



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

‘Mision
<TIC2022’

‘Mision
TIC2022’

Aprendiendo a programar

Diego Iván Oliveros Acosta



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

calapp.co



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

Mision
<TIC2022>



CICLO 1 - FUNDAMENTOS DE PROGRAMACIÓN

Introducción a la Algoritmia e Introducción al **Lenguaje de Programación Python**

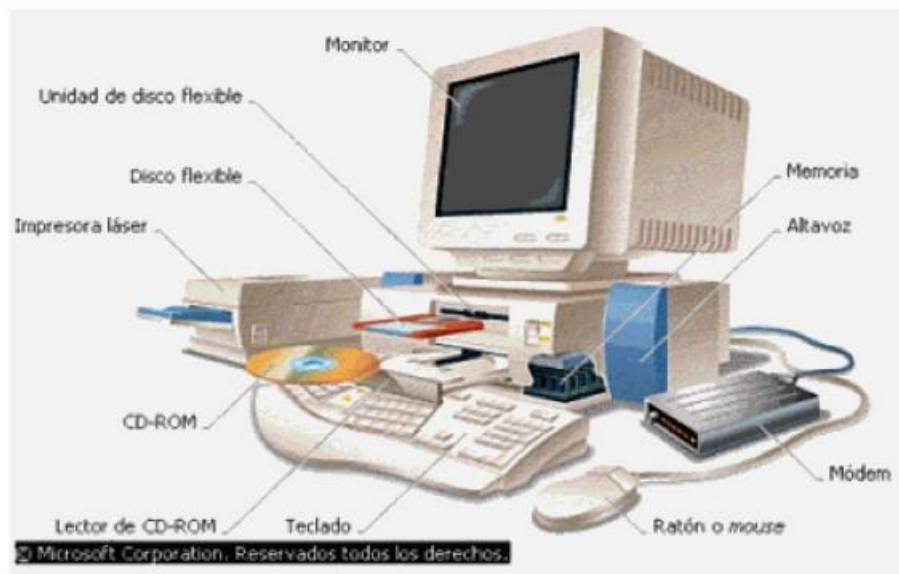


Diego Iván Oliveros Acosta

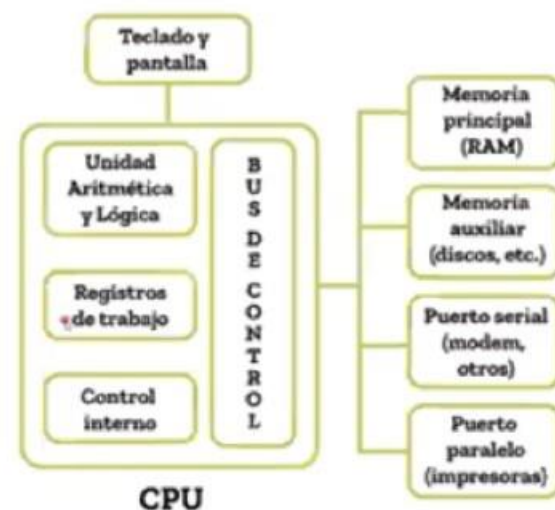
@scalapp.co



Esquema general de un computador



Esquema general de
un computador

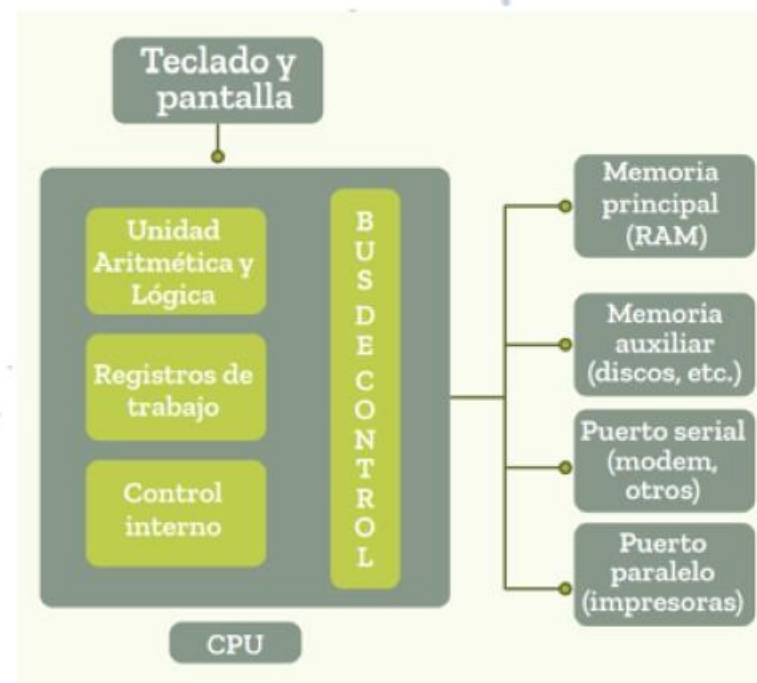




Computadores

¿Qué es un computador?

- Un **computador** es un dispositivo electrónico que almacena y procesa datos.
- Un computador incluye tanto *hardware* como *software*. En general, el **hardware** comprende los elementos físicos visibles de la computadora, y el **software** proporciona las instrucciones invisibles que controlan el hardware y lo hacen realizar tareas específicas.

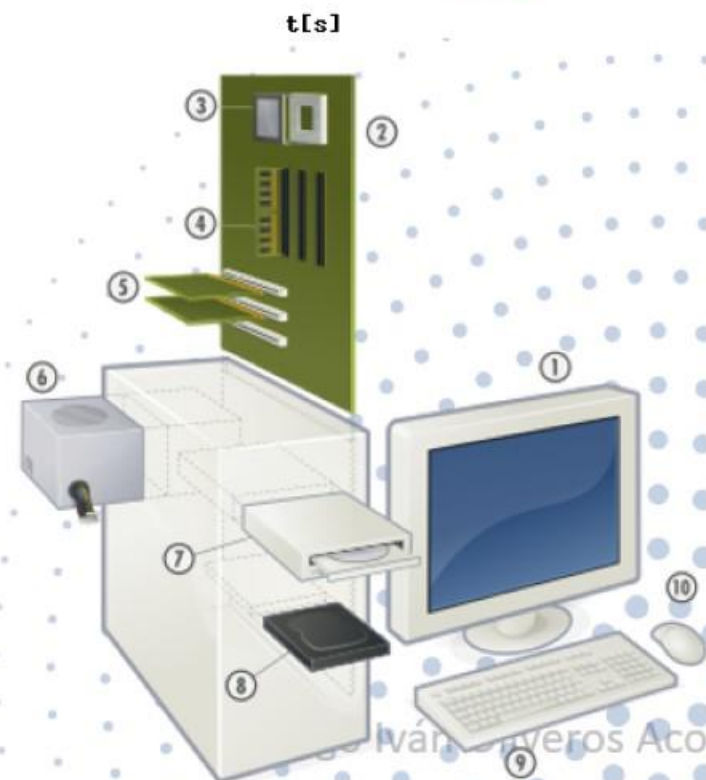
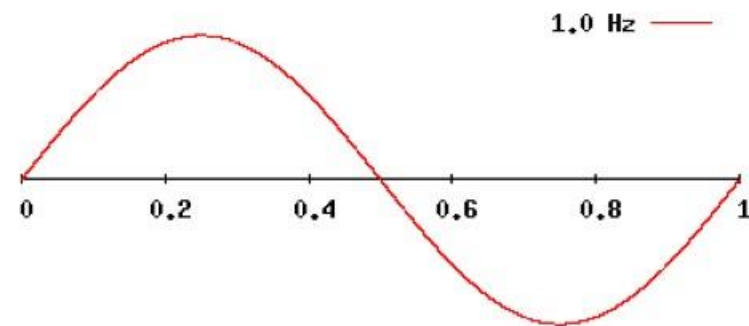




Hardware

En general un computador un computador contiene los siguientes componentes de hardware:

- Una unidad central de procesamiento (CPU)
- Memoria
- Dispositivos de almacenamiento (como discos y CD)
- Dispositivos de entrada (como el mouse y el teclado)
- Dispositivos de salida (como monitores e impresoras)
- Dispositivos de comunicación (como módems y tarjetas de red).





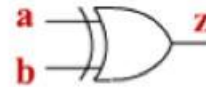

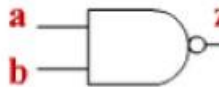

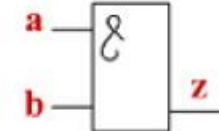
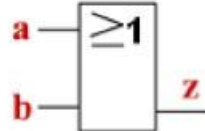
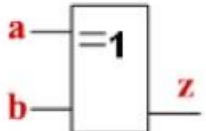
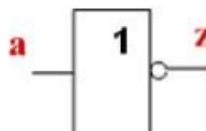
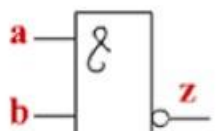
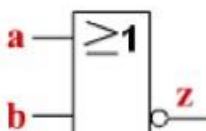
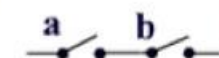







Memoria

- La memoria de una computadora consiste en una secuencia ordenada de bytes para almacenar programas, así como los datos con los que está trabajando el programa.
- Puede pensar en la memoria como el área de trabajo de la computadora para ejecutar un programa.
- Un programa y sus datos deben trasladarse a la memoria de la computadora antes de que la CPU pueda ejecutarlos.

Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01000011	Encoding for character 'C'
2001	01110010	Encoding for character 'r'
2002	01100101	Encoding for character 'e'
2003	01110111	Encoding for character 'w'
2004	00000011	Decimal number 3
.	.	
.	.	

FUNCIONES LÓGICAS BÁSICAS

NOMRE	AND - Y	OR - O	XOR O-exclusiva	NOT Inversor	NAND	NOR																																																																																	
SÍMBOLO																																																																																							
SÍMBOLO																																																																																							
TABLA DE VERDAD	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	z	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	z	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	z	0	0	0	0	1	1	1	0	1	1	1	0	<table><tr><th>a</th><th>z</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	a	z	0	1	1	0	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	z	0	0	1	0	1	1	1	0	1	1	1	0	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	z	0	0	1	0	1	0	1	0	0	1	1	0
a	b	z																																																																																					
0	0	0																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	1																																																																																					
a	b	z																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	1																																																																																					
a	b	z																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
a	z																																																																																						
0	1																																																																																						
1	0																																																																																						
a	b	z																																																																																					
0	0	1																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
a	b	z																																																																																					
0	0	1																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	0																																																																																					
EQUIVALENTE EN CONTACTOS																																																																																							
AXIOMA	$z = a \cdot b$	$z = a + b$	$z = \bar{a} \cdot b + a \cdot \bar{b}$	$z = \bar{a}$	$z = \overline{a \cdot b}$	$z = \overline{a + b}$																																																																																	



Lenguajes de programación

¿Qué son los lenguajes de programación?
¿Por qué son necesarios?

- Las computadoras no entienden los idiomas humanos, por lo cual, los programas deben estar escritos en un idioma que una computadora pueda usar. Hay cientos de lenguajes de programación, y fueron desarrollados para facilitar el proceso de programación para las personas. Sin embargo, todos los programas al final deben convertirse en instrucciones que los computadores puedan ejecutar.





Tipos de lenguaje - I

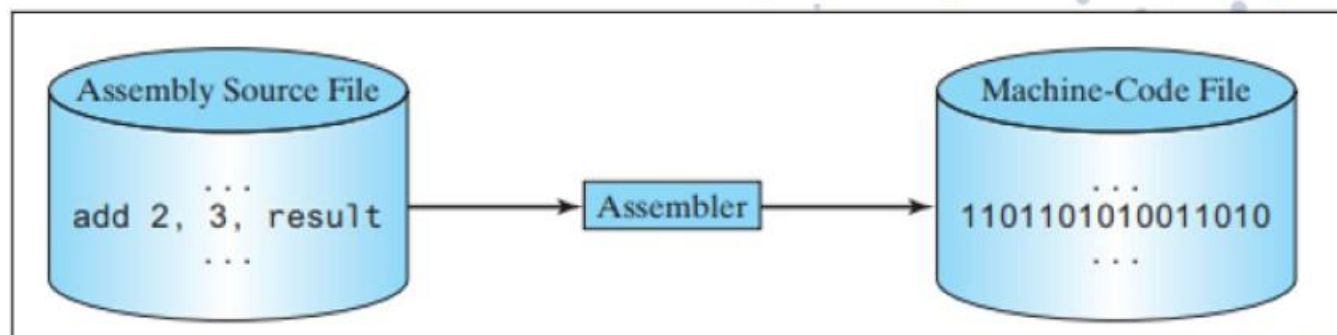
Lenguaje de maquina

Es el idioma nativo de un computador. El cual esta hecho en forma de código binario.



Lenguaje de ensamblador

Programar a punta de unos y ceros es muy tedioso, por lo tanto a inicios de la era de la computación, se inventó el lenguaje de ensamblador. El lenguaje ensamblador utiliza palabras descriptivas cortas para representar un conjunto de instrucciones de maquina

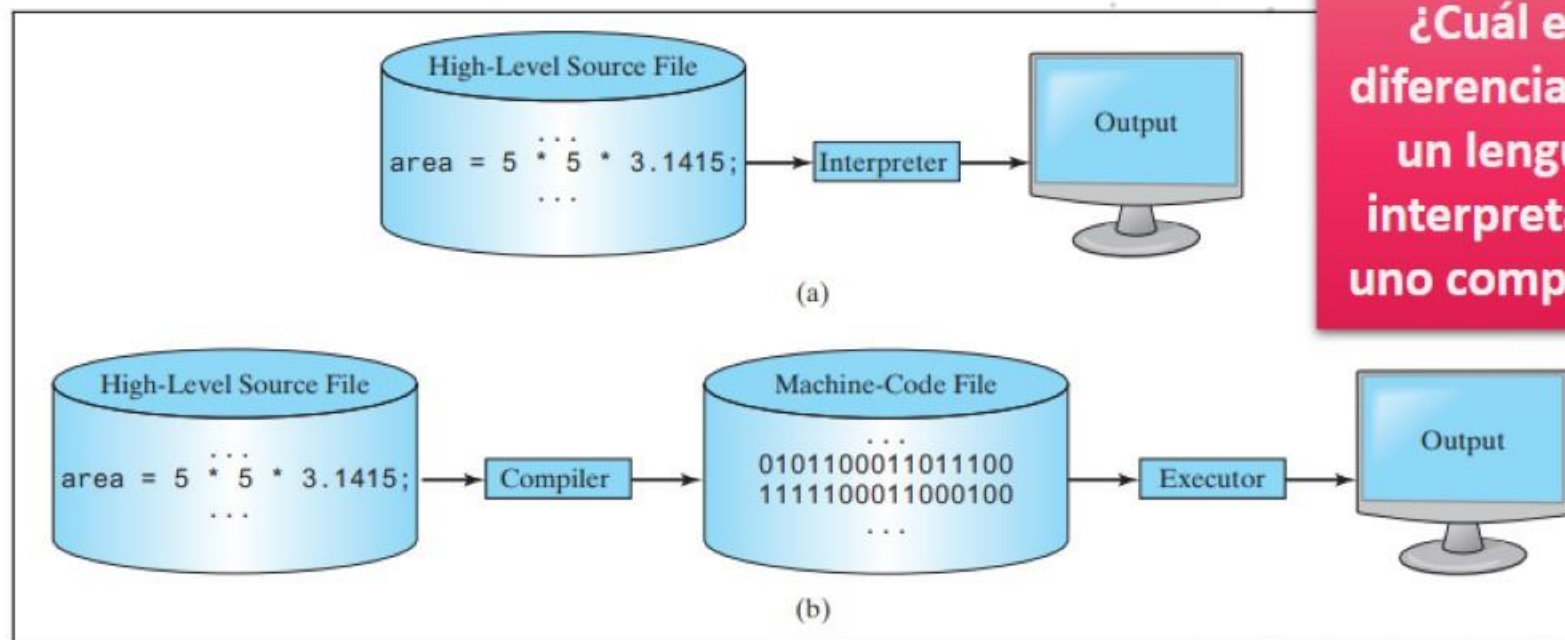




Tipos de lenguaje - II

Lenguaje de alto nivel

En la década de 1950, surgió una nueva generación de lenguajes de programación conocidos como “lenguajes de alto nivel”. Estos lenguajes son independientes de la plataforma, lo que significa que se puede escribir un programa en un lenguaje de alto nivel y ejecutarlo en diferentes tipos de máquinas. Los lenguajes de alto nivel son similares al idioma “Inglés”, y son más fáciles de aprender y de usar.



¿Cuál es la
diferencia entre
un lenguaje
interpretado y
uno compilado?



Metodología de desarrollo de aplicaciones

Pasos en la construcción de una aplicación por computador

1. Análisis del problema
2. Diseño de la solución
3. Implementación de la solución diseñada
 - 3.1 Elaboración de algoritmos
 - 3.1.1 Análisis
 - 3.1.2 Diseño
 - 3.1.3 Construcción del algoritmo
 - 3.1.4 Pruebas de escritorio
 - 3.2 Codificación en algún lenguaje de programación
 - 3.3 Compilación
 - 3.4 Pruebas sistematizadas
4. Pruebas con el usuario
5. Puesta en marcha



Algoritmos

¿Qué es un algoritmo?

- En matemáticas y ciencias de la computación, un algoritmo es una secuencia finita de instrucciones, las cuales están bien definidas y se pueden implementar en una computadora. Típicamente, estas instrucciones se utilizan para resolver un tipo de problema o realizar un cálculo.



¿Qué es un algoritmo?

Algoritmo

- Un algoritmo es una secuencia de instrucciones con las cuales se instruye a *otra persona* para que haga una tarea que uno sabe hacer.
- Con base en esta definición, la gran deducción es que para escribir un algoritmo es absolutamente necesario que uno sepa hacer esa tarea.
- En nuestro contexto, *otra persona* es la máquina, es decir, el computador.

Características de un algoritmo

- **Tiene inicio y fin:** el número de pasos de un algoritmo es finito.
- **Funciona en secuencia:** las instrucciones se ejecutan exactamente en el orden en que están escritas.
- **Cada instrucción es clara y precisa.**



Ejemplo algoritmo

Objetivo: calcular el promedio de edad de los trabajadores de una oficina.

Secuencia de pasos:

- **Paso 1:** Recoger la edad del integrante 1
- **Paso 2:** Recoger la edad del integrante 2
- **Paso 3:** Recoger la edad del integrante 3
- **Paso 4:** Recoger la edad del integrante 4
- **Paso 5:** Recoger la edad del integrante 5
- **Paso 6:** Recoger la edad del integrante 6
- **Paso 7:** Sumar la edad de los integrantes 1, 2, 3, 4, 5 y 6.
- **Paso 8:** Dividir el total de la suma anterior entre 6.



$$12 + 18 + 23 + 40 + 20 + 23 = 136$$

$$136 / 6 = 22.66$$



1. Análisis del
problema

2. Diseño de
la solución

3.
Construcción
del algoritmo

3. Validación
de la
solución/
Prueba de
escritorio

4.
Codificación
en un
lenguaje de
programación



1. Análisis para diseñar un algoritmo

Realizar las siguientes preguntas:

- ¿Cuál es el objetivo buscado?
- ¿Cuáles son los datos de entrada?
- ¿Qué cálculos/procesos deben llevarse a cabo?
- ¿Cuáles son los datos de salida?



2. Diseño de la solución

- **Objetivo:** calcular el promedio de edad de los trabajadores de una oficina.
- **Datos de entrada:** edad integrante 1, edad integrante 2, edad integrante 3, edad integrante 4, edad integrante 5, y edad integrante 6.
- **Procesos / cálculos:** sumar las edades de los integrantes 1, 2, 3, 4, 5, y 6. Y dividir el total por 6.
- **Datos de salida:** promedio de edad de los 6 integrantes.

3. Construcción del algoritmo

Pseudocódigo
(P-código)

inicio

acción 1

acción 2

acción n

...

fin



3. Validación de la solución

- Para validar la solución de un algoritmo, se pueden realizar pruebas de escritorio (las cuales no requieren el uso de computadores). O también se puedan realizar pruebas en computadores, para lo cual se necesita tener codificado el algoritmo en un lenguaje de programación.

Validación manual (prueba de escritorio):

- Determinar los valores finales de las variables suponiendo que se tienen las siguientes instrucciones:
 - $i = 9$
 - $j = 4$
 - $i = i + j$
 - $i = j - 2$

Línea	i	j
1		
2		
3		
4		



4. Codificación en un lenguaje de programación

```
main.py  saved
1  edad1 = 12
2  edad2 = 18
3  edad3 = 23
4  edad4 = 40
5  edad5 = 20
6  edad6 = 23
7
8  suma = edad1+edad2+edad3+edad4+
9  promedio = suma/6
10
11 print(promedio)
```

```
public class Welcome {
    public static void main(String[] args) {
        // Display message Welcome to Java! on the console
        System.out.println("Welcome to Java!");
    }
}
```



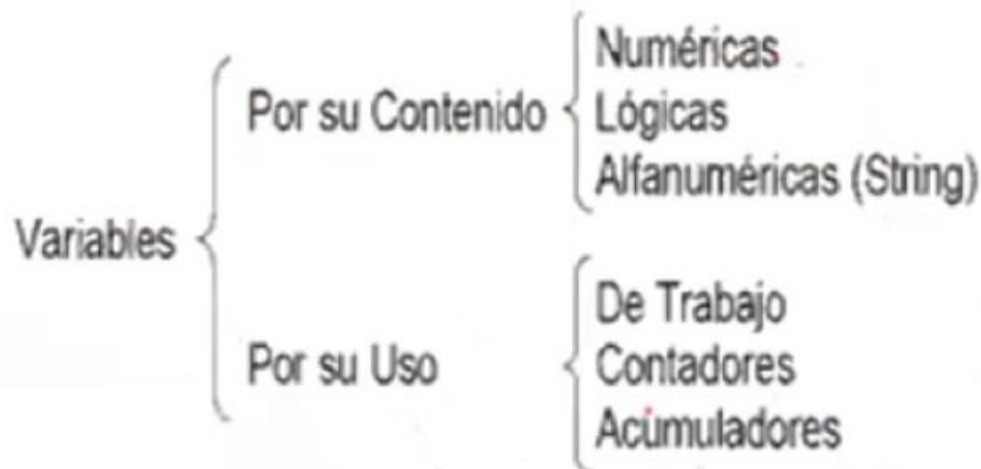
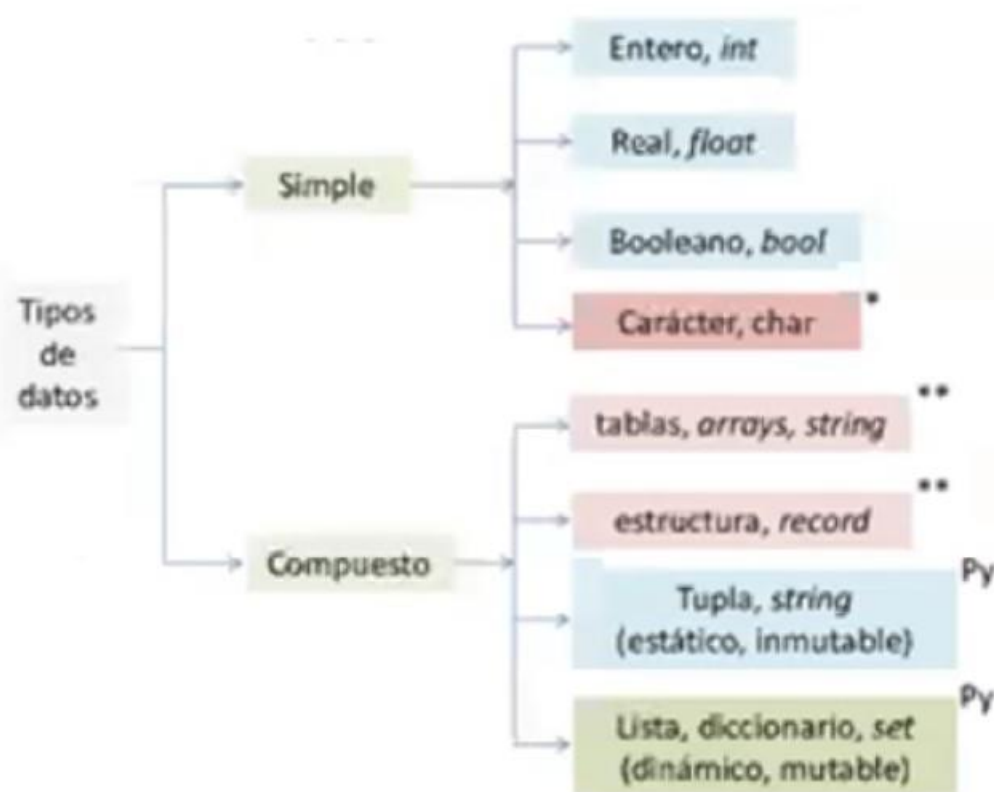
Algoritmo, Datos, Constantes y Variables

Datos

- Los datos son la información con la que trabaja un algoritmo.
- Pueden ser numéricos o no numéricos.
- Los datos numéricos pueden ser:
 - Enteros (int)
 - Reales (tienen parte decimal) (float)
- Los datos no numéricos son de tipo string (hilera o cadena) (str)
- Datos lógicos: pueden ser verdadero (True) o falso (False) (bool)



¿Qué son las variables?





Main data types

boolean = *True / False*
integer = 10
float = 10.01
string = "123abc"
list = [value1, value2, ...]
dictionary = { key1:value1, key2:value2, ... }

Numeric operators

+ addition
- subtraction
* multiplication
/ division
** exponent
% modulus
// floor division

Comparison operators

== equal
!= different
> higher
< lower
>= higher or equal
<= lower or equal

Boolean operators

and logical AND
or logical OR
not logical NOT

Special characters

coment
\n new line
\<char> scape char

String operations

string[i] retrieves character at position i
string[-1] retrieves last character
string[i:j] retrieves characters in range i to j

List operations

list = [] defines an empty list
list[i] = x stores x with index i
list[i] retrieves the item with index i
list[-1] retrieves last item
list[i:j] retrieves items in the range i to j
del list[i] removes the item with index i

Dictionary operations

dict = {} defines an empty dictionary
dict[k] = x stores x associated to key k
dict[k] retrieves the item with key k
del dict[k] removes the item with key k

String methods

string.upper() converts to uppercase
string.lower() converts to lowercase
string.count(x) counts how many times x appears
string.find(x) position of the x first occurrence
string.replace(x,y) replaces x for y
string.strip(x) returns a list of values delimited by x
string.join(L) returns a string with L values joined by string
string.format(x) returns a string that includes formatted x

List methods

list.append(x) adds x to the end of the list
list.extend(L) appends L to the end of the list
list.insert(i,x) inserts x at i position
list.remove(x) removes the first list item whose value is x
list.pop(i) removes the item at position i and returns its value
list.clear() removes all items from the list
list.index(x) returns a list of values delimited by x
list.count(x) returns a string with list values joined by S
list.sort() sorts list items
list.reverse() reverses list elements
list.copy() returns a copy of the list

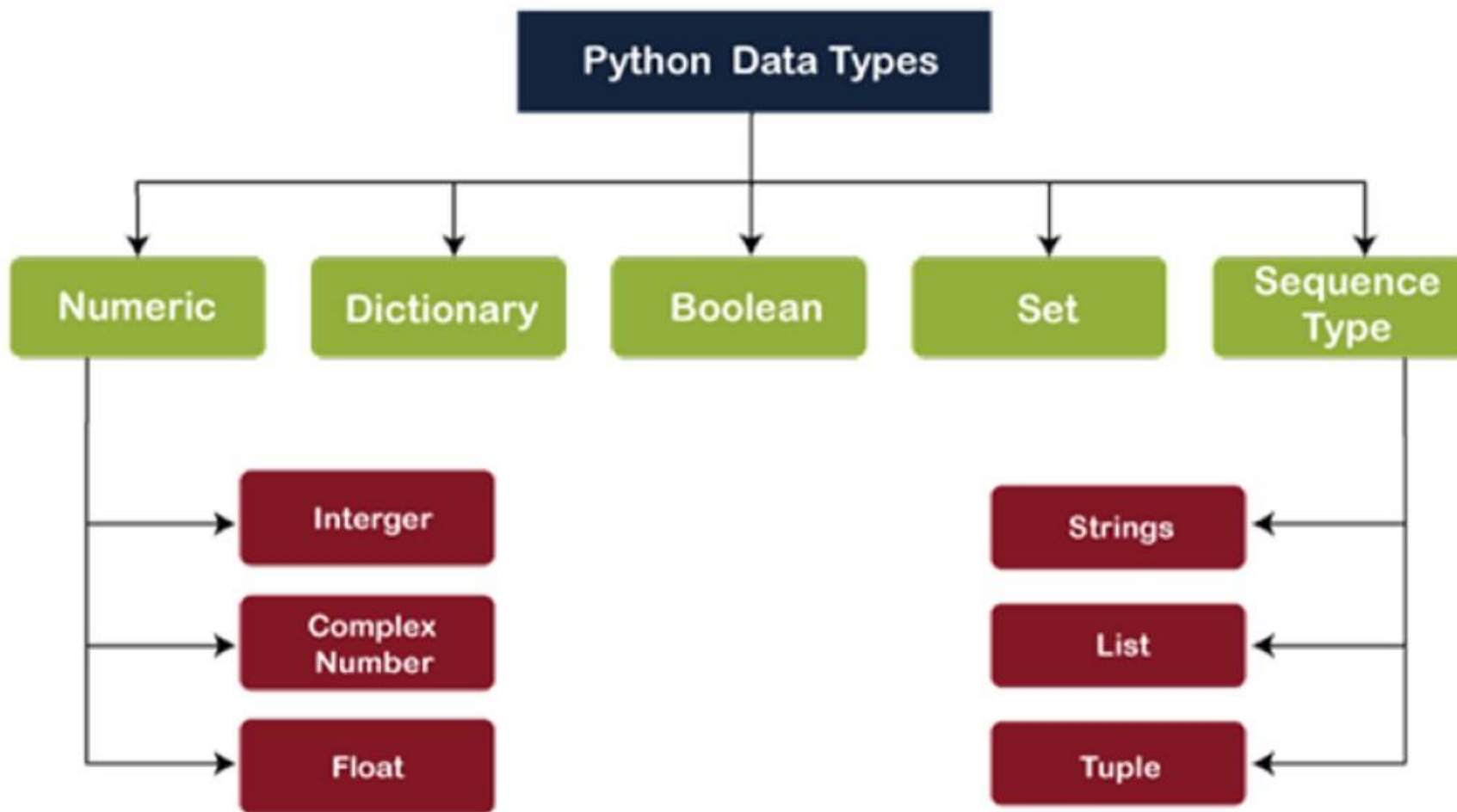
Dictionary methods

dict.keys() returns a list of keys
dict.values() returns a list of values
dict.items() returns a list of pairs (key,value)
dict.get(k) returns the value associated to the key k
dict.pop() removes the item associated to the key and returns its value
dict.update(D) adds keys-values (D) to dictionary
dict.clear() removes all keys-values from the dictionary
dict.copy() returns a copy of the dictionary

Legend: x,y stand for any kind of data values, s for a string, n for a number, L for a list where i,j are list indexes, D stands for a dictionary and k is a dictionary key.

in Oliveros Acosta

@scalapp.co





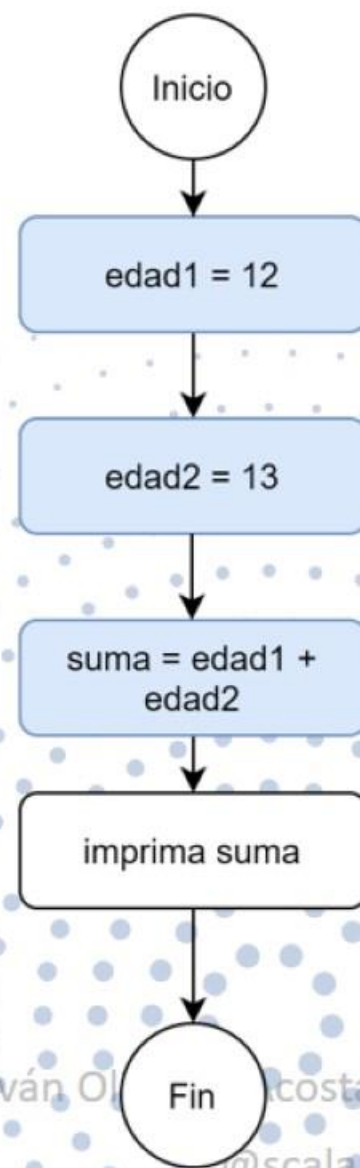
Estructura de un algoritmo

- Entrada de datos
- Procesamiento de datos
- Salida de resultados

Estructuras para la construcción de un algoritmo

- Estructura secuencia
- Estructura decisión
- Estructura ciclo

```
main.py  saved
1  edad1 = 12
2  edad2 = 18
3  edad3 = 23
4  edad4 = 40
5  edad5 = 20
6  edad6 = 23
7
8  suma = edad1+edad2+edad3+edad4+edad5+edad6
9  promedio = suma/6
10
11 print(promedio)
```





Python ¿Cómo lo instalo?

- Es un lenguaje de programación de alto nivel.
- Es interpretado.
- Creado a finales de los 80
inicios de los 90.



python



Ventaja o desventaja

Entrada y Salida de datos

```
public class Welcome {  
    public static void main(String[] args) {  
        // Display message Welcome to Java! on the console  
        System.out.println("Welcome to Java!");  
    }  
}
```

Python

```
1 print("hello world");
```




Expresiones aritméticas

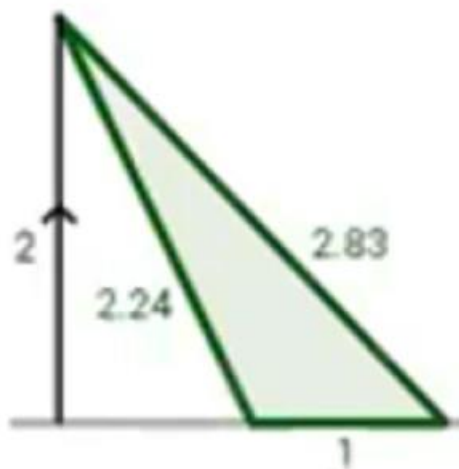
Operador	Operación
**	Potenciación
*	Multiplicación
/	División real
//	División entera
%	Módulo
+	Suma
-	Resta



Primeros pasos

Problema 1

Calcular el área del siguiente triángulo de lados 2.24cm, 2.83cm y 1cm y de altura 2cm.





$$2x^2 - 7x + 3 = 0$$

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- 1 Identificamos los valores de a, b y c

$$a = 2 \quad b = -7 \quad c = 3$$

- 2 Sustituimos en la fórmula general y resolvemos

$$x = \frac{-(-7) \pm \sqrt{(-7)^2 - (4)(2)(3)}}{(2)(2)}$$



Ejercicio 2

El gran gatsby ha decidido comprar 6 luces refractarias, y 3 pares de acetatos, cada uno de diferente color (amarillo, azul y rojo). Esto con el fin de comunicarse con su mansión en Long Island (Nueva York).

La espectacular vivienda está situada en la pequeña localidad de Great Neck, en el condado de Suffolk, a menos de una hora de la isla de Manhattan, y cuenta con siete habitaciones y seis cuartos de baño repartidos en 480 metros cuadrados y un largo muelle donde se colocará un set de refractarias con una señal de estado de funcionamiento, esta última avisará a las embarcaciones que todo está bien por lo que se entiende que la información solo será válida cuando esté encendida.



La segunda mansión, queda justo en frente y se trata de un edificio de estilo mediterráneo construido en 1918 y según consta en los registros de la propiedad, sus actuales dueños lo compraron en 2008 por unos 4,2 millones de dólares, según detalla el periódico neoyorquino. El dueño de la primera mansión se casó a sus 50 años con una hermosa mujer cuando esta tenía tan solo 15 años. Ahora a sus 25 tiene un pretendiente que vive justo al otro lado de la isla.

Ella sabe que el dueño de la mansión **cada semana envía correspondencia membretada Logo y nombre de la empresa. Nombre y apellidos del profesional. Razón social. Dirección y datos de contacto (teléfono, correo electrónico) y además va adornada con unos y ceros. Sabe que si altera la imagen podría enviar el código secreto que indique cómo usar las luces y así poder comunicarse en directo con alguien en la otra mansión.**

Reto 1: Diseñe un programa que imprima todas las posibles combinaciones.



Operadores de asignación en python

Operador	Ejemplo	Lo mismo que
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

Operador	Ejemplo	Lo mismo que
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3



Reto 3: escriba un programa que solucione las siguientes formulas.

$$a + (c - c * (d - (a + b) * (c + b / d * a) \% a + b * c) - (b ** c)) ** b$$

$$e * f + (b - c * (d - (a + b) * (c - e / d * a) / a + b * c) - (e ** f)) ** b$$



Referencias

<https://www.anaconda.com/products/distribution>

Python.org

<https://nbviewer.jupyter.org/>

<https://rise.readthedocs.io/en/stable/installation.html>

<https://jupyter.org/try>

<https://github.com/DiegOliveros/MISION-TIC-2020>

<https://desktop.github.com/>

<https://www.jetbrains.com/es-es/pycharm/download/#section=windows>

<https://notepad-plus-plus.org/downloads/>

<https://www.duolingo.com/>

<https://www.facebook.com/Scalapp.co>