

python

PYTHON

פייתון היא שפת תכנות ברמה גבוהה, שפורסמה לראשונה בשנת 1991. היא ידועה בפשטות ובקריאות שלה. פייתון שמה דגש על קריאות קוד וסינטקס נקי, מה שמאפשר למתכנתים לכתוב פחות שורות קוד בהשוואה לשפות תכנות אחרות.

הקוד של פייתון מתבצע שורה אחר שורה בעזרת מתורגמן, בלי צורך בתהליך קומפילציה (הפיכת הקוד לשפת מחשב). זה מאפשר לבדוק ולתקן קוד במהירות.

כדי להריץ קוד פייתון, צריך להתקין מתורגמן פייתון במחשב. פייתון מציעה גם ממשק אינטראקטיבי שבו אפשר להריץ קוד באופן מיידי וגם אפשרות להריץ סקריפטים של פייתון שנשמרו כקבצים עם הסיומת `py..`

Installation

איך להתקין מתורגמן פייתון ו-IDE סביבת פיתוח משולבת

מתורגמן פייתון הוא תוכנה שנועדה לפרש את הקוד של פייתון. פייתון עצמה היא רק שפה, ולכן יש צורך בתוכנה שתבין את הקוד ותפיק את הפלט.

IDE סביבת פיתוח משולבת: עורך טקסט שבו כותבים את קוד הפייתון. יש לו תכונות שעוזרות בתהליך הכתיבה.

PyCharm היא סביבת פיתוח IDE שתוכננה במיוחד עבור פייתון.

Installation - Python interpreter

[Python.org](https://python.org)

Downloads → View the full list of downloads
We will download one version before the last to avoid eventually problems from the last version.
Release version : 3.10.10 → Download
Files → Mac, Linux or Windows

Launch the .exe file
Check the "add python.exe to PATH" button
→ Install Now
Test in the command prompt "py -3.10" >>> 2+2

```
Microsoft Windows [version 10.0.22000.1936]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\>py -3.10
Python 3.10.10 (tags/v3.10.10:aad5f6a, Feb 7 2023, 17:20:36) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>>
```

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.10.11	April 5, 2023	Download	Release Notes
Python 3.11.3	April 5, 2023	Download	Release Notes
Python 3.10.10	Feb. 8, 2023	Download	Release Notes
Python 3.11.2	Feb. 8, 2023	Download	Release Notes
Python 3.11.1	Dec. 6, 2022	Download	Release Notes
Python 3.10.9	Dec. 6, 2022	Download	Release Notes
Python 3.9.16	Dec. 6, 2022	Download	Release Notes

[View older releases](#)

Files

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later
Windows embeddable package (32-bit)	Windows	
Windows embeddable package (64-bit)	Windows	
Windows help file	Windows	
Windows installer (32-bit)	Windows	
Windows installer (64-bit)	Windows	Recommended

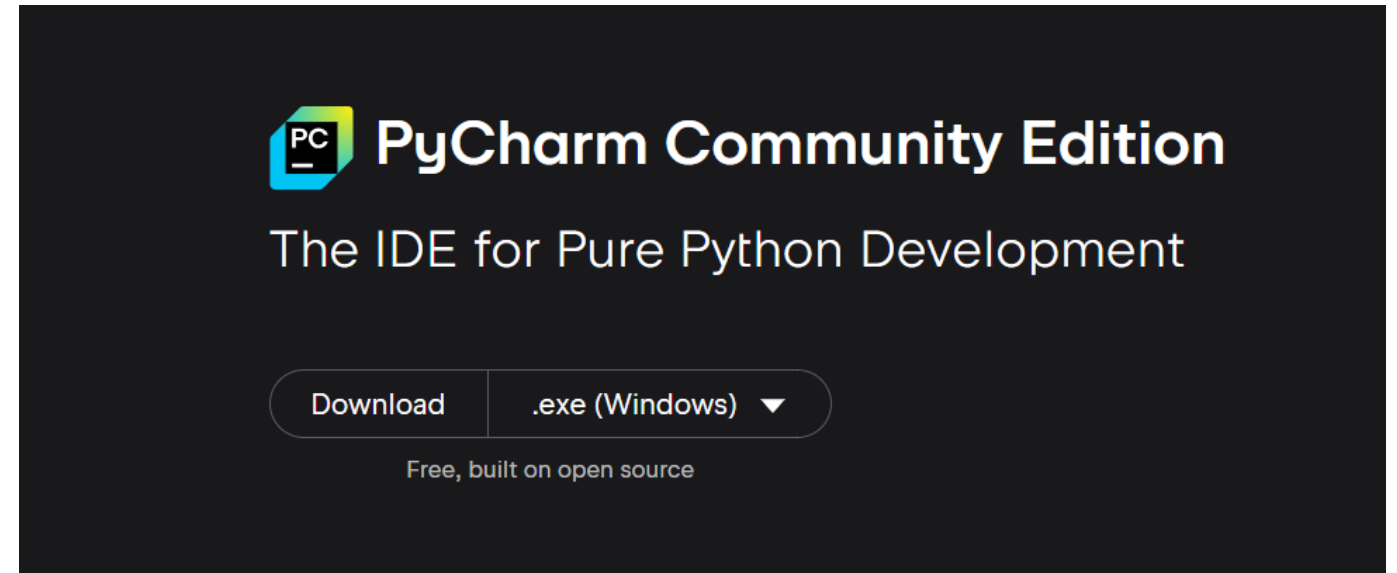
Installation - PyCharm

<https://www.jetbrains.com/pycharm/>

Downloads → **Community** version
Mac, Linux or Windows

Launch the .exe file
Check all the options

Launch PyCharm
Do not import settings



Variable Declaration

בפייתון, אין צורך להגדיר משתנה מראש לפני שמתמשים בו:
 $a = 7.4039$

שמות של משתנים חייבים להתחיל באות או בקו תחתון.
שמות יכולים להכיל אותיות, מספרים ותו קו תחתון.
אסור שיהיו רווחים בשמות המשתנים.

שמות משתנים רגישים לאותיות גדולות וקטנות: שלושת השמות האלה שונים זה מזה:
`name_1` , `Name_1` , `NAME_1`
אסור להשתמש במילות מפתח של פייתון לשמות משתנים (כמו `for`, `if`, `def`, `True` וכו').
כמו כן, כדאי להימנע משימוש בשמות של פונקציות מובנות.

הדרך המומלצת לקרוא למשתנים היא להשתמש באותיות קטנות בלבד ולהפריד בין מילים בקו תחתון.

String Manipulation

מחרוזת בפייתון היא רשימה של תווים בסדר מסוים. תו הוא כל דבר שניתן להקליד במכה אחת במקלדת, כמו אות, מספר, או סלאש. מחרוזות יכולות גם לכלול רווחים, טאבים ותווים של שורה חדשה. לדוגמה,

```
myStr="hello world"
```

כאן, הקצנו את המחרוזת "hello world" למשתנה בשם myStr. ניתן גם להגדיר מחרוזת ריקה, שאין בה תווים, כך:

```
myStr=""
```

כדי ליצור מחרוזת עם תווים מסוימים, ניתן להקצות את התווים למשתנה לאחר שסוגרים אותם במרכאות כפולות או מרכאות יחידות, כפי שמוצג כאן:

```
word = "Hello World"  
print("The word is:",word)
```

String Manipulation

גישה לתווים במחרוזת בפייתון

כדי לגשת לתווים במחרוזת, ניתן להשתמש באופרטור המיקום של פייתון [], כלומר סוגריים מרובעים, כדי לגשת לתווים במחרוזת כפי שמוצג כאן. המיקום מתחיל מ-0.

```
word = "Hello World"  
print("The word is:",word)  
letter=word[0]  
print("The letter is:",letter)
```

Output:

The word is: Hello World

The letter is: H

String Manipulation

למצוא את האורך של מחרוזת בפייתון

כדי למצוא את האורך של מחרוזת בפייתון, ניתן להשתמש בפונקציה `len()`. הפונקציה `len()` מקבלת מחרוזת כארגומנט ומחזירה את האורך של המחרוזת, כפי שמוצג כאן:

```
word = "Hello World"  
print("The string is:",word)  
length=len(word)  
print("The length of the string is:",length)
```

Output:

The string is: Hello World

The length of the string is: 11

String Manipulation

We can add strings together

```
firstname = "Bob"  
lastname = "Lebob"  
print(firstname + " " + lastname)
```

Sub String

```
initial1 = "Donald"  
initial2 = "Duck"  
print(initial1[0] + initial2[0])
```

Format

```
age = 44  
name = "bob"  
occupation = "plumber"
```

```
print(f"Hi my name is {name} and I  
am {str(age)} years old and my  
occupation is {occupation}")
```

Python Data Types

משתנים בפייתון יכולים להיות מסוגים שונים:

- Integer: int
- Floating-point number: float
- Alpha-numeric string: str

פייתון תומכת בסוגי נתונים רבים, מספריים ולא-מספריים. לדוגמה:

Numeric types

- int
- float
- complex

Non-numeric types

- str
- list
- tuple
- set
- dict
- bool

Python String Data Type

מחרוזת היא רצף של תווים שמיוצג על ידי מרכאות יחידות או כפולות.

```
name = 'Python'  
print(name)  
message = 'Python for beginners'  
print(message)
```

Output

```
Python  
Python for beginners
```

Python Numeric Data Type

בפייתון, סוג נתונים מספרי משמש לאחסון ערכים מספריים.

מספרים שלמים integers מספרים עשרוניים floating-point ומספרים מרוכבים complex נכללים בקטגוריית המספרים של פייתון. הם מוגדרים באמצעות המחלקות int, float, ו-complex בפייתון.

- int מחזיק מספרים שלמים חיוביים או שליליים, ללא הגבלת אורך.
- float מחזיק מספרים עשרוניים ודיוקו עד 15 מקומות אחרי הנקודה העشرונית.
- complex מחזיק מספרים מרוכבים.

ניתן להשתמש בפונקציה type() כדי לדעת לאיזו מחלקה (סוג) משתנה או ערך שייך.

Python Numeric Data Type

Example :

```
x = int('100')  
print(x)
```

```
f=float('5.5')  
print(f)
```

```
s=str('1')  
print(s)
```

```
s = '1' + '5'  
print(s)
```

Output :

100

5.5

"1"

15

Python Numeric Data Type

Example :

```
num1 = 5  
print(num1, 'is of type', type(num1))
```

```
num2 = 2.0  
print(num2, 'is of type', type(num2))
```

```
num3 = 1+2j  
print(num3, 'is of type', type(num3))
```

Output :

5 is of type <class 'int'>

2.0 is of type <class 'float'>

(1+2j) is of type <class 'complex'>

The Python Boolean Type

סוג הנתונים הבוליאני Boolean בפייתון יכול להכיל רק שני ערכים אפשריים: True או False. אין ערכים אחרים שיהיו מסוג bool.

בדרך כלל, הוא משמש לייצוג ערכי אמת של ביטויים. לדוגמה, הביטוי $1==1$ הוא True, בעוד שהביטוי $1>2$ הוא False.

Boolean Operators

פעולות בוליאניות הן חישובים פשוטים של ערכי True ו-False. ניתן לבצע חישובים אלו באמצעות אופרטורים בוליאניים, הכוללים AND, OR, NOT. הנה כמה מהפעולות הבוליאניות הנפוצות:

or

and

not

== (equivalent)

!= (not equivalent)

The Python None Type

המילה השמורה None משמשת להגדרת ערך ריק, או חוסר ערך בכלל. היא יכולה לעזור לנו להימנע מ-SyntaxError.

None אינו זהה ל-0, False, או מחרוזת ריקה. None הוא סוג נתונים בפני עצמו `NoneType` ורק None יכול להיות None.

```
Total_score = None  
print(total_score)
```

```
x = None  
print(x)  # ידפיס None  
print(type(x))  # ידפיס <class 'NoneType'>
```

Input fonction

פונקציית `input()` בפייתון משמשת לקבלת קלט מהמשתמש. כברירת מחדל, היא מחזירה את הקלט של המשתמש בצורת מחרוזת.

אם ברצוננו לקבל קלט כערך מספרי `int` או `float` פשוט צריך לבצע המרה `typecasting`

דוגמה לקבלת קלט כערך מספרי:

Taking input as string

```
color = input("What color is rose?: ")  
print(color)
```

Taking input as int

```
n = int(input("How many roses?: "))  
print(n)
```

Taking input as float

```
price = float(input("Price of each rose?: "))  
print(price)
```

Exercises

Exercises

Exercise 1 :

Write a program that asks the user for his first name, his second name and his last name.

Then print :

"Your initials are _ _ _ "

Exercise 2 :

Write a program that asks the user to enter two numbers a and b and display their sum: $a + b$

Exercise 3 :

Write a program that asks the user to enter his name and display his name with "ish" at the end

Exercises

Exercise 4 : Extra

Create a program to calculate the area and circumference of a circle.
Ask the user for the radius (r).

$$\text{Area} = 3.14 * r^2$$

$$\text{Circumference} = 2 * 3.14 * r$$

Bonus : around the number (with round)

Q&A