



PARADIGMAS DE PROGRAMACIÓN

REGLAS Y ENTORNOS

LENGUAJE C VS. PYTHON

INTEGRANTES:

Fernando Haro Calvo
Andrea Rivas Gomez
Teresa Rivas Gomez
Danna Gpe. Sandez Islas
Angel Daniel Solano Meza

INTRODUCCIÓN

En la siguiente presentación se hablaban sobre las reglas de nombramiento y los entornos de C y Python, así como se realizara una comparación entre ambos.



NOMBRES

NOMBRES

Los nombres son la forma más fundamental de abstracción, proveen un mecanismo para referirse a cualquier cosa; desde simples valores, hasta complejos conjuntos de datos y comportamientos en programación orientada a objetos, a librerías enteras en la forma de módulos.



REGLAS DE NOMBRES

REGLAS DE NOMBRES EN C

Las reglas de nombre en el lenguaje C:

- 1.Una variable, siempre debe iniciar con una letra (mayúscula o minúscula) ó un guión bajo (_).
- 2.Una variable, puede contener números, solamente después de la primer letra (siguiendo la regla anterior).
- 3.No es permitido dejar un espacio en blanco a lo largo de la variable.
- 4.Lo recomendable es que sea una variable corta (regularmente entre 20 y 30 caracteres como máximo).
- 5.No puedes utilizar palabras reservadas para la declaración de una variable.
- 6.El nombre de una variable es case sensitive (es decir, a lo largo de tu programa debe escribirse exactamente igual). Todos los nombres de variables son sensibles a mayúsculas y minúsculas.
- 7.Utiliza un nombre que exprese algo del contexto en el cual la estás declarando.



REGLAS DE NOMBRES EN PYTHON

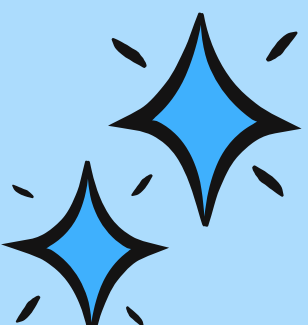
```
break  
class  
continue  
def  
del  
elif  
else  
except  
finally
```

Sólo pueden contener los siguientes caracteres:

- Letras minúsculas.
- Letras mayúsculas.
- Dígitos.
- Guiones bajos (_).

Deben empezar con una letra o un guión bajo, nunca con un dígito.

No pueden ser una palabra reservada del lenguaje («keywords»).



C

El tipo de datos de una variable se debe especificar claramente, lo que proporciona una mayor seguridad en cuanto a la manipulación de datos

```
int numero;  
float precio;  
char letra;
```

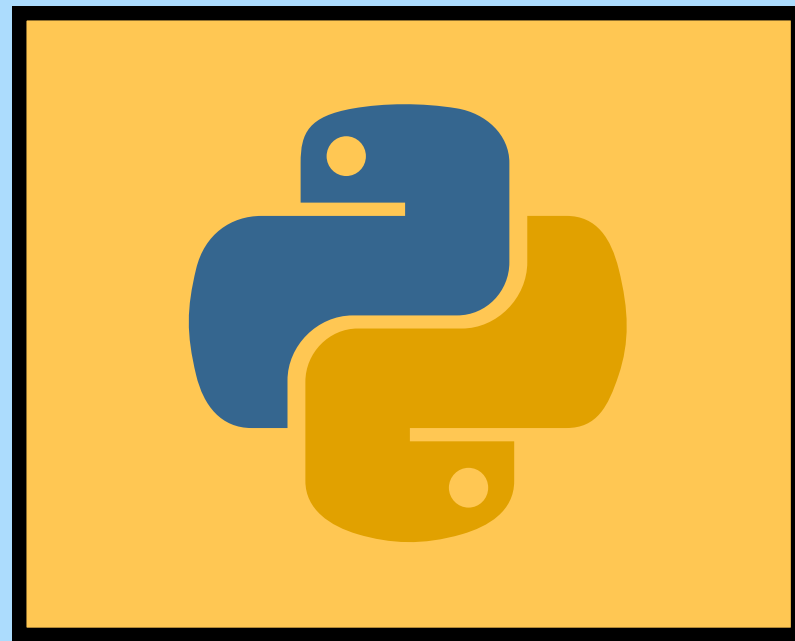
PYTHON

El tipo de datos de una variable se determina automáticamente en tiempo de ejecución según el valor que se le asigne.

```
numero = 10  
precio = 25.5  
letra = 'a'
```




AMBITOS



EN PYTHON

- Ambito de funcion local
- Ambito a nivel modulo
- Ambito incorporado

EN C

- Locales
- Globales
- Externas



Built-in namespace

Módulo: espacio de nombres global

Función/Clase: espacio
de nombres local



ENTORNOS

ENTORNOS

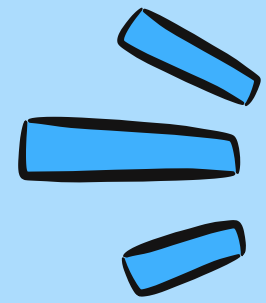
El conjunto colectivo de marcos activos en un programa se le llama entorno. Para buscar el valor de una variable se buscan en los entornos más cercanos, en este caso sería primero local, no local y luego global.



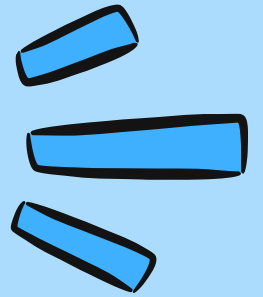
ENTORNO EN C

Los entornos en C son estáticos, lo que significa que se definen en el momento de la compilación y no se pueden cambiar durante la ejecución del programa. Los entornos en C también son locales, lo que significa que son únicos para cada función y no se pueden compartir entre funciones.





ENTORNO EN PYTHON



Los entornos en Python son dinámicos, lo que significa que se definen en el momento de la ejecución del programa y pueden cambiar durante la ejecución. Los entornos en Python también son globales, lo que significa que se pueden compartir entre funciones y módulos.

LENGUAJE C

```
1  #include <stdio.h>
2
3  // Variable global
4  int y = 0;
5
6  void foo()
7  {
8      // Variable local a foo
9      int x = 0;
10     x++;
11     y++;
12
13     printf("Dentro de foo: x = %d\n", x);
14     printf("Dentro de foo: y = %d\n", y);
15 }
16
17 int main()
18 {
19     // Variable local a main
20     int x = 0;
21
22     foo();
23
24     y++;
25
26     printf("\nEn main: x = %d\n", x);
27     printf("En main: y = %d\n", y);
28
29     return 0;
30 }
```

Dentro de foo: x = 1
Dentro de foo: y = 1

En main: x = 0
En main: y = 2

PYTHON

```
1  def foo():  
2      x = 10  
3  
4      def bar():  
5          print("Valor de x dentro de bar:", x)  
6  
7          bar()  
8  
9  foo()
```

Valor de x dentro de bar: 10

REFERENCIAS

- 1.6. Expresiones. (s. f.). Python documentation. Recuperado 28 de febrero de 2024, de <https://docs.python.org/3/reference/expressions.html>
- 2.4. Modelo de ejecución. (s. f.). Python documentation. Recuperado 28 de febrero de 2024, de <https://docs.python.org/3/reference/executionmodel.html>
3. Datos. (s. f.). Aprende Python. Recuperado 28 de febrero de 2024, de <https://aprendepython.es/core/datatypes/data/>
4. Aprende Python (26 febrero 2024). Sergio Delgado Quintero. Aprendepython.es. Recuperado el 29 de febrero de 2024, de https://aprendepython.es/_downloads/907b5202c1466977a8d6bd3a2641453f/aprendepython.pdf
5. Ámbito de las variables — Fundamentos de Programación en C++. (s. f.). Uva.es. Recuperado 28 de febrero de 2024, de https://www2.eii.uva.es/fund_inf/cpp/temas/2_tipos_variables/ambito_variables.html (ambito en C)
6. Reglas de nombres en C y significado de nombres:
7. Pérez, H. (2019, octubre 27). ▷ Reglas para Nombrar 【 Variables en Programación 】 . El Camino Dev. <https://elcamino.dev/reglas-para-nombrar-variables-en-programacion/>
8. Rivera, D. (2022, julio 30). Declaración de Variables en C. Pleets.org. <https://blog.pleets.org/article/es/declaraci%C3%B3n-de-variables-en-c>