Με λίγα λόγια τι κάνει το κάθε αρχείο στον φάκελο `xml-php-lesson/`:

`index.php`

- Είναι η **κεντρική σελίδα** (μενού).
- Περιέχει συνδέσμους (links) για να ανοίξεις τα υπόλοιπα scripts.
- Ξεκινάς από εδώ → `http://localhost/xml-php-lesson/index.php`

`create_xml.php`

- Δημιουργεί (ή αντικαθιστά) το αρχείο `books.xml`.
- Περιέχει 3 βιβλία ως δείγμα και τα αποθηκεύει σε μορφή XML.
- Τρέχεις το αρχείο **μία φορά στην αρχή**, για να έχεις δεδομένα.

`books.xml`

- Το XML αρχείο που περιέχει τα δεδομένα των βιβλίων.
- Δημιουργείται αυτόματα από το `create_xml.php`.
- Χρησιμοποιείται από τα υπόλοιπα αρχεία για εμφάνιση.

`read_simplexml.php`

- Διαβάζει το `books.xml` με την απλή μέθοδο **SimpleXML**.
- Εμφανίζει τα βιβλία (τίτλος, συγγραφέας, έτος).
- Ιδανικό για αρχάριους.

`read_dom.php`

- Διαβάζει το `books.xml` με τη μέθοδο **DOMDocument**.
- Κάνει το ίδιο πράγμα με το `read_simplexml.php`, αλλά με πιο «τεχνικό» τρόπο.
- Κατάλληλο για πιο προχωρημένη επεξεργασία XML.

`ασκηση1.php`

- Μια άσκηση-παράδειγμα που **χρησιμοποιεί SimpleXML** για εμφάνιση βιβλίων.
- Δείχνει τα ίδια δεδομένα με λίγο διαφορετικό στυλ.
- Χρησιμοποιείται για πειραματισμό από τους μαθητές (π.χ. να αλλάξουν την εμφάνιση, να φιλτράρουν κ.λπ.).

🔁 Πώς τα χρησιμοποιείς:

1. Τρέχεις **`create_xml.php`** για να δημιουργηθεί το `books.xml`
2. Μετά τρέχεις:
    - `read_simplexml.php` ή
    - `read_dom.php` ή
    - `ασκηση1.php`
3. Όλα παίρνουν δεδομένα από το `books.xml`

ΕΞΗΓΗΣΗ ΕΝΤΟΛΩΝ

Το `DOMDocument` είναι μια κλάση στην PHP (και γενικά σχετίζεται με την τεχνολογία XML/HTML DOM) που επιτρέπει την ανάγνωση, επεξεργασία και δημιουργία εγγράφων XML ή HTML χρησιμοποιώντας το **DOM API** (Document Object Model).

Αναλυτικά:

- **DOM (Document Object Model):** Είναι ένα πρότυπο που αναπαριστά ένα έγγραφο (όπως XML ή HTML) ως δέντρο κόμβων (nodes), όπου κάθε στοιχείο, γνώρισμα, κείμενο κ.λπ., είναι ένας κόμβος.
- **DOMDocument:** Είναι η κύρια κλάση για την εργασία με DOM στην PHP.

Συνήθεις χρήσεις:

Φόρτωση XML ή HTML:

Η συγκεκριμένη `foreach` σε PHP χρησιμοποιείται για να **διαβάσει δεδομένα από ένα XML αρχείο** που έχει ήδη φορτωθεί ως `SimpleXMLElement` αντικείμενο (και **όχι μέσω DOMDocument** σε αυτή την περίπτωση).

Ας το εξηγήσουμε βήμα-βήμα:

```
foreach ($xml->book as $book) {
  echo "Τίτλος: " . $book->title . "<br>";
  echo "Συγγραφέας: " . $book->author . "<br>";
  echo "Έτος: " . $book->year . "<br><br>";
}
```

1. `$dom = new DOMDocument();`

- Δημιουργεί ένα **νέο αντικείμενο DOMDocument**.
- Αυτό είναι μια ενσωματωμένη κλάση της PHP που χρησιμοποιείται για να διαχειρίζεσαι XML ή HTML έγγραφα με το **DOM (Document Object Model)**.
- Στην ουσία, σου επιτρέπει να διαβάσεις, να τροποποιήσεις ή να δημιουργήσεις XML/HTML.

◇ 2. `$dom->load("books.xml");`

- Φορτώνει το αρχείο `books.xml` και το διαβάζει μέσα στο αντικείμενο DOM.
- Το `DOMDocument` αναλύει (parses) τη δομή του XML, δημιουργώντας ένα **δέντρο κόμβων** (DOM tree), όπου κάθε στοιχείο (tag), γνώρισμα (attribute) και κείμενο είναι κόμβος (node).

```
$dom = new DOMDocument();
$dom->load("books.xml");
```

Μπορείς να έχεις πρόσβαση στα στοιχεία του XML με μεθόδους όπως:

```
$books = $dom->getElementsByTagName("book");

foreach ($books as $book) {
   $title = $book->getElementsByTagName("title")->item(0)->nodeValue;
   $author = $book->getElementsByTagName("author")->item(0)->nodeValue;
```

```
    echo "Τίτλος: $title<br>";
    echo "Συγγραφέας: $author<br><br>";
}
```

☑ Προσθέτει έναν κόμβο ως «παιδί» σε έναν άλλον κόμβο.

Πιο συγκεκριμένα:

- **`$book`**: Είναι ένας κόμβος DOM (πιθανόν τύπου `<book>`).
- **`$title`**: Είναι επίσης ένας κόμβος DOM (π.χ. ένας `<title>` κόμβος).
- **`appendChild()`**: Είναι μέθοδος της DOM API που προσθέτει τον `$title` μέσα στον `$book`, δηλαδή τον κάνει **παιδικό κόμβο** του.

```
$book->appendChild($title);
```

☑ «Για κάθε `<book>` στοιχείο μέσα στο XML αντικείμενο `$xml`, εκτέλεσε...»

Αυτός ο κώδικας χρησιμοποιεί τη **SimpleXML** της PHP για να περιηγηθεί εύκολα σε ένα XML αρχείο.

🔍 Αναλυτικά:

- **`$xml`**: Είναι ένα αντικείμενο τύπου `SimpleXMLElement`, που φορτώθηκε με `simplexml_load_file()` ή `simplexml_load_string()`.
- **`$xml->book`**: Πρόσβαση σε όλα τα στοιχεία `<book>` που βρίσκονται **μέσα στο ριζικό στοιχείο**.
- **`foreach (...)`**: Επαναλαμβάνει τον κώδικα μέσα στα `{}` για κάθε `<book>` που υπάρχει στο XML.

```
foreach ($xml->book as $book)
```

Παίρνει το περιεχόμενο (κείμενο) του κόμβου `<title>` από ένα στοιχείο `<book>` όταν δουλεύουμε με `DOMDocument` στην PHP.

🔍 Αναλυτικά:

1. **`$book`**: Ένα στοιχείο DOM (π.χ. ένα `<book>` κόμβος που έχει φορτωθεί με `DOMDocument`).
2. **`getElementsByTagName("title")`**: Ψάχνει και επιστρέφει **όλους τους `<title>` κόμβους** μέσα στο `$book`. Αυτό επιστρέφει ένα αντικείμενο τύπου `DOMNodeList`.
3. **`item(0)`**: Παίρνει **τον πρώτο κόμβο** από αυτή τη λίστα (τον πρώτο `<title>`).
4. **`nodeValue`**: Παίρνει το **κείμενο που περιέχει** αυτός ο `<title>` κόμβος.

$title = $book→getElementsByTagName("title")→item(0)→nodeValue;

Τα αρχεία `read_simplexml.php` και `read_dom.php` κάνουν **παρόμοια δουλειά** — διαβάζουν XML — αλλά χρησιμοποιούν **διαφορετικές προσεγγίσεις** και **διαφορετικά API της PHP**.

Ας δούμε αναλυτικά τη **διαφορά ανάμεσα στο SimpleXML και το DOMDocument**:

1. `read_simplexml.php` → Διαβάζει XML με SimpleXML

☑ Χαρακτηριστικά:

- Πολύ **εύκολο στη χρήση**.
- Ιδανικό για **απλό και μικρό XML**.
- XML δεδομένα προσπελάζονται σαν **αντικείμενα PHP** (όπως ιδιότητες).

💻 Παράδειγμα:

```
$xml = simplexml_load_file("books.xml");

foreach ($xml->book as $book) {
   echo "Τίτλος: " . $book->title . "<br>";
   echo "Συγγραφέας: " . $book->author . "<br>";
```

```
}
```

Πλεονεκτήματα:

- Καθαρός και απλός κώδικας.
- Ιδανικό για διάβασμα και απλή επεξεργασία.

❌ Μειονεκτήματα:

- Δεν υποστηρίζει καλά πολύπλοκες δομές.
- Περιορισμένος έλεγχος.

📄 2. `read_dom.php` → Διαβάζει XML με DOMDocument

☑ Χαρακτηριστικά:

- Πιο **ισχυρό** εργαλείο.
- Χρήσιμο για **περίπλοκα XML**, επεξεργασία κόμβων, μεταβολές, XPath κ.λπ.
- XML δεδομένα προσπελάζονται μέσω **DOM (Document Object Model)**.

💻 Παράδειγμα:

```
$dom = new DOMDocument();
$dom->load("books.xml");

$books = $dom->getElementsByTagName("book");

foreach ($books as $book) {
   $title = $book->getElementsByTagName("title")->item(0)->nodeValue;
   $author = $book->getElementsByTagName("author")->item(0)->nodeValue;

   echo "Τίτλος: $title<br>";
   echo "Συγγραφέας: $author<br>";
}
```

🎯 Πλεονεκτήματα:

Πλήρης έλεγχος της XML δομής.

Υποστήριξη για XPath.

Μπορεί να τροποποιήσει το XML (π.χ. προσθήκη/διαγραφή κόμβων).

❌ Μειονεκτήματα:

Περισσότερος και πιο "βαρύς" κώδικας.

Όχι τόσο απλός όσο το SimpleXML.

Τελικό Συμπέρασμα:

Χαρακτηριστικό	**SimpleXML (`read_simplexml.php`)**	**DOMDocument (`read_dom.php`)**
Ευκολία	☑ Πολύ εύκολο	❌ Πιο περίπλοκο
Ευελιξία	❌ Περιορισμένη	☑ Υψηλή
Απόδοση	☑ Γρήγορο για μικρά αρχεία	❌ Βαρύτερο
Τροποποίηση κόμβων	❌ Περιορισμένη	☑ Ναι
Κατάλληλο για	Απλό διάβασμα XML	Επεξεργασία, XPath, μεγάλα XML