

ZAJĘCIA SPECJALISTYCZNE

JavaScript



TEMAT 3-4: Obsługa zdarzeń.

Autor dokumentu: Wojciech Galiński

poniedziałek, 6 października 2014 r.

312[01]/T,SP/MENiS/2004.06.14

ŹRÓDŁA WIEDZY: <http://pl.wikipedia.org/>, <http://webmaster.helion.pl/index.php/kurs-javascript>,
<http://krook.org/jsdom/>, <http://www.dynamicdrive.com/>, <http://www.w3schools.com>.

Zagadnienia obowiązkowe

1. **Zdarzenie** – (ang. event) „zapis zajścia w systemie komputerowym określonej sytuacji”, np. kliknięcie klawiszem myszy (click), najechanie myszą na element (mouseover), itd.
2. **Procedura obsługi zdarzenia** – procedura wykonywana po zajściu określonego zdarzenia, np. onclick, onmouseover, itd.
3. **Przypisywanie procedury (funkcji) do obsługi zdarzenia** – aby przypisać procedurę (funkcję) do obsługi zdarzenia, należy dodać do wybranego znacznika odpowiedni atrybut, np.

<znacznik onzdarzenie="nazwa_procedury(parametry)" />

Procedura (funkcja) może być predefiniowana albo zdefiniowana przez użytkownika.

4. **Lista ważniejszych zdarzeń w języku JavaScript** – podzielone są na następujące kategorie:
 - ➔ **obsługa zdarzeń dotyczących dokumentu jako całości** – definiujemy je jako atrybuty znacznika „body”. Wyróżniamy:
 - ✓ **onload** – wywoływana bezpośrednio po wczytaniu dokumentu HTML;
 - ✓ **onbeforeunload** – wywoływana bezpośrednio przed rozpoczęciem zamykania dokumentu HTML. Obecnie obsługiwane przez **Firefox** i **Chromium**: wyświetla się okienko, w którym trzeba potwierdzić zamknięcie okna (zabezpieczenie przed przypadkowym zamknięciem dokumentu), np.
`window.onbeforeunload = function() { return ''; };`
 - ✓ **onunload** – wywoływana bezpośrednio po zamknięciu dokumentu HTML;
 - ✓ **onresize** – wywoływana bezpośrednio po zmianie rozmiaru obszaru dokumentu HTML.
 - ➔ **obsługa zdarzeń dotyczących aktywnego elementu** – wyróżniamy:
 - ✓ **onfocus** – wywoływana bezpośrednio po zaznaczeniu obiektu;
 - ✓ **onblur** – wywoływana bezpośrednio po odznaczeniu obiektu.
 - ➔ **obsługa zdarzeń dotyczących formularzy** – wyróżniamy:
 - ✓ **onsubmit** – wywoływana po zatwierdzeniu danych wewnątrz formularza;
 - ✓ **onreset** – wywoływana bezpośrednio przed resetowaniem kontroltek formularza;
 - ✓ **onchange** – wywoływana bezpośrednio po wykryciu zmian w kontrolce formularza (bezpośrednio przed zdarzeniem „blur”, a nie po każdym naciśnięciu klawisza klawiatury) (dotyczy kontroltek formularza: **input**, **select**, **textarea**);
 - ✓ **onselect** – wywoływana bezpośrednio po zaznaczeniu (fragmentu) tekstu (dotyczy znaczników formularza: `<input type="text" />` i `<textarea></textarea>`).
 - ➔ **obsługa zdarzeń dotyczących obsługi klawiatury** – wyróżniamy:
 - ✓ **onkeydown** – wywoływana bezpośrednio po naciśnięciu klawisza klawiatury;
 - ✓ **onkeypress** – wywoływana bezpośrednio po zajściu zdarzenia „keyup”, umożliwia także odczytanie kodu wciśniętego klawisza, np.
`<script type="text/javascript">
function wyswietl_kod_klawisza(e)
{ var kod=e.keyCode? e.keyCode : e.charCode; alert(kod); }
</script>
<form><input type="text" size="2" maxlength="1"`

```
onkeyup="wyswietl_kod_klawisza(event); this.select()" /></form>
```

- ✓ **onkeyup** – jest wywoływana bezpośrednio po zwolnieniu klawisza klawiatury.

➔ **obsługa zdarzeń dotyczących ruchu myszy** – wyróżniamy:

- ✓ **onmouseover** – wywoływana jednokrotnie po najechaniu myszą nad wybrany obszar;
- ✓ **onmousemove** – wywoływana jednokrotnie po opuszczeniu myszą wybranego obszaru;
- ✓ **onmouseout** – wywoływana po każdorazowym poruszeniu myszą nad wybranym obszarem, np. (zawartość znaczników <body></body>)

```
<script type="text/javascript"> function addText(tekst)
{ document.getElementById('out').innerHTML += ' ' + tekst; }</script>
<div id="out" style="min-height: 50px; background: red;"
  onmouseover="addText('over')" onmousemove="addText('move')"
  onmouseout="addText('out')"></div>
```

➔ **obsługa zdarzeń dotyczących klikania myszą** – wyróżniamy:

- ✓ **onmousedown** – wywoływana bezpośrednio po naciśnięciu przycisku myszy, umożliwia odczytanie wciśniętego przycisku myszy, np.

```
document.onmousedown = function onmousedownHandler(e)
{ if(window.event) { mousenum = event.button; } // IE
  else if(e.which) { mousenum = e.which; } // F, O, CH
  alert(mousenum); }
```
- ✓ **onmouseup** – wywoływana bezpośrednio po zwolnieniu przycisku myszy;
- ✓ **onclick** – wywoływana bezpośrednio po zajściu zdarzenia „mouseup”;
- ✓ **ondblclick** – wywoływana bezpośrednio po dwukrotnym kliknięciu myszą w krótkim odstępie czasu.

➔ **obsługa zdarzeń dotyczących obrazów** – wyróżniamy:

- ✓ **onabort** – wywoływana bezpośrednio po tym, gdy wczytywanie obrazka zostanie przerwane z przyczyn niezależnych od użytkownika przeglądarki internetowej;
- ✓ **onerror** – wywoływana bezpośrednio po tym, gdy wystąpi problem z załadowaniem obrazka do znacznika , np. z powodu usunięcia go z lokalizacji podanej atrybutem „src”.

5. **Dynamiczne przypisywanie procedur obsługi zdarzeń** – to przypisywanie procedur obsługi zdarzeń za pomocą instrukcji języka JavaScript:

obj.onzdarzenie = nazwa_procedury(parametry);

PRZYKŁAD:

```
function onloadHandler() { alert('Dokument HTML został wczytany.');
```

```
window.onload = onloadHandler();
```

W przykładzie procedury obsługi zdarzenia „**keypress**” można znaleźć przykład definicji i dynamicznego przypisania procedury w jednej instrukcji.

Zadania

1. Zapoznaj się z przedstawionymi przykładami. Uruchom je we własnych skryptach i przeanalizuj ich działanie.
2. Wypełnić dokument HTML 4 akapitami tekstu wygenerowanego pod adresem: <http://pl.lipsum.com>. Po wczytaniu dokumentu do przeglądarki wyświetlić informację „Zakończono wczytywanie dokumentu”. Przed zamknięciem dokumentu HTML wyświetlić okienko potwierdzające tą operację.
3. Wstawić informację o bieżących rozmiarach dokumentu HTML, która aktualizuje się po zmianie rozmiaru przeglądarki.
4. Wstawić do przeglądarki prostokąt, które po kliknięciu prawym klawiszem myszy przesunie się w prawo, a po kliknięciu lewym klawiszem myszy przesunie się w lewo. Po podwójnym kliknięciu prostokąt wraca do początkowej pozycji.