

## פרסום האתר שלכם

האתר שבנינו סוף-סוף נראה טוב ומוכן לפירסום! איפה נפרסם אותו? אפשר לפרסם על שרת פרטי שלנו, או באתר ציבורי כלשהו. קודם-כל נבצע כמה פעולות הכנה.

### א. הכנה

1. נוודא, שהקובץ **requirements.txt** בשורש של האתר שלנו כולל את רשימת כל החבילות הדרושות כדי להפעיל את האתר, כגון: flask, wtforms, numpy וכו'.
  2. נבחר פורט (port) כלשהו עבור היישום שלנו. ברירת המחדל היא פורט 5000, אבל כשכמה אנשים מריצים יישומים על אותו שרת, כל אחד צריך לרוץ על פורט אחר. אפשר לבחור כל מספר בין 4 ספרות.
- נשנה את הפקודה שמריצה את היישום ל:

```
app.run(debug = False, host="0.0.0.0", port = NNNN)
```

- כאשר NNNN הוא מספר הפורט שבחרנו. הארגומנט `host="0.0.0.0"` נועד לאפשר גישה מכתובת חיצונית כלשהי.
3. נגיש את השינויים לגיטהאב.

### ב. פרסום על שרת שלנו

אם יש לנו שרת לינוקס המחובר לאינטרנט, אפשר פשוט להיכנס אליו ולהפעיל את השרת שלנו.

1. נוריד את הקוד שלנו בעזרת `git clone`:

```
git clone https://github.com/<username>/<repositoryname>
```

2. ניכנס לתיקייה ונתקין את הספריות הדרושות, למשל כך:

```
cd <repositoryname>
```

```
python3 -m pip install -r <repositoryname>/requirements.txt
```

3. נריץ את היישום:

```
python app.py
```

אם קיבלנו הודעת שגיאה שהפורט שבחרנו תפוס, נבחר פורט אחר ונריץ שוב.

היישום אומר לנו באיזה כתובות אפשר לגשת אליו, למשל, אם הפורט הוא 5000, נראה משהו כמו:

\* Running on all addresses (0.0.0.0)

\* Running on **http://127.0.0.1:5000**

\* Running on **http://178.128.241.205:5000**



ד"ר סגל הלוי דוד אראל

הכתובת הראשונה היא כתובת מקומית. הכתובת השניה אמורה להיות כתובת חיצונית של השרת שלנו. נכוון את הדפדפן לכתובת זו ונוודא שהיישום עובד.

4. כדי לוודא שהיישום ימשיך לעבוד גם כשנסגור את החלון, נסגור את היישום ונפעיל אותו שוב באופן הבא:

```
nohup python app &
```

הפקודה nohup, והסימן & בסוף, ידאגו לכך שהיישום ימשיך לעבוד גם כשנצא מהחלון. אם רוצים גם לכתוב את הפלט של התוכנית לקובץ לוג, ניתן להריץ כך:

```
nohup python app.py > app.log 2>&1&
```

כל ההודעות, שהיו אמורות להיכתב למסך, ייכתבו לתוך הקובץ app.log, ונוכל לקרוא אותם כשניכנס שוב לשרת.

5. לאחר שנריץ את הפקודה, נראה במסוף את מספר התהליך, למשל:

```
# nohup python3 app.py > app.log 2>&1 &
```

[2] **2202959**

נזכור את המספר הזה, כך שאם נרצה להעלות גירסה חדשה, נוכל להפסיק את הגירסה הישנה בעזרת:

**kill 2202959**

## ג. פרסום באתר PYTHONANYWHERE

אפשר לפרסם באתר PythonAnywhere.com. לשם כך יש לפתוח חשבון באתר. החשבון החינמי מספיק לצורך הקורס הנוכחי; אם רוצים יישומים כבדים יותר, צריך לשלם בהתאם למספר השניות הכולל שהיישום שלנו אמור לעבוד.

לאחר שפתחנו חשבון, יש לגשת לטאב "Consoles", לפתוח bash, להוריד את הקוד של האתר שלנו ולהתקין את הספריות הדרושות – ראו בסעיף הקודם, שלבים 1, 2.

לאחר מכן, ניכנס לטאב "Web", נוסיף יישום חדש בעזרת **Add a new Web App** "", ונבחר בקינפוג ידני – **Manual**.

ייווצר לנו, בתיקיית הבית שלנו, קובץ בשם כמו

```
/var/www/erelsgl_pythonanywhere_com_wsgi.py
```

(שם הקובץ יהיה שונה בהתאם לשם המשתמש שלכם). ניתן למצוא את הקובץ תחת **WSGI configuration file**.

צריך ללחוץ על הקובץ הזה ולערוך אותו, כך שיצביע לתיקיה שבה נמצא האתר שלכם. לדוגמה, הקובץ שלי מכיל את שלוש השורות הבאות:

```
import sys
```

```
sys.path.append('/home/erelsgl/research-algorithms-flask-example')
```

```
from flask_example import app as application
```

עכשיו האתר זמין לכולם! הקישור לאתר שלי הוא:

<http://erelsgl.pythonanywhere.com/>

הקישור לאתר שלכם יהיה דומה, רק עם שם המשתמש שלכם.



## ד. HEROKU

לאתרים קצת יותר כבדים, אפשר להשתמש באתר הרוקו: <http://herokuapp.com>. היתרון של האתר הזה הוא, שהוא מאפשר עדכון אוטומטי דרך גיט: בכל פעם שאתם דוחפים עדכון לגיט, האתר שלכם בהרוקו יתעדכן. החיסרון: האתר לא חינמי – כדי להעלות לשם אתרים, צריך כרטיס אשראי בתוקף.

כדי להשתמש בהרוקו, נשים את היישום שלנו במאגר-גיט נפרד.

בשורש של מאגר זה, נשים קובץ בשם Procfile (מתחיל באות גדולה), ובתוכו נשים פקודה האומרת להרוקו מה להריץ. בקובץ run.py נמחק את פקודת ההרצה, כך שעכשיו הוא מכיל רק שורה אחת:

```
from flask_example import app
```

עכשיו נשים בקובץ Procfile את השורה הבאה:

```
web: gunicorn run:app
```

הפקודה gunicorn היא פקודה המריצה יישומי-ווב. run זה שם הקובץ הראשי של היישום שלנו, ו-app זה המשתנה של היישום.

כעת נפתח חשבון חינמי באתר herokuapp. נוריד ונתקין את הכלי לשורת הפקודה (Heroku CLI) בהתאם למערכת ההפעלה שלנו. נכתוב בשורת הפקודה:

**heroku login**

וניכנס לחשבון דרך הדפדפן.

בשורש של המאגר שלנו, ניצור יישום חדש ע"י הפקודה:

```
heroku create my-application
```

כאשר "my-application" הוא שם משמעותי כלשהו שניתן ליישום שלנו.

כעת נגיש בעזרת git commit, ונדחוף את היישום לאתר הרוקו בעזרת הפקודה:

**git push heroku main**

כאשר main הוא שם הענף הראשי בפרוייקט שלנו (ניתן גם להגיש ענף אחר).

בכל פעם שנרצה לעדכן את היישום שלנו, נגיש (commit) ונדחוף (push) באותו אופן.