

פרק 1 : פתח דבר

החומר לפרקים הבאים נלקח מהאתר <https://python.swaroopch.com> . הספר נקרא **A Byte of Python** . ניתן להוריד קובץ pdf בקישור : <https://github.com/swaroopch/byte-of-python/releases/latest> להלן פרק 1 בתוכנית הלימודים שפת פייתון להנדסאים של משרד החינוך החל משנת תשפ"ב :

פרק 1 : פתח דבר

יעדים

היכרות ראשונית עם סביבת הפיתוח של Python.

תכנים

1. פייתון היא שפה עתירת יכולות המאפשרת כתיבה מהירה של תוכניות פשוטות וכן תוכניות מורכבות.
2. השפה אף תומכת בתכנות מונחה עצמים (Object-oriented programming).
3. השפה תומכת בהגדרת טיפוסים נתונים מופשטים (abstract data type) ותכנות צד שרת (server side).
4. התלמידים יכירו ויתרגלו תוכנית בסיסית הכוללת ביצוע קלט ופלט.
5. התלמידים יבינו את ההבדל בין מהדר (Compiler) לבין מפענח/מפרש (interpreter).
6. התלמיד יבין מהו ByteCode ומה ההבדל בינו לבין תהליך קומפילציה רגיל.

סיכום שעות ההוראה : 2 שעות עיוניות.

1.1 אודות פייתון

ההקדמה הרשמית לפייתון היא:

פייתון היא שפת תכנות קלה ורבת עוצמה. יש לה מבני נתונים יעילים ברמה גבוהה וגישה פשוטה אך יעילה לתכנות מונחה עצמים. התחביר האלגנטי וההקלדה הדינמית של פייתון, יחד עם אופיו המפורש, הופכים אותו לשפה אידיאלית לתסריטים ולפיתוח יישומים מהירים בתחומים רבים ברוב הפלטפורמות. נדון ברוב התכונות האלה בפירוט רב יותר בסעיף הבא.

1.1 ב. הסיפור מאחורי השם

גואידו ואן רוסום - Guido van Rossum , יוצר שפת הפייתון, קרא לשפה על שם תוכנית מפורסמת ב - BBC – תאגיד השידור הבריטי הציבורי - "Monty Python's Flying Circus", "הקרקס המעופף של מונטי פייתון". זוהי סדרת טלוויזיה ששודרה בערוץ ה BBC בשנים 1969–1974. התוכנית נכתבה על ידי קבוצת קומיקאים בריטים צעירים שנודעו מאז בשם מונטי פייתון ואין לה קשר לנחש הפייתון הענק.

1.g תכונות של פייתון

1.g.1 שפה פשוטה Simple

פייתון היא שפה פשוטה ומינימליסטית. קריאת תוכנית פייתון טובה מרגישה כמעט כמו לקרוא אנגלית (או עברית עבורנו). זה מאפשר להתרכז בפתרון לבעיה ולא בשפה עצמה.

2.g.1 קלה ללמידה Easy to learn

פייתון היא שפה שמאוד קל להתחיל איתה. לפייתון יש תחביר פשוט במיוחד.

3.g.1 קוד חופשי ופתוח Free and Open Source

פייתון הוא דוגמה ל- *FLOSS* (Free/Libre and Open Source Software) - חופשי/ליברה - חופשי בשפה הצרפתית - ותוכנת קוד פתוח). במילים פשוטות, ניתן להפיץ בחופשיות עותקים של תוכנה זו, לקרוא את קוד המקור שלה, לבצע שינויים בתוכנה ולהשתמש בחלקים ממנה בתוכניות חינוך חדשות. *FLOSS* מבוססת על הרעיון של קהילה החולקת ידע. זו אחת הסיבות לכך שפייתון היא כל כך טובה - היא נוצרה והיא משתפרת כל הזמן על ידי קהילה שרק רוצה לראות פייתון טובה יותר.

4.g.1 שפה ברמה גבוהה High-level Language

כאשר כותבים תוכניות ב Python, אף פעם לא צריך לטרוח על פרטים ברמה נמוכה כגון ניהול הזיכרון המשמש את התוכנית שלנו וכו'.

5.g.1 ניידת Portable

בשל אופייה בקוד פתוח, פייתון הועברה (כלומר שונתה כדי לגרום לה לעבוד עליה) פלטפורמות רבות. תוכניות Python יכולות לעבוד על כל אחת מהפלטפורמות האלה מבלי לדרוש שינויים כלשהם אם זהירים מספיק כדי למנוע תכונות תלויות מערכת.

אפשר להשתמש פייתון על גנו / לינוקס, Windows, FreeBSD, מקינטוש, OS/2, Solaris, אמיגה, AROS, OS / 390, z / OS, BeOS, AS/400, פאלם Psion, OS, QNX, VMS, בלוט VxWorks, RISC OS, פלייסטיישן, שארפ זארוס, Windows CE ו PocketPC!

אפשר אפילו להשתמש בפלטפורמה כמו Kivy כדי ליצור משחקים עבור iPhone, iPad ו-Android.

6.g.1 מפורש Interpreted

תוכנית שנכתבה בשפה מהודרת (מקומפלט) כמו C או ++C מומרת משפת המקור כלומר C או ++C לשפה שהמחשב מבין - שפת מכונה - (כלומר 0 ו-1) באמצעות מהדר עם דגלים ואפשרויות שונות. בעת הפעלת התוכנית, תוכנת המקשר/מטעין - Linker/Loader מעתיקה את התוכנית מדיסק קשיח לזיכרון ומתחילה להפעיל אותה.

פייתון, לעומת זאת, איננה זקוקה להידור (קומפילציה) בינארי. בפייתון מריצים את התוכנית ישירות מקוד המקור. באופן פנימי, Python ממירה את קוד המקור לטופס/צורת ביניים הנקראת `bytecodes` ולאחר מכן מתרגמת זאת לשפת הבסיסית/מכונה של המחשב ולאחר מכן מריצה אותה. כל זה עושה את השימוש בפייתון הרבה יותר קל כי לא צריך לדאוג להידור התוכנית, לוודא כי הספריות הנכונות מקושרות וטעונות וכו'. זה גם הופך את תוכנית הפייתון להרבה יותר ניידת שכן אפשר פשוט להעתיק את תוכנית הפייתון למחשב אחר וזה פשוט עובד!

7.ג.1 מונחה אובייקט Object Oriented

פייתון תומכת בתכנות מונחה פרוצדורת וכן בתכנות מונחה עצמים (OOP). בשפות מונחות פרוצדורות התוכנית בנויה סביב הליכים או פונקציות שאינן אלא חלקי תוכניות הניתנות לשימוש חוזר. בשפות מונחות אובייקטים, התוכנית בנויה סביב אובייקטים המשלבים נתונים ופונקציונליות. בפייתון יש דרך חזקה מאוד אבל פשטנית לעשות OOP, במיוחד בהשוואה לשפות גדולות כמו ++C או Java.

8.ג.1 מורחב Extensible

אם צריך קטע קוד קריטי כדי לרוץ מהר מאוד או רוצים שיהיה חלק כלשהו של אלגוריתם שלא להיות קוד פתוח ניתן לקודד את החלק הזה של התוכנית ב-C או ++C ולאחר מכן להשתמש בו מתוכנית Python שלך.

9.ג.1 ניתן לשיבוץ Embeddable

באפשרותך לשבץ את Python בתוכניות ++C/C כדי להעניק יכולות Scripting עבור משתמשי התוכנית.

10.ג.1 ספריות נרחבות - Extensive Libraries

הספרייה הסטנדרטית של פייתון היא ענקית. זה עוזר לעשות דברים שונים מעורבים כמו ביטויים רגילים, יצירת תיעוד, בדיקת יחידות, שרשור, מסדי נתונים, דפדפני אינטרנט, FTP, CGI, דואר אלקטרוני, XML, XML-RPC, HTML, קבצי WAV, קריפטוגרפיה, GUI (ממשקי משתמש גרפיים), ודברים אחרים תלויי מערכת. כל זה זמין תמיד בכל מקום שבו פייתון מותקן. קורים לזה הפילוסופיה של פייתון "Batteries Included" - "הסוללות כלולות" (הכוונה היא שקונים מכשיר המופעל עם סוללות הן מגיעות באותה החבילה ואין צורך לקנות סוללות בנפרד - ובהתאמה פייתון היא חבילה שלמה).

מלבד הספרייה הסטנדרטית, ישנן מספר ספריות איכותיות אחרות שתוכלו למצוא ב Python Package Index.

11.ג.1 סיכום

פייתון היא שפה מרגשת ורבת עוצמה. יש לה את השילוב הנכון של ביצועים ותכונות שהופכים תוכניות כתיבה בפייתון גם לכיף וגם לקלה.

ד.1 פייתון 3 לעומת 2

ניתן להתעלם מפרק זה אם לא מעוניינים בהבדל בין Python גרסה 2 לבין פייתון גרסה 3 . הפרקים כאן נכתבו עבור פייתון גרסה 3.

כדאי ורצוי ללמוד להשתמש בגרסה האחרונה של פייתון כי בה יש את כל האפשרויות המתקדמות האחרונות. החלק הקשה הוא ללמוד תכנות ולהבין את היסודות של שפת פייתון עצמה. זו המטרה שלנו בפרקי הספר הזה, וברגע שמשיגים את המטרה ניתן בקלות להשתמש בפייתון 2 או בפייתון 3 בהתאם לרצוננו.