Javascript nie posiada wprost funckji sleep (tak jak to jest w PHP , czy C). Na szczęście nie wszystk stracone. Nie oznacza to wcale, że nie można na chwilę wstrzymac kodu, lub wykonać go po pewnym czasie. Są na to dwie metody.

Sposoby na akcję po czasie

- window.setTimeout
- window.setInterval

window.setTimeout

Metoda ta przyjmuje dwa parametry - pierwszy jest to ciąg znaków lub referencja do funkcji, która ma się wykonać 1 raz. Drugi parametr to czas za jaki ma zostać wywołany podany wcześniej kod.

Przekazanie ciągu znaków

{codecitation class='brush: js'}window.setTimeout("alert('działa')", 1000);{/codecitation}

Podany kod jest evalowany . Ja niezbyt lubię to rozwiązanie. Lepiej używać referencji do funkcji lub funkcji anonimownych .

Przekazanie funkcji anonimowej

{codecitation class='brush: js'}window.setTimeout(function() { alert('działa'); },

1000);{/codecitation}

Wynik widoczny dla użytkownika jest taki sam. Ja polecam ten sposób.

window.setInterval

Wywołanie metody `setInterval' daje podobny efekt do setTimeout. Różnica polega na tym, że tu podany jako pierwszy parametr kod będzie się wykonywał wielokrotnie (w teorii do nieskończoności), co określony interwał.

{codecitation class='brush: js'}window.setInterval(function() { alert('działa'); }, 5000);{/codecitation}

Przerywanie ciągu wykonań window.clearInterval

Aby zakończyć powtarzające się wykonywanie zadanej akcji należy użyć metody window.clearInterval(int).

Jako parametr przekazujemy identyfikator zwracany przez setInterval.

{codecitation class='brush: js'} var count = 0; var intId = window.setInterval(function() { alert('działa'); count++; if (count >= 5) { window.clearInterval(intId); } }, 5000); {/codecitation}

Taki kod pozwoli wykonać się zadanej akcji 5 razy.