

מסדי נתונים – 89-281

תרגול 1,2 – XML + DTD

אור כדראוי בשיתוף עם עמיעד רוזנברג

כללי

- סילבוס הקורס :
http://www.biu.ac.il/syllabus/y73/Syll_d89/8928101.rtf
- אתר התרגול :
<https://sites.google.com/site/dbbarilan/>
- דרכי תקשורת :
מייל – or.kadrawi@gmail.com
- שעות קבלה – בתיאום מראש (דרך המייל).

מטלות

- תרגילים תיאורטיים / מעשיים – בהמשך הקורס
- פרויקט גמר
- הגשה – בעזרת מערכת ה-submit.
- הפרויקט יהיה בשפת .net c# בלבד!
- למי שלא מכיר – מומלץ כבר עכשיו להתחיל ולהכיר את השפה.
- בחינה

מה יהיה לנו בתרגול

- חומר ההרצאה הוא בעיקר חומר תיאורטי על מסדי נתונים. מטרת התרגול היא להקנות לכם ידע פרקטי בעבודה מול מאגרי נתונים ומערכות ניהול בסיסי נתונים. ישנם נושאים חופפים בין ההרצאה והתרגול, אך יילמד גם חומר בנפרד שיש לדעת אותו לבחינה.
- חומר התרגול זהה בכל 3 התרגולים. ניתן להגיע לכל אחת מהתרגולים (גם זו שאינכם רשומים אליה). עם זאת, במידה ותיווצר הרצאה עמוסה במיוחד – הנכם מתבקשים להיכנס אליה רק במידה ואתם רשומים אליה!

בתרגול נתעסק ב...

- מבנה נתונים - XML ושותפיו. (3 שיעורים)
 - XML.
 - DTD – סכמה עבור קבצי XML.
 - XPath 1.0 – שפת שאילתות עבור קבצי XML.
 - פרויקט קורס.
- מודלים שונים של בסיסי נתונים.
- אלגברת יחסים. (2 שיעורים)
- שפת שאילתות – SQL. (4 שיעורים)
 - Data Definition Language – DDL.
 - Data Manipulate Language – DML.
- נרמול בסיס נתונים (חומר של ההרצאה)
- מבחנים ישנים. (בזמן הנותר)
- הערה : מסד הנתונים עליו נלמד ונעבוד הוא MySql.

לפני שמתחילים - שאלות?



SGML -> XML

- XML הוא סוג של צמצום של SGML (Standard Generalized Markup Language).
- Markup Language = המטה-דאטה (מידע נוסף המתאר / מסביר את המידע המועבר) מועבר יחד עם המידע עצמו.
- eXtensible Markup Language = XML
- "הרחבה" = השפה אינה מורכבת מתגיות ספציפיות שהוגדרו מראש, אלא מאפשרת לנו להרחיב אותה ולהגדיר תגיות לפי הצורך הפרטי שלנו. בעצם, אנחנו יכולים להוסיף סוגי מידע שונים לפי מה שנרצה.
- אילו יתרונות יש לנו באוסף קבוע של תגיות?
 - האוסף הקבוע מוכר ע"י הצרכנים השונים (כותבי וקוראי המסמך).
 - אילו חסרונות?
 - בעיתיות במקרה של דרישה לתגיות שאינן חלק מאוסף.

למה XML ?

- קובץ XML הוא קובץ טקסט פשוט.
כל תוכנה בכל מערכת הפעלה יכולה לקרוא את המידע.
- מאפשר למידע בפורמט XML להיות בעל ייצוג זהה בכל מיני סביבות שונות ולכן מאפשר להעביר מידע בין סביבות שונות שלא תוכננו כלל עבוד אחת עם השניה. (למשל: בין מסד נתונים של יוניקס למסד נתונים של חלונות)

שימושי XML

- 2 שימושים עיקריים (כיום).

- איחסון מידע – בד"כ בפורמט ידוע כלשהו. (ללא צורך בתוכנה מיוחדת)

- למשל, קובץ קונפיגורציה לאפליקציות שונות.

- קבצי שפה במערכת android.

- תחביר להעברת מידע – מאפשר העברת נתונים בין סביבות שונות שלא תוכננו מראש לעבודה בצורה מתואמת.

- למשל, שמירת מחירון בפורמט קבוע כלשהו המאפשר ללקוחות השונים לגשת למידע ולהציג אותו בצורה שונה.

למשל, אתרי השוואות מחירים שונים, אתרי המכירות מאפשרים לקבל את המחירונים שלהם בקובץ XML. אתרי השוואת המחירים "עוטפים" את המידע ומציגים אותו לפי העיצוב של האתר.

מידע מובנה ומובנה למחצה

- מסד נתונים רלציוני (טבלאי) = מסד נתונים המבוסס על טבלאות (ORACLE, SQL SERVER, ACSESS וכו')
- המידע במסד נתונים רלציוני (טבלאי) הוא מידע מובנה.
- במסד נתונים רלציוני (טבלאי) אנחנו נדרשים להגדיר "סכמה" – המבנה שבו יופיעו הנתונים במסד הנתונים (אופן חלוקת המידע על פני הטבלאות השונות).
- כאשר אנחנו משתמשים במסד נתונים כזה, אנחנו "מוגבלים" לסכמה שייצרנו. לא ניתן לסטות ממנה. הגבלה זו מחייבת אותנו לעבוד בפורמט שלעיתים הוא פחות טבעי ונוח.

מידע מובנה ומובנה למחצה

- מידע המופיע בספר הטלפונים במכשיר הסלולרי שלנו :
- יש לנו מידע שיופיע עבור כל כניסה :
 - שם פרטי
 - שם משפחה
 - קישור לתמונה
- לכן, ברור שבמודל הרלציוני יהיה לנו צורך בטבלה שתכיל את השדות הללו.
- אך המידע שנשמור עבור כל אדם יהיה שונה לחלוטין.
- מספר פלאפון (אחד או יותר) \ מספר בבית \ מספר בעבודה \ כתובת דוא"ל \ קבוצות אליהם הרשומה שייכת (אחת או יותר) וכו'
- בעצם, חלק מהמידע שלנו הוא בפורמט קבוע, אבל חלקו הוא בפורמט משתנה. לכן, המידע שלנו הוא "מובנה למחצה".
- פורמט XML מאפשר לנו לתת ייצוג טבעי למידע בפורמט זה.

יכולת הביטוי של XML

- XML אינה מוגבלת מבחינת התגיות.
אין כלל הגדרה מראש של התגיות. (לגבי כלל קבצי ה-XML)
התגיות יוגדרו ע"י כותב ה-XML עצמו. מאפשר יכולת ביטוי גבוהות מאוד עבור מידע הנשמר בפורמט XML.
- היכולת שלנו לקבוע לעצמנו את התגיות מאפשרת לנו לתת את הייצוג הטבעי עבור המידע המובנה למחצה. אנחנו יכולים להגדיר בעצמנו האם להוסיף מידע על תחום מסוים (ואיזה מידע להוסיף) וכן לא להוסיף מידע כאשר הוא אינו רלוונטי.
- תגיות ה-XML נמצאות במטרה לתאר את (מבנה) המידע.
הפוקוס של המטה-דאטה (הנתונים הנוספים הנשלחים עם המידע) ניתן על (מבנה) המידע עצמו שמועבר.

XML - ייצוג המידע

תג פותח

- XML מתרכז באיחסון והעברת מידע וכן במבנה שלו. למשל:

```
<?xml version="1.0"?>
<message>
  <from> CS Office </from>
  <to> 2nd year students </to>
  <title> Test grades arrived </title>
  <info> The grades of course 89281 was published </info>
</message>
```

תג סוגר

- אז איך הדפדפן יודע לקרוא את המידע ולהציג אותו?
- הוא לא! קריאת המידע וניתוחו היא באחריות ה-parser הקורא את המידע. (לדפדפנים יש parser מובנה לקריאת קבצי XML וכן קוד להצגת הקובץ)
- ישנם 3 גורמים בתהליך: הקובץ עצמו, התוכנה שקוראת את המידע מתוך הקובץ והתוכנה שמציגה \ משתמשת במידע.
- נרחיב על הנושא בהמשך. בשלב זה נתרכז במידע שאנחנו רוצים לשמור בקובץ.
- קובץ ה-XML אינו עושה דבר מלבד אחסון המידע והמבנה.

XML - ייצוג המידע

• שימו לב שהמידע בקובץ

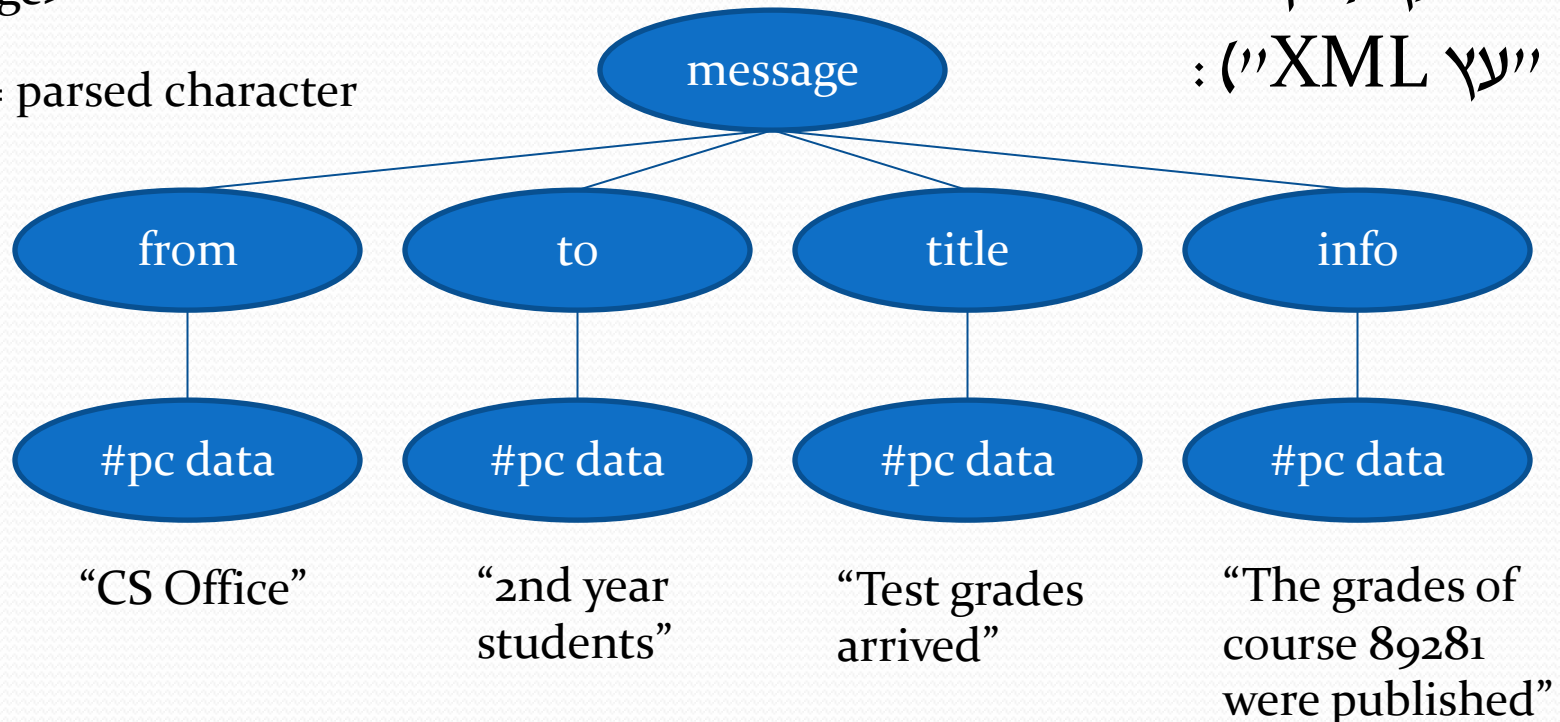
XML מיוצג בפורמט

של עץ (נקרא גם

"עץ XML") :

```
<message>  
  <from> CS Office </from>  
  <to> 2nd year students </to>  
  <title> Test grades arrived </title>  
  <info> The grades of course 89281  
    were published </info>  
</message>
```

➤ pc = parsed character



XML - ייצוג המידע

• נשים לב לכמה נגזרות מעניין זה :

• קיים שורש אחד ויחיד לקובץ ה-XML.

• צומת "רגיל" בעץ מקביל לתג (פותח וסוגר).

שאר תת העץ יופיע בין תגים אלו.

• פתיחת התגים וסגירתם חייבת להתבצע

בסדר הפוך (הראשון שנפתח הוא האחרון שנסגור).

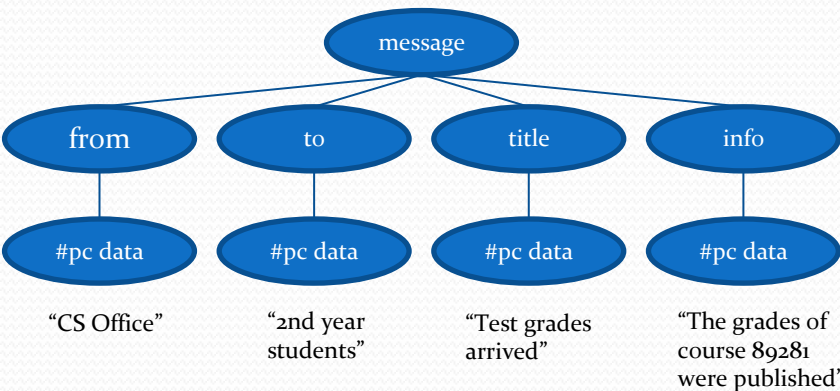
מצב כזה : ` text ` **אינו תקין!**

• צמתי הטקסט ייכתבו בצורה מפורשת במסמך (ללא התגים).

• ע"מ לציין תג ללא בנים נוכל להשתמש בסימון המקוצר `<tag/>`

למשל, אם נרצה להציג הודעה ללא טקסט בכותרת, במקום לרשום

`<title></title>` נוכל לרשום `<title/>` בלבד.



`<message>`

`<from> CS Office </from>`

`<to> 2nd year students </to>`

`<title> Test grades arrived </title>`

`<info> The grades of course 89281
were published </info>`

`</message>`

XML - ייצוג המידע

- XML הוא חצי מובנה.
- עד כה ראינו רק מידע בצורה מובנית (כלומר, מידע שיש דרך קבועה להצגה שלו).
- נראה כיצד נראה חלק מקובץ XML של ספר טלפונים:
- ```
<Phone-Book>
 <entry>
 <name> Avi Cohen </name>
 <tel> 03-9242356 </tel>
 <tel> 054-6325843 </tel>
 </entry>
 <entry>
 <name> Beni Levi </name>
 <tel> 052-3458974 </tel>
 </entry>
</Phone-Book>
```
- אלמנטים \ תגים  
ערכי טקסט
- לרשומה של אבי יש 2 מספרים, בעוד לרשומה של בני יש רק מספר אחד.  
ייתכן מצב שבו ישנם אנשים עם יותר מ-2 מספרים.
- כאשר אנחנו נדרשים להציג בטבלה מידע כזה – יש לנו בעיה (משום שאנחנו לא יודעים כמה שדות להקצות מראש עבור מספר הטלפון).



# XML - ייצוג המידע - תכונות

- לפעמים, לאלמנטים שונים יש תכונות שונות.  
נרצה בספר הטלפונים להגדיר מספר טלפון מועדף.

- ```
<Phone-Book>
  <entry>
    <name> Avi Cohen </name>
    <tel preferred="true"> 03-9242356 </tel>
    <tel> 054-6325843 </tel>
  </entry>
  <entry>
    <name> Beni Levi </name>
    <tel> 052-3458974 </tel>
  </entry>
</Phone-Book>
```

תכונה (Attribute)

ערך התכונה

XML - ייצוג המידע - תכונות

<entry>

<name> Avi Cohen </name>

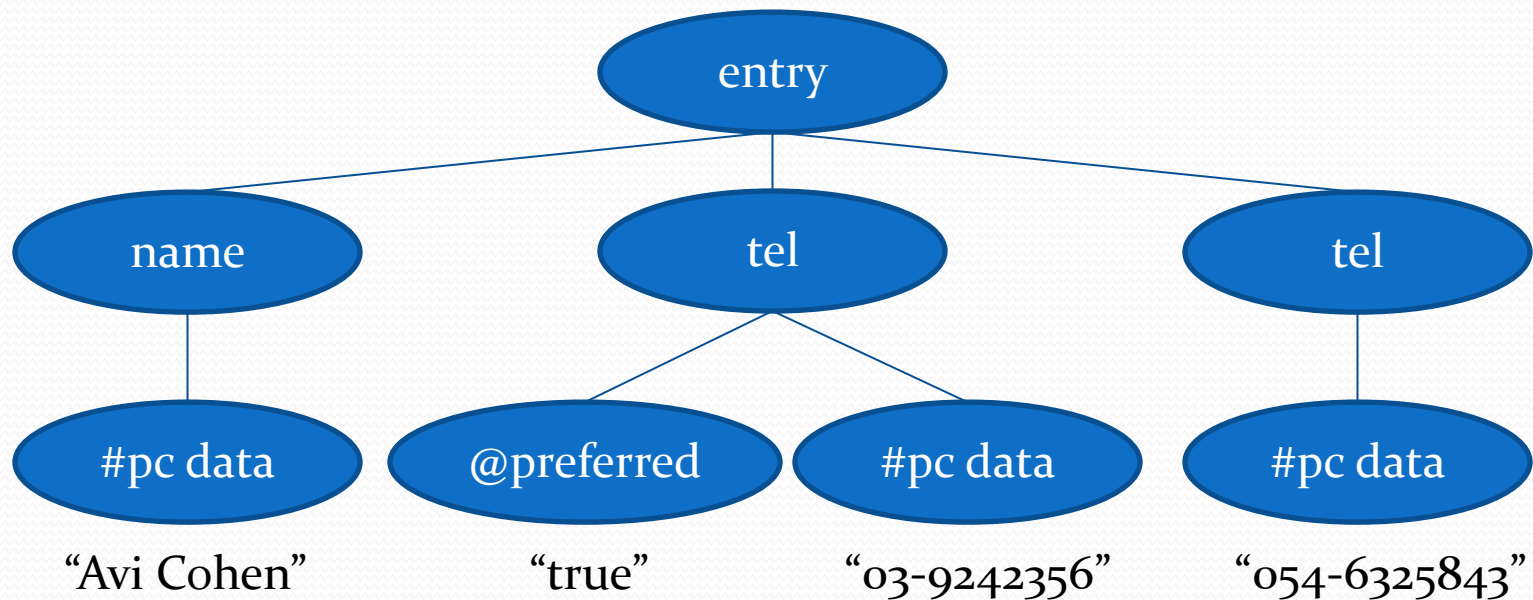
<tel preferred="true"> 03-9242356 </tel>

<tel> 054-6325843 </tel>

</entry>

• בעץ XML נייצג זאת כך :

• (בספרות ישנן גישות נוספות)



XML - ייצוג המידע

- תחילת קובץ של XML תכיל שורת הצהרה :
- `<?xml version="1.0"?>`
- במידת הצורך - יש להוסיף את הקידוד של המסמך :
- `<?xml version="1.0" encoding="UTF-8"?>`
- שאלה : לפי איזה קידוד נדע לקרוא את שדה הקידוד?
- <http://www.w3.org/TR/REC-xml/#sec-guessing>
- הערות בקובץ יופיעו עם `"<!--"` ו- `"-->"` (הערות לא נקראות ע"י ה-parser של XML. הן מופיעות בשביל הקריאות של מי שקורא ישירות את הקובץ) :
- `<!-- comment -->`
- אין לשים רווחים בתוך שמות התגים השונים!
- יש לשים לב לכך ש-XML היא case sensitive ולכן tel ו-Tel הן תגיות שונות!
- קובץ שנכתב ע"פ חוקי התחביר התקינים נקרא "well formed" (בנוי היטב).

XML - ייצוג המידע

• דוגמא לקובץ well formed :

```
• <?xml version="1.0"?>
  <Phone-Book>
    <entry>
      <!--first entry-->
      <name> Avi Cohen </name>
      <tel preferred="true"> 03-9242356 </tel>
      <tel> 054-6325843 </tel>
    </entry>
    <entry>
      <!--second entry-->
      <name> Beni Levi </name>
      <tel> 052-3458974 </tel>
    </entry>
  </Phone-Book>
```

• בעיה :

• מה נעשה כאשר אנחנו רוצים להעביר מידע שכולל תגיות XML?
או למשל כאשר המידע שנרצה להעביר הוא "3 < 2 & 1"? כיצד נבדיל בין המידע לבין התגיות?

XML - ייצוג המידע

- לצורך זה הוגדרו מספר "ישויות" (Entities) בשפה שמשמשות אותנו לצורך זה. דוגמאות:

• "<" = <

• ">" = >

• "&" = &

• &#DecimalUnicodeValue;

• &#xHexadecimalUnicodeValue;

- ```
<?xml version="1.0"?>
<Formulas>
 <Formula> 1 & 2 < 3</Formula>
</ Formulas >
```

  - זו המשמעות של pcdeta (parsed character) – מידע שעובר "פירסור" ומתחשבים בישויות הנמצאות בו.
  - אפשרות נוספת (לעיתים נוחה יותר) היא להגדיר מקטע CDATA – קטע מידע ש"מועבר" כפי שהוא.
- ```
<?xml version="1.0"?>
<Formulas>
  <Formula> <![CDATA[1 & 2 < 3]]> </Formula>
</ Formulas >
```

XML-הגדרת סכמה

- מבנה של XML נקבע מראש ע"י המתכנת...
- אין לנו פורמט קבוע שאנחנו צריכים להתאים את עצמנו אליו...
- ובכל זאת – איך נוכל לדעת מה הפורמט של המסמך שקיבלנו?
בסופו של דבר – התוכנה שלנו צריכה לדעת לקרוא ולנתח את המידע המופיע בו...
- איך נדע שהקובץ אכן מכיל את המידע שאנחנו מצפים לקבל?
איך נדע שהקובץ נמצא בפורמט שאנחנו מצפים לקבל?
איך נדע מהם שמות התגים שבהם נמצא המידע שאנו מחפשים?
- יש צורך להגדיר בצורה כלשהי מהם המסמכים התקינים עבור התוכנה שלנו (מסמכים התואמים למה שאנחנו מצפים לקבל).


XML-הגדרת סכמה - DTD

- שפת Document Type Definition – DTD.
- מגדירה את קב' המסמכים ה"תקינים" (valid).
- היכולת להגדיר קב' מסמכים תקינים מאפשרת לקבוצות מוסמכות להגדיר את הפורמט של מסמכי ה-XML המתאימים להם...
- מתמטיקאים יכולים להגדיר תגיות מיוחדות עבורם (ליצור תגית עבור העלאת ערך מסוים בריבוע, עבור הוצאת שורש וכו').
- DTD תגביל בד"כ את סוגי הצמתים והתוויות שלהם (התגים והתכונות) אבל לא את ערכי הצמתים.

XML-הגדרת סכמה - DTD

- מסמך DTD הוא מסמך הצהרות. כל הצהרה היא מהצורה `<!.....>`
- ניתן להוסיף הערות בפורמט `<!-- comment -->`
- הצהרות ה-DTD ייכתבו בתוך קובץ ה-XML או כקובץ נפרד...
- בתוך הקובץ:

- ```
<!DOCTYPE root [
 declaration 1
 declaration 2
 ...
 final declaration
>
```

 declarations

- בקובץ נפרד:

- `<!DOCTYPE root SYSTEM "FileName">`

כאשר `FileName` הוא הפניה (יחסית או ישירה) לקובץ בו נמצאים ההצהרות.

- כזכור, XML תקין הוא בעל שורש יחיד. שורש זה הוא ה-`root` המוגדר בהצהרה `DOCTYPE`.

- יופיע ישירות אחרי שורת ההצהרה של מסמך XML.

• תזכורת: `<?xml version="1.0" encoding="UTF-8"?>`



# XML-הגדרת סכמה - DTD

- מסמך DTD מאפשר לנו להגדיר את התגיות השונות והמבנה ההיררכי שלהם כפי שאנחנו מצפים לראות במסמך ה-XML.
- במסמך נצהיר על:
  - אלמנטים – התגים של קובץ ה-XML.
  - מאפיינים – התכונות של התגים במסמך XML.
  - ישויות – מאפשר לנו ליצור ישויות חדשות מעבר לאלה הקבועות.
  - PCDATA – טקסט שנרצה שיפוענח (שהישויות בו יהפכו לקבועים שאותם הם מייצגים והאלמנטים הנמצאים בקטע זה יזוהו).  
שדות טקסטואליים תמיד יהיו מסוג PCDATA.

# XML - הגדרת סכמה - DTD - תחביר

• ניזכר במסמך ה-XML המייצג ספר טלפונים :

- ```
<?xml version="1.0"?>
<Phone-Book>
  <entry>
    <name> Avi Cohen </name>
    <tel preferred="true"> 03-9242356 </tel>
    <tel> 054-6325843 </tel>
  </entry>
  <entry>
    <name> Beni Levi </name>
    <tel> 052-3458974 </tel>
  </entry>
</Phone-Book>
```

XML - הגדרת סכמה - DTD - תחביר

- הפורמט של הגדרת האלמנטים הוא :
- `<!ELEMENT name (content)>`
- הגדרת האלמנט Phone-Book תראה כך :
- `<!ELEMENT Phone-Book (entry+)>`
- שם האלמנט הוא Phone-Book.
- הוא מכיל (כבנים) מספר מופעים של האלמנט entry.
- אבל... כמה בנים?

XML - הגדרת סכמה - DTD - תחביר

- סימנים מיוחדים :
 - + - אלמנט בעל מופע אחד לפחות
 - * - אלמנט בעל 0 מופעים או יותר
 - ? - אלמנט בעל מופע אחד לכל היותר.
- כיצד ייראה האלמנט ?entry
- `<!ELEMENT entry (name, tel+)>`
 - מה נעשה אם נרצה לאפשר גם הכנסת אנשים ללא מס' טלפון?
- `<!ELEMENT entry (name, tel*)>`
 - מה מחייבת אותנו ההגדרה הבאה :
- `<!ELEMENT elem1 (elem2?, elem3)+>`
 - בכל מופע של אלמנט 1 יהיה לפחות בן אחד של אלמנט 3, כאשר אם יש בן של אלמנט 2 – יופיע אחריו מופע של אלמנט 3.

XML - הגדרת סכמה - DTD - תחביר

- האלמנט name ייראה כך :
- `<!ELEMENT name (#PCDATA)>`
- ניתן לאפשר למשתמש לבחור בין מספר אלמנטים ע"י שימוש ב-"|"
- `<!ELEMENT elem1 (elem2| elem3)>`
- במקרה של בחירה –חייבים לאפשר מצב שבו המפענח יודע איזה חלק של המודל הוא בוחר... כלומר :
- ~~`<!ELEMENT elem1 ((elem2, elem3) | (elem2, elem4))>`~~
- `<!ELEMENT elem1 (elem2, (elem3 | elem4))>`

XML - הגדרת סכמה - DTD - תחביר

- נניח שנרצה לאפשר טיפה גמישות במסמך...
נניח שנרצה להגדיר שם גם בצורה הבאה:

- ```
<name>
 <fname>Avi</fname>
 <lname>Cohem</lname>
</name>
```

- נשתמש ב- Mixed Content Element

- ```
<!ELEMENT name (#PCDATA | fname | lname)*>
```

 - נוצרת לנו כאן חופשיות יתר. (ניתן להגדיר כמה שמות שנרצה)
אם אנחנו רוצים להשתמש בתוכן מעורב (טקסט + אלמנטים) אנחנו לא יכולים לקבוע את הסדר או את מספר החזרות.
 - אם אנחנו רוצים למנוע את עודף החופשיות הזו – נאלץ לבחור שיטה אחת מהשתיים.

XML - הגדרת סכמה - DTD - תחביר

- כיצד ייראה האלמנט `?tel`
- `<!ELEMENT tel (#PCDATA)>`
- אלמנט ללא בנים ייראה בצורה הבאה :
- `<!ELEMENT noSon (EMPTY)>`
- קיימת גם האפשרות הבאה :
- `<!ELEMENT elem (ANY)>`
- תחת `ANY` נכנס כל טקסט או אלמנט הקיים במסמך.
ברוב המקרים של המקרים לא נשתמש באפשרות זו כי מטרתנו היא בעצם להגביל את המסמך ולא להיפך.

XML - הגדרת סכמה - DTD - תחביר

- מבנה הצהרה כללית על תכונה :
- `<!ATTLIST name att1 type1 def1
att2 type2 def2 ... >`
 - name – שם התג
 - att – שם התכונה
 - type – סוג התכונה (בד"כ – CDATA – מחרוזת תווים.
לא קשור לקטעי CDATA שראינו בקבצי XML!)
 - def – ברירת המחדל
- הצהרה על התכונה preferred במסמך שלנו תיראה כך :
- `<!ATTLIST tel preferred (true | false) "false">`

XML - הגדרת סכמה - DTD - תחביר

- אפשרויות נוספות עבור ערך ברירת המחדל:
 - #REQUIRED – מחייב הכנסת ערך כלשהו.
 - #IMPLIED – כאשר אנחנו לא רוצים לחייב את המשתמש להכניס ערך כלשהו ואין לנו ערך ברירת מחדל משלנו.
 - "value" #FIXED – הערך value המופיע חייב להיות הערך של אותה התכונה. (בד"כ נקבל אזהרה אם מופיע ערך אחר)
 - לדוגמא – אם יש לנו אלמנט של "מחיר" אשר תמיד יהיה בדולרים, אזי ניתן להגדיר:
 - `<!ELEMENT price #PCDATA>`
`<!ATTLIST price currency CDATA #FIXED "usd">`
 - במצב זה כאשר התוכנית תקרא `<price>19.99</price>` היא בפועל תציג:
`<price currency="usd">19.99</price>`

XML-הגדרת סכמה - DTD - תחביר

- אז איך נראה מסמך ה-DTD שלנו?
- `<!ELEMENT Phone-Book (entry+)>`
`<!ELEMENT entry (name, tel*)>`
`<!ELEMENT name (#PCDATA)>`
`<!ELEMENT tel (#PCDATA)>`
`<!ATTLIST tel preferred (true | false) "false">`
- שורות אלו יופיעו בקובץ נפרד שנטען בעזרת פקודת SYSTEM, או שיופיע בחלק ה- **declarations** של המסמך.

XML - הגדרת סכמה - DTD - תחביר

- מזהים ב-XML.
לעיתים נרצה ליצור קשרים בין אלמנטים שונים.
שלב ראשון – הוספת תכונה שתהווה "מפתח" עבור האלמנט
(במטרה שנוכל להתייחס לאותו מפתח):
• `<!ATTLIST entry taz ID #REQUIRED>`
- בעקבות העובדה שמדובר ב"מפתח" – הערך שיופיע עבור אלמנט זה
לא יכול לחזור על עצמו פעמיים באותו המסמך!
- ערכי ברירת המחדל היחידים האפשריים עבור ID הם
`#REQUIRED` , `#IMPLIED`
• (למה `#FIXED` אינו אפשרי?)

XML-הגדרת סכמה - DTD - תחביר

- שלב שני – יש לתת לאלמנט מסוים תכונה שערכה הוא IDREF, כלומר – תכונה שערכה צריך להיות אחד מה-ID הקיימים במסמך.
- `<!ATTLIST entry father IDREF #REQUIRED>`
- נוכל לאפשר רשימה של הפניות בכך שניתן לתכונה את הערך IDREFS (הערכים ב-XML צריכים להיות מופרדים ע"י רווחים)
- גם לתכונות אלו ערכי ברירת המחדל היחידים האפשריים הם `#REQUIRED` או `#IMPLIED`

XML - הגדרת סכמה - DTD - תחביר

• דוגמא לקובץ DTD עם הפניות:

- `<!ELEMENT family (person)+>`
`<!ELEMENT person (name) >`
`<!ATTLIST person`
 `idnum ID #REQUIRED`
 `gender (male | female) #REQUIRED`
 `father IDREF #IMPLIED`
 `mother IDREF #IMPLIED`
 `children IDREFS #IMPLIED >`
`<!ELEMENT name (#PCDATA)>`

• למה אין `#REQUIRED` עבור התכונות `father / mother`?

XML - הגדרת סכמה - DTD - תחביר

• כך ייראה קובץ XML מתאים :

```
<?xml version="1.0"?>
<!DOCTYPE family SYSTEM "Family.dtd">
<family>
  <person idnum="T11" gender="male" children="T13 T14 T15">
    <name>11</name></person>
  <person idnum="T12" gender="female" children="T13 T14 T15">
    <name>12</name></person>
  <person idnum="T13" gender="male" father="T11" mother="T12">
    <name>13</name></person>
  <person idnum="T14" gender="male" father="T11" mother="T12">
    <name>14</name></person>
  <person idnum="T15" gender="female" father="T11" mother="T12" children="T33">
    <name>15</name></person>
  <person idnum="T21" gender="male" children="T23">
    <name>21</name></person>
  <person idnum="T22" gender="female" children="T23">
    <name>22</name></person>
  <person idnum="T23" gender="male" father="T21" mother="T22" children="T33">
    <name>23</name></person>
  <person idnum="T33" gender="female" father="T23" mother="T15">
    <name>33</name></person>
</family>
```

XML - הגדרת סכמה - DTD - תחביר

- יישויות של XML. משמש בד"כ כסוג של מאקרו.
2 סוגי יישויות ב-XML:
- יישויות כלליות – עבור ב-DATA במסמך XML.
- ראינו מספר ישויות כאלה מקודם.
השימוש בישויות אלו הוא בד"כ בקטעי ה-PCDATA.
- יישויות פרמטר – עבור תיאור המידע בקובץ ה-DTD.
- נשתמש בהן בצורה דומה (לא זהה!) לשימוש של ישויות כלליות, אך השימוש בהן יתבצע בחלק הגדרת ה-DTD בלבד!

XML - הגדרת סכמה - DTD - תחביר

- יישות כלליות תוגדר בצורה הבאה :
- `<!ENTITY name 'value'>`
 - חשוב : לא לשכוח את הגרש שמופיע לפני ואחרי הערך!
 - דוגמא :
- `<!ENTITY eliCohen`
`'<entry>`
`<fname>Cohen</fname> <lname>Eli</lname>`
`<tel>03-9485246</tel>`
`</entry>'>`
 - השימוש בקובץ XML יהיה כך :
- `<Phone-Book>`
`&eliCohen;`
`</PhoneBook>`
 - חשוב : לא לשכוח את הנקודה-פסיק המופיע אחרי הערך!
 - כמובן שניתן להגדיר גם ישויות קצרות יותר...
 - ניתן גם לייבא ישויות מקבצים שונים :
- `<!ENTITY name SYSTEM "file.ent">`

XML - הגדרת סכמה - DTD - תחביר

• דוגמאות לשימוש בישויות פרמטר :

- ```
<!ENTITY % p "(#PCDATA)">
<!ELEMENT student (id, firstname, lastname, (subject)*)>
<!ELEMENT id %p;>
<!ELEMENT firstname %p;>
<!ELEMENT lastname %p;>
<!ELEMENT subject %p;>
```
- ```
<!ENTITY % inline "cite | emphasis | br">
<!ELEMENT note (#PCDATA | %inline;)*>
```

• דגש חשוב!

- כאשר הישות מוגדרת בתוך ה-DTD - השימוש בישויות פרמטר אפשרי רק מחוץ להגדרות האלמנטים. שימוש בצורה שהנכם רואים כאן (השימוש בישויות מתבצע בתוך הצהרת האלמנט) אפשרי אך ורק ע"י הגדרת הישויות בקובץ חיצוני וייבוא הקובץ אל המסמך (ע"י SYSTEM)

XML - הגדרת סכמה - DTD - תחביר

- נניח שהגדרנו קובץ עם ישויות כלליות עבור מדינות:

- `<!ENTITY be "belgium">`
`<!ENTITY de "Germany">`
`<!ENTITY il "Israel">`
`<!ENTITY it "Italy">`
`<!ENTITY uk "United Kingdom">`
`<!ENTITY us "United States">`
`<!-- and some more -->`

- ברור שהכנת קובץ כזה היא משהו ארוך ואנחנו רוצים להכין את הרשימה פעם אחת ולהשתמש בה בהרבה מקומות...

XML-הגדרת סכמה - DTD - תחביר

- לכן, נשתמש בשיטה של יישויות פרמטר ע"מ להגדיר את הישויות הכלליות האלה במסמכים השונים:
- ```
<?xml version="1.0"?>
<!DOCTYPE address SYSTEM "address.dtd" [
<!ENTITY % countries SYSTEM "countries.ent">
%countries;
]>
<address>
 <street> ... </street>
 ...
 <country>&us;</country>
</address>
```

הגדרת היישות הפרמטרית countries כך שהערך שלה יהיה המידע שמופיע בקובץ

שימוש ביישות הפרמטרית שבעצם מייבא את הקובץ לתוך ה-DTD של המסמך שלנו.

# XML-הגדרת סכמה - DTD - תחביר

- בשיטה זו אנחנו יכולים להגדיר חלקים מהגדרות ה-DTD בקבצים שונים ולהשתמש בהם במידת הצורך...

