

JS Browser

ΜΑΘΗΜΑ 3.8
LOCATION

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Το αντικείμενο window.location
2. Μέθοδοι Φόρτωσης Νέας Σελίδας
3. Παράμετροι GET Αιτήματος
4. Παράδειγμα

ΜΑΘΗΜΑ 3.8: location

1. Το αντικείμενο window.location

window.location:

- Αντικείμενο με πληροφορίες για το URL και ενέργειες για την τροποποίησή του.
- Properties του window.location:

Property	Επεξήγηση
href	Πλήρες URL (π.χ. https://example.com:8080/path?search=value#hash)
protocol	Πρωτόκολλο (http, https, ftp) (στο παράδειγμα: https)
hostname	domain (στο παράδειγμα: example.com)
port	port (στο παράδειγμα: 8080)
pathname	path (στο παράδειγμα: /path)
search	query string (στο παράδειγμα: ?search=value)
hash	anchor (στο παράδειγμα: #hash)
origin	protocol, domain, port (στο παράδειγμα: https://example.com:8080)

- Το href είναι read-write property (write: Κάνει μετάβαση στο url που δίνουμε)

Παράδειγμα 1: location

```
console.log("Full URL:", window.location.href);
console.log("Protocol:", window.location.protocol);
console.log("Hostname:", window.location.hostname);
console.log("Port:", window.location.port);
console.log("Pathname:", window.location.pathname);
console.log("Search:", window.location.search);
console.log("Hash:", window.location.hash);
console.log("Host:", window.location.host);
console.log("Origin:", window.location.origin);
```

```
Full URL: http://localhost:8080/jsBrowser3.browser.objects/jsBrowser3.8.location/location.html?index=1&index=2#section1
Protocol: http:
Hostname: localhost
Port: 8080
Pathname: /jsBrowser3.browser.objects/jsBrowser3.8.location/location.html
Search: ?index=1&index=2
Hash: #section1
Host: localhost:8080
Origin: http://localhost:8080
INS: successfully injected instart
```

Περισσότερο ενδιαφέρον έχει αν το τρέξουμε μέσω server:

```
python -m http.server 8080
```

Μέθοδοι Φόρτωσης Νέας Σελίδας:

- Με τις ακόλουθες μεθόδους του location, μπορούμε να αλλάξουμε τη σελίδα που πλοηγούμαστε:

Μέθοδος	Επεξήγηση
reload([hard])	Φορτώνει την ίδια σελίδα Αν χρησιμοποιηθεί το προαιρετικό όρισμα hard=true, τότε γίνεται hard reload (server), αλλιώς γίνεται soft reload (cach)
replace(url)	Γίνεται μετάβαση στο url

Ιστορικό στον browser:

- Οι δύο μέθοδοι δεν δημιουργούν εγγραφή στο ιστορικό του browser (δεν είναι δυνατό να πάμε Back/Forward) σε αντίθεση με το property href που δημιουργεί εγγραφή στο ιστορικό.

Αλλαγή Σελίδας με	Ιστορικό
reload([hard])	Όχι
replace(url)	Όχι
href	Ναι

Παράδειγμα 2: navigate

HTML

```
<button onclick="redirectToNewPage()">Go to Example.com (New History Entry)</button>
<button onclick="replaceWithNewPage()">Replace with New Page (No History Entry)</button>
<button onclick="reloadPage()">Reload Current Page</button>
```

JS

```
function redirectToNewPage() {
    window.location.href = "https://example.com";
}

function replaceWithNewPage() {
    window.location.replace("https://example.com/new-page");
}

function reloadPage() {
    window.location.reload(); // Soft reload
}
```

Παράμετροι GET αιτήματος:


- Είδαμε ότι με το location.search παίρνουμε το search μέρος του query.
- Πρόσθετα, μπορούμε να διοχετεύσουμε το location.search στον κατασκευαστή του αντικειμένου URLSearchParams:

Κατασκευαστής	Επεξήγηση
URLSearchParams(search)	Επιστρέφει αντ/νο URLSearchParams


- Το αντικείμενο αυτό, περιέχει βολικές μεθόδους για την διαχείριση των παραμέτρων του GET αιτήματος, όπως:

Μέθοδος	Επεξήγηση
get(name)	Επιστρέφει την τιμή της παραμέτρου με όνομα name
has(name)	True, αν περιέχει παράμετρο με όνομα name
delete(name)	Διαγράφει την παράμετρο με όνομα name
entries()	Επιστρέφει iterator επί όλων των key-value ζευγαριών
set(name, value)	Θέτει στο ζεύγος με όνομα name, την τιμή value
append(name, value)	Προσθέτει το ζεύγος name-value στις παραμέτρους

Παράδειγμα 3: get-attributes



```
<body onload="displayQueryParameters()">
<h1>Parsed Query String Parameters</h1>
<div id="query-params"></div>
</body>
```



```
function displayQueryParameters() {
  // Get the query string parameters
  let params = new URLSearchParams(window.location.search);
  let output = "";

  // Loop through and display each key-value pair
  params.forEach((value, key) => {
    output += `<p><strong>${key}</strong>: ${value}</p>`;
  });

  // Display the results in the HTML
  document.getElementById('query-params').innerHTML = output;
}
```

Αλλαγή πρωτοκόλλου:

- Συχνή Λειτουργία:
 - Ελέγχουμε αν το πρωτόκολλο είναι http
 - Αν ναι, προσαρμόζουμε σε https
- Αλλάζουμε τα στοιχεία του url ώστε να χρησιμοποιεί το https
 - και επισυνάπτουμε το υπόλοιπο url, με άλλες τροποποιήσεις που κρίνονται απαραίτητες

Παράδειγμα 4: protocol

```
function enforceHTTPS() {
    // Check if the protocol is 'http:'
    if (window.location.protocol === "http:") {
        // Construct the HTTPS URL starting with 'https://'
        let newURL = "https://" + window.location.hostname;

        // Only append the port if it's not a standard port (80 for HTTP)
        if (window.location.port && window.location.port !== "80") {
            newURL += ":" + window.location.port;
        }

        // Append the pathname (the part of the URL after the hostname)
        newURL += window.location.pathname;

        // Append the search query if it exists
        if (window.location.search) {
            newURL += window.location.search;
        }

        // Append the hash fragment if it exists
        if (window.location.hash) {
            newURL += window.location.hash;
        }

        // Redirect to the HTTPS version
        window.location.href = newURL;
    }
}

// Call the function when the page loads
enforceHTTPS();
```