

מהי רשת אינטרנט? מה זה www? מהו מודל שרת - לקוח? מהו יישום של אינטרנט (או web)

מאגר של שרתים שמכילים דפים ובהם טקסט, מוסיקה, וידאו, אתרי קניות ועוד. כאשר אנו מקישים כתובת של אתר ולוחצים <enter> - אנחנו פונים לשרת מסוים ומבקשים דף מסוים.

כתובת אינטרנט מתורגמת לסידרה של מספרים שמובילה לדף ייחודי בין מיליארדי וטריליוני הדפים קיימים ברשת.

הנה סרטון אנימציה קצר שמדבר על התהליך.

<http://www.learnthenet.com/learn-the-basics/>

מה זה URL? Universal Resource Locator (נראה דוגמה בהמשך)

פרוטוקול HTTP

**יישום Web - מודל שרת לקוח**

יישום Web טיפוסי מורכב מהרכיבים האלה:

- א- דפדפנים (יישום הלקוח);
- ב- שרתי Web (יישום השרת)
- ג- פרוטוקול HTTP

פרוטוקול המגדיר את המבנה ואת הסדרה של ההודעות שמועבות בין הדפדפן לשרת ה-Web;

- ד- תקן להצגת מסמכים (HTML).

הדיאגרמה מראה את המערכת האישית (מחשב האישי) שדפדפן ומבקש דף ושרת מחשב מרוחק. חץ מעלה משרת לקוח ומחץ מלקוח לשרת. חץ מעגלי מלקוח לשרת ומחץ משרת ללקוח.



## כתובות באינטרנט (I)

- לכל מחשב שמחובר לאינטרנט יש כתובת ייחודית המאפשרת לזהות אותו באופן ייחודי, כפי שמספר זהות מאפשר לזהות אזרח באופן ייחודי.
- ישנן שתי דרכים לייצג כתובת כזו:
  - כתובת IP (IP address). כתובת IP היא מספר, שלפי התקן הנוכחי מורכב מ-32 סיביות
    - מייצגים את הכתובת על-ידי רישום ארבעת הבתים, מופרדים על-ידי נקודה.
    - דוגמאות: 127.0.0.1, 218.60.104.117, 126.17.54.110.
  - שמות תחומים (domain names). שמות תחומים מאפשרים לזהות מחשבים באופן ייחודי.
    - לכל שם תחום יש כתובת IP ייחודית.
    - למשל שם תחום: www.police.gov.il.
  - השימוש בשם www מופיע לעיתים קרובות כשם של מחשב שמריץ שרת Web



## כתובות באינטרנט (II)

- יישום רשת שנקרא DNS (שירות שמות תחומים – Domain Name System) יישום זה כולל שרתי שמות רבים שתפקידם לתרגם שמות תחומים לכתובות IP.
  - יישום זה כולל שרתי שמות רבים שתפקידם לתרגם שמות תחומים לכתובות IP.
- מפתח (port) מאפשר למערכת ההפעלה לנתב את המידע המגיע מהאינטרנט אל היישום המתאים.
  - מאחר שלכל יישומי הרשת המורצים באותו מחשב יש אותה כתובת IP (או אותו שם תחום), קובעים לכל יישום כזה מספר נוסף שנקרא מפתח (port).
  - ליישומים ידועים יש מספרי מפתח ידועים (well-known ports).
    - למשל, שרתי Web (פרוטוקול HTTP) מאזינים למפתח 80;
    - שרתי דוא"ל (פרוטוקול SMTP) מאזינים למפתח 25.
    - רשימת המפתחים של כל השירותים התקניים באינטרנט נמצאת באתר – <http://www.iana.org>.
  - יישומי משתמש משתמשים במספרי מפתח גדולים מ-1023

כתובת האי י פי היא סידרת 4 מספרים בין 0 ל-255. מי מתרגם בין שם הדף והכתובת?

הפורטים עד 1023 תפוסים על ידי ישומים קיימים, למשל פורט 80 להודעות מסוג HTTP ופורט 25 להודעות דואר.

URL

**מטח**

## URL שימוש במאתר משאבים אחיד לזיהוי משאב

- מתכנני האינטרנט הגדירו דרך לזיהוי ייחודי של משאבים:
- מבנה שנקרא מאתר משאבים אחיד (Unified Resource Locator) - URL. בהינתן URL, הדפדפן יכול להגיע מיד לדף המבוקש בלי שיהיה צורך לעבור דרך מסמכים אחרים.
- URL הוא מחרוזת תווים המורכבת מחמשת החלקים האלה:
  - הפרוטוקול שבו משתמש השרת
  - שם המחשב או כתובת IP
  - מספר המפתח
  - שם הקובץ (כולל נתיב התיקיות שבו נמצא הקובץ)
  - נקודת הייחוס (reference) בתוך הקובץ
- דוגמה:  
`http://www.openu.ac.il:80/main/subjects/education.html#java`

סירטון המחשה:

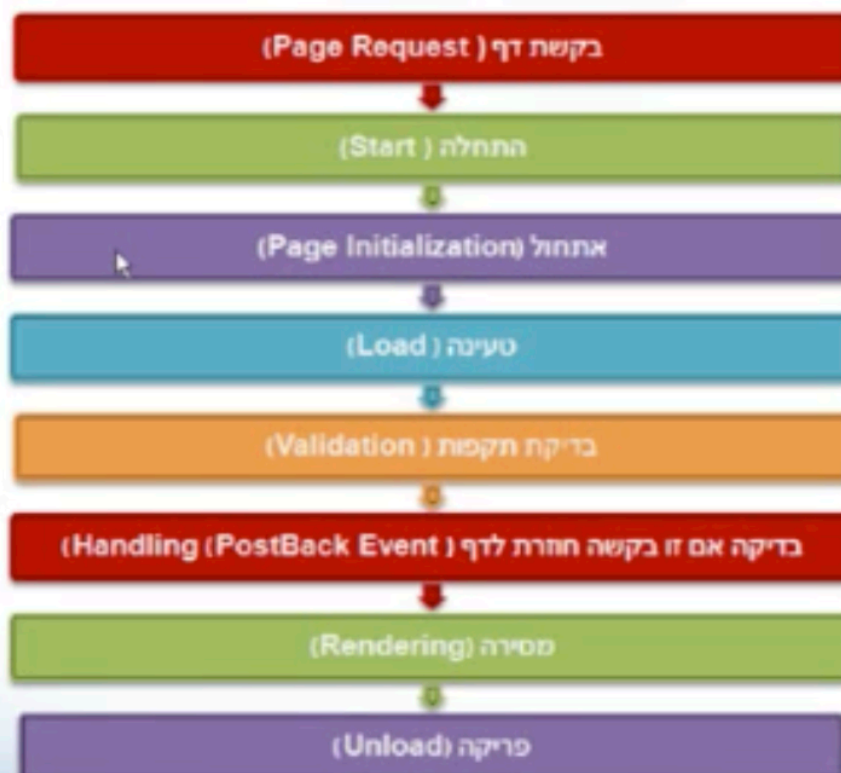
<http://www.warriorsofthe.net>

## יישום שרת-לקוח בטכנולוגיית ASP

- שיחה בין לקוח לשרת מתחילה בשליחה של הודעת בקשה מאת הלקוח (הנקראת HTTP Request) ומסתיימת בשליחת הודעת תגובה מהשרת (HTTP Response).
  - שיחה זאת נקראת מעבר הלוך-ושוב (Round Trip), והיא חייבת לכלול לפחות מעבר הלוך-ושוב אחד.
- בקשת דף המכיל תמונות או קובצי קול תגרום למעבר הלוך-ושוב גם בעבור הדף עצמו וגם בעבור כל קובץ (קול או תמונה) בנפרד, וזאת לפני ניתוק הקשר בין השרת ללקוח.

מחזור החיים של דף asp

## מחזור חיים של דף asp



## תכנות מצב חסר

המודל של שרת-לקוח אינו שומר מצב, כלומר כל בקשה היא עצמאית ללא תלות באחרות, ומטופלת מההתחלה ועד הסוף. היתרון - טיפול בהרבה בקשות ללא שמירת היסטוריה. החיסרון שכל בקשה מתחילה מאפס.

תהיינה דרכים (כמו עוגיות) לשמור מצב (או שימוש בבסיס נתונים).



## תכנות מצב-חסר (stateless)

- שרת ה-Web אינו שומר כל מידע על התקשורת בינו ובין הלקוח
- בכל בקשה המגיעה מלקוח (דפדפן) פותח השרת קשר אל הלקוח ששלח את הבקשה ובונה בעבור אותו לקוח דף HTML מבוקש.
- בסיום הטיפול בבקשה, השרת סוגר את הקשר עם הלקוח ומתפנה לטפל בבקשות של לקוחות אחרים.
- אם הלקוח רוצה לבקש בקשה נוספת מהשרת, עליו להפעיל תהליך זה מחדש, שכן השרת לא שמר כל מידע על השיחה האחרונה, ולפיכך טיפול בבקשה חדשה של אותו לקוח מתבצע ללא כל תלות בבקשה הקודמת.



## תכנות מצב-חסר (stateless)

- צורת ההתקשרות הזאת בין שרת ללקוח נקראת מצב-חסר (Stateless)
- לתכנות בסביבה זו קוראים תכנות מצב-חסר (Stateless Programming).
- יתרונות
  - שיטה זו, שבה השרת אינו שומר מידע על ההתקשרות, מאפשרת לו לטפל באלפי בקשות ביעילות רבה יותר.
- חסרונות
  - מידע לא נשמר בין התקשרות להתקשרות, המשתמש שגלש לאותו אתר צריך להזדהות בכל פעם מחדש.
- פתרונות
  - שמירת מידע בשיטות שונות בצד שרת או בצד לקוח

סביבת העבודה – Visual Web developer 2008 Express edition (או - visual studio)

New -> website

מצד ימין תהיינה כמה תיקיות

In the solution explorer: Config - not used for now, Web\_config, app\_data, default.aspx

איך ניראה דף HTML (דוגמה)

Head  
Body

עבודה עם תגיות - כותרות, טבלאות וכד'.

דוגמה לדף של aspx

ירוך בשרת, תהיה בו אינדיקציה של איזו שפה בשימוש (C#)  
הקוד לשרת מופיע בין התגיות:

<% %>

אם מנסים להציג את הדף כיצד יראה בדפדפן, רק התגיות של HTML יראו, ולא מה שהוא תוצאת הקוד עבור השרת. אפשר לעמוד על הדף, לחצן ימני - הצג בדפדפן.

אם בדף יש רק קוד של HTML (דף סטטי), הדפדפן יציג אותו על פי התגים הנתונים.

כדי להריץ דף של aspx. אפשר ללחוץ על החץ הירוק - זה יפעיל את השרת (על המחשב המקומי). השרת יחליף את חלקי הקוד שאינם HTML ויתרגם הכל ל-HTML.

אם מסתכלים על המקור (source) של הדף שהורץ, רק את הקוד ב-HTML ניראה ולא את קוד המקור שיצר אותו. דוגמה לתאריך ושעה.