

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

Czasem przydatna staje się wiedza o tym, które ze znanych rozwiązań jest szybsze. Oczywiście można polegać na teoretycznej wiedzy, podierać się nawet złożonością algorytmów itd. itd. Często jednak starczy kilka prostych testów. Oto obiekt pozwalający w banalny sposób sprawdzić jak szybko wykonał się skrypt. Nazwałem go yProfiler.

Kod

```
{codecitation class='brush: js'}// obiekt yProfiler - zliczanie czasu wykonania skryptu
```

```
// autor: Patryk yarpo Jar
```

```
var yProfiler = function() {
```

```
// private:
```

```
    var nTimeStart = 0;
```

```
    var nTimeStop = 0;
```

```
    var nLastDuration = -1;
```

```
    var oDate = null;
```

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
function fStart() {  
  
    oDate = new Date();  
  
    nTimeStart = oDate.getTime();  
  
    return nTimeStart;  
  
}  
  
function fStop() {  
  
    oDate = new Date();  
  
    nTimeStop = oDate.getTime();  
  
    nLastDuration = nTimeStop-nTimeStart;  
  
    return nLastDuration;  
  
}
```

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
}
```

```
function fTime() {
```

```
    // nie wystartowano jeszcze :/
```

```
    if (!lnTimeStart) {
```

```
        return false;
```

```
    }
```

```
    // nie skonczone jeszcze
```

```
    if (!lnTimeStop) {
```

```
        fStop();
```

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
}
```

```
if (nLastDuration < 0 || isNaN(nLastDuration)) {
```

```
    nLastDuration = nTimeStop - nTimeStart;
```

```
}
```

```
return nLastDuration;
```

```
}
```

```
function fGetStart() {
```

```
    return (nTimeStart) ? nTimeStart : false;
```

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
}
```

```
function fGetStop() {
```

```
    return (nTimeStop) ? nTimeStop : false;
```

```
}
```

```
function fLoop(func, n) {
```

```
    fStart();
```

```
    for(i = 0; i < n; i++) {
```

```
        func();
```

```
    }
```

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
fStop();
```

```
return nLastDuration;
```

```
}
```

```
// konstruktor:
```

```
fStart(); // domyslnie startuje
```

```
// public
```

```
return {
```

```
    start : fStart,
```

```
    stop  : fStop,
```

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
time : fTime,  
  
getStart: fGetStart,  
  
getStop : fGetStop,  
  
loop : fLoop  
  
};  
  
};{/codecitation}
```

W tym obiekcie można dostrzec właściwości `private`, o których więcej przeczytasz na stronie [Douga Crockforda](#)

. Wykorzystałem tu także wzorzec modułowy do stworzenia obiektu (kiedyś opiszę kilka sposobów na produkcję obiektów) dzięki czemu

nie

trzeba, a nawet

[nie należy używać operatora `new`](#)

(przeczytaj dlaczego - blog YUI).

Zastosowanie:

Mierzenie czasu wykonania skryptu - własny obiekt

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

```
{codecitation class='brush: js'}var time = yProfiler();
```

```
function test1() {
```

```
    var n = 10000;
```

```
    for(var i=0; i != n; i++) {
```

```
        // as
```

```
    }
```

```
    var n = 10000;
```

```
    for(var i=n; i>0; i-- ) {
```

```
        // as
```

```
    }
```

```
{/codecitation}
```

Powyższy kod pozwala na wykorzystanie metody `yProfiler.loop()`. Przyjmuje ona dwa parametry - pierwszy to referencja na funkcję, która ma być wywołana, drugi - liczba wywołań tej funkcji

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 22:28

(patrz prywatną metodę `yProfiler.fLoop()`). Na koniec wszystko jest wyświetlane za pomocą alertów. Jest jeszcze drugi sposób wywołania.

Zastosowanie 2:

```
{codecitation}    var time = yProfiler();
```

```
    time.start();
```

```
    function test1() {
```

```
        var n = 10000;
```

```
        for(var i=0; i <= n; i++) {
```

```
    }/codecitation}
```

Jeśli zajrzysz do środka tego obiektu to pierwsza metoda jest niejawnym wywołaniem drugiej. Przyjemnych testów wydajności. Pamiętaj, że nie można traktować wyników jako wyroczni. Wiele zależy od przeglądarki, sprzętu i innych czynników. Nie gwarantuję także, że obiekt ten jest wolny od błędów - używasz na swoją odpowiedzialność.