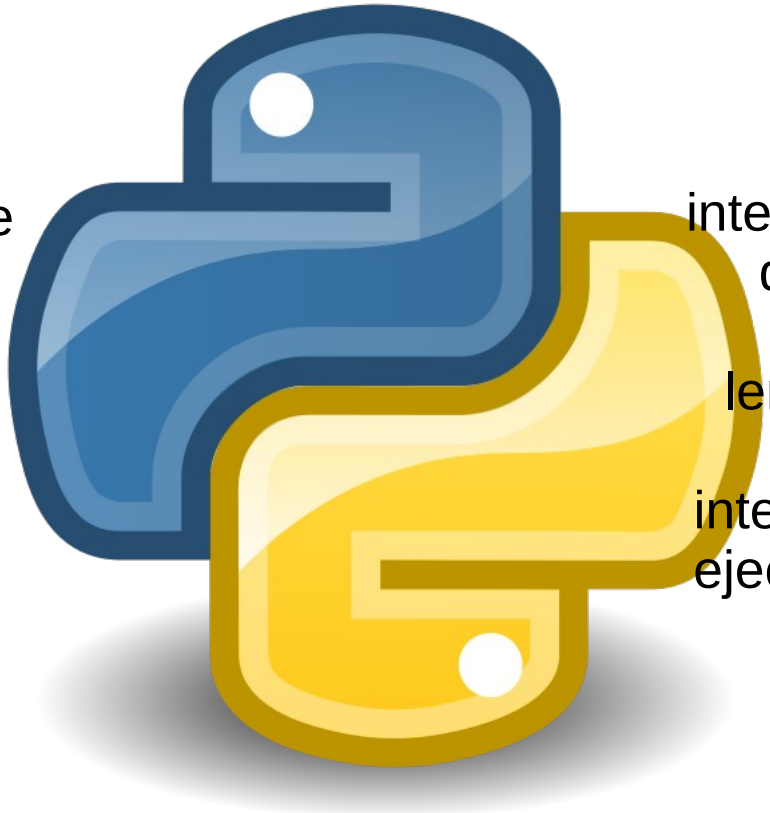
PYTHON

Python es un lenguaje de programación multiplataforma, consistente y maduro. Es potente, fácil de usar y de aprender. Es un lenguaje de propósito general



Es un lenguaje interpretado, esto significa que no se compila a diferencia de otros lenguajes como Java o C/C++, sino que es interpretado en tiempo de ejecución. Además, es de tipado dinámico.

PYTHON

Es un lenguaje de programación de código abierto, diseñado inicialmente por el neerlandés Guido Van Rossum a finales de los años 1980.

La primera versión Python 0.9.0 se lanzó en 1991.

Última versión estable: Python 3.11.5.

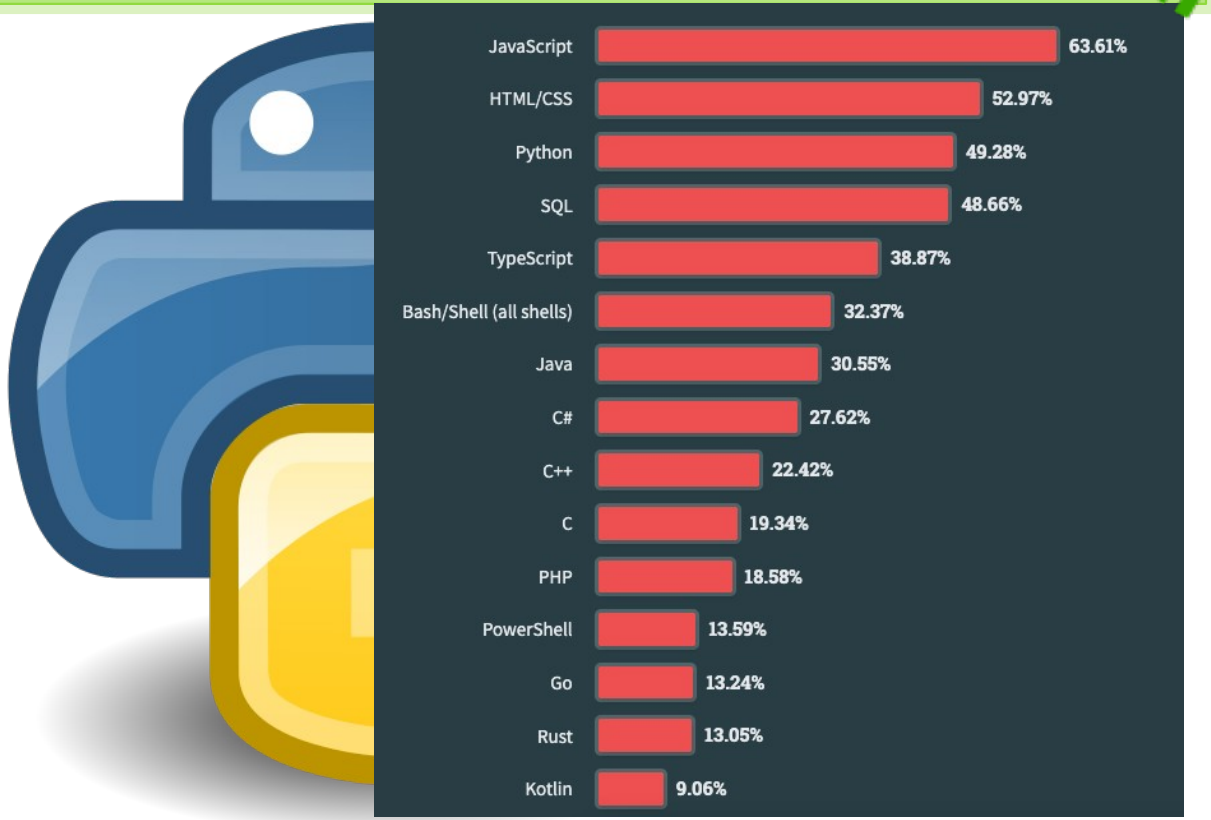
PYTHON

Se usa en muchos campos como el aprendizaje automático, la automatización y la computación científica. También puede servir como un lenguaje para aplicaciones web.

Hay diversos *frameworks* de Python como Django, TurboGears y Flask, que dan soporte a los programadores en el desarrollo de aplicaciones complejas.

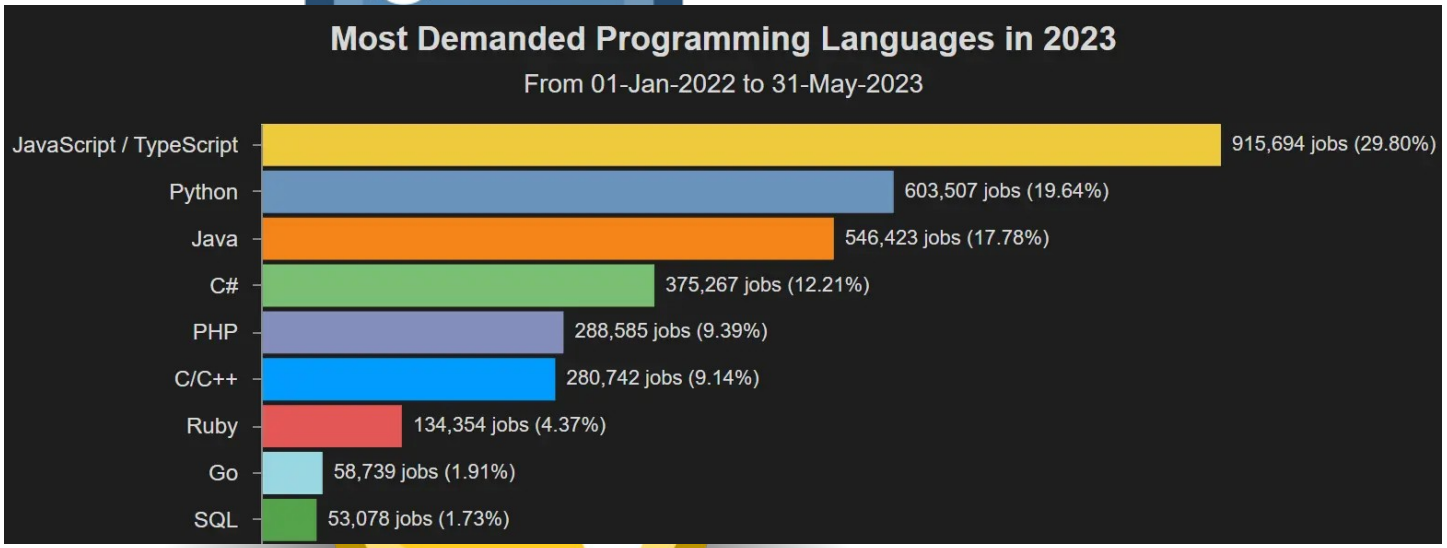
PYTHON

Lenguajes más demandados en el 2023



PYTHON

Lenguajes más demandados para el 2024



PYTHON

Identificadores

Un identificador es una secuencia de símbolos que se utilizan como nombres de variables, funciones, arreglos, clases y otras estructuras de los lenguajes de programación.

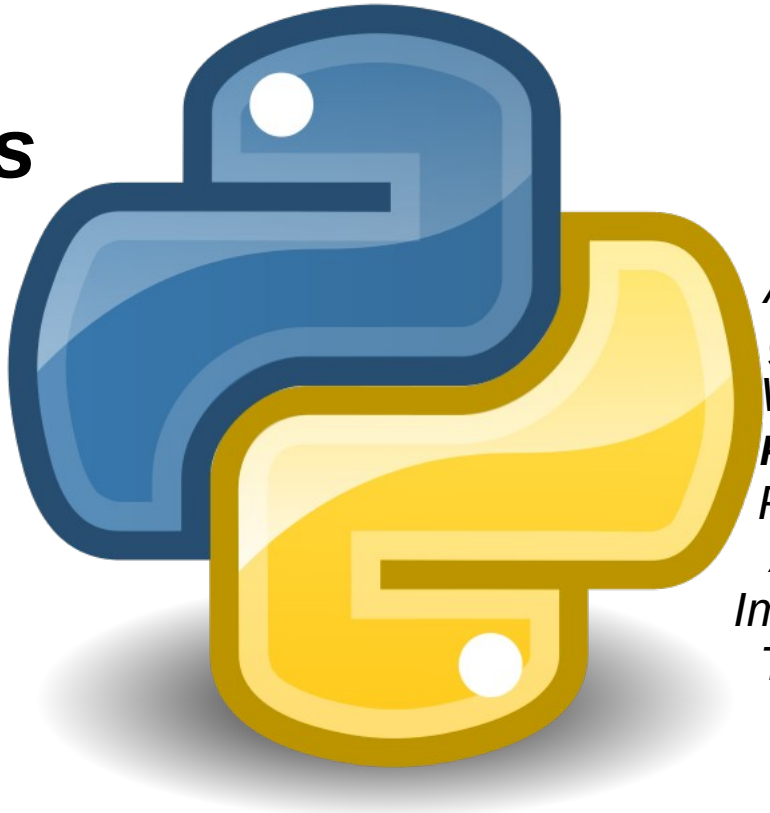
En Python se escriben como secuencias de caracteres alfanuméricos.

Un identificador en Python puede tener cualquier longitud que sea positiva.

PYTHON

Identificadores

Un identificar valido debe cumplir con la condición adicional de que no pertenezca a las palabras reservadas para el lenguaje, a continuación se listan las palabras reservadas del lenguaje Python:



And, continue, except
global, lambda, pass
With, as, def, if, del, is
***Finally, nonlocal, for,** in*
Raise, while, break, or
Assert, elif, False, try
Import, None, not, return
True, yield, class, else
From

PYTHON

Identificadores - Ejemplos

x
y
num1
suma
Promedio
Edad
paisDeNacimiento

_nombre
area_circulo
PI
PAIS_NACIMIENTO
valorMaximo

Tener en cuenta que Python es sensible a mayúsculas y minúsculas

PYTHON

Variables

Una variable es un espacio en memoria donde se almacena un dato, un espacio donde se almacena la información necesaria para realizar las acciones que ejecutan los programas.

Fuertemente Tipado

No se puede sumar datos enteros
Con datos de tipo String

Tipado Dinámico

En Python las variables se pueden declarar asignándoles un valor, no es necesario declararlas previamente, tampoco es necesario determinar el tipo de dato.

nombre = “Pedro”
num1 = 12
altura = 1,72

PYTHON

Tipos de datos primitivos

Numéricos: enteros (int)
reales (float)

Booleanos: true o false

Caracteres y cadena de caracteres
string (str)

nombre = “Luis”

variable1 = "La casa blanca"

variable2 = """Ahora tengo texto multilinea esto significa que pude escribir texto en varias lineas"""

edad = 23
peso = 12.6
mayor_edad = true

PYTHON

Operadores

Los operadores son símbolos que le indican al intérprete que realice una operación específica, como aritmética, comparación, lógica, etc.

Tipos de operadores en Python:

Operadores aritméticos
Operadores relacionales
Operadores de asignación
Operadores lógicos
Operadores de pertenencia
Operadores de identidad

PYTHON

Operadores aritméticos

Un operador aritmético toma dos operandos como entrada, realiza un cálculo y devuelve el resultado.

+	Suma
-	Resta
*****	Multiplicación
/	División
%	Módulo
******	Potencia
//	División entera

PYTHON

Operadores Relacionales

Un operador relacional se emplea para comparar y establecer la relación entre los operandos. Devuelve un valor booleano (true o false) basado en la condición.

>	Mayor que
<	Menor que
==	Igualdad
>=	Mayor o igual que
<=	Menor o igual que
!=	Diferencia

PYTHON

Operadores Lógicos

Se utiliza un operador lógico para tomar una decisión basada en múltiples condiciones. Los operadores lógicos utilizados en Python son and, or y not.

and	Devuelve True sin ambos son verdaderos
or	Devuelve True si alguno de los operandos es verdadero
not	Devuelve True si alguno de los operandos False

PYTHON

Operadores de Asignación

Se utilizan para asignar valores a una variable. generalmente se combina con otros operadores donde la operación se realiza en los operandos y el resultado se asigna al operando izquierdo.

=	//=
+=	&=
-=	|=
*=	^=
/=	«=
%=	»=
**=	

PYTHON

Operadores de Pertenencia

Se emplean para identificar pertenencia en alguna secuencia (listas, strings, tuplas).

in	not in
Devuelve True si el valor especificado se encuentra en la secuencia. En caso contrario devuelve False.	Devuelve True si el valor especificado no se encuentra en la secuencia. En caso contrario devuelve False.

PYTHON

Operadores de Identidad

Se usan para comprobar si dos variables emplean la misma ubicación en memoria..

is	is not
Devuelve True si los operandos se refieren al mismo objeto. En caso contrario devuelve False.	Devuelve True si los operandos no se refieren al mismo objeto. En caso contrario devuelve False.

PYTHON

Concatenar

En Python existen varias formas de concatenar dos o más objetos:

1) Empleando el signo +
2) Usando la coma (,)
3)Empleando la f y las { }

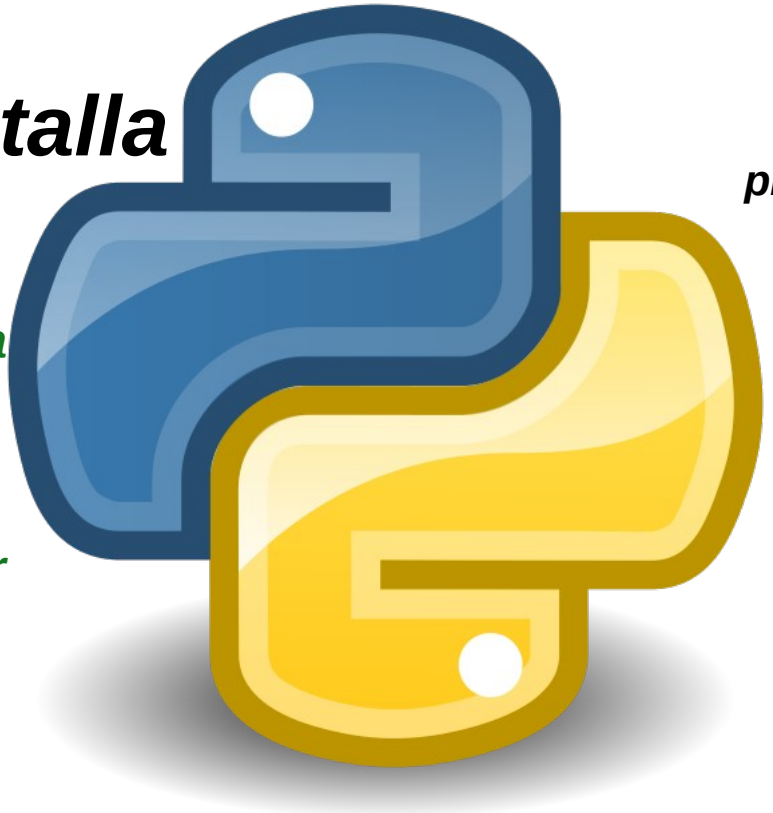
```
nombre = "Andres"
apellidos = "Sarmiento"

print ("El nombre completo es: " + str(nombre) + " " + str(apellidos) )

print ("El nombre completo es: ", nombre, " ", apellidos)

print (f"El nombre completo es: {nombre} {apellidos} ")
```

PYTHON

Imprimir en pantalla

Para imprimir en pantalla en Python vasta con usar la función
print()

Con print podemos mostrar en pantalla texto el cual va como argumento de la función y debe ir entre comillas.



print("Esta es una línea de texto")
print('Viernes')

También se puede imprimir el valor de una variable.

print(nombre)
print(suma)

PYTHON

Imprimir en pantalla

Python añade un salto de línea al final de cada print. Si quisiéramos evitar eso lo que se debe hacer es usar el argumento end

```
print("Hola", end" ")
print("Python")
```

```
Hola Python
```

PYTHON

Entrada por teclado

Para obtener un dato ingresado por teclado se debe usar la función input()

```
print("¿Cuál es tu nombre? ")
nombre = input()
```

```
print("¿Cuál es tu nombre? ",  end" ")
```

Cuando el programa llega a la función input se detiene esperando recibir un dato y que se pulse la tecla enter.

Podemos usar el argumento end para que el dato ingresado por el usuario se escriba en la misma línea y no en la proxima.

PYTHON

Entrada por teclado

A la función input() se le puede enviar un argumento que se escribe en la pantalla de esta forma también evitaríamos el salto de línea

nombre = input(“¿Cuál es tu nombre? ”)

De forma predeterminada, la función input() convierte la entrada en una cadena, aunque escribamos un número. Si intentamos hacer operaciones, se producirá un error.

Si se quiere que Python interprete la entrada como un número entero, se debe utilizar la función int() de la siguiente manera:

sueldo = int(input(“Ingrese un valor: ”))