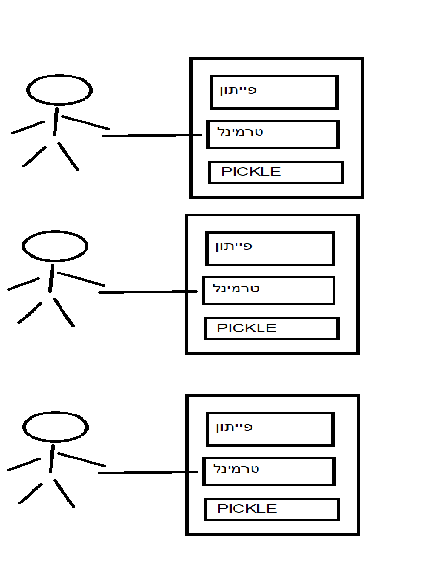
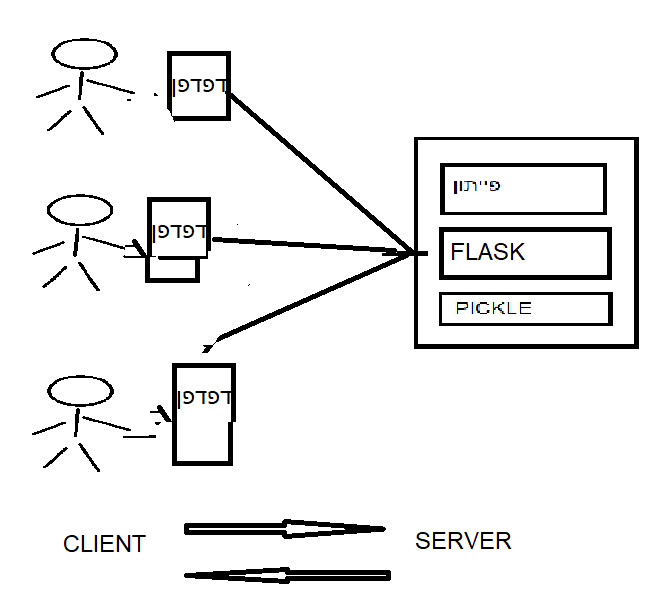
# ­מתחילים FLASK

מה עשינו עד עכשיו בפרוייקט?

## אפליקציית Desktop



## אפליקציית Client Server



FLASK

1. **התקנה והגדרות**
2. **Route, URL** 
   1. נתיב פשוט: /discount/coupon/123456
   2. העברת נתונים עם args:

/discount?coupon=123456&dis=10

* 1. העברת נתונים בתוך הנתיב:

/discount/coupon/123456

/discount/<coupon>

1. **Jinja templates**
   1. **Template Inheritance (blocks)**
2. **Forms**
   1. **validations**

## Jinja Templates

1. ליצור template עם שם וטלפון, כל אחד ב div משלו
2. להוסיף גם מייל contacts.html, כל אחד ב div משלו
3. צרו מחלקה contact עם תכונות שם, מייל, טלפון וכתובת והציגו אותה ב template
4. כנ"ל רק עם מילון עם התכונות הנ"ל

contact={"name":"tal", "address":"haifa"}

1. לעשות רשימה של contacts (אובייקטים או מילונים) ולהציג ב template

כל איש קשר ב div משלו

לעצב עם CSS

1. כנ"ל רק רשימה של Tasks כמו בפרוייקט 1
2. לקחת את הקובץ pickle מפרוייקט 1 ולהציג את הנתונים (המשימות) שיש שם

ב Template

## שליחת נתונים לשרת

יש כמה דרכים:

1. על ה URL

<http://127.0.0.1:5000/contacts?name=tal&phone=054-222222>

http://127.0.0.1:5000/coupon?code=123456

1. על ה URL בתוך הנתיב עצמו:

http://127.0.0.1:5000/code/123456

**תרגיל:**

יש 3 קופונים, עם הנחות של 10, 15, 20 אחוז. כתבו route שמציג את ההנחה המתאימה לפי הקופון, או הודעה שהקופון לא נכון.

<http://127.0.0.1:5000/code/122>

**תרגיל:**

להוסיף תמונה לפרופיל, השם של התמונה אותו דבר כמו שם המשתמש, למשל:

<http://127.0.0.1:5000/profile/tal>

tal.jpg

להוסיף עוד משתמש כך שיעבוד:

http://127.0.0.1:5000/profile/shani

תרגיל:

1. Route שמקבל שני מספרים ומחזיר את הסכום
2. Route שמקבל שם ומחזיר טלפון (לפי מילון על השרת)

{"tal":"54354353"}

1. משחק ניחושים: route שמקבל מספר בין 1 ל 10 ובודק אם הוא שווה למספר בין 1 ל 10 שנבחר רנדומלית. אם גדול מהמספר מחזיר too large, אחרת too small

מושגים:

אתר סטטי: הכל מוגדר מראש, כל התוכן כבר כתוב. לא מגיב לקלט.

**אפליקציית ווב** (מערכת ווב): אינטראקטיבית, כלומר מגיבה לקלט. לא הכל מוכן מראש, התוכן נוצר במיוחד לפי הקלט.

אפליקציות מובייל: כמו אפליקציות ווב מבחינת ה backend. ההבדל הוא ב front, כלומר בחלק שהמשתמש רואה שרץ על הטלפון במקום על המחשב.

סוגי אפליקציות מובייל:

* Native
  + כתובה בשפה שהמכשיר מדבר
    - IOS: Objective C, Swift
    - Android: Java, Kotlin
* Hybrid
  + בעצם האפליקציה היא דפדפן קטן

תרגיל forms:

1. להוסיף מה שצריך כדי לקבל גם hello עם השם וגם תאריך של היום
2. להוסיף בדיקה אם השם שווה לשם שלכם, להחזיר hello, אחרת להחזיר unauthorized
3. להוסיף עוד input לשם משפחה ולהחזיר hello עם השם ושם משפחה
4. לעשות form שמקבל מחיר ומחזיר מחיר + מעמ
5. לעשות form עם שני inputs כך שאם מכניסים שני מספרים מקבלים חזרה את הסכום
6. לעשות form שמקבל קופון ואם הקופון נכון מחזיר שקיבלת הנחה 10%
7. כנ"ל רק להוסיף input עבור מחיר ואם הקופון נכון להחזיר את המחיר פחות 10%

תרגיל templates:

1. לסיים את כל ה routes עם ה templates המתאימים עם template inheritance כלומר blocks כמו בדוגמא ב add.html

בדיקות תקינות של forms (ולידציה validations)

1. על השרת: יש לוודא שמקבלים את כל הערכים שצריך, אחרת להחזיר חיווי למשתמש.
2. על הדפדפן: להשתמש ב HTML required

תרגול blocks

1. צרו base.html עם כל ה html הרגיל כולל title, קישור ל css, וסרגל ניווט עם 3 קישורים ולוגו של חברה ותיבת חיפוש.
2. צרו ממנו (כלומר extends) דף עם כמה כתובות מייל
3. כנ"ל רק טלפונים
4. דף שמכיל גם את הכתובות מייל וגם את הטלפונים
5. העבירו את תיבת החיפוש לדף חיפוש שנוצר מ base.html
6. צרו דף עם רשימת המיילים, הטלפונים ותיבת החיפוש בבלוקים מתוך base.html