# מבוא ל Django

Django זה framework (דומה מאד ל flask מבחינת פיתוח אפליקציית web)

* Routes
* Templates
* Static
* Sessions
* **ORM**
* **אפליקציות מובנות (למשל אפליקציה ברירת מחדל שאומרת שהכל בסדר)**

**תרגיל Django views and URLs**

**(path זה כמו route ב flask)**

1. במקום המחרוזת Hello World תכתבו שידפיס Hello Django
2. הוסיפו view כך שלפונקציה שקשורה ל path /polls/ תדפיס Hello Again
3. הוסיפו path ל index כך שאם נכנסים ל /polls/hello מקבלים את ה Hello World המקורי
4. שנו את הנתיבים כך שה URL מתחיל ב johnbryce:

/johnbryce/polls/

/johnbryce/polls/hello

# ORM

## מודלים models

מודל זה Class שמחובר לטבלה ב database

עשינו את זה ידנית (בלי ORM) עם מוצרים בחנות.

ב Django החיבור בין ה class לטבלה נעשה אוטומטית ע"י Model

שלבי עבודה עם models:

1. **כותבים class שיורש מ Model**
2. **python manage.py makemigrations polls**
3. **python manage.py migrate**

הפקודה הזאת מייצרת SQL מתוך ה classes

ב database של Django רואים שנוצרו הטבלאות לפי ה classes

Db.sqlite3

הערה לגבי תכנון:

אפליקציית סקרים עם שאלות ותשובות.

מהו הקשר שאנו רוצים בין שאלה לתשובה?

1. לכל שאלה יש בדיוק תשובה אחת
2. לכל שאלה יש כמה תשובות
3. לכל שאלה יש כמה תשובות, ולכל תשובה יש כמה שאלות

מבחינת המושגים, ל3 האפשרויות הנ"ל יש שמות:

1. One to One, 1:1
2. One to Many, 1:N
3. Many to Many, M:N

תרגיל תאוריה:

מה הקשר בין תלמיד לכיתה?

מה הקשר בין תלמיד לקורס?

טכנית, ממשים את היחסים עם FOREIGN KEY

**תרגיל 1 (עד 19:20):**

בפרוייקט של הסטודנטים:

1. ליצור 2 מודלים עבור תלמיד וקורס עם יחס One to Many בינהם (כלומר בקורס יכולים להיות כמה תלמידים)
   * קורס:
     1. שם הקורס
     2. שם המורה
   * תלמיד
     1. שם התלמיד
     2. לאיזה קורס משוייך
2. להכניס ל DB 3 קורסים
3. לעשות pathשמציג את כל הקורסים שכרגע יש ב DB

דוגמת הרצה:

http://localhost:8000/courses

python

javascript

html

**תרגיל 2 (עד 20:30):**

1. הוסיפו 3 תלמידים לקורס פייתון
2. הציגו אותם ב path

דוגמת הרצה:

<http://localhost:8000/students>

tal

gal

Yossi

**תרגיל בית (המשך):**

1. להוסיף path שמציג את התלמידים בקורס שמופיע ב url:

דוגמת הרצה:

<http://localhost:8000/students/python>

רק תלמידים בקורס פייתון

<http://localhost:8000/students/javascript>

רק תלמידים בקורס JS

1. להוסיף path שמציג את שם תלמיד לפי ה id שלו:

דוגמת הרצה:

<http://localhost:8000/students/1>

מציג את שם התלמיד עם id 1

1. להוסיף path שמציג שם קורס עבור תלמיד מסויים:

<http://localhost:8000/course/student/1>

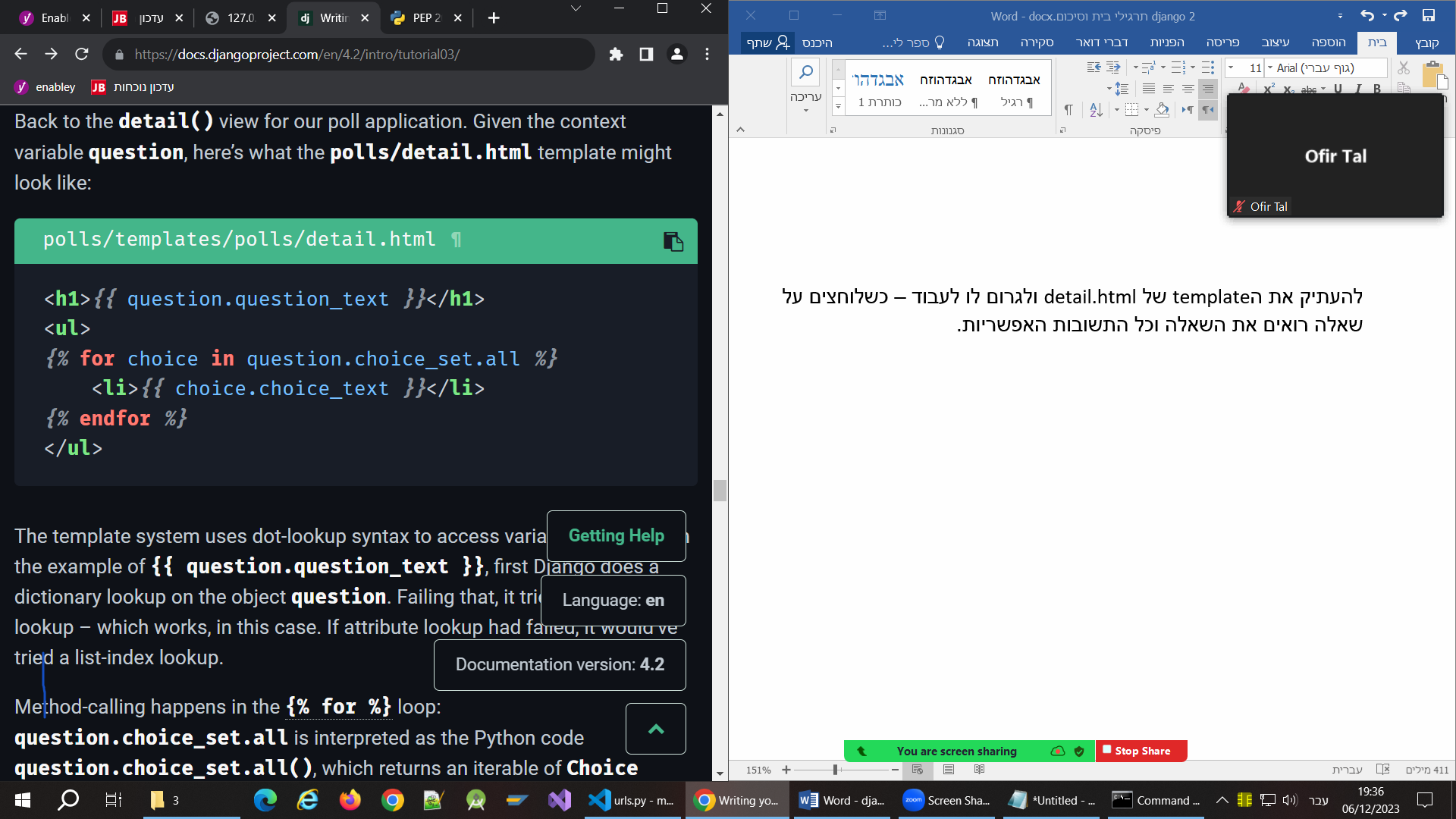
מציג את שם הקורס של תלמיד עם id=1

<http://localhost:8000/course/student/student1>

מציג את שם הקורס של תלמיד עם שם student1

תרגיל:

להעתיק את הtemplate של detail.html ולגרום לו לעבוד – כשלוחצים על שאלה רואים את השאלה וכל התשובות האפשריות.



שלבים לפתרון:

1. להבין מה רוצים מאיתנו. במקרה הזה לוקחים Question ועוברים על כל ה Choices בתוך ה choice\_set שלו.
2. להחזיר Question לתוך ה template
3. /polls/1 יחזיר את שאלה עם id=1 ל Template, וה template יודע לעשות לולאה על כל ה choices
   1. לחבר את /polls/1 עם view – מחובר ע"י detail שזאת פונקציה בתוך views
   2. לכתוב את detail כך שתעשה render עם השאלה עם id=1 ל detail.html
   3. להעביר ל detail.html משתנה שנקרא question שבתוכו יש אובייקט מסוג Question

תרגילים:

1. באפליקציית students, הוסיפו דף בית עם template שמציג את כל התלמידים
2. כנ"ל את כל הקורסים ב Template
3. הוסיפו endpoint שמוסיף תלמיד בצורה הבאה:

/students/add/tal

ומחזיר template עם ההודעה הבאה:

The student **tal** was added successfully with id 2

1. צרו אפליקציה חדשה בשם cars בפרוייקט חדש cars\_project.

* צרו Model בשם Car עם תכונות שם, צבע ומחיר
* צרו endpoint שמוסיף רכב חדש עם תכונות רנדומליות (למשל שם car1, צבע אדום מתוך 5 צבעים ומחיר 120000(
* צרו endpoint שמחזיר את רשימת שמות כל הרכבים עם template
* כנ"ל רק מחזיר את כל התכונות של הרכב לא רק את השם בצורה של טבלה עם template
* **בונוס**: להוסיף גם קישור לתמונה ולהציג אותה
* **בונוס**: לשנות את צבע הרקע של ה div שבתוכו נתוני הרכב לצבע הרכב (עדיף עם cards)