**Obiekt Window**

wtorek, 16 marzec 2010 05:58 Administrator

[Email](http://webmaster.helion.pl/index.php/component/mailto/?tmpl=component&link=aHR0cDovL3dlYm1hc3Rlci5oZWxpb24ucGwvaW5kZXgucGhwL2t1cnNqcy13c3BvbHByYWNhLXotcHJ6ZWdsYWRhcmthLW1vZGVsLWRvbS9rdXJzanMtb2JpZWt0LXdpbmRvdy8zNTgta3Vyc2pzLW9iaWVrdC13aW5kb3c%3D)[Drukuj](http://webmaster.helion.pl/index.php/kursjs-wspolpraca-z-przegladarka-model-dom/kursjs-obiekt-window/29-kurs-javascript/358-kursjs-obiekt-window?tmpl=component&print=1&page=)[PDF](http://webmaster.helion.pl/index.php/kursjs-wspolpraca-z-przegladarka-model-dom/kursjs-obiekt-window/29-kurs-javascript/358-kursjs-obiekt-window?format=pdf)

Obiekt window znajduje się na szczycie hierarchii obiektów i reprezentuje okno przeglądarki. Jest też obiektem domyślnym, tzn. do jego metod i właściwości można się odwoływać bezpośrednio, z pominięciem jego nazwy. Na przykład zastosowana na początku kursu instrukcja alert("tekst") wyświetlająca okno dialogowe ze zdefiniowanym tekstem to nic innego jak metoda obiektu window, wywołanie mogłoby zatem wyglądać również następująco:

window.alert("tekst");

Tak samo jest z pozostałymi składowymi tego obiektu. Listę składowych dla danej przeglądarki można uzyskać po zastosowaniu skryptu z listingu 4.1.

Listing 4.1. Wyświetlenie składowych obiektu window

<script type="text/javascript">

for(index in window){

  document.write(index +" = " + window[index] + "<br />");

}

</script>

**Właściwości obiektu window**

Obiekt window zawiera wiele właściwości, niestety każda przeglądarka ma własny ich zestaw. W tabeli 4.1 zostały przedstawione najpopularniejsze z nich, wraz z informacją, które z przeglądarek je obsługują, a także czy dana właściwość występuje w standardzie rekomendowanym przez organizację W3C.

Tabela 4.1. Wybrane właściwości obiektu window

|  |  |  |
| --- | --- | --- |
| Nazwa | Znaczenie | Dostępność |
| closed | Ustawiona na true oznacza, że okno zostało zamknięte. | FF, IE, NN, OP, W3C |
| defaultStatus | Domyślny tekst wyświetlany na pasku stanu. | FF, IE, NN, OP, W3C |
| document | Obiekt document zawierający elementy wyświetlanej witryny. | FF, IE, NN, OP, W3C |
| event | Zawiera odniesienie do obiektu typu Event opisującego ostatnie zdarzenie. Używany w modelu zdarzeń przeglądarki Internet Explorer. | IE, OP |
| frames | Tablica zawierająca odniesienia do ramek danego okna. | FF, IE, NN, OP |
| history | Obiekt History zawierający historię odwiedzin. | FF, IE, NN, OP, W3C |
| innerWidth | Wewnętrzna wysokość obszaru okna. | NN, OP, FF |
| innerHeight | Wewnętrzna szerokość obszaru okna. | NN, OP, FF |
| length | Liczba ramek potomnych zawartych w oknie. | FF, IE, NN, OP, W3C |
| location | Obiekt Location zawierający URL aktualnego dokumentu. | FF, IE, NN, OP, W3C |
| locationbar | Obiekt pozwalający na określanie, czy ma być widoczny pasek adresu. Posiada właściwość visible, której można przypisywać wartości true i false. | NN, FF |
| menubar | Obiekt pozwalający na określanie, czy ma być widoczny pasek menu. Posiada właściwość visible, której można przypisywać wartości true i false. | NN, FF |
| name | Nazwa bieżącego okna. | FF, IE, NN, OP, W3C |
| opener | Jeżeli bieżące okno zostało otwarte za pomocą metody open, właściwość opener zawiera odniesienie do okna źródłowego. | NN, OP, FF, W3C |
| outerHeight | Zewnętrzna wysokość obszaru okna. | NN, OP, FF, W3C |
| outerWidth | Zewnętrzna szerokość obszaru okna. | NN, OP, FF, W3C |
| pageXOffset | Przesunięcie bieżącej strony w poziomie względem lewego górnego rogu okna (szerokość zasłoniętego obszaru). | NN, OP, FF, W3C |
| pageYOffset | Przesunięcie bieżącej strony w pionie względem lewego górnego rogu okna (wysokość zasłoniętego obszaru). | NN, OP, FF, W3C |
| parent | Odniesienie do okna źródłowego. | FF, IE, NN, OP, W3C |
| personalbar | Obiekt pozwalający na określanie, czy ma być widoczny pasek ustawień osobistych. Posiada właściwość visible, której można przypisywać wartości true i false. | NN, FF |
| scrollbars | Obiekt pozwalający na określanie, czy mają być widoczne paski przewijania. Posiada właściwość visible, której można przypisywać wartości true i false. | NN, FF |
| self | Odniesienie do bieżącego aktywnego okna lub ramki. | FF, IE, NN, OP, W3C |
| status | Tekst wyświetlany na pasku stanu. | FF, IE, NN, OP, W3C |
| statusbar | Obiekt pozwalający na określanie, czy ma być widoczny pasek stanu. Posiada właściwość visible, której można przypisywać wartości true i false. | NN, FF |
| toolbar | Obiekt pozwalający na określanie, czy ma być widoczny pasek narzędziowy. Posiada właściwość visible, której można przypisywać wartości true i false. | NN, FF |
| top | Odniesienie do okna znajdującego się najwyżej w hierarchii. | FF, IE, NN, OP, W3C |
| window | Odniesienie do bieżącego okna lub ramki | FF, IE, NN, OP |

Gdyby zatem istniała potrzeba, aby podczas ładowania witryna umieszczała na pasku stanu własny tekst, należałoby w sekcji body umieścić prosty skrypt w postaci widocznej na listingu 4.2.

Listing 4.2. Zmiana napisu na pasku stanu

<script type="text/javascript">

window.status = "Nowy tekst...";

</script>

Aby natomiast poznać rozmiary okna wskazywane przez obiekt window (w przeglądarkach Firefox, Netscape Navigator i Opera), można użyć skryptu z listingu 4.3.

Listing 4.3. Wyświetlenie rozmiarów okna przeglądarki

<script type="text/javascript">

document.write("innerWidth = " + window.innerWidth + "<br />");

document.write("innerHeight = " + window.innerHeight + "<br />");

document.write("outerWidth = " + window.outerWidth + "<br />");

document.write("outerHeight = " + window.outerHeight + "<br />");

</script>

**Metody obiektu Window**

Poniżej zebrane zostały najpopularniejsze metody obiektu window, część z nich została omówiona dokładniej wraz z odpowiednimi przykładami zastosowania, a pozostałe umieszczono w tabeli znajdującej się w podpunkcie „Pozostałe popularne metody”.

**Metoda alert**

Wywołanie: alert(argumenty)

Dostępność: FF, IE4, NN3, OP9

Metoda alert była już prezentowana i powoduje wyświetlenie okna dialogowego ze zdefiniowaną zawartością. Podobnie jak w przypadku document.write, można wyświetlać zarówno ciągi znaków (które muszą być ujęte w znaki cudzysłowu lub apostrofu), jak i wartości liczbowe oraz zawartość zmiennych. Gdy argumentem będzie obiekt, zostanie wyświetlona wraz z jego nazwą taka informacja, jak na rysunku 4.2.

|  |
| --- |
| http://webmaster.helion.pl/images/kursjs/r04_03_pliki/image002.jpg |
| Rysunek 4.2. Obiekt jako argument metody alert |

**Metoda clearInterval**

Wywołanie: clearInterval(id)

Dostępność: FF, IE4, NN4, OP9

Zatrzymuje timer o identyfikatorze id uruchomiony przez wywołanie setInterval. Przykład wykorzystania zostanie podany przy opisie metody setInterval.

**Metoda clearTimeout**

Wywołanie: clearTimeout(id)

Dostępność: FF, IE4, NN4, OP9

Zatrzymuje timer o identyfikatorze id uruchomiony przez wywołanie setTimeout. Przykład wykorzystania zostanie podany przy opisie metody setTimeout.

**Metoda confirm**

Wywołanie: confirm(argumenty)

Dostępność: FF, IE4, NN4, OP0

Metoda confirm wyświetla okno dialogowe zawierające przyciski OK i Anuluj (Cancel). Postać tego okna będzie podobna we wszystkich przeglądarkach (rysunek 4.3).

|  |
| --- |
| http://webmaster.helion.pl/images/kursjs/r04_03_pliki/image003.jpg |
| Rysunek 4.3. Okno wyświetlane przez metodę confirm |

Metoda pozwala stwierdzić, który przycisk został kliknięty przez użytkownika, zwraca bowiem wartość true, jeśli wybrany został przycisk OK, oraz false, jeśli był to przycisk Anuluj bądź okno dialogowe zostało zamknięte w inny sposób. W jaki sposób odczytać informację zwróconą przez wywołanie confirm, ilustruje skrypt z listingu 4.4.

Listing 4.4. Użycie metody confirm

<script type="text/javascript">

var result = confirm("Czy wyświetlić wszystkie dane?");

if(result == true){

  document.write("Został kliknięty przycisk OK.");

}

else{

  document.write("Został kliknięty przycisk Anuluj.");

}

</script>

Instrukcja if bada tu wartość zwróconą przez metodę confirm, a zapisaną w zmiennej pomocniczej. W zależność od tego, czy jest to true (kliknięto przycisk OK), czy false (kliknięto przycisk Anuluj), wyświetla na ekranie odpowiedni tekst.

Nie ma przy tym formalnej konieczności używania w instrukcji if warunku w postaci result == true, z powodzeniem można użyć konstrukcji skróconej typu if(result) — efekt działania będzie taki sam. Można również zrezygnować ze zmiennej pomocniczej result i wywołanie metody confirm umieścić bezpośrednio w if. Takie rozwiązanie zostało zaprezentowane na listingu 4.5.

Listing 4.5. Bezpośrednie użycie wartości zwróconej przez confirm

<script type="text/javascript">

if(confirm("Czy wyświetlić wszystkie dane?")){

  document.write("Został kliknięty przycisk OK.");

}

else{

  document.write("Został kliknięty przycisk Anuluj.");

}

</script>

**Metoda moveBy**

Wywołanie: moveBy(px, py)

Dostępność: FF, IE4, NN4, OP9

Metoda moveBy pozwala przesunąć okno przeglądarki (względem bieżącego położenia) o zadaną liczbę pikseli w pionie i (lub) w poziomie. Przesunięcie w poziomie jest określane przez argument px, a w pionie — przez py. Jeśli argument px jest dodatni, następuje przesunięcie w prawo, a jeśli jest ujemny — w lewo. Analogicznie: jeśli argument py jest ujemny, następuje przesunięcie w górę, a kiedy dodatni — w dół. Na listingu 4.6 został zaprezentowany prosty skrypt ilustrujący możliwości metody. Przesuwa on okno przeglądarki o 100 pikseli w prawo, następnie o 200 pikseli w lewo, a później o kolejne 100 pikseli w prawo, przywracając okno do położenia pierwotnego. Ponieważ zmiana położenia odbywa się w pętli for w krokach co jeden piksel, osiągamy płynną animację ruchu.

Listing 4.6. Ilustracja działania metody moveBy

<script type="text/javascript">

for(i = 0; i < 100; i++){

  window.moveBy(1, 0);

}

for(i = 0; i < 200; i++){

  window.moveBy(-1, 0);

}

for(i = 0; i < 100; i++){

  window.moveBy(1, 0);

}

</script>

**Metoda moveTo**

Wywołanie: moveTo(px, py)

Dostępność: FF, IE4, NN4, OP9

Metoda przesuwa okno tak, aby jego lewy górny róg znalazł się w punkcie wskazywanym przez argumenty px i py, przy czym pierwszy określa współrzędną x, a drugi współrzędną y. Aby zatem ustawić okno w lewym górnym rogu ekranu, należy zastosować wywołanie:

window.moveTo(0, 0);

**Metoda open**

Wywołanie: open(URL, nazwa [, właściwości[, zamiana]]])

Dostępność: FF, IE4, NN4, OP9

Metoda open otwiera nowe okno o nazwie nazwa, z dokumentem wskazywanym przez URL, o ile taka możliwość nie została zablokowana w opcjach konfiguracyjnych przeglądarki (większość współczesnych przeglądarek udostępnia taką możliwość). Wygląd okna określa argument właściwości. Może on przyjmować parametry przedstawione w tabeli 4.2. Poszczególne parametry należy oddzielać od siebie wyłącznie znakami przecinka (przykład przedstawiony poniżej).

Tabela 4.2. Parametry dla metody open

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parametr | Znaczenie | Dopuszczalne wartości | Przykład | Dostępność |
| left | Współrzędna x lewego górnego rogu okna. | liczby całkowite | left=100 | FF, IE, NN, OP |
| top | Współrzędna y lewego górnego rogu okna. | liczby całkowite | top=10 | FF, IE, NN, OP |
| height | Wysokość obszaru okna zawierającego treść strony, włącznie z wysokością poziomego paska przewijania. | liczby całkowite, minimalna wartość: 100 | height=200 | FF, IE, NN, OP |
| width | Szerokość obszaru okna zawierającego treść strony, włącznie z wysokością pionowego paska przewijania. | liczby całkowite, minimalna wartość: 100 | width=200 | FF, IE, NN, OP |
| outerHeight | Zewnętrzna wysokość całego obszaru okna, włącznie z wszystkimi paskami narzędziowymi. | liczby całkowite, minimalna wartość: 100 | outerHeight=200 | FF, NN, OP |
| outerWidth | Zewnętrzna szerokość całego obszaru okna, włącznie z wszystkimi paskami narzędziowymi. | liczby całkowite, minimalna wartość: 100 | outerWidth=200 | FF, NN, OP |
| innerHeight | Wysokość obszaru okna zawierającego treść strony (odpowiednik height), włącznie z wysokością poziomego paska przewijania. | liczby całkowite, minimalna wartość: 100 | innerHeight=200 | FF, NN, OP |
| innerWidth | Szerokość obszaru okna zawierającego treść strony (odpowiednik width), włącznie z wysokością pionowego paska przewijania. | liczby całkowite, minimalna wartość: 100 | innerWidth=200 | FF, NN, OP |
| menubar | Określa, czy ma być widoczny pasek menu. | yes, no | menubar=yes | FF, IE, NN |
| toolbar | Określa, czy ma być widoczny pasek narzędziowy (nawigacyjny). | yes, no | toolbar=yes | FF, IE, NN |
| location | Określa, czy ma być widoczny pasek adresu. | yes, no | location=yes | FF, IE, NN, OP |
| directories | Określa, czy ma być widoczny pasek ustawień osobistych (Personal Toolbar). Jego zawartość zależy od konkretnej przeglądarki. | yes, no | directories=yes | FF, IE, NN, OP |
| personalbar | Określa, czy ma być widoczny pasek ustawień osobistych (Personal Toolbar). Odpowiednik opcji directories. Jego zawartość zależy od konkretnej przeglądarki. | yes, no | personalbar=yes | FF, NN |
| status | Określa, czy ma być widoczny pasek stanu (statusu). | yes, no | status=yes | FF, IE, NN, OP |
| resizable | Określa, czy wymiary okna mogą być zmieniane. | yes, no | resizable=yes | FF, IE, NN |
| scrollbars | Określa, czy w przypadku, kiedy dokument nie mieści się w oknie, mają być wyświetlane paski przewijania. | yes, no | scrollbars=yes | FF, IE, NN |
| dependent | Określa, czy nowe okno ma być zależne od okna otwierającego. | yes, no | dependent=yes | FF, NN |
| chrome | Pozwala otworzyć okno nieposiadające żadnego interfejsu. | yes, no | chrome=yes | FF, NN |
| modal | Określa, czy okno ma być modalne. | yes, no | modal=yes | FF, NN |
| dialog | Określa, czy okno ma być tworzone jako dialogowe. | yes, no | dialog=yes | FF, NN |
| minimizable | W przypadku okna dialogowego pozwala określić, czy ma ono posiadać przycisk minimalizacji. | yes, no | minimizable=yes | FF, NN |

Opcjonalny argument zamiana ustawiony na true określa, że otwierany dokument ma się pojawić w historii otwieranych witryn jako nowy wpis, a ustawiony na false — że ma zamienić aktualny wpis. Metoda zwraca obiekt wskazujący nowe okno, pozwalający na wykonywanie na tym oknie innych operacji (wywoływanie metod, zmiana właściwości itp.).

Argument nazwa zamiast nazwy okna może też zawierać jedną z poniższych wartości:

\_blank — odnośnik zostanie otwarty w nowym oknie lub nowej karcie (jest to zachowanie domyślne);

\_parent — odnośnik zostanie otwarty w oknie (karcie) nadrzędnym;

\_self — odnośnik zostanie otwarty w oknie (karcie) bieżącym;

\_top — odnośnik zostanie otwarty w oknie (karcie) znajdującym się na szczycie hierarchii.

Aby zatem otworzyć nowe okno o rozmiarach 800×600 pikseli, o nazwie okno2, umiejscowione w lewym górnym rogu ekranu, zawierające stronę o adresie http://marcinlis.com, posiadające pasek nawigacyjny oraz adresu, należy zastosować wywołanie:

window.open("http://marcinlis.com", "okno2", "width=800,height=600,left=0,top=0,toolbar=yes,location=yes");

**Metoda prompt**

Wywołanie: prompt(str[, defStr])

Dostępność: FF, IE4, NN4, OP9

Metoda prompt wyświetla okno dialogowe z tekstem zdefiniowanym przez argument str, przyciskami OK i Anuluj (Cancel) oraz polem tekstowym, umożliwiające wprowadzenie danych przez użytkownika, tak jak jest to widoczne na rysunku 4.4.

|  |
| --- |
| http://webmaster.helion.pl/images/kursjs/r04_03_pliki/image004.jpg |
| Rysunek 4.4. Okno wyświetlane przez metodę prompt |

W polu tekstowym wyświetlony zostanie ciąg defStr, o ile argument ten został podany w wywołaniu metody. Jeżeli nie zostanie podany, większość przeglądarek w polu tekstowym nie wyświetli nic (formalnie rzecz ujmując — wyświetli pusty ciąg znaków), jedynie Internet Explorer (nawet w najnowszych wersjach) wyświetla wtedy ciąg undefined. Dlatego też jeśli nie chcemy, aby w polu pojawił się jakikolwiek tekst, najlepiej stosować wywołanie w postaci:

prompt(tekst, "");

Wprowadzony przez użytkownika ciąg znaków jest zwracany jako wynik działania funkcji. Jeżeli nie zostaną wprowadzone żadne dane bądź w oknie zostanie kliknięty przycisk Anuluj (Cancel), wynikiem działania będzie wartość null. Przykładowy skrypt, który prosi użytkownika o podanie imienia, a następnie używa wprowadzonych danych do wyświetlenia powitania, mógłby wyglądać tak, jak przedstawiono to na listingu 4.7.

Listing 4.7. Ilustracja działania metody prompt

<script type="text/javascript">

var result = prompt("Podaj swoje imię:", "");

if(result == null || result == ""){

  document.write("Czemu nie chcesz podać swojego imienia?");

}

else{

  document.write("Cześć " + result + ". Miło Cię powitać na naszej stronie!");

}

</script>

**Metoda setInterval**

Wywołanie: setInterval(exp, mtime)

Dostępność: FF, IE4, NN4, OP9

Metoda setInterval przetwarza wyrażenie (lub wywołuje funkcję) wskazane przez exp co mtime milisekund. Zwraca identyfikator, który może być użyty do zatrzymania zliczania czasu przez wywołanie clearInterval. Przykład wykorzystania metody setInterval jest widoczny na listingu 4.8.

Listing 4.8. Proste użycie metody setInterval

<script type="text/javascript">

var i = 0;

function changeStatusBarText()

{

  window.status = i++;

}

setInterval("changeStatusBarText();", 1000);

</script>

Wywołanie setInterval("changeStatusBarText();", 1000); powoduje, że co 1000 milisekund będzie wywoływana funkcja changeStatusBarText. W tej funkcji znajduje się instrukcja zmieniająca napis na pasku stanu na wartość reprezentowaną przez zmienną i. Ponieważ i jest przy każdym wywołaniu zwiększane o 1, na pasku co sekundę pojawi się nowa wartość większa o jeden od poprzedniej. Czynność ta będzie powtarzana dopóty, dopóki nie zostanie opuszczona strona zawierająca skrypt. Gdyby wyświetlanie miało zakończyć się na przykład po osiągnięciu przez i wartości 10, należałoby zmodyfikować funkcję changeStatusBarText tak, by wywoływała metodę clearInterval. Skrypt miałby wtedy postać przedstawioną na listingu 4.9.

Listing 4.9. Zatrzymanie zegara uruchomionego przez setInterval

<script type="text/javascript">

var i = 0, id;

function changeStatusBarText()

{

  window.status = i++;

  if(i > 10) clearInterval(id);

}

id = setInterval("changeStatusBarText();", 1000);

</script>

Pojawiła się tu nowa globalna zmienna id, która przechowuje identyfikator zwrócony przez wywołanie metody setInterval. W funkcji changeStatusBarText została natomiast umieszczona instrukcja warunkowa badająca stan zmiennej i. Kiedy wartość tej zmiennej przekroczy 10, zegar jest wyłączany przez wywołanie clearInterval(id).

Napiszmy jeszcze skrypt, który po uruchomieniu zacznie wyświetlać co 200 milisekund na pasku stanu wartości rosnące, po osiągnięciu wartości 10 zmieni kierunek odliczania, po osiągnięciu zera ponownie go zmieni, a po wykonaniu 5 takich zmian zatrzyma się. Tak działający skrypt znajduje się na listingu 4.10.

Listing 4.10. Sterowanie zmianami cyklicznymi

<script type="text/javascript">

var i = 0, id, j = 1, changes = 0;

function changeStatusBarText()

{

  window.status = i;

  i += j;

  if(i > 10 || i < 0){

    j = -j;

    i += j;

    if(++changes >= 5) clearInterval(id);

  }

}

id = setInterval("changeStatusBarText();", 200);

</script>

Pojawiły się tu dwie nowe zmienne. Pierwsza to j, która zawiera wartość, o jaką będzie powiększana zmienna i przy każdym wywołaniu funkcji changeStatusBarText. Druga to changes — zliczająca liczbę zmian kierunku odliczania. Wywołanie funkcji setInternval ma praktycznie taką samą postać jak w poprzednim przykładzie, inny jest jedynie drugi argument. Tym razem jest to 200, co oznacza, że funkcja changeStatusBarText będzie wywoływana co 200 milisekund.

W funkcji changeStatusBarText najpierw właściwości status obiektu przypisywany jest stan zmiennej i. Przy pierwszym wywołaniu będzie to wartość 0, gdyż taka wartość została przypisana zmiennej przy inicjacji. Następnie do wartości i dodawana jest aktualna wartość j. Ponieważ początkowo j jest równe 1, na pasku stanu zaczną się pojawiać kolejne wartości całkowite, począwszy od 0.

Instrukcja warunkowa if bada złożony warunek i > 10 || i < 0. Wnętrze tej instrukcji jest zatem wykonywane albo wtedy, gdy i jest większe od 10, albo też, gdy i jest mniejsze od zera. W takiej sytuacji niezbędne jest odwrócenie kierunku zliczania. Czyli jeżeli i rosło, to musi zacząć maleć, bądź też, jeśli i malało, to musi zacząć rosnąc. Jest to osiągane w bardzo prosty sposób, przez odwrócenie znaku zmiennej j (instrukcja j = -j;).

Występująca wewnątrz if instrukcja i += j; powoduje, że będą wykonywane pełne cykle, czyli odliczanie od 0 do 10, następnie od 10 do 0 itd. (a także, że nie pojawią się wartości –1 i 11). Czyli zarówno 0 (począwszy od 2. cyklu), jak i 10 będą wyświetlane przez 400 milisekund (po 200 na każdy cykl). Jako ćwiczenie do samodzielnego wykonania można polecić wykonanie takiej modyfikacji skryptu, aby 0 i 10 zawsze były wyświetlane tylko przez 200 milisekund.

Wewnętrzna instrukcja warunkowa if bada warunek ++changes >= 5. Najpierw jest tu zwiększana wartość zmiennej changes, a następnie jest ona porównywana z wartością 5. Jeżeli changes osiągnie 5, oznacza to, że została wykonana założona przez nas liczba cykli i należy zakończyć wywoływanie funkcji changeStatusBarText. Dokonujemy tego przez wyzerowanie timera za pomocą wywołania funkcji clearInterval.

Wykonajmy jeszcze jeden przykład. Przy użyciu poznanego wcześniej obiektu Date oraz metody setInterval można pokusić się o napisanie skryptu, który na pasku statusu będzie wyświetlał aktualny czas. Jak widać na listingu 4.11, nie jest to skomplikowane zadanie.

Listing 4.11. Wyświetlenie zegara pokazującego aktualny czas

<script type="text/javascript">

function showTime()

{

  var data = new Date();

  var godziny = data.getHours();

  var minuty = data.getMinutes();

  var sekundy = data.getSeconds();

  var czas = godziny;

  czas += ((minuty < 10) ? ":0" : ":") + minuty;

  czas += ((sekundy < 10) ? ":0" : ":") + sekundy;

  window.status = czas;

}

showTime();

setInterval("showTime();", 1000);

</script>

Czas jest pobierany w funkcji showTime za pomocą wywołań metod getHours, getMinutes i getSesonds i jest zapisywany w zmiennej czas, której zawartość jest następnie przypisywana właściwości status obiektu window. Funkcja showTime jest natomiast wywoływana co sekundę dzięki metodzie setInterval.

**Metoda setTimeout**

Wywołanie: setTimeout(exp, mtime)

Dostępność: FF, IE, NN, OP

Przetwarza wyrażenie (lub wywołuje funkcję) exp po czasie mtime milisekund. Zwraca identyfikator, który może być użyty do zatrzymania zliczania czasu przez wywołanie clearTimeout. Warto w tym miejscu wspomnieć o różnicy, jaka występuje pomiędzy setTimeout a setInterval. Choć wyglądają podobnie, mają nieco inne działanie. Otóż setInterval powoduje działanie co zadany czas, a setTimeout po zadanym czasie. A więc w pierwszym przypadku mamy do czynienia z działaniem cyklicznym, a w drugim — z jednorazowym.

Jeśli zatem w pięć sekund po wczytaniu strony miałoby się wyświetlić okno dialogowe, to należy zastosować przykładowe wywołanie:

setTimeout("alert('tekst');", 5000);

Metoda setTimeout może być przy tym użyta tak, by wywoływania pewnej funkcji były cykliczne. Po prostu oprócz startowego wywołania należy w funkcji umieścić kolejne wywołanie. Schemat takiej konstrukcji jest następujący:

function funkcja([argumenty])

{

  //instrukcje

  setTimeout("funkcja([argumenty]);", czas);

}

setTimeout("funkcja([argumenty]);", czas);

Korzystający z tej metody skrypt wyświetlający na pasku stanu kolejne wartości całkowite miałby zatem postać taką, jaka została zaprezentowana na listingu 4.12.

Listing 4.12. Symulacja wywołań cyklicznych za pomocą metody setTimeout

<script type="text/javascript">

var i = 0;

function changeStatusBarText()

{

  window.status = i++;

  setTimeout("changeStatusBarText();", 1000);

}

setTimeout("changeStatusBarText();", 1000);

</script>

**Pozostałe popularne metody**

W tabeli 4.3 zostały zebrane pozostałe popularne metody obiektu window.

Tabela 4.3. Wybrane metody obiektu window

|  |  |  |
| --- | --- | --- |
| Metoda | Opis | Dostępność |
| back() | Odpowiada wciśnięciu przycisku Back (Wstecz) w przeglądarce. | FF, NN, OP |
| blur() | Usuwa fokus z bieżącego okna. | FF, IE, NN, OP |
| close() | Zamyka okno przeglądarki. | FF, IE, NN, OP |
| find([str[, caseSensitive, backward]]) | Umożliwia przeszukanie treści bieżącego okna pod kątem występowania ciągu str. Jeżeli nie zostanie podany żaden argument, przeglądarka wyświetli okno dialogowe umożliwiające wprowadzenie danych. Parametr caseSensitive ustawiony na true oznacza uwzględnianie wielkości liter, natomiast backward ustawiony na true oznacza przeszukiwanie od końca dokumentu. | FF, NN |
| focus() | Ustawia fokus na bieżącym oknie. | FF, IE, NN, OP |
| forward() | Odpowiada wciśnięciu przycisku Forward (W przód) w przeglądarce. | FF, NN, OP |
| home() | Odpowiada wciśnięciu przycisku Home (Start, Główna itp.) w przeglądarce. | FF, NN, OP |
| print() | Drukuje zawartość okna. | FF, IE, NN, OP |
| resizeBy(px, py) | Przemieszcza prawy dolny róg okna do punktu o współrzędnych px, py. | FF, IE, NN, OP |
| resizeTo(szerokość, wysokość) | Zmienia rozmiary okna. | FF, IE, NN, OP |
| scroll(x, y) | Przesuwa zawartość okna do współrzędnych x, y. Metoda przestarzała — o ile to możliwe, należy korzystać z metody scrollTo. | FF, IE, NN, OP |
| scrollBy(px, py) | Przesuwa zawartość okna o px pikseli w poziomie i py pikseli w pionie. Metoda użyteczna tylko w przypadku, kiedy zawartość dokumentu nie mieści się w oknie. | FF, IE, NN, OP |
| scrollTo(x, y) | Przesuwa zawartość okna do współrzędnych x, y. | FF, IE, NN, OP |
| stop() | Odpowiada wciśnięciu przycisku Stop w przeglądarce. | FF, NN, OP |