

Contenido

- Problemas y soluciones
- Python
- Variables
- Operadores básicos
- Instrucciones básicas
 - asignación
 - entrada y salida
 - funciones

PROBLEMAS Y SOLUCIONES

"Es tonto responder una pregunta que no se entiende"

George Polya

PROBLEMA

Retirar dinero de un cajero automático

Análisis

El objetivo es especificar, entender el problema

Contexto

Sistema Bancario

Condiciones iniciales

- ☐ Se conoce el banco donde se tiene el dinero
- ☐ Se conoce el saldo
- ☐ Se conoce la cantidad a retirar

Condiciones finales

- ☐ Se obtiene el dinero

SOLUCIÓN

PROBLEMAS Y SOLUCIONES

"Es tonto responder una pregunta que no se entiende"

George Polya

PROBLEMA

Retirar dinero de un cajero automático

Diseño

El objetivo es detallar con **dibujos, diagramas, texto, etc**, las características que tendrá la solución antes de ser construida.

El diseño permite **mostrar la solución antes de comenzar el proceso de construcción**

SOLUCIÓN

PROBLEMAS Y SOLUCIONES

"Es tonto responder una pregunta que no se entiende"

George Polya

PROBLEMA

Retirar dinero de un cajero automático

Construcción

Tiene por objetivo realizar el algoritmo solución

Algoritmo

1. Insertar la tarjeta
2. Seleccionar el idioma
3. Ingresar la clave
4. Seleccionar la operación por realizar
5. Ingresar los datos de acuerdo con la operación seleccionada
6. Obtener el resultado de la operación

SOLUCIÓN

Conceptos básicos

Programa:

conjunto de funciones o rutinas
es la implementación de un algoritmo



Phyton

Python es un **lenguaje de programación:**

- ☐ interpretado cuya filosofía hace hincapié en la legibilidad de su código.
- ☐ multiparadigma, ya que soporta **orientación a objetos, programación imperativa** y, en menor medida, **programación funcional**.
- ☐ dinámico
- ☐ multiplataforma.
- ☐ sintaxis visual
- ☐ poderoso y fácil de aprender.

Fue creado a finales de los ochenta por **Guido van Rossum** en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una **notación indentada** (con márgenes) de obligado cumplimiento.



Variables

✓ Lugares de memoria en los que se almacenan datos.

✓ Su contenido puede variar

Tipo

- ✓ entero
- ✓ real
- ✓ carácter
- ✓ cadena de caracteres

Nombre

Solo debe contener solo caracteres alfanuméricos y el carácter subrayado(_) y tiene que empezar con un carácter alfabético

Contenido

Ligado al tipo

Phyton

Tipos de datos básicos

• Enteros	3	int
• Reales	3.8	float
• Complejos	1j	complex
• Cadenas de caracteres	"Martha"	str
• Booleano	True o False	bool

En Phyton no se declaran las variables pero en la mayoría de los lenguajes si

Tipos de dato estructurados

Listas

Diccionarios

Tuplas

Conjuntos

Rango



Phyton

Operaciones básicas

- Suma +
- Resta -
- Multiplicación *
- División / Resultado es un real
- Cociente de división // Resultado número entero
- Residuo %
- Potencia **

Ejemplo:

$$2+3 =5$$

$$3-5 =-2$$

$$4*5 =20$$

$$3/2 =1,5$$

$$5//2 = 2$$

$$4\%2 =0$$

$$3**2 =9$$



Phyton

Instrucciones básicas

- **Asignación**

=

variable = expresión aritmética

ejemplo

base=7

altura= 8.2

area = base x altura



Orden de evaluación de una expresión aritmética

Para evaluar una expresión aritmética hay que saber en qué orden se ejecutan las operaciones.

Los operadores tienen una jerarquía y pueden estar más de una vez en una expresión. Los paréntesis () son los operadores de agrupación y tienen la más alta jerarquía o mayor prioridad. En una expresión, el orden de evaluación es:



1. Lo que está entre paréntesis.

2. Menos unario: indica que se invierte el signo de la variable, constante o expresión a la que precede. Ejemplo: $-a$, -11 , $-(b*c)$.

3. Multiplicación y división.

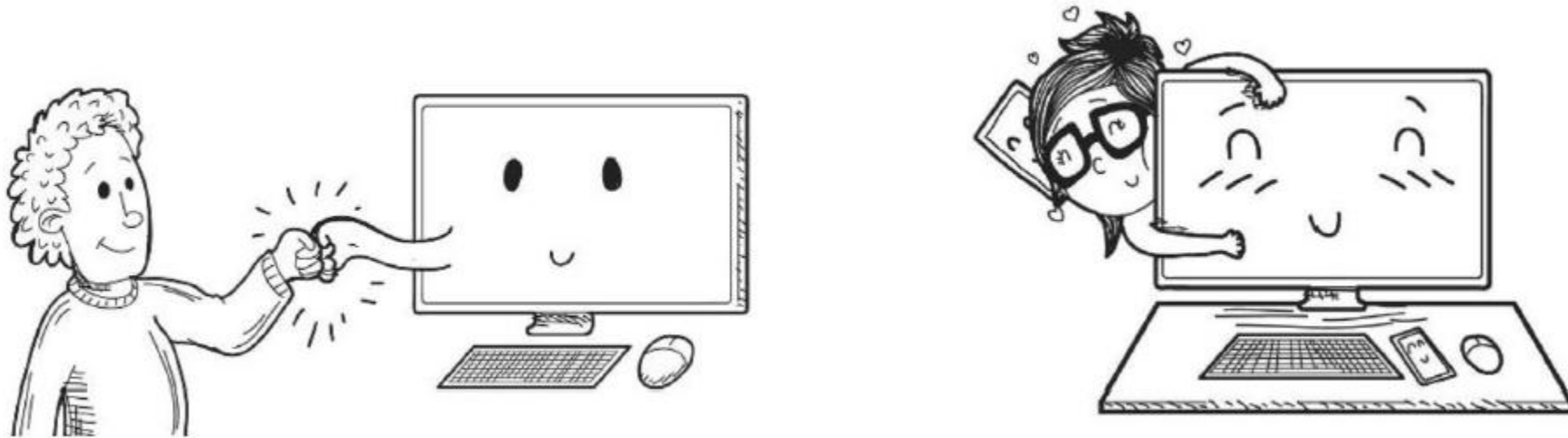
4. Suma y resta.

5. La asignación. Debido a que tiene la más baja prioridad, primero se evalúa toda la expresión del lado derecho y luego se asigna el valor a la variable que está en la izquierda de la expresión.

— Si dos operadores tienen la misma prioridad, entonces primero se realiza el operador que está a la izquierda y luego el que está a la derecha.

— Si dos operadores son de diferente jerarquía, se ejecuta el de mayor jerarquía y se deja pendiente el operador de menor jerarquía, hasta que se encuentre otro operador de la misma jerarquía o ya no haya operaciones pendientes por realizar.

Instrucciones de entrada y salida



El usuario es quien hace los requerimientos y se beneficia del algoritmo.

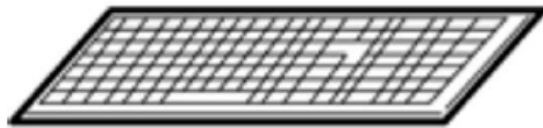
Las instrucciones que permiten la interacción entre el usuario y la máquina son las llamadas instrucciones de entrada y salida.

Permiten que éste **“hable”** con la máquina y que ésta le **“responda”** y viceversa.

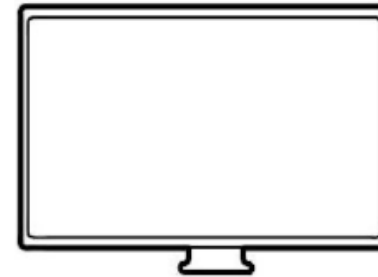
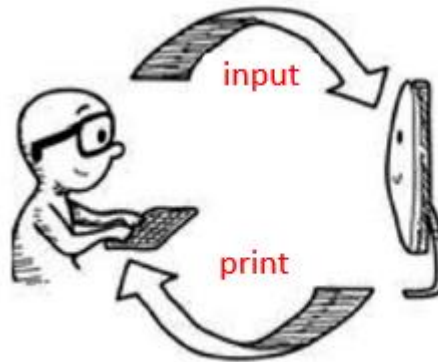
Instrucciones de entrada y salida

Estas instrucciones son las de lectura y escritura y están estrechamente relacionadas con los dispositivos de entrada y salida

El **teclado** como dispositivo de **entrada** y con la **pantalla** como dispositivo de **salida**.



comunicación entre el usuario y la máquina, a través del usuario, se ejecuta una instrucción de **lectura**; el usuario “tiene la palabra” (**teclado**)



comunicación se hace a través de la máquina, se ejecuta una instrucción de **salida**; la máquina “tiene la palabra” (**pantalla**).

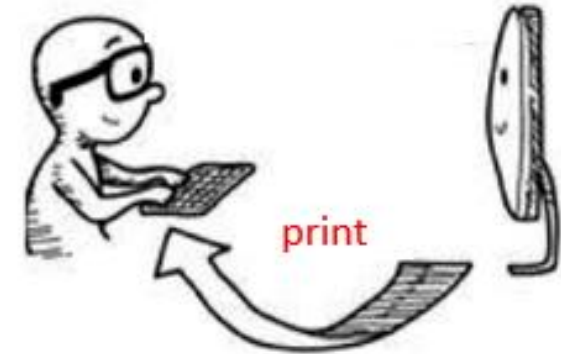
Phyton

Instrucciones básicas

Cómo escribir

- Mostrar por pantallas

`print()`



La función `print()` permite mostrar texto en pantalla.

Para escribir mensajes

Para escribir una expresión

Para escribir combinado

```
print ("Hola")
```

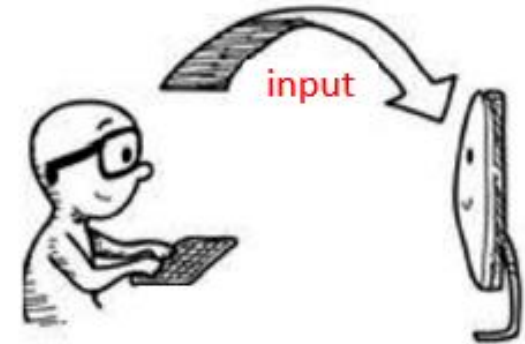
```
print( cont+1)
```

```
print ("El resultado es:", numero)
```

```
print("La suma de", a, "y", b, "es", suma)
```

Cómo leer

- **Ingreso por teclado** `input()`
La función `input()` permite obtener texto escrito por teclado.
- `print("Mensaje")` `print("¿Cómo se llama?")`
- `variable = input()` `nombre = input()`
- `variable = input("Mensaje")`
 `nombre = input("¿Cómo se llama?")`
- `variable = int(input("Mensaje"))`
 `edad = int(input("¿Cuál es tu edad?"))`
- `variable = float(input("Mensaje"))`
 `sueldo = float(input("¿Cuál es tu sueldo?"))`



Instrucciones que modifican contenido de una variable

- ✓ Asignación =
- ✓ Instrucción de entrada `input`

Estructura de una función

Definición

def nombre función (lista_de_parámetros):

cuerpo

return o return expresión  Cuando se requiera

Invocación o llamado Para que se ejecute
nombre_función (lista_de_argumentos)

Phyton

Función

Estructura de una función

Definición

def nombre función ():

 cuerpo

 return o return expresión  Cuando se requiera

Invocación o llamado Para que se ejecute

nombre_función ()

Ejemplos

Phyton

Sumar dos reales

Datos de entrada: los dos números (reales)

Dato de salida: suma de los números (Real)

def main():

print("Bienvenido")

print("Yo sumo dos números reales")

print("Ingrese el primer número")

num1=float(input())

print("Ingrese el segundo número")

num2=float(input())

suma=num1+num2

print(num1, "+", num2, "=", suma

print("Tarea finalizada")

Encabezado de la función

main()

Invocación o llamado Para que se ejecute

Ejemplos

Phyton

Calcular definitiva de un curso en la Escuela

Datos de entrada: las notas de tercio (reales)

Dato de salida: definitiva (real)

def main():

print("Calculo la definitiva de un curso")

print("Ingrese la nota del primer tercio")

nota1=float(input())

print("Ingrese la nota del segundo tercio")

nota2=float(input())

print("Ingrese la nota del primer tercio")

nota1=float(input())

print("Ingrese la nota del tercer tercio")

nota2=float(input())

definitiva=nota1*0.3+nota2*0.3+nota3*0.4

print("La nota definitiva del curso con notas", "nota 1=", nota1, "nota 2=", nota2, "nota 3=", nota3, "es=", def)

print("Tarea finalizada")

main()

Encabezado de la función

Invocación o llamado Para que se ejecute

Phyton

Instalación de Phyton (Versión 3.7 o superior)

[**https://www.python.org/downloads/**](https://www.python.org/downloads/)