

Browser

ΜΑΘΗΜΑ 1.9

Document

ΠΕΡΙΕΧΟΜΕΝΑ:

1. document ως ρίζα του DOM
2. Υπενθυμίσεις - Επεξεργασία του DOM
3. Πληροφορίες Σελίδας
4. Φόρμες
5. Cookies
 1. Ιδιότητες Cookie
 2. Παράδειγμα

ΜΑΘΗΜΑ 1.9: Document

1. document ως ρίζα του DOM

Υπενθύμιση από προηγούμενα μαθήματα:

- Το αντικείμενο document μοντελοποιεί τον κόμβο - ρίζα. Αλυσίδα Πρωτοτύπων:
 - **HTMLDocument**: Διατηρείται για λόγους συμβατότητας
 - **Document**: Μέθοδοι για events κ.α.
 - **Node**: Είναι κόμβος του Δένδρου. Μέθοδοι Επιλογής π.χ. με σχέσεις συγγένειας
 - **EventTarget**: Μπορεί να είναι στόχος σε events
 - **Object**: Το γενικό αντικείμενο της JS

Παράδειγμα 1: reminder

```
console.log("Node Type of document:", document.nodeType);  
console.log("Node Name of document:", document.nodeName);
```

```
console.log("First Child of document:", document.firstChild);  
console.log("Last Child of document:", document.lastChild);
```

```
if (document.hasChildNodes()) {  
    console.log("The document has child nodes.");  
} else {  
    console.log("The document has no child nodes.");  
}
```

```
let documentChildren = document.childNodes;  
console.log("Child nodes of the document (NodeList):",  
documentChildren);
```

```
documentChildren.forEach(node => {  
    console.log("Child Node:", node);  
    console.log("Node Type:", node.nodeType);  
    console.log("Node Name:", node.nodeName);  
});
```

```
console.log("Parent Node of document:", document.parentNode);  
console.log("Next Sibling of document:", document.nextSibling);  
console.log("Previous Sibling of document:", document.previousSibling);
```

```
let rootElement = document.documentElement;  
console.log("Root element of the document:", rootElement);
```

```
Node Type of document: 9  
Node Name of document: #document  
  
    <html>  
      <head></head>  
      ▶ <body> ... </body>  
First Child of document: </html>  
  
    <html>  
      <head></head>  
      ▶ <body> ... </body>  
Last Child of document: </html>  
The document has child nodes.  
Child nodes of the document (NodeList): ▶ NodeList [html]  
  
    <html>  
      <head></head>  
      ▶ <body> ... </body>  
Child Node: </html>  
Node Type: 1  
Node Name: HTML  
Parent Node of document: null  
Next Sibling of document: null  
Previous Sibling of document: null  
Root element of the document:  
    <html>  
      <head></head>  
      ▶ <body> ... </body>  
    </html>
```

Υπενθύμιση από προηγούμενα μαθήματα:

- Το document περιέχει σημαντικές μεθόδους για την διαχείριση της σελίδας:
 - Επιλογείς:** Μέθοδοι που επιλέγουν στοιχεία από το DOM (μάθημα 1.1)

Property	Επεξήγηση
getElementById(id)	Επιστρέφει το Element το οποίο έχει το συγκεκριμένο id.
getElementsByClassName(class)	Επιστρέφει HTMLCollection το οποίο περιέχει τα Element με τη συγκεκριμένη κλάση
getElementsByTagName(tag)	Επιστρέφει HTMLCollection το οποίο περιέχει τα Element του συγκεκριμένου tag
getElementByName(name)	Επιστρέφει NodeList το οποίο περιέχει τα Element με το χαρ/κο name
querySelector(query)	Επιστρέφει το πρώτο Element που ταιριάζει με το CSS query
querySelectorAll(query)	Επιστρέφει NodeList που περιέχει όλα τα Element που ταιριάζουν με το CSS query

- Πρόσθετα περιέχει τα εξής βολικά properties:

Μέθοδος	Επεξήγηση
documentElement	To root element (<html>)
body	To στοιχείο <body>
head	To στοιχείο <head>

Παράδειγμα 2: DOM

```
const rootElement = document.documentElement;
console.log("Root Element (HTML):", rootElement);

const headElement = document.head;
console.log("Head Element:", headElement);

const bodyElement = document.body;
console.log("Body Element:", bodyElement);

const infoDiv = document.querySelector("#info");

const newParagraph = document.createElement("p");
newParagraph.textContent = "This paragraph was dynamically added using createElement() and appended to the #info div.";
newParagraph.classList.add("highlight");

infoDiv.appendChild(newParagraph);

const newHeading = document.createElement("h2");
newHeading.textContent = "This is a dynamically created heading!";
newHeading.classList.add("highlight");

infoDiv.insertBefore(newHeading, newParagraph);
```

- Δημιουργία Στοιχείων (μάθημα 1.4):

Μέθοδος	Επεξήγηση
createElement(tag)	Κατασκευάζει ένα αντικείμενο τύπου tag

- Events:** (Μπορεί να είναι στόχος Events - βλ. μάθημα 1.1 και ενότητα 2)
- Έλεγχος ορατότητας** (properties που δείχνουν αν η σελίδα είναι ορατή/κρυμμένη κ.λπ.) - βλ. μάθημα 2.4

Πληροφορίες Σελίδας:

- Το αντικείμενο document περιέχει properties που δίνουν πληροφορίες για τη σελίδα:

Property	Επεξήγηση
title	Τίτλος Σελίδας
URL	URL της Σελίδας
domain	Domain της Σελίδας
referrer	Προηγούμενη σελίδα
characterSet	Κωδικοποίηση
readyState	Τρέχουσα κατάσταση της φόρτωσης της σελίδας (get): Τιμές: loading: Ακόμη φορτώνεται το document interactive: Έχει γίνει parse και φορτώνονται δευτερευοντα στοιχεία όπως εικόνες, complete: Η φόρτωση έχει ολοκληρωθεί
lastModified	Ημερομηνία Τελευταίας Τροποποίησης

Παρατήρηση:

- Όλα τα παραπάνω properties είναι read-only,
 - εκτός του title, το οποίο μπορούμε να χρησιμοποιήσουμε για να αλλάξουμε τον τίτλο της σελίδας.

Παράδειγμα 3: info

```
// Log all document information properties
console.log("Initial Document Information:");
console.log("Title:", document.title);
console.log("URL:", document.URL);
console.log("Domain:", document.domain);
console.log("Referrer:", document.referrer);
console.log("Character Set:", document.characterSet);
console.log("Ready State:", document.readyState);
console.log("Last Modified:", document.lastModified);

// Change the document title
document.title = "New Title for the Document";

// Log the updated title
console.log("Updated Title:", document.title);
```

```
Initial Document Information:
Title: Psounis JS Browser Tutorial
URL:
file:///C:/Users/psoun/OneDrive/Work/edu/WEB/c
Domain:
Referrer:
Character Set: UTF-8
Ready State: loading
Last Modified: 10/07/2024 09:57:22
Updated Title: New Title for the Document
```

Φόρμες:

- Στο document περιέχεται συλλογή από τις φόρμες της σελίδας που μπορούμε να χρησιμοποιήσουμε για να τις στείλουμε προγραμματιστικά στο server:

Μέλος	Επεξήγηση
forms	Συλλογή από φόρμες

- Έπειτα έχουμε πρόσβαση σε κάθε φόρμα με τα properties/μεθόδους:

Μέλος	Επεξήγηση
elements	Τα στοιχεία της φόρμας
submit()	Στέλνει την φόρμα
reset()	Διαγράφει τα δεδομένα της φόρμας

Παρατήρηση:

- Η διαφορά με τους επιλογείς είναι:
 - Η συλλογή από φόρμες είναι live HTMLCollection, που σημαίνει ότι ενημερώνεται αυτόματα αν υπάρξουν αλλαγές στις φόρμες της σελίδας
 - Ενώ οι επιλογείς επιστρέφουν στατικό αποτέλεσμα, που δεν ενημερώνεται για τις αλλαγές της φόρμας

Παράδειγμα 4: form

```
<form name="loginForm">
  <label for="username">Username:</label>
  <input type="text" name="username" id="username">
  <br>
  <label for="password">Password:</label>
  <input type="password" name="password" id="password">
  <br>
  <input type="submit" value="Login">
  <input type="reset" value="Reset">
</form>

<button id="submitForm">Submit Form Programmatically</button>
```

```
// Access the first form and its elements
let firstForm = document.forms[0];
console.log("First form:", firstForm);

// Access form elements
let formElements = firstForm.elements;
console.log("Form elements:", formElements);

// Access the username input field by name
let usernameInput = formElements['username'];
console.log("Username input element:", usernameInput);

// Programmatically submit the form when the button is clicked
document.getElementById("submitForm").addEventListener('click', function() {
  firstForm.submit();
});

// Programmatically reset the form after 5 seconds
setTimeout(function() {
  firstForm.reset();
  console.log("Form reset to initial state.");
}, 5000);
```


Cookies:

- Δεδομένα που αποθηκεύονται στον browser:
 - Η βασική τους ιδιότητα είναι ότι χρησιμοποιούνται στην επικοινωνία με τον server.
 - Συγκεκριμένα ο server στέλνει cookies που θέλει να αποθηκευτούν στον browser
 - Και ο client στέλνει στον server όλα τα cookies που έχει αποθηκευμένα με κάθε αίτημα προς αυτόν.
- Συνηθισμένες Χρήσεις:
 - Session Management: Ένα cookie που αποθηκεύει ένα μοναδικό αναγνωριστικό του χρήστη, ανταλλάσσεται με κάθε επικοινωνία client-server.
- Τα cookies χρησιμοποιούνται για ανταλλαγή ελάχιστης πληροφορίας(συνήθως το πολύ 4KB)
 - Για αποθήκευση περισσότερων δεδομένων χρησιμοποιούνται τα storage API και IndexedDB (βλ. ενότητα 3)

Δημιουργία Cookie:

- Χρησιμοποιούμε το property του document:

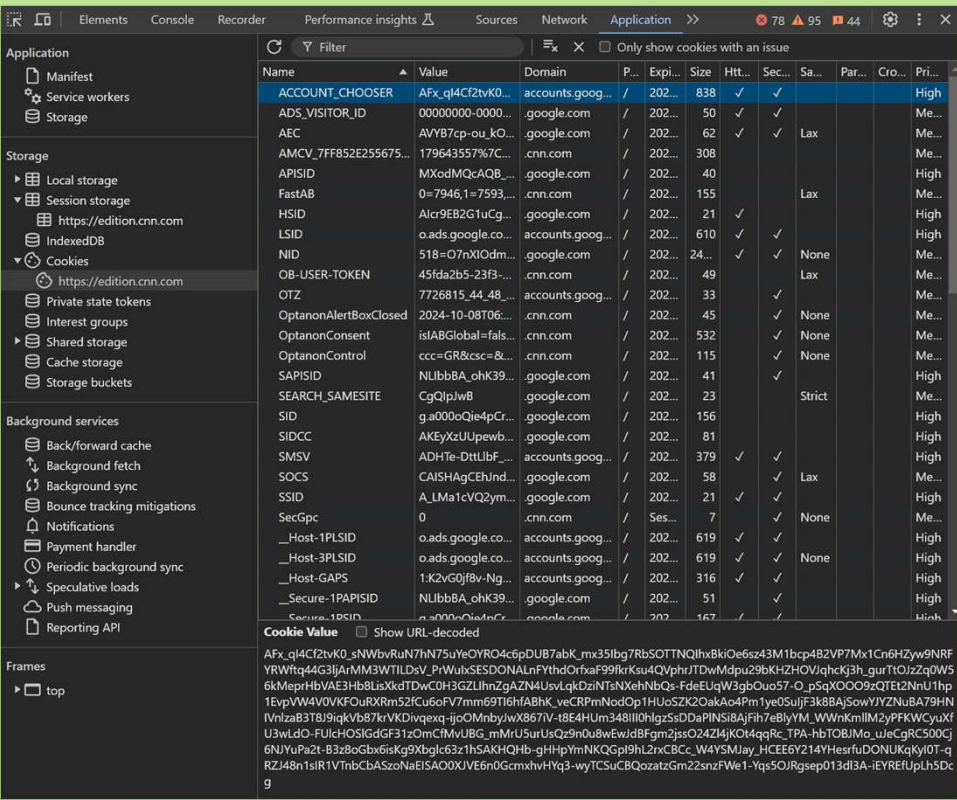
property	Επεξήγηση
cookie	Συμβολοσειρά που περιγράφει όλα τα cookies

Παράδειγμα 5: cookie

```
<script>
  console.log(document.cookie);
</script>
```

Cookies στα Developer Tools:

- F12->Application->Cookies



Ιδιότητες Cookie:

- Ένα cookie έχει τις εξής ιδιότητες:

Ιδιότητα	Επεξήγηση
expires	Χρονική στιγμή που το cookie λήγει (οπότε και σβήνεται από τον browser)
domain	Το domain που μπορεί να έχει πρόσβαση στο cookie
path	Το path του domain που έχει πρόσβαση στο cookie (π.χ. / (όλα), /admin)
secure	Το cookie ανταλλάσσεται μόνο σε HTTPS
HttpOnly	Η JS δεν έχει πρόσβαση στο cookie

Παρατήρηση:

- Η διαχείριση των cookies αποκλειστικά ως μία συμβολοσειρά φαίνεται παρωχημένη και δυσκολεύει τη χρήση τους.
- Ωστόσο μοντέρνα API όπως τα Storage API και IndexedDB μπορούν να χρησιμοποιηθούν για την αποθήκευση δεδομένων στον browsers και τα cookies να παραμείνουν αποκλειστικά για διαχείριση από τη μεριά του server.
 - Έτσι η χρήση τους περιορίζεται για τέτοιους λόγους: Session Management, Tracking, Analytics κ.λπ.
- Βλέπουμε στο παράδειγμα μία προγραμματιστική διαχείριση των cookies.

Παράδειγμα 6: form

```

<!-- Input fields for key and value -->
<label for="key">Cookie Key:</label>
<input type="text" id="key" placeholder="Enter cookie name">

<label for="value">Cookie Value:</label>
<input type="text" id="value" placeholder="Enter cookie value">

<!-- Buttons to interact with cookies -->
<button onclick="addCookie()">Add Cookie</button>
<button onclick="removeCookie()">Remove Cookie</button>
<button onclick="printCookies()">Print All Cookies</button>

<!-- Area to display cookies -->
<div id="cookieOutput" style="margin-top: 20px;"></div>
    
```

```

// Function to add a cookie
function addCookie() {
    let key = document.getElementById('key').value;
    let value = document.getElementById('value').value;

    if (key && value) {
        // Set cookie with 7-day expiration
        let expiryDate = new Date();
        expiryDate.setTime(expiryDate.getTime() + (7 * 24 * 60 * 60 * 1000)); // 7
        days
        let expires = "expires=" + expiryDate.toUTCString();

        document.cookie = key + "=" + value + ";" + expires + "; path=/";
        console.log(`Cookie added: ${key}=${value}`);
    } else {
        console.log("Please enter both a key and a value.");
    }
}
    
```

...ΣΥΝΕΧΕΙΑ...

```
// Function to remove a cookie
function removeCookie() {
  let key = document.getElementById('key').value;

  if (key) {
    // Set the cookie with an expiration date in the past
    document.cookie = key + "=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";
    console.log('Cookie removed: ${key}');
  } else {
    console.log("Please enter a key to remove the cookie.");
  }
}

// Function to print all cookies
function printCookies() {
  let allCookies = document.cookie;
  let output = document.getElementById('cookieOutput');

  if (allCookies) {
    console.log("Cookies: " + allCookies);
    output.textContent = "Cookies: " + allCookies;
  } else {
    console.log("No cookies found.");
    output.textContent = "No cookies found.";
  }
}
```

Cookie Manager

Cookie Key:

Cookie Value:

Cookies: cookie1=value1

Cookie Manager

Cookie Key:

Cookie Value:

Cookies: cookie1=value1; cookie2=value2

Σημείωση:

- Το παράδειγμα αυτό απαιτεί να τρέξει μέσω server (βλ. βίντεο)
- π.χ. μέσω python: **python -m http.server 8080**