

Hacki, czyli jak obejść niedoskonałości przeglądarek

Jak z pewnością spostrzegłeś podczas kursu wielokrotnie pisałem, że tego nie obsługuje jedna przeglądarka, a tego inna. Najczęściej jednak problemy stwarza MSIE, dlatego głównie na nim skupimy się podczas tej lekcji. Hacki najczęściej wykorzystują dziurę w jakiejś przeglądarce. Zobacz przykłady:

Na początek ukrywanie fragmentów kodu CSS dla MSIE:

```
/* w pliku css */ table { width: 100px; voice-family: ""}""; // 1
voice-family: inherit; // 2 width: 200px; } // 3
```

Opis:

* - ten hack nazywa się "box model hack";

1 - 'voice-family' jest analogiczny do 'font-family', z tą różnicą, że jest wykorzystywane do odczytywania - na głos - przez przeglądarkę strony [w praktyce chyba tylko Opera 8.0 korzysta z tego, chyba:-)]. Przypisana wartość nic nie znaczy - powoduje, że MSIE przestaje czytać style od tego miejsca [zaczyna czytać dopiero kolejną regułę].

2 - przypisujemy do 'voice-family' wartość domyślną, żeby "naprawić" wcześniejszy - specjalny błąd.

3 - po raz kolejny definiujemy właściwości, które chcemy ukryć przed MSIE Oczywiście napisałem jedynie przykładowo. Pamiętaj, że jeśli określimy coś raz, a później jeszcze raz to ta ostatnia opcja jest ważniejsza.

```
/* tak to czyta MSIE: */ table { width: 100px; voice-family: ""}""; // i już
kończy, dalej nie będzie sprawdzał tej reguły - skoczy do następnej voice-family: inherit; width:
200px; } /* a nowoczesne przeglądarki: */ table { width: 100px; voice-family: ""}""; //
nieprawidłowa wartość, ale czyta dalej, bo to przecież nie koniec świata:-) voice-family: inherit;
width: 200px; }
```

Chyba wszystko jasne. Efekt jest bardzo ciekawy. Choć w przykładzie może się wydawać, że daje niewiele możliwości, to można dzięki niemu czasem rozwiązać pewne problemy. Ja osobiście rzadko używam tego hacka, gdyż częściej zdarza mi się sytuacja, kiedy to MSIE wymaga więcej kodu niż nowoczesne przeglądarki. A teraz kolejny ciekawy efekt:

```
table { właściwości: wartości czytane przez wszystkie przeglądarki; } /* Komentarz
wieloliniowy, jednak przez dodanie " na końcu MSIE przeoczy koniec i będzie uważał cały
pozostały kod za komentarz */ table { właściwości: czytane tylko przez nowsze przeglądarki;
} /* Komentarz wieloliniowy zamknięty normalnie MSIE będzie uważał zamknięcie tego
komentarza jako zamknięcie poprzedniego, a otwierające "*" zignoruje */
```

* - ten hack nazywa się "commented backslash hack";

Oczywiście można - używając hacków - tworzyć reguły dla innych selektorów niż 'table'. Zasada działania tego hacku jest podobna, gdyż ukrywa się fragmenty kodu dla MSIE. Z tą różnicą, że poprzedniego używa się do ukrywania pojedynczych właściwości, a tego do całych

reguł.

I jeszcze jeden hack, którego zadaniem jest ukrywanie kawałków kodu przed MSIE.

```
table {   właściwości: wartości czytane przez wszystkie przeglądarki; }   body>table {  
właściwości: czytane tylko przez nowsze przeglądarki; }
```

Jest to jeden z ciekawszych hacków. M.in. dlatego, że pozwala na obejście niedoskonałości MSIE **6.0**. Pierwsza reguła jest czytana całkowicie normalnie - przez wszystkie przeglądarki. Druga reguła jednak jest rozpoznawana jedynie przez nowoczesne przeglądarki - choć jest w pełni zgodna ze standardami. Wskazuje się po prostu, że body jest "starsze" od table. Oczywiście można stosować taki zapis do wszystkich selektorów - należy jednak pamiętać, o tym, aby "ta nierówność" była spełniona.

Osobny arkusz stylów dla MSIE:

No już próbowałeś wszystkiego - prośbą, groźbą - a on nic! To "jak nie młotem, to pałą"; i daj mu osobny plik CSS

```
.. <head> <link rel="stylesheet" href="style-normal.css"  
type="text/css" /> <!-- [if IE]>           // 1 <style type="text/css">  
// 2 table { width: 100px; }           // 2 </style>           // 2 <![endif] -->           // 3  
</head> ..
```

* - ten hack nazywa się "warunkowy komentarz";

1 - w HTML/XHTML taki zapis: '<!-- cośtam cośtam -->' jest traktowany jako komentarz i nie jest w żaden sposób interpretowany. Jeśli jednak umieścimy '[if IE]' MSIE zaczyna interpretować zawartość tego komentarza.

2 - W powyższym przykładzie dodaliśmy jedną linię kodu CSS, ale moglibyśmy dodać jej dużo więcej, lub nawet po raz kolejny załączyć plik css z dodatkowym opisem stylu dla MSIE. Pamiętaj, żeby jednak odwołanie do stylu dla wszystkich przeglądarek umieścić nad tym zapisem.

3 - zamykamy obszar czytany przez MSIE, a także cały komentarz.

Ten sposób jest używany przez mnie najczęściej, ponieważ daje mi to, czego ja najbardziej chcę - opis właściwości tylko dla MSIE. Teraz wróć do lekcji "Pozycjonowanie - gotowa strona"; i dodaj tam odpowiednie linie kodu [HTML i CSS], aby sprawić, by strona w Operze/FF była "elastyczna";, a w MSIE prezentowała się nieźle.

Osobny arkusz stylów, ale za pomocą JS

Hacki, czyli jak obejść niedoskonałości przeglądarek

Wpisany przez Patryk yarpo Jar
piątek, 18 września 2009 02:52

```
... <head> <script type="text/javascript">                                // 1  if (-1 !=
navigator.userAgent.indexOf("&quot;MSIE&quot;")) {      // 2  document.write('<link
rel="stylesheet" type="text/css" href="msie.css" />'); } // 3
</script>                                // 1  </head> ...
```

Opis:

- 1 - obszar skryptu JS.
- 2 - sprawdzam, czy internauta używa MSIE. Jeśli tak...
- 3 - używając metody write() piszę dodatkową linię kodu wczytującą inny arkusz stylów.

Można oczywiście dopisać tym sposobem tylko fragment kodu - jedną regułę, a można wstawić cały plik. Zasada jest taka sama jak w przykładzie powyżej - i efekt też identyczny.

Uwaga: niektórzy mają wyłączoną obsługę JS, przez co ten sposób wydaje się być gorszy. Jednak jeśli ktoś używa MSIE, to raczej nie wyłącza obsługi JS [to domena bardziej doświadczonych internautów, którzy używają lepszych programów niż MSIE].

Uwaga: niektórzy po zainstalowaniu Opery nie zmieniają w jej ustawieniach opcji 'Przedstawiaj jako MSIE', przez co nawet Opera będzie czytać ten plik/regułę, co może spowodować, że strona nie będzie się w niej prezentować tak dobrze jak w FF - raczej nie ma możliwości, aby prezentowała się gorzej niż w MSIE:-/.