

תכנות מונחה עצמים

Python





PYTHON

פייתון היא שפת תכנות דינמית ופשוטה, הנמצאת בשימוש נרחב בתחום התכנות. השפה נוצרה בשנת 1991 והתפתחה מאז בהתבסס על עקרונות תכנות פשוטים וקריאים. המטרה העיקרית של פייתון היא להקל על התכנות ולעשות את הקוד קריא ופשוט להבנה. בעבור מתכנתים, זה יוצר מהירות בכתיבת הקוד ומפשט את התהליך

אנו נכתוב בפייתון בתוכנת PyCharm, עם הורדה והתקנה פשוטה

סרטון הדרכה להורדה

מטרות עיקריות של PYTHON

- קריאה וקלה להבנה: פייתון מתאפיינת בקוד שקריא ופשוט להבנה.
- דינמיות: פייתון היא שפת תכנות דינמית, שמאשרת התעלמות מטיפוסים במהלך הכתיבה ומאפשרת החלפת טיפוסים במהלך הריצה, מה שמאפשר כתיבת קוד גמיש ומהיר יותר.
- מגוון פלטפורמות: פייתון זמינה על מגוון רחב של מערכות הפעלה ופלטפורמות ויש לה מערכת ניהול חבילות עוצמתית (pip) המאפשרת התקנת וניהול ספריות חיצוניות בקלות.
- מגוון רחב של שימושים: פייתון מתאימה למגוון רחב של יישומים, כולל פיתוח אפליקציות ואתרי אינטרנט, תכנות מדעי, פיתוח משחקים, תכנות מערכות, תוכנות תחזוקה, ועוד.
- קהילה רחבה ותמיכה עוצמתית: פייתון נמצאת בשימוש נרחב עם קהל משתמשים גדול, קהילה פעילה של מפתחים ומשתמשים שמספקים תמיכה, וכמו כן קיימים מגוון ספריות ותוספות תוכנה מוכנות (packages) שמסייעות בפיתוח.

PYTHON VS JAVA

- בדומה ל java גם פייתון תומכת בתכנות מונחה עצמים ומשתמשת במחלקות ואובייקטים כדי לארגן ולנהל את הקוד.
- בשתי השפות נוכל ליצור מחלקות (class) וליצור אובייקטים מהן, ולהשתמש בתכונות ובשיטות שנמצאות במחלקות בהתאם.
- פייתון יותר קצרה בפשטות ובירידה לפרטים בהגדרת המחלקות והאובייקטים, ומאפשרת להתמקד יותר במימוש המתודות והאלגוריתמים.

הבדלים חשובים בין פייתון לג'אווה		
קוד	פייתון	ג'אווה
קוד	פחות שורות קוד	יותר שורות קוד
Framework	בהשוואה לג'אווה, לפייתון יש פחות Frameworks, כלומר, הוא פחות מבוסס על חלקים שונים שנבנים אחד על השני (היררכיה) כדי ליצור מערכת אחת מגובשת. (כמו Collections בג'אווה).	הרבה יותר.
סינטקס	קל לזכור, מאוד דומה לשפה אנשוית.	מורכב, זורק שגיאה על כל עניין קטן בין אם פספסת ';' או '{'.
תכונות עיקריות	פחות שורת קוד, תכנות מהיר והקלדה דינמית.	ניהול זיכרון עצמי, חזק, בלתי תלוי בפלטפורמה אחרת.
מהירות	פייתון איטי יותר מכיוון שהוא משתמש במתורגמן (interpreter) וקובע גם את סוג הנתונים (הטיפוסים) ממש בזמן הריצה.	ג'אווה מהירה יותר בהשוואה לפיתון, יש לג'אווה קומפיילר לעומת פייתון שאין לו, הוא משתמש במתורגמן.
טיפוסים	מוקלד דינמי, כלומר, אנחנו לא מחוייבים לציין את הטיפוס לפני כל משתנה.	מוקלד באופן סטטי, כלומר אנחנו מחוייבים לציין את הטיפוס לפני כל משתנה.
הרצה	פייתון היא שפה ללא קומפיילר, יש לה Interpreter.	ג'אווה היא שפת קימפול וגם Interpreter ולכן היא מהירה יותר מפייתון.
בלוק	פייתון משתמש ב-Tab כדי להפריד קוד לבלוקים נפרדים	שפת ג'אווה משתמשת בסוגריים מתולתלים כדי להגדיר את ההתחלה והסוף של כל פונקציה והגדרת מחלקה
ממשקים וירשות	פייתון תומך בירשות בודדות ומרובות ולא תומך בממשקים (אין לו interface).	ללא ירשות מרובות, כלומר לא יכולה להרחיב יותר ממחלקה אחת.
ניידות ונוחות הרצה	תוכניות Python זקוקות להתקן של ה-interpreter במחשב כדי לתרגם קוד Python. פייתון פחות נייד.	תוכנית Java יכולה לפעול בכל מחשב או מכשיר נייד המסוגלים להריץ את המכונה הווירטואלית של של ג'אווה - JVM.

Why Python?

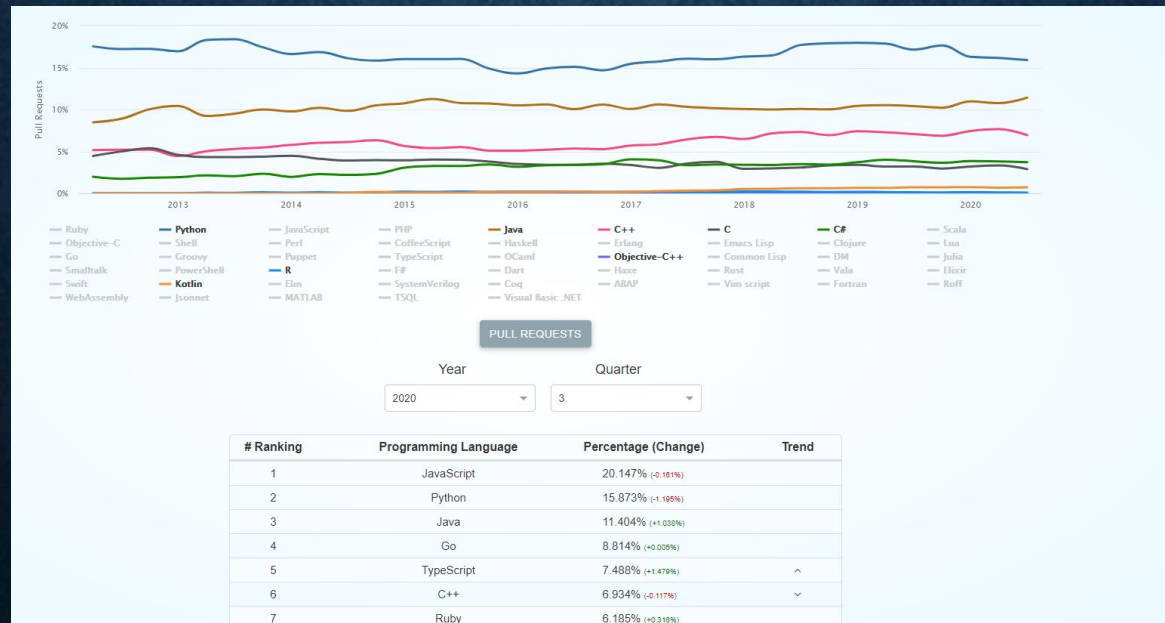
Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). •

Python has a simple syntax similar to the English language. •

Python has syntax that allows developers to write programs with fewer lines than some other programming languages. •

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. •

Python can be treated in a procedural way, an object-oriented way or a functional way. •



Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:  
    print("Five is greater than two!")
```

Try it Yourself »

Python will give you an error if you skip the indentation:

Example

Syntax Error:

```
if 5 > 2:  
print("Five is greater than two!")
```

Try it Yourself »