

JAVASCRIPT EVENTS

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος

Επιμέλεια διαφανειών: Π. Αγγελάτος

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών



Στόχος της ώρας

- Βαθύτερη ανάλυση των γεγονότων
 - Πως δηλώνονται χωρίς frameworks
 - Ειδικές συμπεριφορές
 - Η ειδική μεταβλητή this
 - Είδη γεγονότων
- Πολλά παραδείγματα

Γεγονότα

- Προσφέρουν διαδραστικότητα στη σελίδα
- Ενέργεια χρήστη → Αντίδραση σελίδας
- Η Javascript είναι σχεδιασμένη να λειτουργεί με γεγονότα

Πως χρησιμοποιούνται

- Χειριστές
 - Κώδικας που τρέχει όταν γίνεται κάποιο γεγονός
- Γεγονότα
 - Συμβαίνουν όταν αλληλεπιδρά ο χρήστης με τη σελίδα
 - Αφορούν κάθε φορά ένα στοιχείο
- Αντιστοιχίζουμε χειριστές με γεγονότα
 - Στο HTML
 - Με παραδοσιακό τρόπο
 - Με τη μέθοδο του W3C
- Όταν συμβεί το γεγονός τρέχει ο χειριστής

Διαφορετικότητα

- Μεγάλες διαφορές από browser σε browser
- Τα παρακάτω ισχύουν σε browsers όπως:
 - Firefox
 - Chrome
- Χρησιμοποιείτε frameworks όπως jQuery για γεγονότα



Δήλωση γεγονότων

- Μέσα στο HTML
 - Με ιδιότητα στο στοιχείο του οποίου τα γεγονότα μας ενδιαφέρουν
 - Ξεπερασμένη
 - Μπλέκει περιεχόμενο με συμπεριφορά

HTML δήλωση γεγονότων

- Ιδιότητες έχουν όνομα **on + όνομα γεγονότος**
 - **onclick**
 - **onkeypress**
 - **onsubmit**
 - κλπ
- Τιμή της ιδιότητας ο κώδικας που θα τρέξει (χειριστής)

Παράδειγμα

Στοιχείο που
αφορά το
γεγονός

Δήλωση ενδιαφέροντος
για κάποιο γεγονός

• `Click me`

Είδος
γεγονότος

Τι θέλουμε να
γίνει όταν συμβεί
(χειριστής)

ΠΡΟΣΟΧΗ!
ΜΗΝ ΤΟ
ΔΟΚΙΜΑΣΕΤΕ
ΣΤΟ ΣΠΙΤΙ

Δήλωση γεγονότων

- Με παραδοσιακό τρόπο
 - Ορίζοντας ιδιότητα στο αντικείμενο της DOM αναπαράστασης
 - Ορίζονται μέσα από την Javascript
 - Δε μπλέκουν περιεχόμενο με συμπεριφορά

Παραδοσιακή δήλωση γεγονότων

- Βρίσκουμε την DOM αναπαράσταση
 - Μέσω jQuery (`$('span')[0]`)
 - Μέσω DOM (`document.getElementById('foo')`)
- Ορίζουμε την ιδιότητα **on + όνομα γεγονός**
- Τιμή της ιδιότητας μία συνάρτηση

Παράδειγμα

- `Click me`

```
function doSomething() {  
    alert( 'doSomething invoked' );  
    return true;  
}  
var a = document.getElementById( 'foo' );
```

Στοιχείο που αφορά το
γεγονός

↑
`a.onClick = doSomething;`
↓
Είδος
γεγονότος

Τι θέλουμε να
γίνει όταν συμβεί
(χειριστής)

Προσοχή

- **Λάθος**

- `a.onclick = doSomething();`
- Εκτελεί την `doSomething` και το αποτέλεσμα αποθηκεύεται στο `onclick`

- **Σωστό**

- `a.onclick = doSomething`
- Στο `onclick` αποθηκεύεται η ίδια η συνάρτηση

Παραδοσιακή δήλωση γεγονότων

- Κύριο μειονέκτημα
 - Μόνο έναν χειριστή ανά γεγονός

```
a.onclick = doSomething;
```

```
a.onclick = doSomethingElse;
```

- Εκτελείται μόνο η `doSomethingElse`

Δήλωση γεγονότων

- W3C δήλωση γεγονότων
 - Λύνει το κύριο πρόβλημα του παραδοσιακού τρόπου
 - Ορίζουμε πολλούς χειριστές στο ίδιο γεγονός εύκολα

W3C δήλωση γεγονότων

- Μέθοδος `addEventListener`
 - Υπάρχει στη DOM αναπαράσταση οποιουδήποτε στοιχείου
- Παράμετροι
 - Γεγονός
 - Το γεγονός στο οποίο θέλουμε να αντιστοιχίσουμε τον χειριστή
 - Χειριστής
 - Η συνάρτηση/κώδικας που θα τρέξει όταν συμβεί το γεγονός
 - Χρησιμοποίηση φάσης δέσμευσης
 - True/False. Συνήθως θα είναι false

W3C δήλωση γεγονότων

- `Click me`

```
function doSomething() {  
    alert( 'doSomething invoked' );  
    return true;  
}
```

```
var a = document.getElementById( 'foo' );
```

```
a.addEventListener( 'click', doSomething, false );
```



Στοιχείο που
αφορά το
γεγονός



Είδος
γεγονότος



Τι θέλουμε να
γίνει όταν συμβεί
(χειριστής)

Δήλωση πολλών χειριστών

```
a.addEventListener( 'click', doSomething, false );  
a.addEventListener( 'click', doSomethingElse, false );
```

- Δουλεύει όπως θα περιμέναμε, τρέχουν και τα δύο
- Μπορούμε να αφαιρέσουμε χειριστές εύκολα
- ```
a.removeEventListener('click', doSomething, false);
```

# Λέξη κλειδί `this`

- Αναφέρεται πάντα στον ιδιοκτήτη της συνάρτησης που εκτελείται
- Ιδιοκτήτης μιας συνάρτησης είναι το αντικείμενο του οποίου είναι μέθοδος

# Πείραμα

```
Click me
Click me
Click me
```

```
function wtf_is_this() {
 console.log(this);
}
```

```
var b = $('#bar')[0];
b.onclick = wtf_is_this;
```

```
var a = document.getElementById('foo');
a.addEventListener('click', wtf_is_this, false)
```

# Συμπέρασμα

- Το `this` αναφέρεται στη DOM αναπαράσταση του στοιχείου του οποίου πυροδοτήθηκε το γεγονός
  - Με εξαίρεση τον HTML τρόπο (μη το χρησιμοποιείτε!)
- Με τον ίδιο κώδικα μπορούμε να αλλάζουμε ιδιότητες του στοιχείου του γεγονότος

# Παράδειγμα

```
w3C method
```

```
function clickHandler() {
 $(this).css('color', 'red');
}
```

```
var a = $('#foo')[0];
```

```
a.addEventListener('click', clickHandler, false)
```

# Το αντικείμενο γεγονός

- Ένα αντικείμενο που περιέχει πληροφορίες για το γεγονός
- Πληροφορίες όπως:
  - Που ήταν το ποντίκι όταν συνέβει
  - Τι πλήκτρα είχε πατήσει ο χρήστης
  - Ποιο κουμπί του ποντικιού είχε πατήσει



# Το αντικείμενο γεγονός

- Πρόσβαση σε αυτό
  - Γίνεται μέσω του χειριστή
  - Αρκεί να ορίσουμε σε αυτόν μία παράμετρο
  - Θα περάσει σε αυτή αυτόματα το αντικείμενο γεγονός

# Χειριστές με παράμετρο

```
w3C method
```

```
function doSomething(e) {
 //Το e είναι το αντικείμενο γεγονόςτος
}
```

```
var a = $('#foo')[0];
```

```
a.addEventListener('click', doSomething, false)
```

# jQuery to the rescue

- Η κατάσταση του αντικειμένου γεγονόςτος μεταξύ των browser προκαλεί πονοκέφαλο
- Η jQuery:
  - Διαχειρίζεται όλες τις διαφορές μεταξύ των browser
  - Μας προσφέρει «κανονικοποιημένο» αντικείμενο γεγονόςτος παντού
  - Έχει, όπως η W3C μέθοδος, το **this** να αναφέρεται στο στοιχείο

# jQuery to the rescue

- Δήλωση γεγονότων με jQuery
- `$( 'επιλογέας' ).γεγονός( function( e ) { ... } )`
- Παράδειγμα

```
$('span').click(function(e) {
 alert('Link clicked');
});
```

# Ανάδυση γεγονότων

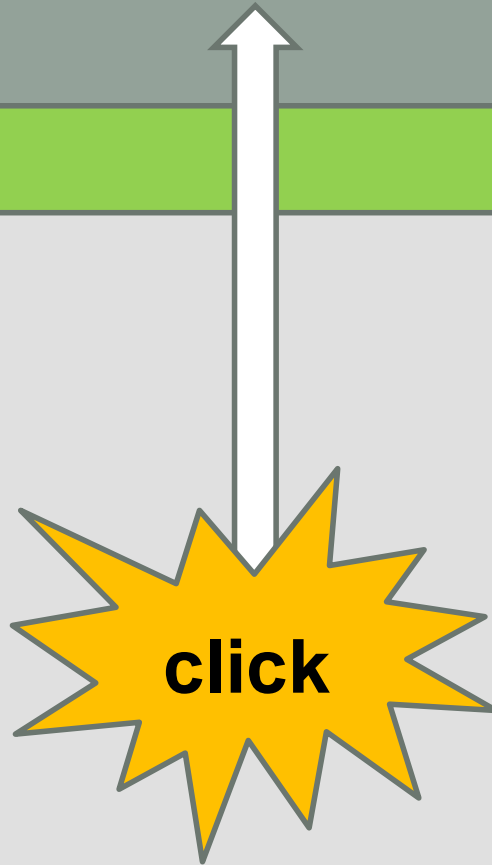
- Ένα γεγονός σε ένα στοιχείο «αναδύεται» μέχρι τον τελευταίο πατέρα
- Λύνει προβλήματα όπως
  - Πατέρας και παιδί έχουν χειριστές για το κλικ. Ποιο τρέχει πρώτα;
  - Σωστό drag 'n drop
- Αναλυτική εξήγηση  
[http://www.quirksmode.org/js/events\\_order.html](http://www.quirksmode.org/js/events_order.html)

document

body

div

click



# Πόσα γεγονότα υπάρχουν;

- Ποντικιού
  - mousedown
  - mouseup
  - click
- Πληκτρολογίου
  - keydown
  - keypress
  - keyup
- Φόρμας
  - submit
  - reset
- *Και άλλα πολλά..*

# Γεγονός mousedown

- Συμβαίνει όταν ο χρήστης πατήσει ένα κουμπί από το ποντίκι του και ο κέρσοντας είναι πάνω σε ένα στοιχείο
- Μπορεί να χρησιμοποιηθεί με κάθε στοιχείο



# Γεγονός mouseup

- Συμβαίνει όταν ο χρήστης αφήσει ένα κουμπί από το ποντίκι του και ο κέρσοντας είναι πάνω σε ένα στοιχείο
- Μπορεί να χρησιμοποιηθεί με κάθε στοιχείο

# Γεγονός mousemove

- Συμβαίνει όταν ο χρήστης κουνήσει το ποντίκι του και ο κέρσοντας είναι πάνω σε ένα στοιχείο
- Μπορεί να χρησιμοποιηθεί με κάθε στοιχείο

# Αντικείμενο γεγονός

- Ιδιότητες σχετικές με το ποντίκι
  - pageX
    - Η απόσταση του ποντικιού από την αριστερή πλευρά του document
  - pageY
    - Η απόσταση του ποντικιού από την πάνω πλευρά του document
  - which
    - Ποιο κουμπί του ποντικιού ήταν πατημένο
    - 1 – Αριστερό κουμπί
    - 2 – Μεσαίο κουμπί
    - 3 – Δεξί κουμπί

# Συνδυασμός

- Παράγουμε το εφέ drag n' drop χρησιμοποιώντας
  - mousedown
  - mousemove
  - mouseup
  - Αντικείμενο γεγονόςτος

# Γεγονός click

- Συμβαίνει συμβούν συνεχόμενα mousedown και mouseup στο ίδιο στοιχείο
- Μπορεί να χρησιμοποιηθεί με κάθε στοιχείο
- Υλοποίηση κουμπιών διεπαφής χρήστη

# Γεγονός keydown

- Συμβαίνει όταν ο χρήστης πιέσει ένα πλήκτρο στο πληκτρολόγιό του
- Πυροδοτείται στο στοιχείο που είναι εστιασμένο
- Τα στοιχεία που μπορούν να εστιαστούν διαφέρουν ανά browser

# Γεγονός keyup

- Συμβαίνει όταν ο χρήστης αφήσει ένα πλήκτρο στο πληκτρολόγιό του
- Πυροδοτείται στο στοιχείο που είναι εστιασμένο
- Τα στοιχεία που μπορούν να εστιαστούν διαφέρουν ανά browser

# Γεγονός keypress

- Συμβαίνει όταν ο χρήστης πιέσει ένα πλήκτρο στο πληκτρολόγιό του
- Πυροδοτείται στο στοιχείο που είναι εστιασμένο
- Τα στοιχεία που μπορούν να εστιαστούν διαφέρουν ανά browser



# Αντικείμενο γεγονός

- Ιδιότητες σχετικές με το πληκτρολόγιο
  - altKey
    - Είναι true αν ήταν πατημένο το πλήκτρο alt
  - ctrlKey
    - Είναι true αν ήταν πατημένο το πλήκτρο control
  - which
    - Ο κωδικός του κουμπιού του πληκτρολογίου

# Τι μπορούμε να κάνουμε

- Συντομεύσεις πληκτρολογίου
  - Twitter
  - Gmail

# Γεγονός submit

- Συμβαίνει όταν ο χρήστης υποβάλει μία φόρμα
- Χρησιμοποιείται μόνο σε φόρμες
- Επιτρέπει validation στην μεριά του χρήστη

# Γεγονός reset

- Συμβαίνει όταν ο χρήστης κάνει reset μία φόρμα
- Χρησιμοποιείται μόνο σε φόρμες

# Γεγονός load

- Συμβαίνει όταν:
  - Φορτώσει όλο το body
  - Κατέβουν όλες οι εικόνες
- Χρήσιμο όταν ο κώδικάς μας προυποθέτει να υπάρχει το document

# Έλεγχος προεπιλεγμένης λειτουργίας

- Προεπιλεγμένες λειτουργίες

| Γεγονός         | Λειτουργία             |
|-----------------|------------------------|
| Υποβολή Φόρμας  | Υποβολή                |
| Κλικ συνδέσμου  | Μετάβαση στον σύνδεσμο |
| Κλικ σε κείμενο | Επιλογή κειμένου       |

- Αν ο χειριστής κάποιου γεγονότος επιστρέφει false διακόπτεται η προεπιλεγμένη λειτουργία

# Μάθαμε

- Τι είναι τα γεγονότα
- Πως ορίζω χειριστή σε κάποιο γεγονός
- Τι είναι το this
- Πόσα είδη γεγονότων υπάρχουν
- Ανάδυση γεγονότων

# Την επόμενη φορά

- Ανάπτυξη web εφαρμογών για συσκευές mobile
  - Προσκεκλημένος διδάσκοντας Ανδρέας Νομικός