

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

W JavaScriptcie istnieje kilka funkcji globalnych, które można wywoływać bezpośrednio (są dostępne w każdym miejscu skryptu). Najpopularniejsze to `eval`, `isNaN`, `parseFloat` i `parseInt`. Są one opisane poniżej. Użyte zostały następujące oznaczenia przeglądarek:

IE — Internet Explorer,

NN — Netscape Navigator,

OP — Opera,

FF — Firefox.

Jeżeli przy określeniu przeglądarki występuje numer, oznacza to minimalną wersję, niezbędną, aby dana funkcja (właściwość, obiekt) była dostępna. **Funkcja `eval`**

Składnia: `eval(str)`

Dostępność: od JavaScript 1.0 i JScript 1.0

Przeglądarki: IE3, FF, NN, OP

Jeżeli parametr `str` jest ciągiem znaków tworzącym wyrażenie, metoda `eval` zwraca obliczoną wartość tego wyrażenia, jeżeli natomiast `str` jest instrukcją lub ciągiem instrukcji, są one wykonywane. Przykładowo: jeśli jako argumentu użyjemy ciągu:

```
(6 + 7 * 2) / 5
```

to wynikiem będzie 4, a jeśli argumentem będzie ciąg `func()`, a w skrypcie będzie dostępna funkcja o nazwie `func`, to zostanie ona wykonana. W tym drugim przypadku metoda `eval` nie zwraca żadnej wartości (dokładniej — zwraca wartość `undefined`), o ile ostatnia przetwarzana przez nią instrukcja nie zwróciła żadnej wartości. Jeżeli ostatnia instrukcja zwróciła jakąś wartość, ta wartość staje się wynikiem działania `eval`. Użycie funkcji `eval` zostało zilustrowane w przykładzie widocznym na listingu 3.1.

Listing 3.1. Użycie funkcji `eval`

```
<script type="text/javascript">

function func1()
{
    document.write("funkcja func1");
}

function func2()
{
    document.write("funkcja func2");

    return 8;
}
```

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

```
}  
  
var result = eval("&quot;(6 + 7 * 2) / 5&quot;);  
  
document.write(result);  
  
document.write("&quot;<br />&quot;);  
  
result = eval("&quot;func1();&quot;);  
  
document.write("&quot;<br />&quot;);  
  
document.write(result);  
  
document.write("&quot;<br />&quot;);  
  
result = eval("&quot;func2();&quot;);  
  
document.write("&quot;<br />&quot;);  
  
document.write(result);  
  
</script>
```

Pierwsza wykonywana instrukcja tworzy zmienną `result` i przypisuje jej wynik działania funkcji `eval`, której argumentem jest ciąg znaków zawierający wyrażenie arytmetyczne. Oznacza to, że wartością zmiennej stanie się obliczona wartość wyrażenia (będzie to 4). Wartość ta jest wyświetlana na ekranie za pomocą instrukcji `document.write`.

Drugie wywołanie funkcji `eval` zawiera ciąg znaków `func1()`. To oznacza, że zostanie wywołana funkcja `func1`. W jej wnętrzu znajduje się instrukcja wyświetlająca na ekranie napis z nazwą funkcji (pojawi się on więc na ekranie), funkcja nie zwraca jednak żadnej wartości. Wynikiem działania `eval` będzie więc `undefined` i ta wartość zostanie przypisana zmiennej `result`.

Trzecie wywołanie `eval` zawiera ciąg znaków `func2()`, co oznacza, że zostanie wywołana funkcja `func2`. W kodzie tej funkcji znajduje się instrukcja wyświetlająca jej nazwę na ekranie oraz instrukcja `return` zwracająca wartość 8. To oznacza, że wynikiem działania `func2` będzie 8, a tym samym wynikiem działania `eval` będzie 8. Taka też wartość zostanie przypisana zmiennej `result`. **Funkcja `isNaN`**

Składnia: `isNaN(wartość)`

Dostępność: od JavaScript 1.1 i JScript 1.0

Przeglądarki: IE3, FF, NN2, OP

Ta funkcja zwraca wartość `false`, jeżeli parametr `wartość` przedstawia wartość liczbową (może być przekonwertowany na wartość liczbową), lub `true` w przeciwnym razie. Parametr `wartość` może zawierać złożone wyrażenie. A zatem wykonanie instrukcji:

```
isNaN("&quot;abc&quot;);
```

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

spowoduje zwrócenie wartości false, a:

```
isNaN("123");
```

zwrócenie wartości true. Ilustruje to przykład z listingu 3.2.

Listing 3.2. Ilustracja działania funkcji isNaN

```
<script type="text/javascript">
var zmienna1 = "abc";
var zmienna2 = "123";
var nieJestLiczba = isNaN(zmienna1);
if(nieJestLiczba){
    document.write("Wartość zmiennej zmienna1 nie może być interpretowana jako liczba.<br />");
}
else{
    document.write("Wartość zmiennej zmienna1 może być interpretowana jako liczba.<br />");
}
nieJestLiczba = isNaN(zmienna2);
if(nieJestLiczba){
    document.write("Wartość zmiennej zmienna2 nie może być interpretowana jako liczba.<br />");
}
else{
    document.write("Wartość zmiennej zmienna2 może być interpretowana jako liczba.<br />");
}
</script> Funkcja parseInt
```

Składnia: parseInt(str[, podstawa])

Dostępność: od JavaScript 1.0 i JScript 1.0

Przeglądarki: IE3, FF, NN2, OP

Funkcja parseInt przetwarza ciąg znaków podany jako argument str na wartość całkowitą. Opcjonalny argument podstawa pozwala na ustalenie podstawy systemu liczbowego; może on przyjmować wartości od 2 do 36. W przypadku, kiedy argument podstawa zostanie pominięty lub kiedy będzie miał wartość 0, przyjęte zostaną następujące zasady:

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

1.

Jeżeli ciąg str rozpoczyna się od znaków 0x, będzie traktowany jako wartość szesnastkowa.

2.

Jeżeli ciąg str rozpoczyna się od znaku 0, będzie traktowany jako wartość ósemkowa (nie jest to określone w standardzie, ale w ten sposób zachowują się wszystkie popularne przeglądarki).

3.

Jeżeli nie zachodzą przypadki 1. i 2., ciąg str będzie traktowany jako wartość dziesiętna.

Jeżeli ciąg str nie będzie reprezentował prawidłowej wartości całkowitej, funkcja zwróci wartość NaN.

Sposób użycia funkcji `parseInt` został zobrazowany w skrypcie z listingu 3.3.

Listing 3.3. Ilustracja działania funkcji `parseInt`

```
<script type="text/javascript">
```

```
var ciąg1 = "12";
```

```
var ciąg2 = "34";
```

```
var wynik = ciąg1 + ciąg2;
```

```
document.write("Zwykle dodawanie ciągów: ciąg1 + ciąg2 = ");
```

```
document.write(wynik);
```

```
document.write("<br />");
```

```
ciąg1 = parseInt(ciąg1);
```

```
ciąg2 = parseInt(ciąg2);
```

```
wynik = ciąg1 + ciąg2;
```

```
document.write("Dodawanie po konwersji na liczby: ciąg1 + ciąg2 = ");
```

```
document.write(wynik);
```

```
document.write("<br />");
```

```
</script>
```

Należy pamiętać, że `parseInt` przetwarza ciąg znaków, zamieniając go na liczbę, tylko do napotkania pierwszego

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

znaku spoza danego systemu liczbowego. Dalsza część ciągu jest odrzucana. Tabela 3.1 zawiera wyniki kilku przykładowych wywołań `parseInt`.

Tabela 3.1. Przykładowe wywołania funkcji `parseInt`

Wywołanie

Wynik

`parseInt(20, 16)`

32

`parseInt(20, 10)`

20

`parseInt(10, 2)`

2

`parseInt("2F", 16)`

47

`parseInt("2F", 10)`

2

`parseInt(9, 2)`

NaN

### **Funkcja `parseFloat`**

Składnia: `parseFloat(str)`

Dostępność: od JavaScript 1.1 i JScript 1.0

Przeglądarki: IE2, FF, NN2, OP

Funkcja `parseFloat` przetwarza ciąg znaków podany jako argument `str` na wartość rzeczywistą. Jeżeli ciąg `str` nie będzie reprezentował prawidłowej wartości rzeczywistej, funkcja zwróci wartość NaN. Wartość rzeczywista może być przedstawiona w zapisie z separatorem dziesiętnym, np.:

```
var wartość = parseFloat("3.14");
```

jak również w notacji wykładniczej:

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

```
var wartość = parseFloat("&quot;1.234e2&quot;");
```

W tym drugim przypadku zmiennej wartość została przypisana wartość rzeczywista 123,4, zapis 1.234e2 oznacza bowiem  $1.234 * 10^2$ . Ogólnie zapis X.YeZ należy rozumieć jako  $X.Y * 10^Z$ . **Pozostałe funkcje**

### globalne

#### Funkcja decodeURI

Składnia: decodeURI(URI)

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE5.5, FF, NN6, OP

Funkcja decodeURI dekoduje ciąg znaków URI zakodowany przez funkcję encodeURI. **Funkcja decodeURIComponent**

Składnia: decodeURIComponent(URI)

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE5.5, FF, NN6, OP

Funkcja decodeURIComponent dekoduje ciąg znaków URI zakodowany przez funkcję encodeURIComponent. **Funkcja encodeURI**

Składnia: encodeURI(str)

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE5.5, FF, NN6, OP

Funkcja encodeURI przekształca ciąg str w taki sposób, aby był poprawnym identyfikatorem zasobów (URI, Uniform Resource Identifier). Kodowanie odbywa się przez zamianę wybranych znaków na odpowiadające im sekwencje zgodne z kodowaniem UTF-8. Wszystkie znaki niestandardowe (czyli te, które nie mogą się znaleźć w adresie) zamienia na odpowiadające im kody Unicode, a następnie zapisuje je w formacie szesnastkowym. Funkcja nie zmienia znaków: -, \_, ., !, ~, \*, ', (, ), ;, /, ?, :, @, &, =, +, \$, ,, #. Jeżeli mamy przykładowy adres:

```
http://mojadomena.com/skrypt.php?parametr=żółty liść
```

to nie może być przekazany w standardowej postaci, gdyż zawiera znaki spoza podstawowego zestawu ASCII oraz zawiera spację. Wszystkie te znaki muszą być zakodowane. Jeżeli taki ciąg znaków poddamy działaniu funkcji encodeURI, to zostanie on przetworzony na:

```
http://mojadomena.com/skrypt.php?parametr=%C5%BC%C3%B3%C5%82ty%20li%C5%9B%C4%87 Funkcja encodeURIComponent
```

Składnia: encodeURIComponent(str)

Dostępność: od JavaScript 1.5 i JScript 5.5

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

Przeglądarki: IE5.5, FF, NN6, OP

Funkcja `encodeURIComponent` przekształca ciąg str w taki sposób, aby był poprawnym identyfikatorem zasobów (URI, Uniform Resource Identifier). Kodowanie odbywa się przez zamianę wybranych znaków na odpowiadające im sekwencje zgodne z kodowaniem UTF-8. Kodowane są również znaki `;`, `/`, `?`, `:`, `@`, `&`, `=`, `+`, `$`, `,`, `#` — przeciwnie niż w przypadku użycia metody `encodeURI`. **Funkcja `escape`**

Składnia: `escape(str)`

Dostępność: od JavaScript 1.0 i JScript 1.0

Przeglądarki: IE3, FF, NN2, OP

Funkcja `escape` zwraca przetworzony ciąg str, w którym wszystkie znaki inne niż znaki alfabetu łacińskiego oraz znaki `@`, `*`, `-`, `_`, `+`, `.`, `/` zostały zamienione na sekwencje znaków specjalnych (inaczej sekwencje ucieczki — stąd też nazwa funkcji, z ang. `escape` — uciekać). Dla znaków o kodach poniżej 256 sekwencja specjalna ma postać `%xx`, gdzie `xx` to heksadecymalny kod znaku, natomiast dla znaków o kodach powyżej 255 sekwencja ma postać `%unnnn`, gdzie `nnnn` to kod znaku w formacie Unicode. Przykładowo tekst „żółty liść” po przetworzeniu przez funkcję `escape` będzie miał postać:

```
%u017C%F3%u0142ty%20li%u015B%u0107
```

Funkcji `escape` nie należy używać do kodowania adresów URL (ani w całości, ani we fragmentach). Do tego celu służą funkcje `encodeURI` i `encodeURIComponent`. **Funkcja `isFinite`**

Składnia: `isFinite(wartość)`

Dostępność: od JavaScript 1.3 i JScript 3.0

Przeglądarki: IE4, FF, NN4, OP

Funkcja `isFinite` zwraca wartość `true`, jeżeli parametr `wartość` przedstawia wartość skończoną, lub `false` w przeciwnym razie. **Funkcja `unescape`**

Składnia: `unescape(str)`

Dostępność: od JavaScript 1.0 i JScript 1.0

Przeglądarki: IE3, FF, NN2, OP

Funkcja `unescape` to odwrotność funkcji `escape`. Zwraca ciąg, w którym wszystkie sekwencje specjalne w postaci `%uxxxx` lub `%xx` zostały zamienione na odpowiadające im znaki. **Właściwości globalne**

Oprócz globalnych funkcji istnieją również globalne właściwości. W rzeczywistości są to definicje typów `Infinity`, `NaN` i `undefined`. Zostały one zebrane w tabeli 3.2.

Tabela 3.2. Właściwości globalne

Właściwość

Znaczenie

## Funkcje globalne

Dodał Administrator  
niedziela, 14 marzec 2010 16:52

---

Dostępność

Infinity

Reprezentuje nieskończoność. Jest tożsama z wartością właściwości `POSITIVE_INFINITY` obiektu `Number`.

Od JavaScript 1.3 i JScript 3.0

NaN

Reprezentuje nieliczbę. Jest tożsama z wartością właściwości `NaN` obiektu `Number`.

Od JavaScript 1.3 i JScript 3.0

undefined

Reprezentuje wartość niezdefiniowaną.

Od JavaScript 1.3 i JScript 5.5