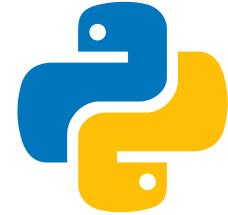


Conceptes bàsics sobre Phython



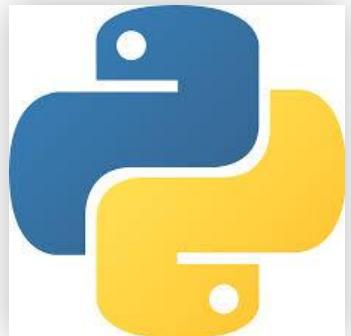
1. Sobre Phyton
2. Programació en Phyton
3. Més info sobre Phyton

Sobre Phyton

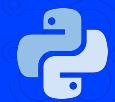


Introducció

- És un llenguatge de programació creat per Guido van Rossum l'any 1991 durant les seves vacances de Nadal 89-90
- OpenSource
- Llenguatge d'alt nivell
- Interpretat, no compilat
- Tipatge de dades dinàmic i fort
- Orientat a Objectes OOP
- Propòsit general i multiplataforma
- Utilitzat per Google, Yahoo, la NASA, etc



1. Sobre Phyton



TIOBE

Sep 2025	Sep 2024	Change	Programming Language	Ratings	Change
1	1		Python	25.98%	+5.81%
2	2		C++	8.80%	-1.94%
3	4		C	8.65%	-0.24%
4	3		Java	8.35%	-1.09%
5	5		C#	6.38%	+0.30%
6	6		JavaScript	3.22%	-0.70%
7	7		Visual Basic	2.84%	+0.14%
8	8		Go	2.32%	-0.03%
9	11		Delphi/Object Pascal	2.26%	+0.49%
10	27		Perl	2.03%	+1.33%

1. Sobre Phyton



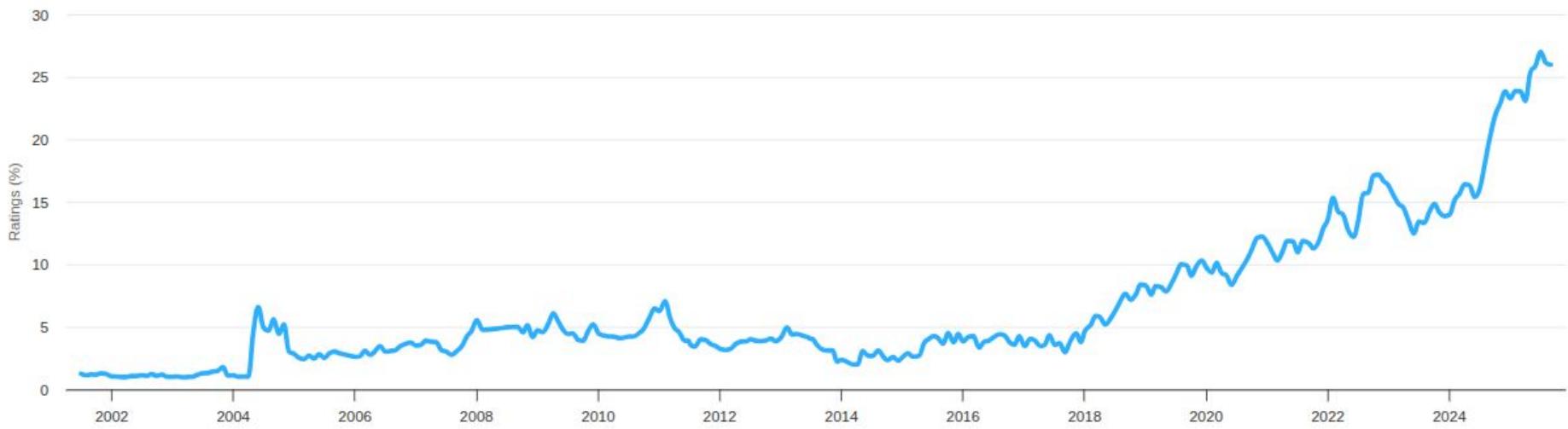
⬆️ Highest Position (since 2001): #1 in Sep 2025

⬇️ Lowest Position (since 2001): #13 in Feb 2003

⭐️ Language of the Year: 2007, 2010, 2018, 2020, 2021, 2024

TIOBE Index for Python

Source: www.tiobe.com



Programació Phyton



Consideracions generals

- La convenció de *naming* de Python ens mana escriure els noms de fitxers en minúscules:

```
hola.py  calculadora.py
```

- Les crides a funcions també:

```
print()  input()
```

- Les variables les escriurem seguint el patró *snake_case*:

```
alumne    alumnes_de_dam2
```



Consideracions generals

- Una bona identació (o tabulació) és obligada:

```
if nom == 'Mateu':  
    print('Hola ', nom)  
    input()
```

- Posarem espais entre enumerats:

```
print('RAM ', quant_ram, 'HDD ', capacitat_hdd)
```

- Hem d'aplicar identació horitzontal també:

```
8 + 2           total += 6
```



Consideracions generals

- No es fa ús de cap caràcter especial per delimitar el codi:
 - No usarem claudàtos {} per delimitar els blocs
 - No usarem punt i coma pel final de sentència ;
- Per indicar els blocs de codi **usarem una tabulació rigurosa.**
- Els comentaris in-line els iniciem amb hashtag #
- Els blocs de comentari, s'enclausen amb tres cometes dobles """



Operadors aritmètics

Operació	Operador	Exemple
Suma	+	$2 + 4 = 6$
Resta	-	$4 - 2 = 2$
Multiplicació	*	$3 * 3 = 9$
Potència	**	$3 ** 3 = 27$
Divisió	/	$50 / 6 = 8.3$
Quocient	//	$50 // 6 = 8$
Resta (mòdul)	%	$50 \% 6 = 2$



Operadors relacionals i lògics

Operació	Operador	Exemple
Igualtat	<code>==</code>	<code>a == b</code>
Diferència	<code>!=</code>	<code>a != b</code>
Major que	<code>></code>	<code>a > b</code>
Major o igual	<code>>=</code>	<code>a >= b</code>
Menor	<code><</code>	<code>a < b</code>
Menor o igual	<code><=</code>	<code>a <= b</code>
Lògics (i, o, no)	And or not	<code>a =3 and b<4 or c == 5 and not (a ==4)</code>



Tipus de dades

Clase	Tipo	Ejemplo
Int	Numèric	42
Long	Numèric	456966786151987643L
Float	Numèric	3.14
Str	Caràcters	'Hola món'
Bool	Booleà	True False
List	Seqüència	["pera", "poma", "plàtan"]
Tuple	Seqüència	("pera", "poma", "plàtan")
Range	Seqüència	Range(6)



Les variables

- No es declara el seu tipatge prèviament
- La funció **type** ens retorna el tipus de dades.

```
>>> c = "Hola Mundo" # cadenes de caràcters
>>> type(c) # comprovar tipus de dades
<type 'str'>
>>> e = 23 # número enter
>>> type(e) # comprovar tipus de dades
<type 'int'>
```



Estructures condicional: if

Syntaxi:

```
if(condició):  
    #accions  
elif(condició):  
    #accions  
else:  
    #accions
```

```
if(dia == "Dilluns"):  
    print("Dilluns un altre cop nooo!!")  
elif(dia == "Dimarts"):  
    print("Els dimarts són pesats")  
elif(dia == "Dimecres"):  
    print("Ja estem a la meitat de la setmana")  
elif(dia == "Dijous"):  
    print("El Dijous ja té bona pinta")  
elif(dia == "Divendres"):  
    print("Per fi divendres :)")  
elif(dia == "Dissabte"):  
    print("We are happy now")  
elif(dia == "Diumenge"):  
    print("Bajona, demà és dilluns")  
else:  
    print("Dia invàlid")
```



Estructura iterativa: **while**

Sintaxi:

while condició:
#accions

```
contador = 1

while contador <= 10:
    print("Pas: ", str(contador))
    contador += 1
```



Estructura iterativa: **for**

Sintaxi:

for variable in Llista:
 #accions

```
# tupla
ciclistes = ('T. Pogaçar', 'J. Vingegaard', 'S. Kuss')
# llista
futbolistes = ['Alexia P.', 'Pedri', 'Graham Hansen',
'Lewandowski']

for nom in ciclistes:
    print(nom)

for futbolista in futbolistes:
    print(futbolista)

# valor prenerà els valors enters de l'1 a 10.
for valor in range(1, 10):
    print(valor)
```



Estructura de Classes/Objectes

```
class Person:  
    def __init__(self, name, age): # Constructor  
        self.name = name # self és equivalent a this de java fa referència a la pròpia  
        classe  
        self.age = age  
  
    def myfunc(self):  
        print("Hello my name is " + self.name)  
  
p1 = Person("John", 36)  
p1.myfunc()
```

Més informació Phyton

3. Més informació Phyton



- [Tutorial w3schools](#)
- [Tutorial Python3](#)
- [Guía d'estil programació Python](#)
- IDE per a desenvolupar en Python: **Visual Studio Code o PyCharm**



"El desembre de 1989, estava buscant un projecte de programació com a hobby que em mantingués ocupat durant les setmanes de Nadal. La meva oficina estaria tancada i només tindria el meu ordinador de casa a mà."

Raimon Izard

