

JS Browser

ΜΑΘΗΜΑ 2.7

INLINE EVENTS

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Inline Events
 1. Events Πληκτρολογίου και Ποντικιού
 2. Events σε Φόρμες και Clipboard
 3. Events Window και Κινητών Συσκευών
2. VS addEventListener()

ΜΑΘΗΜΑ 2.7: Inline Events

1. Inline Events

Inline Events

- Ορίζονται στην HTML
- Προκαλούν την εκτέλεση κώδικα όταν συμβεί το event
- Παράδειγμα inline event:

event	Επεξήγηση
onclick	Κλικ στο στοιχείο

- Μπαίνει ως attribute σε στοιχείο της HTML και αν γίνει κλικ σε αυτό εκτελεί τον κώδικα JS που έχει τεθεί στην τιμή του attribute

Παρατήρηση:

- Είναι «παλιός» τρόπος ενεργοποίησης των events
- Χρήσιμος για τογ examples και πρόχειρο κώδικα
- Σήμερα προτιμάται ο ορισμός των events να γίνεται αποκλειστικά με την JS:
 - Με την addEventListener()
 - (και έτσι ο κώδικας της JS να είναι πλήρως διαχωρισμένος από την HTML)

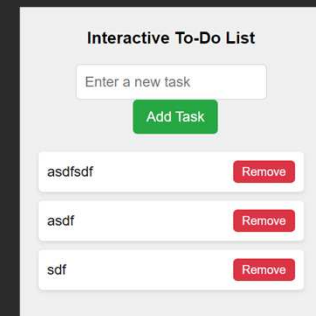
Παράδειγμα 1: inline.html

```
<h3>Interactive To-Do List</h3>
<input type="text" id="taskInput" placeholder="Enter a new task">
<button id="addTaskButton" onclick="addTask()">Add Task</button>
<ul id="taskList"></ul>
```

```
function addTask() {
  const taskInput = document.getElementById('taskInput');
  const taskList = document.getElementById('taskList');
  const newTaskText = taskInput.value;

  if (newTaskText !== '') {
    const newTaskItem = document.createElement('li');
    newTaskItem.innerHTML = `${newTaskText} <button
onclick="removeTask(this)">Remove</button>`;
    taskList.appendChild(newTaskItem);
    taskInput.value = '';
  } else {
    console.log('Please enter a task.');
```

```
function removeTask(button) {
  const taskItem = button.parentElement;
  taskItem.remove();
}
```



Events Ποντικιού

event	Επεξήγηση
onclick	Κλικ στο στοιχείο
ondblclick	Διπλό κλικ στο στοιχείο
onmouseover	Hover στο στοιχείο
onmouseout	Δείκτης μετακινείται έξω από στοιχείο
onmousedown	Το ποντίκι είναι πατημένο
onmouseup	Το ποντίκι απελευθερώνεται
onmousemove	Μετακίνηση πάνω από στοιχείο

Events Πληκτρολογίου

event	Επεξήγηση
onkeydown	Πάτημα κουμπιού
onkeyup	Απελευθέρωση κουμπιού
onkeypress	Πάτημα και απελευθέρωση κουμπιού

Events Drag&Drop

event	Επεξήγηση
ondrag	Το στοιχείο γίνεται drag
ondragstart	Αρχή drag
ondragover	Πάνω από έγκυρο drag στόχο
ondrop	Το στοιχείο γίνεται drop

Παράδειγμα 2: keyboard.html

<h3>Type Something Below:</h3>

<input type="text" onkeydown="showKey(event)" placeholder="Start typing...">

<p id="keyOutput"></p>

```
function showKey(event) {  
  const keyOutput = document.getElementById('keyOutput');  
  keyOutput.textContent = `You pressed: ${event.key}`;  
}
```

Type Something Below:

asdasdfwerwqf

You pressed: f

Events σε Φόρμες

event	Επεξήγηση
onsubmit	Όταν γίνεται submit
onchange	Αλλάζει η τιμή σε στοιχείο
onfocus	Το στοιχείο κερδίζει το focus
onblur	Το στοιχείο χάνει το focus
oninput	Η τιμή στοιχείου τροποποιείται
onreset	Όταν γίνεται reset

Events Clipboard

event	Επεξήγηση
oncopy	Όταν γίνεται copy (Ctrl+C)
onpaste	Όταν γίνεται paste (Ctrl+V)
oncut	Όταν γίνεται cut (Ctrl+X)

Events Media

event	Επεξήγηση
onplay	Αρχίζει να παίζει
onpause	Γίνεται pause
onended	Όταν φτάνει στο τέλος
onvolumechanges	Αλλάζει η ένταση ήχου

Παράδειγμα 3: clipboard.html

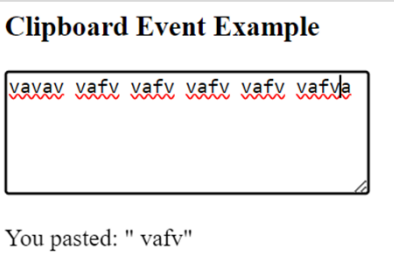
```
<h3>Clipboard Event Example</h3>
<textarea id="textArea" oncopy="handleCopy()" oncut="handleCut()"
onpaste="handlePaste(event)" rows="5" cols="30" placeholder="Try copying,
cutting, or pasting text here..."></textarea>

<p id="eventOutput"></p>

function handleCopy() {
  const textArea = document.getElementById('textArea');
  const selectedText = textArea.value.substring(textArea.selectionStart, textArea.selectionEnd);
  document.getElementById('eventOutput').textContent = `You copied: "${selectedText}"`;
}

function handleCut() {
  const textArea = document.getElementById('textArea');
  const selectedText = textArea.value.substring(textArea.selectionStart, textArea.selectionEnd);
  document.getElementById('eventOutput').textContent = `You cut: "${selectedText}"`;
}

function handlePaste(event) {
  const pastedText = event.clipboardData.getData('text');
  document.getElementById('eventOutput').textContent = `You pasted: "${pastedText}"`;
}
```



Events του Window

event	Επεξήγηση
onload	Όταν η σελίδα έχει φορτωθεί
onunload	Όταν η σελίδα γίνει unload
onresize	Όταν γίνεται resize
onscroll	Όταν γίνεται scroll

Events για Κινητές Συσκευές

event	Επεξήγηση
ontouchstart	Όταν αγγίζεται η οθόνη
ontouchmove	Όταν γίνονται κινήσεις με άγγιγμα
ontouchend	Όταν τελειώνουν οι κινήσεις με άγγιγμα

Παράδειγμα 4: window.html

```
<body onload="handleLoad()" onresize="handleResize()"
onscroll="handleScroll()">
  <div id="output">Window Event Output:</div>
</body>
```

```
function handleLoad() {
  document.getElementById('output').textContent = "Window has
loaded successfully!";
}

function handleResize() {
  document.getElementById('output').textContent = `Window resized
to: ${window.innerWidth} x ${window.innerHeight}`;
}

function handleScroll() {
  document.getElementById('output').textContent = `Scroll position:
${window.scrollY}px`;
}
```



Σημείωση:

- Θα προτιμήσουμε τον ορισμό των events μέσω της `addEventListener()` στο στοιχείο, αντί για τα inline events
- Εκτός από τον διαχωρισμό περιεχομένου (HTML) και συμπεριφοράς (JS) που είναι σημαντικός, έχουμε και το ακόλουθο πλεονέκτημα:
 - Με την `addEventListener()` μπορούμε να προσθέσουμε δύο ή παραπάνω callbacks στο ίδιο event
 - Με inline event, μπορούμε να έχουμε μόνο ένα callback ανά event.

Παράδειγμα 5: vs-add-event.html

```
<button id="myButton"
  onclick="console.log('Inline event handler executed!')">
  Click Me
</button>
```

```
// Select the button element
const button = document.getElementById('myButton');

// Add three different event listeners using addEventListener()
button.addEventListener('click', () => {
  console.log('First addEventListener handler executed!');
});

button.addEventListener('click', () => {
  console.log('Second addEventListener handler executed!');
});

button.addEventListener('click', () => {
  console.log('Third addEventListener handler executed!');
});
```

```
Inline event handler executed!
First addEventListener handler executed!
Second addEventListener handler executed!
Third addEventListener handler executed!
```