

XML = EXTENSIBLE MARKUP LANGUAGE

[01 XML]

קובץ המכיל מידע שהוא "self-describing" – דהיינו לא צריך שום פרוש מעבר לכתוב בו. XML אילו טאגים של מידע על מידע (metadata)

הקובץ מכיל שני רכיבים:

- אלמנטים או אלמנטים עם ערכים - element and values
- תכונות של אלמנטים - attributes

כל האלמנטים מתחילים בטאג <element_name> ונסגרים בטאג-סוגר </element_name> ניתן להוסיף בין הטאגים ערכים או עוד אלמנטים. חשוב לזכור שכל קובץ חייב רק אלמנט שורש אחד – root element – ניתן לכתוב את גירסת הקובץ בראשו (בד"כ כותבים)

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<element_name>value</element_name>
```

```
<element_name element_attribute="attribute_value">value</element_name>
```

XDOCUMENT XELEMENT XATTRIBUTE

[04 XElement XAttribute]

בעזרת שלושת האובייקטים הבאים ניתן ליצור ולקרוא קבצי XML באופן הבא:

יצירה:

- XmlDocument

מכיל את כל המסמך כולל גירסה ו ROOT נשתמש בו בכדי לטעון ולכתוב את ה XML

```
XmlDocument personsXml = XmlDocument.Load("Perons.xml");
```

```
...
```

```
personsXml.Save("Perons.xml");
```

- XElement

מכיל מידע אודות אלמנט וכל התוכן שלו, דהיינו ערך, אלמנטים אחרים ותכונות.

```
new XElement("person",  
    new XElement("Name", "chipopo"),  
    new XElement("Address", "street one 7"),
```

```
new XAttribute("Age", 32)),
```

- XAttribute

בקוד מעל רואים יצירת תכונה חדשה עבור אלמנט ה person

את כל האלמנטים ניתן לאגד לתוך אלמנט השורש אשר נמצא ב XDocument :

```
XDocument persons = new XDocument(  
    new XElement("persons",  
        new XElement("person",  
            new XElement("Name", "chipopo"),  
            new XElement("Address", "street one 7"),  
            new XAttribute("Age", 32)),  
        new XElement("person",  
            new XElement("Name", "yakov"), ...
```

קריאה\שליפת נתונים:

- XDocument

ישנו המאפיין ROOT בעזרתו שולפים את השורש

```
XDocument personsXml = XDocument.Load("persons.xml");  
XElement root = personsXml.Root;
```

- XElement

מכיל שתי שיטות: לשליפת אלמנט בודד מתחתיו או לשליפת מספר אלמנטים שנימצאים מתחתיו.
כל אחת מהשיטות יכולה לקבל את שם האלמנט המסויים שאותו מחפשים.
מספר דוגמאות:

```
IEnumerable<XElement> persons = root.Elements();  
IEnumerable<XElement> names = persons.Elements("Name");  
person.Element("Name").Value
```

בנוסף ישנן שתי שיטות עבור שליפה של התכונות.
לדוגמה:

```
grade.Attribute("course").Value  
person.Attributes()
```

- XAttribute

באותו אופן כמו ל XElement גם לתכונה יש Value אותו ניתן לשלוף

שימו לב שניתן ממש לערוך את ה XML ע"י לולאות ה foreach. אבל רק את המאפיינים של אותו אובייקט עליו הלולאה רצה. את האובייקט עצמו לא ניתן לשנות כי הוא רץ ע"י ממשק IEnumerable

```
foreach (var grade in grades)
{
    //grade = 100; //ERROR because the ienumerator interface is read only
    (but not for inner properties!)
    Console.WriteLine(grade.Value);
}

foreach (var item in students)
{
    foreach (var g in item.Elements("Grade"))
    {
        g.Value = "100";
    }
}
```

XPATH

[07 XPath]

שיטה שבה ה XML הופך לעץ של אובייקטים ועליו ניתן לתשאל באמצעות מחרוזות שאילתה. למשל:

```
doc = new XmlDocument();
...
```

```
XmlNodeList list = doc.SelectNodes(XPathExpression);
foreach (XmlNode node in list)
{
    Console.WriteLine("Node Name : " + node.Name + " " + "Node
value : " +node.InnerText);
}
```

כאשר ה XPathExpression מכיל שאילתה עם פורמט מיוחד :

```
XPathExpression = "/books/book/price"
```

יתן את מחירי הספרים

```
XPathExpression = "/books/book/authors/author/@lname"
```

יתן את שמות המשפחה של הסופרים – אילו תכונות

```
XPathExpression = "/books/book[price>20]/title"
```

יתן את כותרות הספרים שמחירים גדול מ20