# FLASK - אירגון התיקיות

לפני שניכנס לנושאים מתקדמים יותר, נסדר את הספרייה בצורה שתקל עלינו בהמשך, ולא נצטרך להכניס את כל הקוד לקובץ אחד.

ראשית, ניצור תיקייה חדשה שתכיל קובץ \_\_init\_\_.py , תזכורת קובץ \_\_init\_\_.py מגדיר תיקייה כ-namespace עבור כמה מודולים. לצורך הדוגמא נקרא לתיקייה flask\_example .

נעביר את החלק הראשון של הקוד (ייבוא הספריות) מסקריפט הראשי לסקריפט \_\_init\_\_ כלומר בנתיים הוא יראה כך :

from flask import Flask, render\_template

app = Flask(\_\_name\_\_)

נכניס את templates לתיקייה ,נוסיף סקריפט חדש עם השם routes.py , ונעביר את כל שקשור לניתוב (לא כולל הפונקציית main) מהסקריפט הראשי לסקריפט הזה:

users = [

    {'name': 'Joee Javany',

    'email': 'joo@example.com',

    'phone': '111-1111'},

    {'name': 'Tom Pythonovitch',

    'email': 'python\_is\_coool@example.com',

    'phone': '222-2222'},

]

@app.route('/')

def hello\_world():

    return render\_template('home.html' , users = users)

ונשנה את השם של המודול הראשי להיות run.py כדי שיהיה קל לזהות אותו.   
כדי שנוכל להשתמש במשתנה app צריך לייבא אותו מהתיקייה שיצרנו. בסוף הסקריפט run.py צריך להיראות ככה:

from flask\_example import app

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug = True)

נצטרך לייבא את הפונקציה render\_template ואת המשתנה app גם לסקריפט routes.py כדי שנוכל להשתמש בו:

from flask import render\_template

from flask\_example import app

...

ולבסוף נצטרך גם לייבא את routes לסקריפט \_\_init\_\_ אבל את זה נעשה אחרי שניצור את המשתנה app, כי אם נייבא אותו לפני תיזרק שגיאה, שכן הקוד מנסה להפעיל את routes ,אבל routes מסתמך על אובייקט שעדיין לא הוגדר (על app):

from flask import Flask, render\_template

app = Flask(\_\_name\_\_)

from flask\_example import routes

ונריץ את run.py כדי לוודא שהכל עובד כשורה.