

JS Browser

ΜΑΘΗΜΑ 2.4

ΠΕΡΙΣΣΟΤΕΡΑ EVENTS

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Events window
 1. Events Σελίδας
 2. DOMContentLoaded
 3. resize
 4. scroll
 5. Δίκτυο
 6. Visibility
2. Events Clipboard
3. Άλλα Events

Events Σελίδας:

- Τα παρακάτω events σχετίζονται με τη φόρτωση και το κλείσιμο της σελίδας (και αφορούν το **αντικείμενο window**):

event	Επεξήγηση
load	Ενεργοποιείται όταν όλα τα στοιχεία της σελίδας (html, css, js, images) έχουν φορτωθεί
beforeunload	Ενεργοποιείται όταν ο χρήστης πρόκειται να φύγει από τη σελίδα προτού γίνει η ενέργεια
unload	Όταν ο χρήστης φεύγει από τη σελίδα, αφού γίνει η ενέργεια

Παρατηρήσεις:

- Τα beforeunload, unload ενεργοποιούνται σε διάφορα σενάρια:
 - Ο χρήστης γράφει ένα νέο url και πατάει Enter
 - Ο χρήστης κάνει κλικ σε σύνδεσμο σε άλλη σελίδα
 - Ο χρήστης πατάει refresh, back, forward, bookmark, κ.λπ.
 - Ο χρήστης κλείνει το tab ή το παράθυρο
- Το beforeunload χρησιμοποιείται συνήθως για να βγάλει κάποιο μήνυμα, αν π.χ. ο χρήστη δεν έχει σώσει τις αλλαγές που έχει κάνει
- Το unload χρησιμοποιείται συνήθως για ενέργειες όπως logging, clean-up πόρων κ.λπ.

Παράδειγμα 1: event-window.html

```
<textarea id="userInput" placeholder="Type something here..."></textarea>
```

```
window.addEventListener('load', () => {
  console.log('Page has fully loaded.');
```

```
  alert('Welcome! The page has fully loaded.');
```

```
});
```

```
window.addEventListener('beforeunload', (event) => {
  const textarea = document.getElementById('userInput');
```

```
  if (textarea.value !== '') {
```

```
    event.preventDefault(); // Not necessary in modern browsers but kept for
```

```
    compatibility
```

```
    event.returnValue = ''; // This displays the browser's default confirmation
```

```
    dialog
```

```
    // Note: Custom messages are ignored by modern browsers for security
```

```
    reasons
```

```
  }
```

```
});
```

```
window.addEventListener('unload', () => {
  console.log('Page is unloading...');
```

```
});
```

Reload site?

Changes you made may not be saved.

☐ Prevent this page from creating additional dialogs

Reload

Cancel

Reload site?

Changes you made may not be saved.

☐ Prevent this page from creating additional dialogs

Reload

Cancel

Events Σελίδας: DOMContentLoaded

- Τα παρακάτω events σχετίζονται με τη φόρτωση και το κλείσιμο της σελίδας (και αφορούν το αντικείμενο window):

event	Επεξήγηση
DOMContentLoaded	Ενεργοποιείται όταν έχει φορτωθεί και έχει γίνει parse η HTML (δηλαδή έχει κατασκευαστεί το DOM)

Παρατήρηση:

- Το event DOMContentLoaded πυροδοτείται πριν από το event load (που είδαμε στην προηγούμενη διαφάνεια)
 - Περιμένει μόνο την HTML να φορτωθεί και να γίνει parse (ενώ το load περιμένει και τα άλλα μέρη, όπως stylesheets, js και εικόνες)
 - Συχνά προτιμάται για να κάνουμε ενέργειες με το που θα φορτωθεί η σελίδα, αφού επιτρέπει τα scripts να τρέχουν με το που θα φορτωθεί η σελίδα.

Παράδειγμα 2: event-window-dom.html

```
<h1>DOMContentLoaded vs. Load Events</h1>
<p>This page demonstrates the differences between DOMContentLoaded and Load events.</p>

<div id="status"></div>
```

```
// DOMContentLoaded event: Fires when the initial HTML document is loaded and
// parsed
document.addEventListener('DOMContentLoaded', () => {
  const statusDiv = document.getElementById('status');
  statusDiv.textContent = 'DOMContentLoaded: The DOM is fully parsed, but
  images might still be loading.';
  console.log('DOMContentLoaded event fired.');
```

```
// Load event: Fires when the whole page including images, scripts, and CSS is fully
// loaded
window.addEventListener('load', () => {
  const statusDiv = document.getElementById('status');
  statusDiv.textContent = 'Load: The entire page, including images, has fully
  loaded.';
  console.log('Load event fired.');
```

DOMContentLoaded vs. Load Events

This page demonstrates the differences between DOMContentLoaded and Load events.



Load: The entire page, including images, has fully loaded.

Events Σελίδας: resize

- Το παρακάτω event σχετίζονται με το μέγεθος της σελίδας (και αφορούν το αντικείμενο window):

event	Επεξήγηση
resize	Πυροδοτείται όταν γίνεται resize το μέγεθος της σελίδας

Παράδειγμα 3: event-window-resize.html

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 20px;
}

.resize-message {
  bottom: 20px;
  left: 20px;
  background-color: #f1f1f1;
  padding: 10px 15px;
  border: 1px solid #ddd;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  border-radius: 5px;
  font-size: 14px;
  color: #333;
}
```

```
<h1>Resize the Window</h1>
<p>Resize the browser window to see the dimensions update in real time.</p>

<div class="resize-message" id="resizeMessage">Window size: 0px x 0px</div>
```

```
const resizeMessage = document.getElementById('resizeMessage');

// Function to update the resize message with the current window size
function updateResizeMessage() {
  const width = window.innerWidth;
  const height = window.innerHeight;
  resizeMessage.textContent = `Window size: ${width}px x ${height}px`;
}

// Event listener for resize event to update the message with the
window dimensions
window.addEventListener('resize', updateResizeMessage);

// Initial call to set the correct values on page load
updateResizeMessage();
```

Resize the Window

Resize the browser window to see the dimensions update in real time.

Window size: 908px x 1352px

Events Σελίδας: scroll

- Τα παρακάτω event σχετίζονται με το scroll που κάνει ο χρήστης (και αφορούν το αντικείμενο window):

event	Επεξήγηση
scroll	Πυροδοτείται όταν ο χρήστης κάνει scroll

Παρατήρηση:

- Χρησιμοποιούνται τα εξής properties του documentElement (δηλ. του στοιχείου html)

event	Επεξήγηση
scrollTop	Ύψος (σε πλήθος pixels) που έχει γίνει scroll από την αρχή του <html>
scrollHeight	Συνολικό ύψος της σελίδας
clientHeight	Ορατό ύψος της σελίδας (viewport)

Παράδειγμα 4: event-window-scroll.html

```
<h1>Scroll the Page</h1>
<p>Scroll down to see the scroll indicator at the top of the page update as you move.</p>

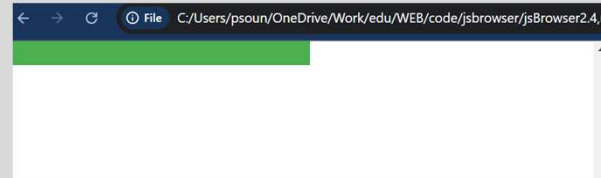
<div class="scroll-indicator" id="scrollIndicator"></div>
```

```
const scrollIndicator = document.getElementById('scrollIndicator');

// Function to update the scroll indicator width based on the scroll position
function updateScrollIndicator() {
  const scrollTop = document.documentElement.scrollTop;
  const scrollHeight = document.documentElement.scrollHeight -
document.documentElement.clientHeight;
  const scrollPercentage = (scrollTop / scrollHeight) * 100;
  scrollIndicator.style.width = scrollPercentage + '%'; // Update the width of the
scroll indicator
}

// Event listener for scroll event to update the scroll indicator
window.addEventListener('scroll', updateScrollIndicator);

// Initial call to set the correct width on page load
updateScrollIndicator();
```



Events Σελίδας: Δίκτυο

- Τα παρακάτω event έχουν να κάνουν με τη σύνδεση δικτύου (και αφορούν το αντικείμενο window):

event	Επεξήγηση
online	Όταν ο browser συνδέεται με το δίκτυο
offline	Όταν ο browser αποσυνδέεται από το δίκτυο

Παρατήρηση:

- Χρησιμοποιούμε και το αντικείμενο navigator που μεταξύ άλλων έχει το property:

event	Επεξήγηση
online	boolean: Av ο browser είναι online

Παράδειγμα 5: event-window-network.html

```
<h1>Network Status Detection</h1>
<div id="networkStatus" class="status">Checking network status...</div>
```

```
const networkStatus = document.getElementById('networkStatus');
// Function to update the network status message
function updateNetworkStatus() {
  if (navigator.onLine) {
    networkStatus.textContent = 'You are online!';
    networkStatus.classList.remove('offline');
    networkStatus.classList.add('online');
  } else {
    networkStatus.textContent = 'You are offline. Some features may be
unavailable.';
    networkStatus.classList.remove('online');
    networkStatus.classList.add('offline');
  }
}
// Event listeners for online and offline events
window.addEventListener('online', updateNetworkStatus);
window.addEventListener('offline', updateNetworkStatus);
// Initial check for network status
updateNetworkStatus();
```

Network Status Detection

You are online!

Events Σελίδας: Visibility

- Το παρακάτω event εντοπίζει αν η σελίδα είναι ορατή (π.χ. αν άλλαξε tab, ή έκανε minimize το παράθυρο). Είναι στο αντικείμενο document):

event	Επεξήγηση
visibilityChange	Όταν αλλάξει η ορατότητα της σελίδας

Παρατήρηση:

- Για τη διαχείριση του event είναι χρήσιμα τα εξής properties του document:

event	Επεξήγηση
hidden	boolean: Η σελίδα είναι ορατή
visibilityState	Η κατάσταση της σελίδας

- Η κατάσταση της σελίδας απεικονίζεται με τις τιμές:
 - visible (έστω και μερικώς ορατή)
 - hidden (δεν είναι ορατή)

Παράδειγμα 6: event-window-visibility.html

```
<h1>VisibilityChange Event Demo</h1>
<div id="timer">0</div>
<div id="status">Page is visible</div>
```

```
let timer = 0;
let interval;

// Function to update the timer
function startTimer() {
  interval = setInterval(() => {
    timer++;
    document.getElementById('timer').textContent = timer;
  }, 1000);
}

// Function to stop the timer
function stopTimer() {
  clearInterval(interval);
}

// Event listener for visibility change
document.addEventListener('visibilitychange', () => {
  const status = document.getElementById('status');
  if (document.visibilityState === 'hidden') {
    stopTimer();
    status.textContent = 'Page is hidden - Timer paused';
    console.log('Page is hidden - Timer paused')
  } else {
    startTimer();
    status.textContent = 'Page is visible - Timer running';
    console.log('Page is visible - Timer running');
  }
});

// Start the timer initially
startTimer();
```

VisibilityChange Event Demo

70

Page is visible - Timer running

Events Clipboard:

- Τα παρακάτω events σχετίζονται με το clipboard (copy - cut - paste)

event	Επεξήγηση
copy	Όταν γίνεται copy (με το ποντίκι ή Ctrl+C)
paste	Όταν γίνεται paste (με το ποντίκι ή Ctrl+V)
cut	Όταν γίνεται cut (με το ποντίκι ή Ctrl+X)

Παρατήρηση:

- Με τα παραπάνω event και την απενεργοποίηση του δεξί κλικ, μπορούμε να απενεργοποιήσουμε τον εύκολο τρόπο αντιγραφής από τη σελίδα μας.

Παράδειγμα 7: event-clipboard.html

```
<h1>Clipboard Events Listener</h1>
<textarea id="textArea" rows="4" cols="50" placeholder="Type or paste some text here..."></textarea>
<div id="output">Try copying, cutting, or pasting text to see the events in action.</div>
```

```
const textArea = document.getElementById('textArea');
const output = document.getElementById('output');

// Listening to the copy event
textArea.addEventListener('copy', (event) => {
  output.textContent = 'Copy event triggered! ';
});

// Listening to the cut event
textArea.addEventListener('cut', (event) => {
  output.textContent = 'Cut event triggered! ';
});

// Listening to the paste event
textArea.addEventListener('paste', (event) => {
  output.textContent = 'Paste event triggered! Aborted! ';
  event.preventDefault();
});
```


Σημείωση:

- Στο επόμενο μάθημα θα ασχοληθούμε αναλυτικά με τα drag & drop events
- Ωστόσο δεν θα ασχοληθούμε αναλυτικά με τα events που αναφέρονται σε αυτήν την διαφάνεια σε αυτή τη σειρά μαθημάτων.

Περισσότερα Events:

- Events για animations:

event	Επεξήγηση
animationstart	Όταν αρχίζει ένα CSS animation
animationend	Όταν τελειώνει ένα CSS animation
animationiteration	Σε κάθε επανάληψη του animation
transitionend	Όταν τελειώνει ένα CSS animation

- Events για pointer (ποντίκι, πένα, αφή)

event	Επεξήγηση
pointerdown	επαφή pointer με στοιχείο
pointerup	διακοπή επαφής pointer με στοιχείο
pointermove	Ο pointer κινείται ενόσω είναι σε επαφή
pointerenter	Ο pointer εισέρχεται σε στοιχείο
pointerleave	Ο pointer εξέρχεται από στοιχείο
pointercancel	Ένα event του pointer ακυρώνεται

- Events για web storage:

event	Επεξήγηση
storage	Όταν γίνεται αλλαγή στο web storage

- Events εκτυπωτή

event	Επεξήγηση
beforeprint	Πριν εμφανιστεί το παράθυρο εκτύπωσης
afterprint	Αφού κλείσει το παράθυρο εκτύπωσης

- Events συσκευής

event	Επεξήγηση
deviceorientation	Όταν αλλάζει η θέση συσκευής (portrait, landscape)
devicemotion	Κίνηση συσκευής (π.χ. tilting ή shaking)

- Events αφής:

event	Επεξήγηση
touchstart	Πρώτη αφή
touchmove	Κίνηση αφής
touchend	Τέλος αφής
touchcancel	Ακύρωση αφής (π.χ. από το σύστημα)