

Browser

ΜΑΘΗΜΑ 2.2

ΤΟ ΑΝΤΙΚ/NO EVENT

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Το αντικείμενο event
 1. Properties για Έλεγχο Propagation
 2. Properties για Mouse Events
 3. Properties για Keyboard Events
 4. Properties για focus

Το αντικείμενο Event:

- Διοχετεύεται αυτόματα σε έναν listener (ο οποίος συντάσσεται με το πρώτο όρισμα να είναι το αντικείμενο Event):

```
(event) => {
  ...
}
```

- Περιέχει χρήσιμα μέλη και μεθόδους:
 - Στο προηγούμενο μάθημα είδαμε τις `stopPropagation()`, `stopImmediatePropagation()` και `preventDefault()`
- Properties του event:

Property	Επεξήγηση
target	Το στοιχείο που προκάλεσε το event
currentTarget	Το στοιχείο του οποίου ο handler ενεργοποιήθηκε
type	Ο τύπος του event (π.χ. 'click', 'keydown', κ.λπ) (επόμενο μάθημα)

Παράδειγμα 1: event-object.html

```
<div class="container1">
  <button class="btn1">Click Me</button>
</div>
```

HTML

```
document.querySelectorAll("button").forEach(button=>{
  button.addEventListener("click", (event)=>{
    console.log("button capture", event.target, event.currentTarget, event.type)
  }, true)
  button.addEventListener("click", (event)=>{
    console.log("button bubble", event.target, event.currentTarget, event.type)
  })
})
document.querySelectorAll("div").forEach(div=>{
  div.addEventListener("click", (event)=>{
    console.log("div capture", event.target, event.currentTarget, event.type)
  }, true)
  div.addEventListener("click", (event)=>{
    console.log("div bubble", event.target, event.currentTarget, event.type)
  })
})
```

JS

```
div capture  <button class="btn1">Click Me</button>
> <div class="container1"> </div> click

button capture  <button class="btn1">Click Me</button>
  <button class="btn1">Click Me</button> click

button bubble  <button class="btn1">Click Me</button>
  <button class="btn1">Click Me</button> click

div bubble  <button class="btn1">Click Me</button>
> <div class="container1"> </div> click
```

Properties για έλεγχο propagation:

- Με τα ακόλουθα properties ελέγχουμε το propagation του event:

Property	Επεξήγηση
bubbles	Boolean: Αν το event θα περάσει από φάση bubbling
cancelable	Boolean: Αν το event μπορεί να ακυρωθεί
defaultPrevented	Boolean: Αν έχει κληθεί η preventDefault()
eventPhase	Σε ποια φάση βρίσκεται το event (capturing, target, bubbling)

Παράδειγμα 2: event-propagation.html

```
<div class="container1">
  <button class="btn1">Click Me</button>
</div>
```



```
document.querySelectorAll("button").forEach(button=>{
  button.addEventListener("click", (event)=>{
    console.log("button capture", event.bubbles, event.eventPhase, event.cancelable)
  }, true)
  button.addEventListener("click", (event)=>{
    console.log("button bubble", event.bubbles, event.eventPhase, event.cancelable)
  })
})
document.querySelectorAll("div").forEach(div=>{
  div.addEventListener("click", (event)=>{
    console.log("div capture", event.bubbles, event.eventPhase, event.cancelable)
  }, true)
  div.addEventListener("click", (event)=>{
    console.log("div bubble", event.bubbles, event.eventPhase, event.cancelable)
  })
})
```



```
div capture true 1 true
button capture true 2 true
button bubble true 2 true
div bubble true 3 true
```

Properties για Mouse Events:

- Ειδικά για Events που προκαλούνται από το ποντίκι, τα ακόλουθα properties είναι χρήσιμα:

Property	Επεξήγηση
clientX και clientY	Συντεταγμένες του pointer στο ορατό παράθυρο του Browser
offsetX και offsetY	Συντεταγμένες του pointer στο στοιχείο target του event
screenX και screenY	Συντεταγμένες του pointer στην οθόνη
button	Το κουμπί του mouse που πατήθηκε (left-middle-right)

Παράδειγμα 3: event-mouse.html

```
<div class="container">
  <button class="btn">Click Me</button>
</div>
```

```
document.querySelector("button")
  .addEventListener("click", (event)=>{
    console.log("client ", event.clientX, " ", event.clientY);
    console.log("offset ", event.offsetX, " ", event.offsetY);
    console.log("screen ", event.screenX, " ", event.screenY);
    console.log("button: ", event.button);
  }, true);
```

```
client  54  47
offset  25  18
screen  1090  230
button:  0
```

Properties για Keyboard Events:

- Ειδικά για Events που προκαλούνται από το πληκτρολόγιο, τα ακόλουθα events είναι χρήσιμα:

Property	Επεξήγηση
key	Η τιμή του κουμπιού που πατήθηκε
code	Η φυσική τιμή του κουμπιού που πατήθηκε
altKey ctrlKey shiftKey metaKey	Αν έχουν πατηθεί τα τροποποιητικά κουμπιά

- Το key είναι η τιμή του κουμπιού όπως πατήθηκε (π.χ. Α που προκλήθηκε από shift+'Α'), ενώ το code είναι το πραγματικό κουμπί που πατήθηκε (επιστρέφει τον κωδικό του)

Παρατήρηση:

- Στο επόμενο μάθημα θα δούμε αναλυτικά τα events πληκτρολογίου
- Για την ώρα ας γνωρίζουμε μόνο το event keydown, το οποίο ενεργοποιείται όταν ο χρήστης πατήσει ένα κουμπί.

Παράδειγμα 4: event-keyboard.html

```
<input type="text" id="inputField" placeholder="Type something..." />
<div id="output" class="output"></div>
```

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

inputField.addEventListener('keydown', (event) => {
  output.textContent = `You pressed key: "${event.key}", Code: "${event.code}"`;

  if (event.ctrlKey && event.code === 'KeyS') {
    event.preventDefault(); // Prevent default save behavior in the browser
    alert('Save action detected (Ctrl+S)');
  }

  // Example: Reject certain characters based on `key`
  if (event.key === '!') {
    event.preventDefault();
    alert('Exclamation marks are not allowed!');
  }
});
```

asdf

You pressed key: "Shift", Code: "ShiftLeft"

Properties για Focus:

- Focus: (κυρίως στις φόρμες) Ένα control γίνεται ενεργό (π.χ. σε ένα textbox, ο χρήστης κάνει κλικ, για να ξεκινήσει να γράφει)

Property	Επεξήγηση
relatedTarget	Στοιχείο που σχετίζεται με την ενέργεια focus που έγινε

Παρατήρηση:

Το παραπάνω property ενεργοποιείται στα events:

- Το event focusin: Το στοιχείο κερδίζει το focus
- Το event focusout: Το στοιχείο χάνει το focus
- Το event focus: Το στοιχείο κερδίζει το focus (χωρίς bubbling)
- Το event blur: Το στοιχείο χάνει το focus (χωρίς bubbling)

Ενεργοποίηση:

- Στα events focusin και focusout, το relatedTarget μαζί με το target έχουν την πληροφορία για το ποιος κέρδισε και έχασε το focus:

Property	event: focusin	event: focusout
target	Το στοιχείο που κέρδισε το focus	Το στοιχείο που έχασε το focus
relatedTarget	Το στοιχείο που έχασε το focus	Το στοιχείο που κέρδισε το focus

Παράδειγμα 5: event-focus-blur.html

```
<form id="exampleForm">
  <input type="text" id="firstName" class="input-field" placeholder="First Name" />
  <input type="email" id="email" class="input-field" placeholder="Email" />
</form>
```

```
document.getElementById('exampleForm').addEventListener('focusin', (event) => {
  console.log(`Focused on: ${event.target.id}`);
  if (event.relatedTarget) {
    console.log(`Focus moved from: ${event.relatedTarget.id}`);
  }
}, true);

document.getElementById('exampleForm').addEventListener('focusout', (event) => {
  console.log(`Blurred from: ${event.target.id}`);
  if (event.relatedTarget) {
    console.log(`Focus moved to: ${event.relatedTarget.id}`);
  }
  // add validation logic
}, true);
```