero
4
4 101 90
```

Ahora, si tuviésemos que hacerlo para 100 números

¿se complica?
¿utilizarían la misma estrategia?

Modificar estrategia

Es necesario implementar otra estrategia de solución que utilice una estructura de datos, es decir un conjunto de datos agrupados de alguna manera.

A esta estructura de datos se la denomina **arreglos**

Definición

Un arreglo es un conjunto de variables agrupadas bajo un solo nombre que contienen de un número finito de datos (elementos) del mismo tipo , y un número de orden que es consecutivo (subíndice), 0, 1, 2, ..., n -1.

Tipos de arreglos

Existen dos tipos de arreglos:

- **Vectores** (unidimensional)
- **Matrices** (multidimensional)

Vectores - Definición desglosada

Conjunto de variables agrupadas bajo un solo nombre...

edades

Vectores - Definición desglosada

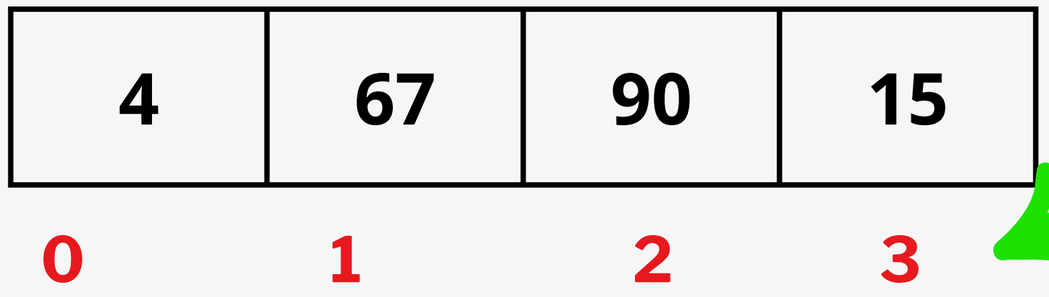
...que contienen de un número finito de datos (elementos) del mismo tipo

4	67	90	15

edades

Vectores - Definición desglosada

...y un número de orden que es consecutivo (subíndice), 0,1,2,3,4, n -1



Un subíndice identifica la posición de cada variable.
Equivale al número de orden de la variable dentro del arreglo.

UADE

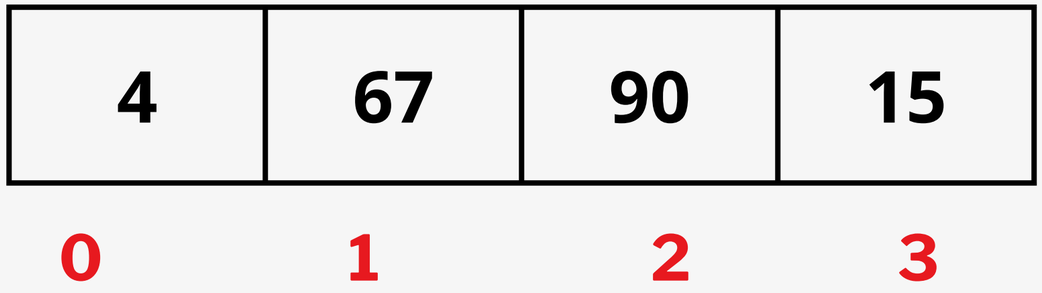
Lic. Julia Monasterio

Subíndices

Subíndices

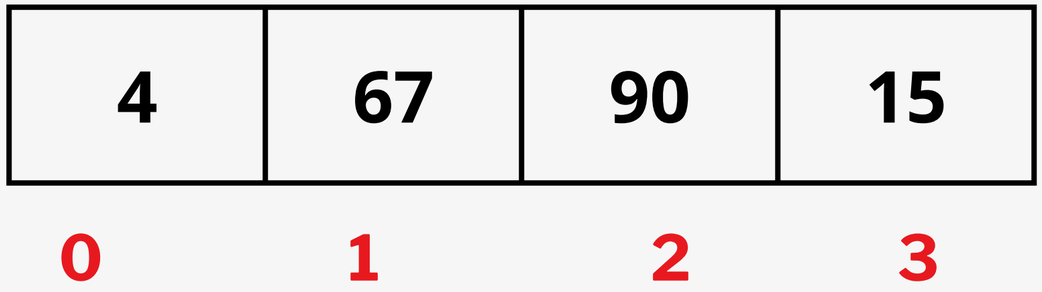
- Deben ser números enteros
- Siempre comienzan por el 0. Terminan uno antes del tamaño del vector
- Se escriben luego del nombre del arreglo, encerrados entre corchetes
- Pueden usarse constantes, variables y expresiones

¿Cómo se usan?




```
edades=[14,18,24,35]

print(edades[2]) #constante

a=1

print(edades[a]) #variable

print(edades[a+2]) #Expresion
```

¿Cómo se usan?




```
edades=[14,18,24,35]

a=2

edades[a] = edades[a]+1 #El 24 se convierte en 25

b=0

edades[b] = edades[b+1] #El 0 se convierte en 1. Va a informar el 18 en lugar del 14
```

UADE

Lic. Julia Monasterio

Listas en Python

Listas en Python

- En Python los vectores se implementan a través de listas.

- Las listas pueden crecer y reducirse , algo que no ocurre con los vectores tradicionales en otros lenguajes de programación (suelen tener tamaño pre definido)

UADE

Lic. Julia Monasterio

Práctica en clase

Realizar en PSEINT y Python

Leer 10 números e imprimirlos en el orden inverso

Solución en Python

```python
numeros=[] #declaro un vector vacio

contador=0

while contador<10:
    n=int(input("Ingrese un número"))
    numeros.append(n)
    contador+=1

contador=9

while contador>=0:
    print(numeros[contador])
    contador-=1
```

Aspectos importantes

- Declaración de listas: al inicializar una variable con dos corchetes sin escribir nada entre ellos, estamos creando una lista vacía

- Método append(): permite agregar nuevos elementos al final de una lista, como si se tratase nuevos vagones del tren.

UADE

Lic. Julia Monasterio

Práctica en clase

Realizar Python

Imprimir por pantalla una lista definida dentro del programa.

Imprimir una lista por pantalla

```
lista = [4,7,2,9,5]



x=0

while x < 5:
    print(lista[x], end=" ")
    x+=1
```

```
4 7 2 9 5
```

Aspectos importantes

- Declaración con elementos: las listas pueden declararse con elementos en su interior, es decir no vacías. Estos elementos pueden ser constantes o variables.

lista = [4,7,2,9,5] o lista1 = [a,3, numero]

- end = “ “ : indica que el renglón no termino y que el próximo print debe ir al lado. Por defecto siempre tiene \n

UADE

Lic. Julia Monasterio

Práctica en clase

Realizar Python

Leer 50 números enteros, calcular su promedio e imprimir aquellos valores leídos que sean mayores al promedio obtenido.

```python
#Primera parte: lectura de datos y cálculo del promedio
ELEMENTOS = 50
lista =[]
contador=0
suma=0

while contador<ELEMENTOS:
    n=int(input("Ingrese un número entero\n"))
    lista.append(n)
    suma+=n
    contador+=1

promedio=suma/ELEMENTOS

print("El primedio es: ",promedio)
```

```
#Segunda parte: imprimir aquellos elementos leídos que sean mayores al promedio

contador=0

while contador<ELEMENTOS:
    if lista[contador]>promedio:
        print(lista[contador])
    contador+=1
```

CONSTANTES

- Si en alguna oportunidad fuera necesario modificar la cantidad de números a ingresar, el único cambio requerido será el valor de la variable **ELEMENTOS**, sin necesidad de ajustar ningún otro aspecto del programa.

- Escribir el nombre de la variable en mayúsculas es una convención utilizada para definir constantes.

UADE

Lic. Julia Monasterio

Práctica en clase

Realizar Python

Leer un conjunto de números y guardarlos en una lista, finalizando la carga con 1. Luego buscar el mayor elemento leído, mostrarlo y eliminarlo del arreglo.

Imprimir por pantalla la lista antes y después del borrado.

```python
#Primera parte: carga de la lista
lista=[]

n= int(input("Ingrese un número o -1 para terminar:"))

while n!=-1:
    lista.append(n)
    n= int(input("Ingrese un número o -1 para terminar:"))
```

```python
#Función para imprimir una lista
def imprimirLista(lista):
    largo=len(lista)
    for i in range(largo):
        print(lista[i], end=" ")
```

```python
#Segunda parte: calculo la cantidad de elementos

largoLista = len(lista)

if largoLista==0:
    print("No se ingresaron valores")
else:
    mayor=lista[0]
    posicion=0

    for i in range(largoLista):
        if lista[i]>mayor:
            mayor=lista[i]
            posicion=i
    imprimirLista(lista)

    print("El maximo de la lista es: ", mayor, "y se encontro en la posición: ",posicion)
```

```
    #Tercer parte: eliminar el mayor
    print("Borrando el mayor que es ", mayor)

    del lista[posicion]

    imprimirLista(lista)
```

NOVEDADES

- **Función len:** devuelve la longitud de una lista, es decir la cantidad de elementos

- **Instrucción for:** como vimos antes, es otra estructura de bucle. Y es la estructura ideal para recorrer una lista

- **Función range:** genera una secuencia de números enteros entre 0 y el valor del parámetro suministrado. El valor final **nunca** esta incluido.
  - range(5) --> 0,1,2,3,4

NOVEDADES

- **Función del:** permite borrar elementos de una lista. También sirve para borrar variables, una lista completa o elementos de una lista.

del lista #Borra la lista completa

del lista[10] #Borra el elemento de la posición 10 de la lista

del edad #Borra la variable edad

UADE

Lic. Julia Monasterio

Ejemplos

UADE

Lic. Julia Monasterio

Práctica en clase

Realizar Python

Escribir una función para ingresar números enteros en una lista y devolverla como valor de retorno.

La cantidad de valores a leer se recibe como parámetro.

```python
#Función para cargar una lista, se debe recibir el tamaño de la misma como parámetro
def cargarLista(tamanio):
    lista=[]
    for elemento in range(tamanio):
        n=int(input("Ingrese un número"))
        lista.append(n)
    return lista
```

```python
#Programa principal
cant= int(input("Ingrese la cantidad de elementos:\n"))
miLista = cargarLista(cant)
print(miLista)
```

```
Ingrese la cantidad de elementos:
3
Ingrese un número
7
Ingrese un número
6
Ingrese un número
5
[7, 6, 5]
```

NOVEDADES

- Una función puede devolver una lista como valor de retorno

- Es posible imprimir una lista sin necesidad que se encuentre en un bucle

Uso de for y range()

```
#Imprimir los numeros del 1 al 100

for i in range(1,101):
    print(i)
```

- Cuando range() se usa con dos parámetros el primero indica el inicio de la secuencia, mientras que el segundo señala el final de la misma. El valor final no esta incluído

range() con incremento

```
for impar in range(1,101,2):
    print(impar)
```

Cuando range() se usa con tres parámetros el tercero actua como incremento. Con solo dos parámetros el valor por defecto del tercer parámetro es 1

range() con incremento negativo

```
for impar in range(101,0,-2):
    print(impar)
```

El incremento puede ser negativo. En este caso el valor inicial debe ser mayor que el valor final.

UADE

Lic. Julia Monasterio

Resúmen de la clase

- Arreglos: listas
- Los subíndices, números enteros de 0 a N
- Asignar valores a listas
- CONSTANTES
- Método append()
- Función len()
- Instrucción FOR
- Instrucción del

UADE

Lic. Julia Monasterio

EJERCITACIÓN

Objetivos

- Introducir el concepto de estructuras de datos
- Familiarizarse con el uso de listas en Python, conocidas como arreglos o vectores en otros lenguajes de programación

UADE

Lic. Julia Monasterio

Ejercitación

- **Ejercicio 1:** Escribir una función para ingresar desde el teclado una serie de números entre A y B y guardarlos en una lista. En caso de ingresar un valor fuera de rango la función mostrará un mensaje de error y solicitará un nuevo número. Para finalizar la carga se deberá ingresar -1. La función recibe como parámetros los valores de A y B, y devuelve la lista cargada (o vacía, si el usuario no ingresó nada) como valor de retorno. Tener en cuenta que A puede ser mayor, menor o igual a B.

En los ejercicios 2 a 5 utilizar la función del ejercicio 1 para ingresar datos en una lista y:

- **Ejercicio 2:** Calcular la suma de los números de la lista.

UADE

Lic. Julia Monasterio

Ejercitación

- **Ejercicio 3:** Determinar si la lista es capicúa (palíndromo). Una lista capicúa se lee de igual modo de izquierda a derecha y de derecha a izquierda. Por ejemplo, [2, 7, 7, 2] es capicúa, mientras que [2, 7, 5, 2] no lo es.

- **Ejercicio 4:** Escribir una función para contar cuántas veces aparece un valor dentro de la lista. La función recibe como parámetros la lista y el valor a buscar, y devuelve un número entero.

- **Ejercicio 5:** Desarrollar una función que reciba la lista como parámetro y devuelva una nueva lista con los mismos elementos de la primera, pero en orden inverso. Por ejemplo, si la función recibe [5, 7, 1] debe devolver [1, 7, 5]

Muchas gracias!

Consultas?