

DOM

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος
Επιμέλεια διαφανειών: Π. Αγγελάτος

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών



Στόχος της ώρας

- Εξοικείωση με το DOM
- Δέντρο DOM:
 - Διάσχιση
 - Τροποποίηση
- Παραδείγματα

Τι είναι το DOM

- Document Object Model
- Στάνταρ του W3C
- Ορίζει έναν τρόπο να έχουμε πρόσβαση στο HTML
- Είναι ανεξάρτητο γλώσσας προγραμματισμού
 - libxml2 για C, C++, Python κ.α.
- Δεν είναι απαραίτητο για την εμφάνιση της σελίδας

Τι είναι το DOM

- Μία αναπαράσταση της σελίδας
- Μας προσφέρει μεθόδους για
 - Διάβασμα δεδομένων του εγγράφου
 - Αλλαγή δεδομένων του εγγράφου
 - Αλλαγή δομής του εγγράφου

DOM δέντρο

- Κατασκευάζεται όταν φορτώνει η σελίδα
- Δε μπορούμε να το χρησιμοποιήσουμε πριν τελειώσει το φόρτωμα της σελίδας
- Αποτελείται από DOM Nodes

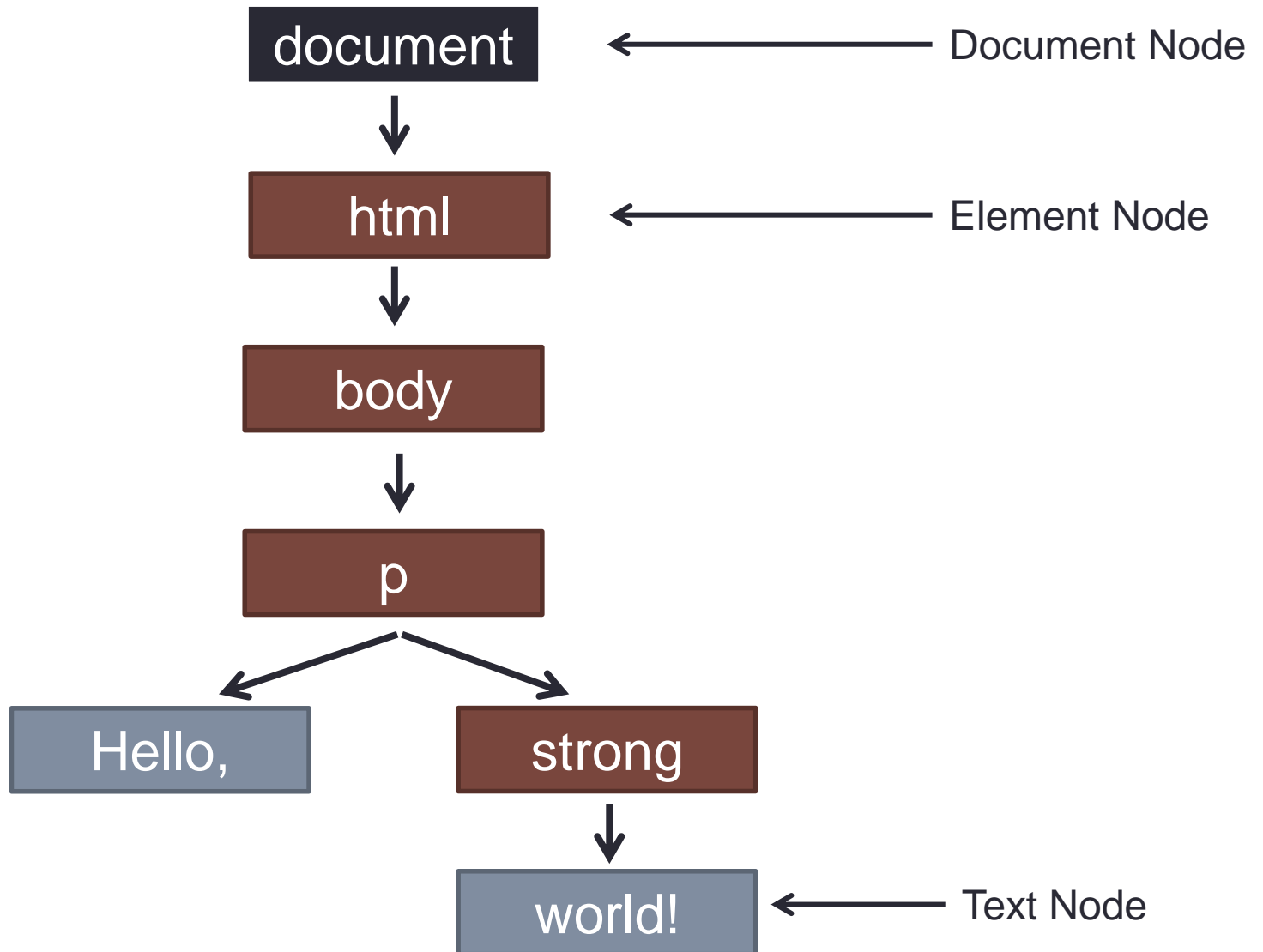
DOM δέντρο

- Το DOM δέντρο έχει περισσότερα στοιχεία από το HTML δέντρο
- Ένα στοιχείο μπορεί να περιέχει
 - Άλλα στοιχεία
 - Κομμάτια κειμένου

HTML δέντρο

```
<html>  
  <body>  
    <p>  
      Hello, <strong>world!</strong>  
    </p>  
  </body>  
</html>
```

DOM δέντρο



Αναπαράσταση DOM

- Κάθε στοιχείο HTML αναπαριστάται με ένα **μοναδικό** αντικείμενο
- Υπάρχουν αντικείμενα DOM που δεν αντιστοιχούν σε στοιχείο HTML

Nodes

- Κάθε στοιχείο HTML είναι ένα Element Node
- Κάθε κομμάτι κειμένου στο HTML είναι ένα TextNode
- Τα HTML σχόλια είναι Comment Nodes

Text Nodes

- Ένα κομμάτι κειμένου του εγγράφου

Element Node



- `<p>This is a text node</p>`



Text Node

Element Node

Element Node



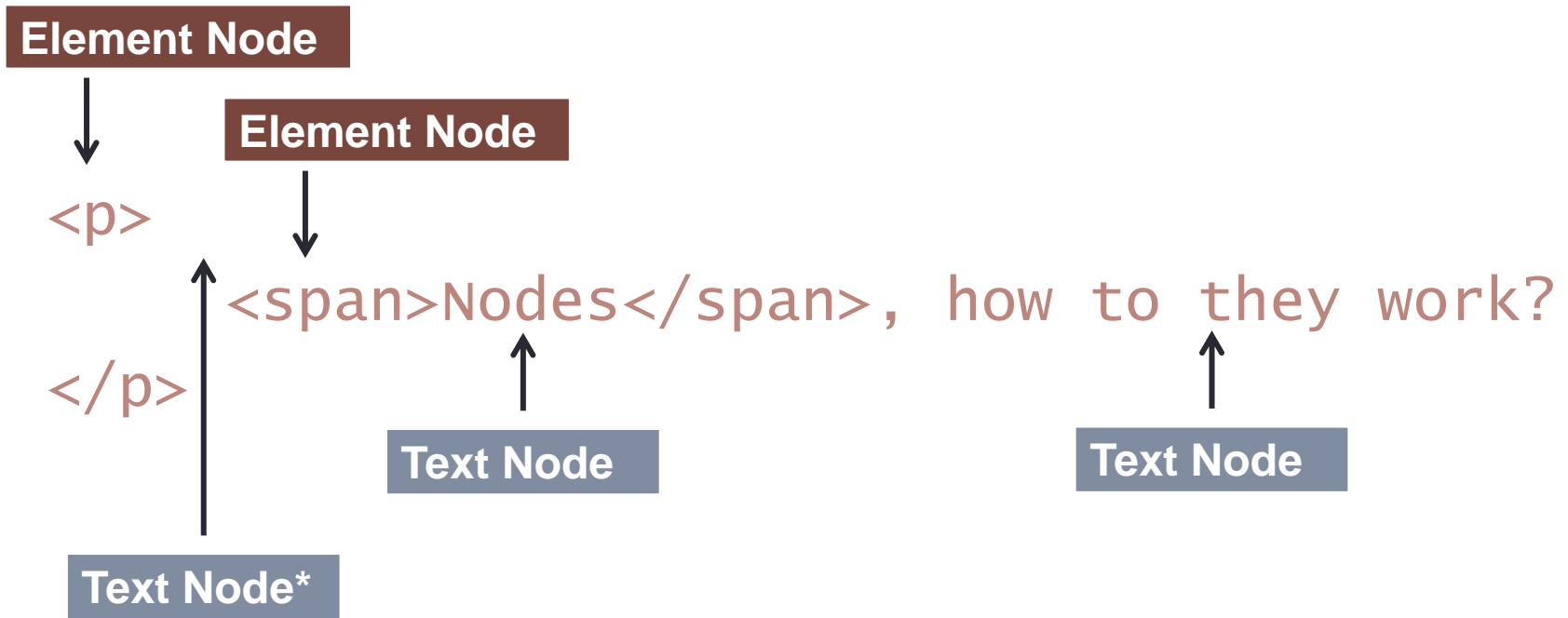
- `<p>Hello, world!</p>`



Text Node

Text Node

Τι nodes βλέπτε;



*Περιέχει μόνο ένα space!

Ιδιότητες των Node

- `x.nodeName`
- Περιέχει το όνομα του Node
 - Για Text Nodes είναι `'#text'`
 - Για Element Nodes είναι το όνομα της ετικέτας
- Η ετικέτα είναι ή σε μικρά ή σε κεφαλαία ανάλογα με τον browser

Ιδιότητες των Node

- `x.nodeValue`
- Περιέχει την τιμή του Node
 - Για Text Nodes είναι το String με το κείμενο του Node
 - Για Element Nodes είναι null

Διάσχιση του DOM

- Σε κάθε Node έχουμε μεθόδους και ιδιότητες για να διασχίσουμε το DOM δέντρο
- Με αυτό τον τρόπο βρίσκουμε το στοιχείο που θέλουμε
- Ως βάση έχουμε το `document.body`

Παιδιά

- `x.childNodes`
- Είναι ένας πίνακας
- Περιέχει όλα τα παιδιά του στοιχείου `x`
- Το πρώτο παιδί είναι το στοιχείο 0
- Το μήκος του είναι `x.childNodes.length`

Παιδιά

```
<body>
```

```
    Hello, <strong>world</strong>
```

```
</body>
```

```
document.body.childNodes[ 0 ] → “Hello, “
```

```
document.body.childNodes[ 1 ] → <strong>
```

Ειδικά παιδιά

- `x.firstChild`
- Περιέχει το πρώτο παιδί του `x` ή `null` αν δεν έχει παιδιά
- `x.lastChild`
- Περιέχει το τελευταίο παιδί του `x` ή `null` αν δεν έχει παιδιά

Ειδικά Παιδιά

```
<body>  
    Hello, <strong>world</strong>  
</body>
```

`document.body.firstChild` → “Hello, “

`document.body.lastChild` → ``

`document.body.lastChild.firstChild` → “world”

Πατέρας

- `x.parentNode`
- Περιέχει τον πατέρα του `x`
- Είναι σίγουρα Element Node

Αδέρφια

- `x.nextSibling`
- Περιέχει το επόμενο Node από τα παιδιά του πατέρα του `x`

Αδέλφια

- `x.previousSibling`
- Περιέχει το προηγούμενο Node από τα παιδιά του πατέρα του `x`

Απευθείας μετάβαση

- Μέθοδοι σε κάθε Element Node
- Αντί για μεγάλες δύσχρηστες αλυσίδες από `childNodes[
].childNodes`
- Πηγαίνουμε κατευθείαν στο στοιχείο ανεξάρτητα από τη δομή του δέντρου

Απευθείας μετάβαση

- `x.getElementById(id)`
- Επιστρέφει το στοιχείο που έχει *id* *id* και είναι απόγονος του *x*
- Μειονέκτημα
 - Σε κάθε στοιχείο που μας ενδιαφέρει χρειάζεται ξεχωριστό *id*

Απευθείας μετάβαση

- `x.getElementsByTagName(ετικέτα)`
- Επιστρέφει ένα **πίνακα** με τους απογόνους του `x` που έχουν ετικέτα *ετικέτα*

Ιδιότητες HTML

- `x.getAttribute(ιδιότητα)`
- Μέθοδος σε κάθε DOM ElementNode
- Επιστρέφει την τιμή της ιδιότητας σε string

Ιδιότητες HTML

- `x.setAttribute(ιδιότητα, τιμή)`
- Μέθοδος σε κάθε DOM ElementNode
- Θέτει την ιδιότητα και την τιμή της στο Node που αναφέρεται

Τιμές πεδίου

- `x.value`
- Περιέχει την τιμή της ιδιότητας `value`
- Είναι ορισμένο μόνο σε ετικέτες
 - `input`
 - `textarea`
 - `select`

CSS μέσω του DOM

- `x.style`
- Αντιπροσωπεύει τους CSS κανόνες που είναι inline στο στοιχείο
- Είναι ένα αντικείμενο
- Περιέχει κάθε CSS ιδιότητα

CSS μέσω του DOM

- Οι ιδιότητες του `.style` έχουν ίδιο όνομα με τις αντίστοιχες CSS με μία διαφορά
- Οι παύλες παλείπονται και το επόμενο γράμμα είναι κεφαλαίο
- `background-color` → `x.style.backgroundColor`
- `border` → `x.style.border`
- `list-style-type` → `x.style.listStyleType`

Προσοχή!

- Η ιδιότητα `style` δε περιέχει CSS κανόνες που προκύπτουν από εξωτερικά CSS
- Οι CSS κανόνες του `style` έχουν την μέγιστη και ίδια ειδικότητα με τους `inline` κανόνες
- Η ιδιότητα `style` δε μπορεί να τεθεί ένα `string`
- `x.style = "color: black"`

Δημιουργία Node

- Δημιουργούμε Node με την συνάρτηση `document.createElement(ετικέτα)`
- Φτιάχνει μία ετικέτα και μας δίνει την DOM αναπαράστασή της
- Δεν ανήκει στο DOM έγγραφο ακόμα
- Το `parentNode` του είναι ορισμένο σε null

Δημιουργία Node

```
var p = document.createElement( 'p' );
```

Το `p` περιέχει την DOM αναπαράσταση μία παραγράφου

Δημιουργία TextNode

- `document.createTextNode(δεδομένα);`
- Δημιουργεί ένα `TextNode` που περιέχει τα *δεδομένα*

Δημιουργία textNode

- `var data = document.createTextNode('Test data');`
- Το data περιέχει τη DOM αναπαράσταση του κειμένου 'Test data'

Πρόσθεση παιδιών

- `x.appendChild(y)`
- Προσθέτει το `y` ως τελευταίο παιδί του `x`

Πρόσθεση παιδιών

- `<p>Test data</p>`
- `var p = document.createElement('p');`
- `var data = document.createTextNode('Test data');`
- `p.appendChild(data);`
- `document.body.appendChild(p);`

Πρόσθεση παιδιών

- `x.insertBefore(y, z);`
- Βάζει το `y` ως παιδί του `x` στη θέση πριν το παιδί `z`
- `x.insertBefore(y, x.firstChild);`
- Βάζει το `y` ως πρώτο παιδί του `x`

Αντικατάσταση παιδιών

- `x.replaceChild(νέο, παλιό);`
- Ορίζει το Node *νέο* να είναι παιδί του *x* στη θέση του παλιού

Διαγραφή παιδιών

- `x.removechild(y);`
- Διαγράφει το στοιχείο `y` από παιδί του `x`

Μάθαμε

- Χρήση DOM για:
 - Πρόσβαση (ανάγνωση) HTML σελίδας
 - Προσθήκη κόμβων
 - Διαγραφή κόμβων
 - Επεξεργασία ιδιοτήτων και περιεχομένου

Τελική εργασία

- Ομάδες 3+ ατόμων
- Δήλωση ομάδων στο web-seminar@softlab.ntua.gr
- Μέχρι 8 Ιανουαρίου
 - Στείλτε μας μία έκδοση που να δουλεύει
- Μέχρι αρχές Φεβρουαρίου
 - Θα έχετε την ευκαιρία να την βελτιώσετε
- Πλήρης web υπηρεσία
- Μας στέλνετε
 - Κώδικα + URL όπου μπορούμε να την δούμε
- Μοιραστείτε στο forum επιλογές hosting
- Αρχές Φεβρουαρίου οι καλύτερες θα παρουσιαστούν

Καλά χριστούγεννα

