

# ΑΝΑΠΤΥΞΗ ΤΗΛΕΜΑΤΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

---

## Document Object Model (DOM) & JavaScript

Κωνσταντίνος Τσερπές

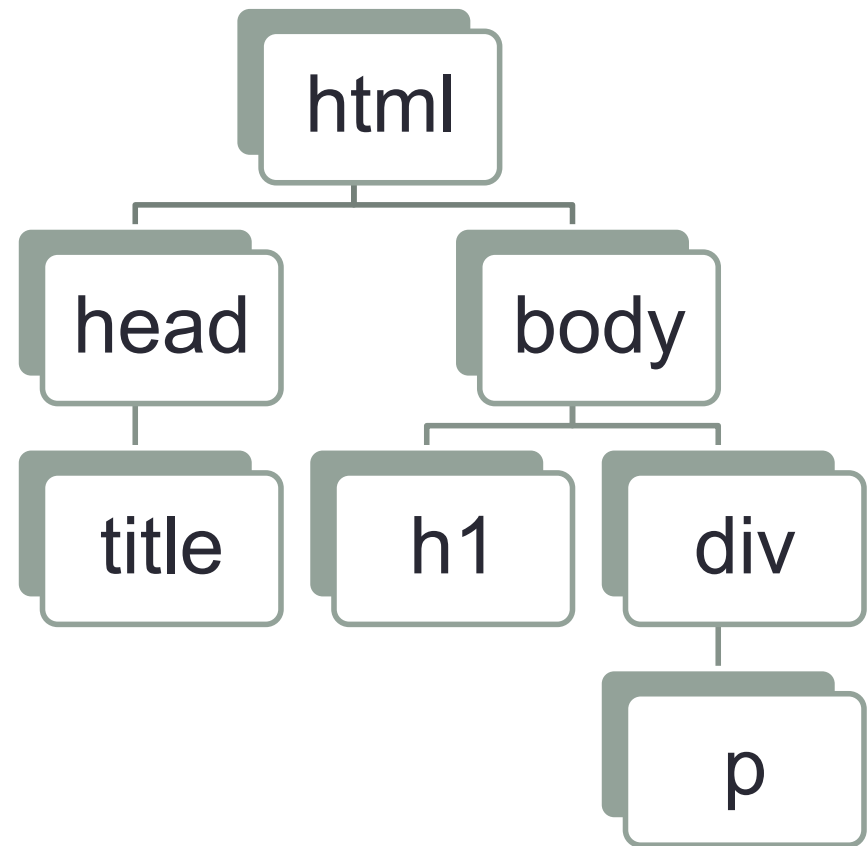
Τμήμα Πληροφορικής και Τηλεματικής

Χαροκόπειο Πανεπιστήμιο



# Επαναληψη: Document Object Model (DOM)

- Μία αναπαράσταση της τρέχουσας ιστοσελίδας σε μία δενδρική δομή από Javascript objects
- Επιτρέπει τον έλεγχο στοιχείων (elements) της σελίδας με χρήση script code αφού φορτωθεί η σελίδα
  - Άμεση αλληλεπίδραση στην πλευρά του πελάτη
  - Browser-independent
  - Επιτρέπει το σταδιακό εμπλουτισμό της σελίδας
- Χρησιμοποιείται επίσης για XML parsing

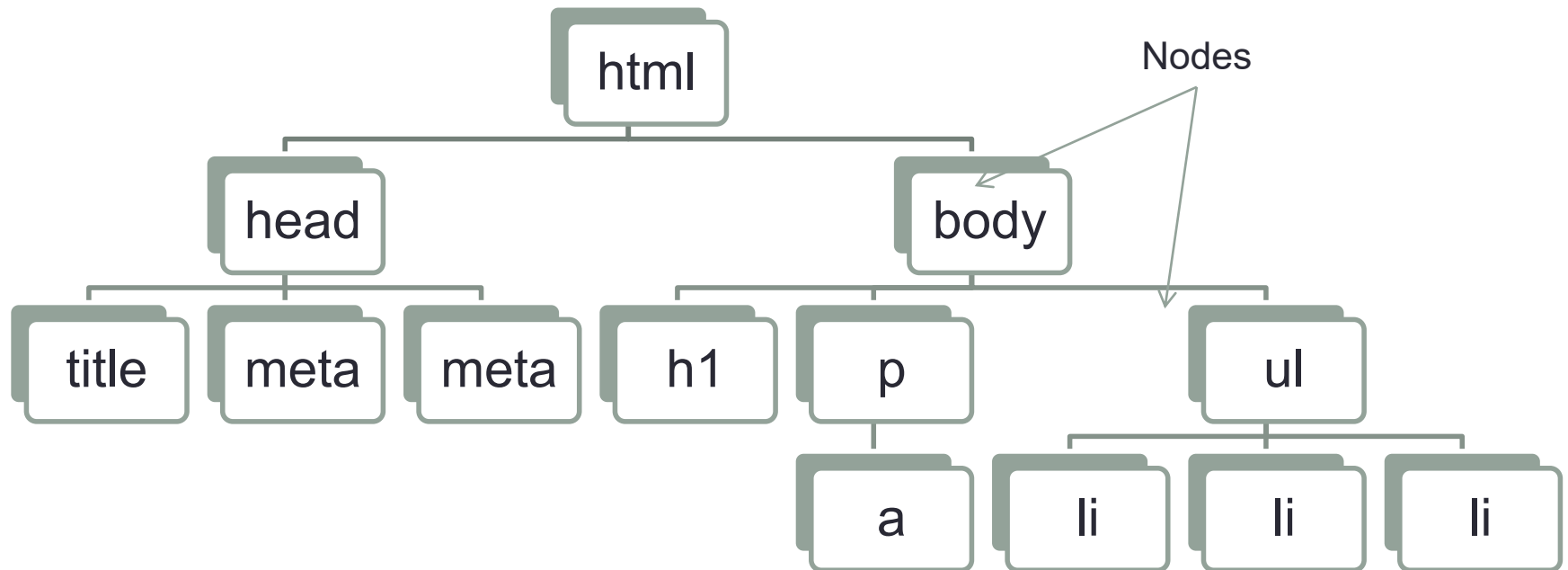


# Παράδειγμα XHTML Σελίδας

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/XHTML1/DTD/XHTML1-strict.dtd">
<html xmlns="http://www.w3.org/1999/XHTML"
xml:lang="en" lang="en">
  <head>
    <title>Page Title</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Language" content="en-us" />
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>A paragraph with a <a href="http://www.google.com/">link</a>.</p>
    <ul>
      <li>a list item</li>
      <li>another list item</li>
      <li>a third list item</li>
    </ul>
  </body>
</html>
```



# Το Δέντρο DOM



# Τύποι Κόμβων (Node Types)

`<p>This is a paragraph of text with a <a href="/path/to/another/page.html">link</a> inside.`



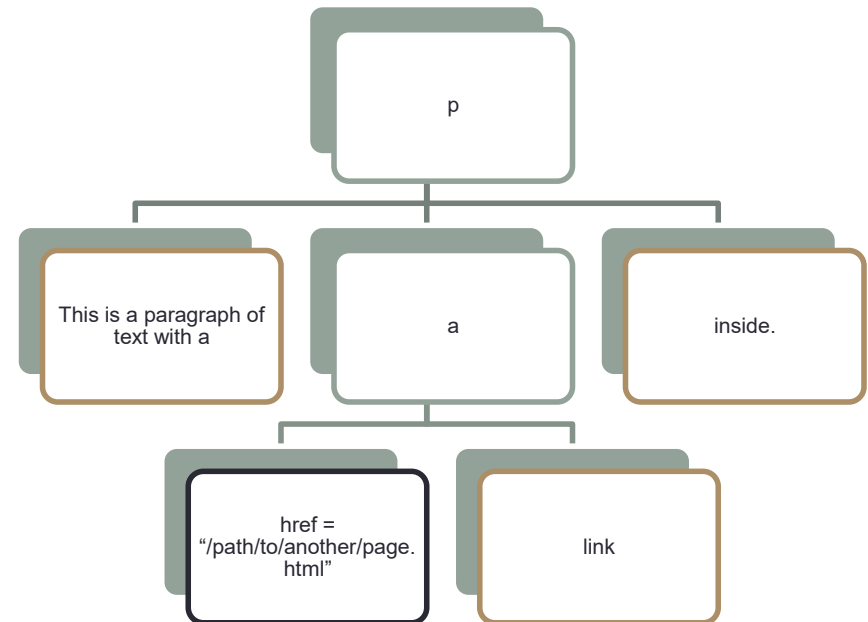
- **element nodes** (HTML tag)
  - can have children and/or attributes



- **text nodes** (text in a block element)
  - a child within an element node
  - cannot have children or attributes



- **attribute nodes** (attribute/value pair inside the start of a tag)
  - a child within an element node
  - cannot have children or attributes



# Nodes Vs Elements

- Quiz
  - Στο παράδειγμα `<a>blah</a>`, ποια είναι τα nodes και ποια τα elements
  - Τι value έχει ένα element node;

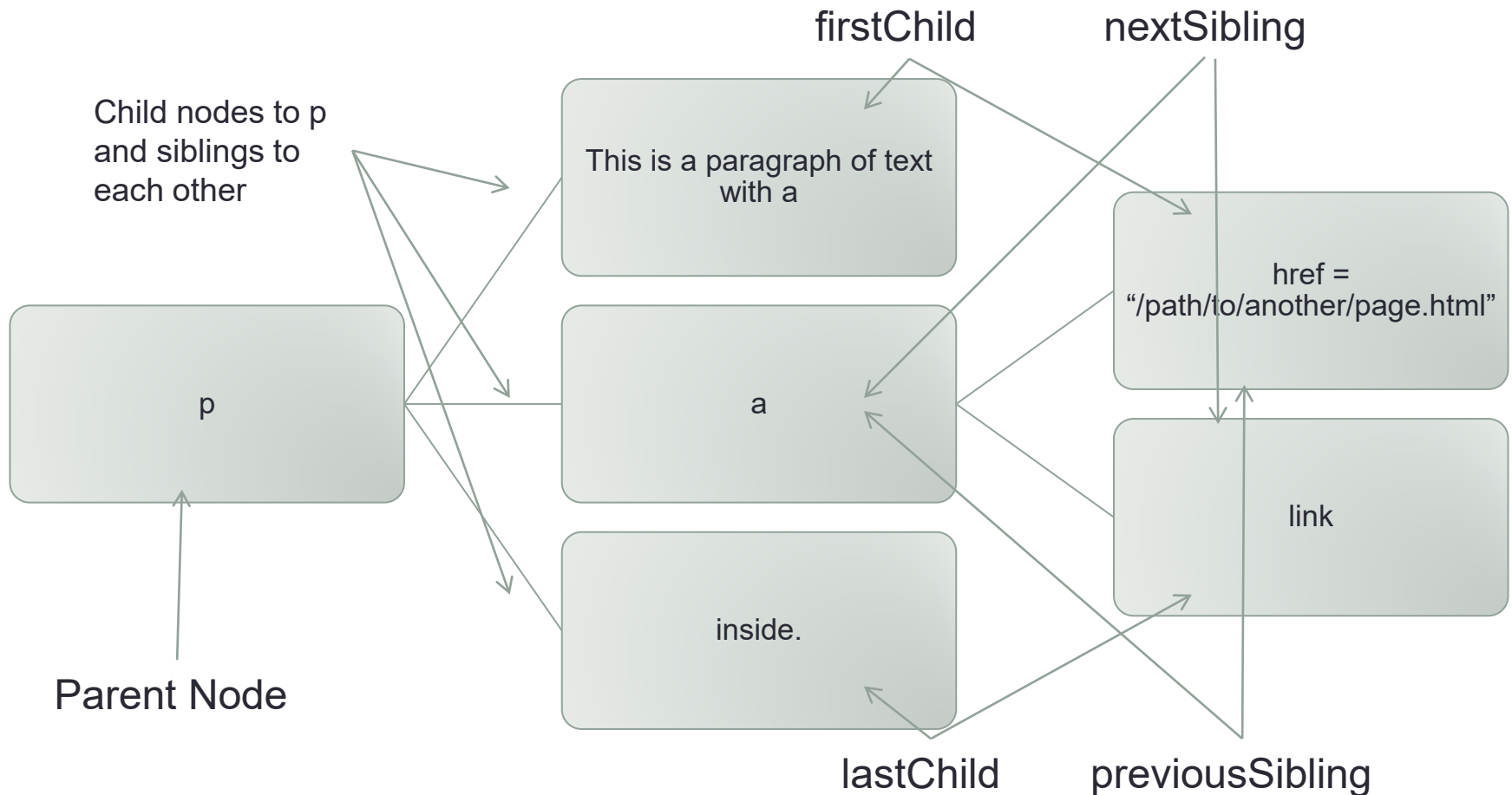


# Πρόσβαση Στο Δέντρο DOM

- Κάθε αντικείμενο DOM σε κόμβο έχει τις ακόλουθες ιδιότητες:
  - `firstChild`, `lastChild` : αρχή/τέλος της λίστας παιδιών αυτού του κόμβου
  - `childNodes` : Όλα τα παιδιά του κόμβου (array)
  - `nextSibling`, `previousSibling` : γειτονικοί κόμβοι οι οποίοι έχουν τον ίδιο γονέα
  - `parentNode` : το στοιχείο το οποίο περιλαμβάνει αυτό τον κόμβο



# Παράδειγμα Πρόσβασης Στο Δέντρο DOM tree





# Τροποποίηση Του Δέντρου DOM

- Κάθε αντικείμενο κόμβου DOM έχει τις ακόλουθες μεθόδους:
  - **appendChild(newNode)** : τοποθετεί το newNode στο τέλος τις λίστας των παιδιών του συγκεκριμένου κόμβου
  - **insertBefore(newChild, oldChild)** : τοποθετεί το νέο κόμβο newChild στη λίστα των παιδιών του συγκεκριμένου κόμβου ακριβώς πριν το oldChild
  - **removeChild(the\_Node)** : αφαιρεί τον κόμβο the\_Node από τη λίστα των παιδιών του συγκεκριμένου κόμβου
  - **replaceChild(newChild, oldChild)** : αντικαθιστά τον κόμβο oldChild με τον newChild



# Δημιουργία Νέων Κόμβων: createElement

```
// create a new <h2> node
var newHeading = document.createElement("h2");
newHeading.style.color = "green";
var newHeadingTxt = document.createTextNode("This is a
  heading");
newHeading.appendChild(newHeadingTxt);

// put it onto the page in the div with id "content"
var contentArea = document.getElementById("content");
contentArea.appendChild(newHeading);
```

- Το `document.createElement("tag")` δημιουργεί ένα νέο άδειο κόμβο DOM που αναπαριστά ένα στοιχείο αυτού του τύπου
- Οι ιδιότητες του κόμβου αυτού μπορούν να οριστούν όπως κάθε άλλου κόμβου DOM
- Όταν οι απαραίτητες ιδιότητες οριστούν ο κόμβος μπορεί να προστεθεί στη σελίδα



# Διαχείριση Κόμβων (1)

- Quiz

- Φτιάξτε 3 textboxes και ένα κουμπί. Το 1<sup>ο</sup> textbox θα αντιστοιχεί σε κείμενο που θέλετε να γράψετε με heading1 (“h1”), το 2<sup>ο</sup> με heading2 (“h2”) και το 3<sup>ο</sup> με το default τρόπο (“p”). Κάθε φορά που θα πατάτε το κουμπί θα προστίθεται στη σελίδα το κείμενο με την αντίστοιχη μορφοποίηση\*. Δείτε παράδειγμα:

Text in h1:

Text in h2:

Text in p:

**Heading 1**

**Heading 2**

Normal text

**Heading 1 (again)**

**Heading 2 (again)**

Normal text (again)

\*Η τιμή που έχει ένα node αποθηκεύεται στο property: `nodeValue`



# Διαχείριση Κόμβων (2)

- Quiz
  - Τι θα γίνει αν το κουμπί είναι τύπου submit; Πώς λύνεται το πρόβλημα;
  - Κάντε το ίδιο με το προηγούμενο με τη διαφορά ότι κάθε φορά που προστίθεται μία νέα καταχώρηση θα διαγράφεται μία παλιά



# innerHTML

- Η innerHTML επιτρέπει την αντικατάσταση του περιεχομένου οποιουδήποτε HTML element με ένα string.
- Το string μπορεί να περιλαμβάνει άλλα HTML elements (δημιουργία elements χωρίς τις μεθόδους DOM)

- Οι ακόλουθες 2 γραμμές:

```
newHeadingTxt = document.createTextNode("This is a  
heading");  
newHeading.appendChild(newHeadingTxt);
```

- ...μπορούν να αντικατασταθούν από:

```
newHeading.innerHTML = "This is a heading";
```



# innerHTML Vs DOM

## innerHTML

- Λιγότερος κώδικας
- Πιο γρήγορη (μάλλον, βλ. [www.quirksmode.org/dom/innerHTML.html](http://www.quirksmode.org/dom/innerHTML.html))
- Δεν είναι μέρος κανενός προτύπου (π.χ. W3C) ωστόσο υποστηρίζεται από όλους τους browsers
- Προσοχή με τα " και '
- Κακό στυλ (*style and JS code embedded within HTML*)
- Σε μία λίστα από childs δεν μπορείς να προσθέσεις nodes στη μέση (μόνο αρχή και τέλος)

## DOM Objects

- Δομημένος κώδικας
- Ευκολότερο να προσθέσουμε event handlers σε DOM object
- Ευκολότερο να πειράξουμε το styles των DOM objects
- still okay to use innerHTML if a node's inner contents are trivial



# innerHTML

- Quiz
  - Υλοποιήστε το προηγούμενο παράδειγμα και με innerHTML



# Quiz (Form Validation)

- Φτιάξτε μία σελίδα με ίδια λειτουργικότητα με την ακόλουθη:

**Εισαγωγή νέου φοιτητή**

Όνομα:

Επώνυμο:

Πατρώνυμο:

A.M.:

Ημέρ/νία εγγραφής:

Ηλικία:

Εξάμηνο:

Όνομα:  (The field "First Name" cannot be empty)





# Events

- The Event interface represents any event of the DOM. It contains common properties and methods to any event.
- A lot of other interfaces implement the Event interface



# addEventListener Method

- The `EventTarget.addEventListener()` method registers the specified listener on the `EventTarget` it's called on.
- The event target may be an `Element` in a document, the `Document` itself, a `Window`, or any other object that supports events (such as `XMLHttpRequest`).
- Syntax
  - `target.addEventListener(type, listener[, options]);`
  - `target.addEventListener(type, listener[, useCapture]);`



# Πηγές

- <http://www.w3schools.com/js/default.asp>
- <http://www.cs.washington.edu/education/courses/cse190m/07sp/lectures/slides/08-dom.html>
- Chris Ullman, Lucinda Dykes, Beginning Ajax, Wiley Publishing Inc, 2007

