

מג'אוה לפייתון חלק א'

מבנה של סקריפט פייתון-

בדומה לתוכניות של ג'אוה, לכל פרויקט יש כמה מסמכים או סקריפטים, ולכל מסמך יש את הסימנים שלו- בג'אוה: '.java' או '.py' בפייתון. לתוכניות בשפות סטטיות יש כמה קבצים וקובץ ראשי המכיל פונקציית main שרץ בתחילת התוכנית "ומנהל" אותה. בפייתון לעומת זאת המפרש מריץ סקריפטים החל מהמסמך הראשון ועד המסמך האחרון לפי סדר מסוים, ואין צורך בפונקציה ראשית שפועלת בתחילת התוכנית, אך על כל פנים, ניתן להגדיר פונקציה ספציפית כנקודת תחילת התוכנית, זה שימושי בעיקר כדי להבין כיצד התוכנית עובדת. הדרך הנהוגה להגדרת פונקציה ראשית היא ע"י `if __name__ == "__main__":`

```
def main():
    print("Hello World!")

if __name__ == "__main__":
    main()
```

סקריפט ו-module -

כדי להבין מה זה `__name__` נצטרך להבין תחילה איך להריץ קבצי פייתון- ישנם שתי דרכים (עיקריות) להורות לפייתון להריץ את קבצי הפייתון: פייתון היא שפה שמפעילה "מפרש" שעובר שורה אחר שורה של הקוד ומבצע אותה או שומר בזיכרון את הפעולה של אותה שורה.

עם התקנת השפה על המחשב אנחנו מתקינים גם מצב "אינטראקטיבי", שהוא תוכנית שרצה בזמן אמת, מחכה לפקודות בשפת פייתון ומבצעת אותם.

כדי להפעיל את המצב האינטראקטיבי, צריך רק להפעיל את הפקודה `python` או `python3` בטרמינל, והוא יקרה למצב הנ"ל.

נוכל לייבא למצב הקיים גם modules של פייתון שכתבנו מראש וכך להפעיל אותם מתוך התוכנית האינטראקטיבית ע"י הפקודה `import` ושם `module` או `path` שלו (במידה והטרמינל לא נמצא באותה תיקייה של ה-module).

דרך נוספת להריץ את הקוד היא כסקריפט, ואז המשתמש צריך להזין בטרמינל, בתיקייה של הקבצים את הפקודה: `python3 name_of_the_file.py` (הפקודה היא `python` או `python3` ואז שם הסקריפט בסימנים '.py').

סקריפט הוא קובץ טקסט של פייתון(קובץ '.py') שמכיל קוד פייתון שמטרתו לרוץ ישירות ע"י המשתמש. לעומתו module הוא קובץ טקסט (קובץ '.py') המכיל קוד פייתון ומטרתו להיות תוכנית שמיובאת לתוכניות פייתון אחרות, אז בעצם ההבדל בניהם הוא שהראשון נועד להרצה והשני נועד לייבוא לתוכנית רצה.

לפעמים נרצה להשתמש בסקריפט אחד בתוך סקריפט אחר כmodules.

בזכות המשתנה `__name__` נוכל לקבוע אם נרצה להריץ את הקובץ כסקריפט או לייבא אותו כmodules. כשאנחנו מריצים את הקובץ כסקריפט המשתנה `__name__` יהיה שווה למחרת `"__main__"` אבל כשמריצים את הקוד כmodules בתוך תוכנית אחרת, אז ערך המשתנה הוא שם הקובץ.

עכשיו מה שיקרא אם נריץ את הסקריפט הוא שהמפרש יבדוק מה ערכו של `__name__` ואם המשתנה יהיה שווה `__main__` הוא יבין שהקוד נקרא כסקריפט, ויריץ אותו החל מנקודה שאותה צוינה כנקודת ההתחלה, למשל למעלה קבענו שהתוכנית תתחיל מהפונקציה `main`.



אך אם המשתנה לא שווה ל- `__main__` המפרש יבין שזהו module בתוך תוכנית אחרת, ובמקרה כזה לא יהיה צורך להגדיר נקודת התחלה, כי התוכנית שמריצה את module תקבע באילו משתנים היא רוצה להשתמש מתוך הקובץ.

