

Estructuras de Datos y Algoritmos

Programación en Python y C++

Rafael De Luna Loredó

Facultad de Ciencias, UASLP



1 Introducción

¿Qué es un
algoritmo?

¿Como
programar
algoritmos?

textquestiondown

Que lenguajes
de
programación
usar?

2 Conceptos Básicos

Bibliotecas y
cabeceras
Tipos de Datos
Números

3 Espacio de nombres

std

4 Operaciones

Operaciones
Matemáticas
Operaciones
Lógicas

5 Condicionales

Introducción
if else elif
Switch
Operador
ternario

6 Ciclos

Introducción
for

Introducción
Declaración

8 Funciones

Introducción

9 Apuntadores

Introducción

10 Estructuras de Datos

Introducción

11 Objetos

Introducción

12 Ordenamiento

Introducción

13 Búsqueda

Introducción

14 Grafos

1 Introducción

- ¿Qué es un algoritmo?
- ¿Como programar algoritmos?

¿Qué es un algoritmo?

¿Como
programar
algoritmos?

textquestiondown

Que lenguajes
de

programación
usar?

2 Conceptos Básicos

3 Espacio de nombres

④ Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

¿Qué es un algoritmo?

Es una serie de pasos a seguir para solucionar un problema.
Pero no solo eso, incluso en nuestra vida diaria usamos algoritmos sin saberlo, por ejemplo al ponernos los zapatos o vestirnos seguimos una serie de pasos para llegar a un resultado final

1 Introducción

¿Qué es un
algoritmo?

¿Como
programar
algoritmos?

textquestiondown

Que lenguajes
de
programación
usar?

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

¿Cómo programar algoritmos?

Antes de empezar a tirar código hay que hacernos unas cuántas preguntas

- ¿sigue una serie de pasos?
- ¿son consecutivos?
- ¿que resultados puedo esperar?

1 Introducción

¿Qué es un
algoritmo?

¿Como
programar
algoritmos?

textquestiondown

Que lenguajes
de
programación
usar?

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

¿Que lenguajes de programación usar?

Podemos hacer uso de cualquier lenguaje, todo dependerá de que tan cómodos nos sintamos con el lenguaje que vayamos a usar o estemos utilizando.

En este curso veremos los ejemplos en Python y C++ , C++ por ser el más utilizado en programación competitiva y Python por su sintaxis sencilla y su amplia utilización en la industria.

① Introducción

② Conceptos Básicos

Bibliotecas y
cabeceras

Tipos de Datos

③ Espacio de
nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

⑨ Apuntadores

⑩ Estructuras de
Datos

⑪ Objetos

⑫ Ordenamiento

⑬ Búsqueda

⑭ Grafos

⑮ Cadenas

⑯ Referencias

Cabeceras

Son archivos que contienen las declaraciones de funciones y/o clases por lo que suelen tener código

Bibliotecas

Una biblioteca o librería podríamos considerarla una colección de cabeceras, además de cabeceras incluye archivos de enlazado dinámico o estático.

1 Introducción

2 Conceptos Básicos

Bibliotecas y
cabeceras

Tipos de Datos

Númericos

Caracteres

Lógicos

Contenedores y
colecciones

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

Enteros

- No aceptan decimales
- Pueden ser negativos o positivos
- En C++ ocupan 4 bytes de memoria y un valor máximo de ± 2147483647
- En Python varía el tamaño en memoria pero puede ser desde 28 bytes hasta 408 bytes o más

Reales

- Aceptan decimales
- Pueden ser negativos o positivos
- En C++ ocupan 4 bytes de memoria y valores entre $1.17549e-38$ y $3.40282e+38$
- En Python varía el tamaño en memoria pero puede ser desde 24 bytes hasta 408 bytes o más

Reales de doble precisión

- Aceptan decimales
- Pueden ser negativos o positivos
- En C++ ocupan 8 bytes de memoria y valores entre $2.22507e-308$ y $1.79769e+308$
- En Python el tipo float hace una implementación a bajo nivel del tipo **double** de C.

caracter

- Representan letras, simbolos o caracteres
- En C++ ocupan 1 byte de memoria y valores de ± 127
- En Python como tal no existe.

Cadena de caracteres

- Representan letras, simbolos, caracteres y/o palabras
- Tanto en C++ como en Python el tamaño varía según el tamaño de la cadena
- En el caso de C++ existen dos tipos, un arreglo de tipo **char** y el tipo **str** a través de la biblioteca **cstring**

Booleanos

- Representan solamente **True** o **False**, **1** o **0**
- Ocupan 1 byte de almacenamiento, por lo que se desperdicia mucho espacio de memoria

Diccionarios

- Se componen de una llave o clave y un valor
- Son elementos ordenados
- Pueden existir múltiples llaves
- Las llaves no se pueden repetir
- En C++ se llaman mapas, para usarlos hay que importar la cabecera *map*

Listas

- Son una secuencia de elementos
- En Python pueden contener múltiples tipos de datos
- Para usarlos en C++ hay que importar la cabecera *list*

Tuplas

- Son una secuencia de elementos
- Pueden contener múltiples tipos de datos
- En Python son inmutables, es decir no se pueden modificar o eliminar elementos
- Para usarlos en C++ hay que importar la cabecera *tuple*

Conjuntos

- Son una secuencia de elementos no repetidos
- Hacen alusión a la definición matemática de conjuntos
- En C++ existen dos tipos, ordenados y no ordenados
- Para usarlos en C++ hay que importar las cabeceras `set` y `unordered_set` respectivamente

① Introducción

② Conceptos Básicos

③ Espacio de nombres std

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

① Introducción

② Conceptos Básicos

③ Espacio de
nombres
std

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

⑨ Apuntadores

⑩ Estructuras de
Datos

⑪ Objetos

⑫ Ordenamiento

⑬ Búsqueda

⑭ Grafos

⑮ Cadenas

⑯ Referencias

1 Introducción

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

Operaciones Matemáticas

Operaciones Lógicas

5 Condicionales

6 Ciclos

7 Arreglos

- | | | |
|-------------------------|-------------------------|-----------------|
| 1 Introducción | 5 Condicionales | 11 Objetos |
| 2 Conceptos Básicos | 6 Ciclos | 12 Ordenamiento |
| 3 Espacio de nombres | 7 Arreglos | 13 Búsqueda |
| 4 Operaciones | 8 Funciones | 14 Grafos |
| Operaciones Matemáticas | 9 Apuntadores | 15 Cadenas |
| Operaciones Lógicas | 10 Estructuras de Datos | 16 Referencias |

Operaciones Matemáticas

- Tienen orden de precedencia, siendo el mismo que conocemos en matemáticas
- Van de izquierda a derecha
- Se realizan en pares
- Se pueden realizar entre diferentes tipos, todo dependera de si se guardan o no en una variable
- Son las mismas que en matemáticas, suma, resta, multiplicación, división y módulo o residuo
- Para uso de funciones mas avanzadas habrá que hacer uso de cabeceras o bibliotecas creadas para dicho propósito

① Introducción

② Conceptos Básicos

③ Espacio de
nombres④ Operaciones
Operaciones
Matemáticas
Operaciones
Lógicas

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

⑨ Apuntadores

⑩ Estructuras de
Datos

⑪ Objetos

⑫ Ordenamiento

⑬ Búsqueda

⑭ Grafos

⑮ Cadenas

⑯ Referencias

Operaciones Lógicas

- Devuelven un valor booleano
- Van de izquierda a derecha
- Se realizan en pares
- Son algunas que ya conocemos en matemáticas, mayor que, menor que, igual a, diferente de, menor o igual, mayor o igual, conjunción y disyunción
- La conjunción y disyunción dependen de las otras

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

Introducción

if else elif

Switch

Operador ternario

⑥ Ciclos

Introducción

- Son un tipo de estructura
- Nos ayudan a tener control según valores o resultados esperados
- Evalúan operaciones lógicas
- Pueden contener o concatenar múltiples estructuras

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales
Introducción
if else elifSwitch
Operador
ternario

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

if

Nos ayuda a evaluar si una condición existe, en caso de que se cumpla, se ejecuta el código que contiene

else

Es la contraparte de **if**, en caso de que la condición de **if** no se cumpla, ejecuta el código que contiene

elif

Es una combinación de **else** con **if**, hace una segunda evaluación de otro posible resultado esperado

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales
Introducción
if else elifSwitch
Operador
ternario

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

elif

- En Python no existe pero se puede hacer con diccionarios o concatenando múltiples **if else**
- Es una estructura que evalúa múltiples casos posibles en base al posible valor de una variable
- También puede contener otras estructuras

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales
Introducción
if else elifSwitch
Operador
ternario

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

Operador ternario

- El funcionamiento es como el de un **if**, en Python de hecho es un **if**
- Sirve para comparaciones simples, donde no necesitamos ejecutar mucho código

1 Introducción

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

Introducción

for

while

do while

① Introducción

② Conceptos Básicos

③ Espacio de
nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

Introducción
for
while
do while

⑦ Arreglos

⑧ Funciones

⑨ Apuntadores

⑩ Estructuras de
Datos

⑪ Objetos

⑫ Ordenamiento

⑬ Búsqueda

⑭ Grafos

⑮ Cadenas

⑯ Referencias

Introducción

- Son un tipo de estructura
- Nos ayudan a ejecutar un código múltiples veces
- Ayudan a reducir código

➊ Introducción

➋ Conceptos Básicos

➌ Espacio de
nombres

➍ Operaciones

➎ Condicionales

➏ Ciclos

Introducción
for
while
do while

➐ Arreglos

➑ Funciones

➒ Apuntadores

➓ Estructuras de
Datos

➑ Objetos

➒ Ordenamiento

➓ Búsqueda

➑ Grafos

➑ Cadenas

➑ Referencias

for

- Se define un contador
- El contador cambia sin necesidad de declarar dicho cambio
- Hay que definir un tope o límite
- Hay que definir como avanza el contador

➊ Introducción

➋ Conceptos Básicos

➌ Espacio de
nombres

➍ Operaciones

➎ Condicionales

➏ Ciclos

Introducción
for
while
do while

➐ Arreglos

➑ Funciones

➒ Apuntadores

➓ Estructuras de
Datos

➑ Objetos

➒ Ordenamiento

➓ Búsqueda

➑ Grafos

➑ Cadenas

➑ Referencias

while

- Se ejecuta mientras dada una condición, dicha condición exista o se cumpla
- No se define un límite, mientras la condición exista, se seguirá ejecutando
- Necesitamos crear una manera de salir, por si la condición sigue existiendo
- Podría no ejecutarse si la condición no existe antes de hacer la evaluación

① Introducción

② Conceptos Básicos

③ Espacio de
nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

Introducción
for
while
do while

⑦ Arreglos

⑧ Funciones

⑨ Apuntadores

⑩ Estructuras de
Datos

⑪ Objetos

⑫ Ordenamiento

⑬ Búsqueda

⑭ Grafos

⑮ Cadenas

⑯ Referencias

do while

- Es muy parecido a **while** con la diferencia de que primero ejecuta y después comprueba si dicha condición existe
- No se define un límite, mientras la condición exista, se seguirá ejecutando
- Necesitamos crear una manera de salir, por si la condición sigue existiendo
- Se ejecuta al menos una vez
- En Python no existe

1 Introducción

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

Introducción

Declaración

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos
Introducción
Declaración

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

Introducción

- Son de un solo tipo
- Puede contener múltiples valores
- Su espacio en memoria varía según el tipo de dato y los valores que pueda tener
- En Python como tal no existen, se usan las listas en su lugar

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos
Introducción
Declaración

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

Componentes

- Son de un solo tipo
- Puede contener múltiples valores
- Su espacio en memoria varía según el tipo de dato y los valores que pueda tener
- En Python como tal no existen, se usan las listas en su lugar

1 Introducción

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

➊ Introducción

➋ Conceptos Básicos

➌ Espacio de
nombres

➍ Operaciones

➎ Condicionales

➏ Ciclos

➐ Arreglos

➑ Funciones
Introducción

➒ Apuntadores

➓ Estructuras de
Datos

➑ Objetos

➒ Ordenamiento

➓ Búsqueda

➑ Grafos

➑ Cadenas

➑ Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

① Introducción

② Conceptos Básicos

③ Espacio de
nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

⑨ Apuntadores
Introducción

⑩ Estructuras de
Datos

⑪ Objetos

⑫ Ordenamiento

⑬ Búsqueda

⑭ Grafos

⑮ Cadenas

⑯ Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

➊ Introducción

➋ Conceptos Básicos

➌ Espacio de
nombres

➍ Operaciones

➎ Condicionales

➏ Ciclos

➐ Arreglos

➑ Funciones

➒ Apuntadores

➓ Estructuras de
Datos

Introducción

➑ Objetos

➒ Ordenamiento

➓ Búsqueda

➒ Grafos

➓ Cadenas

➓ Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

- | | | | | | |
|---|--------------------|----|----------------------|--------------|--------------|
| 1 | Introducción | 6 | Ciclos | Introducción | |
| 2 | Conceptos Básicos | 7 | Arreglos | 12 | Ordenamiento |
| 3 | Espacio de nombres | 8 | Funciones | 13 | Búsqueda |
| 4 | Operaciones | 9 | Apuntadores | 14 | Grafos |
| 5 | Condicionales | 10 | Estructuras de Datos | 15 | Cadenas |
| | | 11 | Objetos | 16 | Referencias |

1 Introducción

2 Conceptos Básicos

3 Espacio de nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento
Introducción

13 Búsqueda

14 Grafos

15 Cadenas

16 Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda
Introducción

14 Grafos

15 Cadenas

16 Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos
Introducción

15 Cadenas

16 Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

1 Introducción

2 Conceptos Básicos

3 Espacio de
nombres

4 Operaciones

5 Condicionales

6 Ciclos

7 Arreglos

8 Funciones

9 Apuntadores

10 Estructuras de
Datos

11 Objetos

12 Ordenamiento

13 Búsqueda

14 Grafos

15 Cadenas
Introducción

16 Referencias

① Introducción

② Conceptos Básicos

③ Espacio de nombres

④ Operaciones

⑤ Condicionales

⑥ Ciclos

⑦ Arreglos

⑧ Funciones

Referencias I

- [1] BHASIN, H.
Python Basics, 3 ed.
David Pallai, Mercury Learning and Information, 2019.
- [2] CAIRÓ, O., AND GUARDATI, S.
Algorithms, 3 ed.
McGraw-Hill Interamericana, 2006.
- [3] DEITEL, P., AND DEITEL, H.
C++ Como Programar, 9 ed.
Pearson Educación de México, 2014.
- [4] DOWNEY, B., A.
Think Python, 2 ed.
O'Reilly, 2016.

Referencias II

- [5] JAWORSKI, M., AND ZIADÉ, T.
Expert Python Programming, 3 ed.
Packt Publishing, 2019.
- [6] LAAKMANN, M., G.
Cracking the Coding Interview, 6 ed.
CarrerCup, 2016.
- [7] MATTHES, E.
Python Crash Course, 2 ed.
No Star Press, 2019.
- [8] RAMALHO, L.
Fluent Python, 1 ed.
O'Reilly, 2014.

Referencias III

- [9] REEK, K.
Pointers On C.
Addison-Wesley Longman, 1997.
- [10] SEDGEWICK, R., AND WAYNE, K.
Algorithms, 4 ed.
Pearson Education, 2011.
- [11] STROUSTRUP, B.
The C++ Programming Language, 4 ed.
Pearson Education, 2013.
- [12] STROUSTRUP, B.
Principles and Practice Using C++, 2 ed.
Pearson Education, 2014.

Referencias IV

[13] UNKNOWN.
Cplusplus.

Thanks!