

## Operatory arytmetyczne

Dodał Administrator  
sobota, 13 marzec 2010 23:03

---

Operatory arytmetyczne służą do wykonywania operacji arytmetycznych, czyli mnożenia, dzielenia, dodawania, odejmowania itp. Występują w tej grupie również takie operatory, jak operator inkrementacji i dekrementacji oraz operatory ustalenia znaku wartości. Operatory  $*$ ,  $/$ ,  $+$ ,  $-$ ,  $\%$  są dwuargumentowe, natomiast  $+$ ,  $-$ ,  $++$  i  $--$  są jednoargumentowe. Zostały zebrane w tabeli 2.2.

Tabela 2.2. Operatory arytmetyczne

Operator

Wykonywane działanie

Przykład

\*

mnożenie

$x * y$

/

dzielenie

$x / y$

+

dodawanie

$x + y$

-

odejmowanie

$x - y$

%

dzielenie modulo (reszta z dzielenia)

$x \% y$

++

inkrementacja (zwiększanie)

## Operatory arytmetyczne

Dodał Administrator  
sobota, 13 marzec 2010 23:03

---

X++, ++X

--

dekrementacja (zmniejszanie)

X--, --X

+

ustalenie znaku wartości

+X

-

ustalenie znaku wartości

-X

### Operatory klasyczne

Działanie przedstawionych w tabeli operatorów sprowadza się do wykonywania typowych działań arytmetycznych, znanych każdemu, kto uczęszczał w szkole na lekcje matematyki (wyjątkami są ++ i --, które zostaną omówione niżej). Jeśli zatem trzeba dodać do siebie dwie zmienne, liczbę do zmiennej lub dwie liczby, należy zastosować operator +, do mnożenia operator \* itd., ilustruje to kod z listingu 2.1.

Listing 2.1. Proste operacje arytmetyczne

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

```
var liczba1 = 2 * 6 + 10;
```

```
var liczba2 = liczba1 / 2;
```

```
var liczba3 = liczba2 - liczba1;
```

```
document.write(liczba1);
```

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

```
document.write(liczba2);
```

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

```
document.write(liczba3);
```

```
</script>
```

Zmiennej liczba1 została przypisana wartość wynikająca z działania  $2 * 6 + 10$ , czyli 22. Zwróćmy uwagę, że

## Operatory arytmetyczne

Dodał Administrator  
sobota, 13 marzec 2010 23:03

---

najpierw zostało wykonane działanie arytmetyczne znajdujące się z prawej strony operatora = i dopiero wynik tego działania został przypisany zmiennej. Tak samo będzie w przypadku innych instrukcji przypisania. Przetwarzanie wyrażeń będzie się odbywało od strony prawej do lewej.

Zmiennej liczba2 została przypisana wartość wynikająca z działania liczba1 / 2, czyli 11, natomiast zmiennej liczba3 wartość wynikająca z działania liczba2 - liczba1, czyli -11. Wartości wszystkich zmiennych zostały ostatecznie wyświetlone na ekranie za pomocą instrukcji document.write. Zwróćmy przy tym uwagę, że jeśli ma być wyświetlona wartość zmiennej, to nie należy ujmować jej w znaki cudzysłowu. Przykład ten pokazuje, że operacje arytmetyczne mogą być wykonywane na samych liczbach, na zmiennych i liczbach oraz na samych tylko zmiennych.

Do operatorów arytmetycznych należy również %, przy czym, jak zostało to zaznaczone w powyższej tabeli, nie oznacza on obliczania procentów, ale dzielenie modulo, czyli resztę z dzielenia. Przykładowo działanie 10 % 3 da w wyniku 1, trójka zmieści się bowiem w dziesięciu 3 razy, pozostawiając resztę 1 ( $3 * 3 = 9$ ,  $9 + 1 = 10$ ). Podobnie:  $21 \% 8 = 5$ , gdyż  $2 * 8 = 16$ ,  $16 + 5 = 21$ . **Więcej o przetwarzaniu wyrażeń**

Tak jak zostało to wspomniane, wyrażenia są przetwarzane od strony prawej do lewej (jest to ogólna zasada dotycząca przetwarzania wyrażeń). Załóżmy, że mamy do czynienia z następującą instrukcją:

```
var suma = 2 + 3 + 4;
```

To złożone wyrażenie składające się z dwóch operacji dodawania i jednej operacji przypisania. Jak będzie ono przetwarzane? Otóż najpierw zostanie wykonane działanie  $4 + 3$ , a w jego miejsce zostanie podstawiona wynikająca z tej operacji wartość, czyli 7. Powstanie więc wyrażenie:

```
var suma = 1 + 7;
```

Następnie zostanie wykonane działanie  $7 + 1$ , a w jego miejsce zostanie podstawiony wynik tej operacji, czyli 8. Powstanie wtedy wyrażenie:

```
var suma = 8;
```

Zostanie ono wykonane już bezpośrednio, czyli ostatecznie zmiennej suma zostanie przypisana wartość 8. Warto też wiedzieć, że wynikiem operacji przypisania (operacja przypisania ma swój wynik!) jest wartość, która została przypisana. A zatem w miejscu wystąpienia instrukcji:

```
liczba = 5;
```

znajdzie się wartość 5.

Stosując opisane zasady, można napisać przykładowe (w pełni poprawne i przydatne) wyrażenie:

```
var wartosc = suma = 2 + 3;
```

Po powyższych wyjaśnieniach z pewnością stało się jasne, że najpierw zostanie wykonane dodawanie, jego wynik zostanie przypisany zmiennej suma (jeżeli nie została zadeklarowana wcześniej, zostanie w tym miejscu utworzona), a wynik tego przypisania, czyli wartość zmiennej suma, zostanie przypisany zmiennej wartosc.

### Operatory inkrementacji i dekrementacji

Osobno należy omówić operatory inkrementacji, czyli zwiększania (++), oraz dekrementacji, czyli zmniejszania (--), które z pewnością są znane osobom znającym języki takie jak C, C++ czy Java. Dla osób, które dotychczas nie programowały, niezbędne są jednak dodatkowe wyjaśnienia. Operator ++ zwiększa po prostu wartość zmiennej o jeden, a -- zmniejsza wartość zmiennej o jeden. Oba mogą występować w formie przedrostkowej lub

## Operatory arytmetyczne

Dodał Administrator  
sobota, 13 marzec 2010 23:03

---

przyrostkowej. Przykładowo: jeśli istnieje zmienna liczba, forma przedrostkowa będzie miała postać: ++liczba, natomiast forma przyrostkowa postać liczba++.

Obie postacie powodują zwiększenie wartości zapisanej w o jeden, ale w przypadku formy przedrostkowej (++liczba) odbywa się to przed wykorzystaniem zmiennej, a w przypadku formy przyrostkowej (liczba++) — dopiero po jej wykorzystaniu. Mimo że dla osób początkujących może się to wydawać niezrozumiałe, wszelkie wątpliwości rozwieje przykład widoczny na listingu 2.2 (dla ułatwienia opisu poszczególne wiersze kodu zostały ponumerowane). Jakie będzie jego działanie?

Listing 2.2. Ilustracja działania operatora inkrementacji

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

```
/*1*/ var liczba1 = 2;
```

```
/*2*/ document.write(liczba1++);
```

```
/*3*/ document.write(++liczba1);
```

```
/*4*/ document.write(liczba1);
```

```
/*5*/ var liczba2 = liczba1++;
```

```
/*6*/ document.write(liczba2);
```

```
/*7*/ liczba2 = ++liczba1;
```

```
/*8*/ document.write(liczba2);
```

```
/*9*/ document.write(++liczba2);
```

```
</script>
```

Po uruchomieniu kodu w oknie przeglądarki pojawi się ciąg liczb 244467. Dlaczego? Otóż w linii 1. zadeklarowano zmienną liczba1 oraz przypisano jej wartość 2. W linii 2. zastosowano formę przyrostkową operatora ++, zatem najpierw została wyświetlona wartość zmiennej liczba1 (liczba1 = 2) na ekranie, a dopiero potem jej wartość zwiększyła się o jeden (liczba1 = 3). W linii 3. — dokładnie odwrotnie, to znaczy, przez zastosowanie formy przedrostkowej najpierw zwiększa się wartość zmiennej liczba1 o jeden (liczba1 = 4), a dopiero potem ta wartość zostaje wyświetlona na ekranie. W linii 4. jedyną operacją jest ponowne wyświetlenie wartości liczba1 (liczba1 = 4).

W linii 5. najpierw zostaje przypisana aktualna wartość liczba1 (liczba1 = 4) zmiennej liczba2 (liczba2 = 4), a dopiero potem zostaje zwiększona liczba1 o jeden (liczba1 = 5). W linii 6. zostaje wyświetlona wartość liczba2. W linii 7. najpierw jest zwiększana liczba1 o jeden (liczba1 = 6), a następnie tak powstała wartość jest przypisywana zmiennej liczba2 (liczba2 = 6). W linii 8. zostaje wyświetlona wartość liczba2 na ekranie. W linii ostatniej, 9., najpierw zostaje zwiększona liczba2 o jeden (liczba2 = 7), a dopiero potem ta wartość jest wyświetlona na ekranie. W ten sposób powstaje wymieniony wyżej ciąg liczb.

Operator dekrementacji (--) działa analogicznie do ++, z tą różnicą, że zmniejsza wartość zmiennej o jeden. Zatem po zmodyfikowaniu programu z poprzedniego listingu w taki sposób, aby wszystkie wystąpienia ++ zostały zamienione na --, powstanie skrypt zaprezentowany na listingu 2.3.

Listing 2.3. Ilustracja działania operatora dekrementacji

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

## Operatory arytmetyczne

Dodał Administrator  
sobota, 13 marzec 2010 23:03

---

```
/*1*/ var liczba1 = 2;  
  
/*2*/ document.write(liczba1--);  
  
/*3*/ document.write(--liczba1);  
  
/*4*/ document.write(liczba1);  
  
/*5*/ var liczba2 = liczba1--;  
  
/*6*/ document.write(liczba2);  
  
/*7*/ liczba2 = --liczba1;  
  
/*8*/ document.write(liczba2);  
  
/*9*/ document.write(--liczba2);  
  
</script>
```

Tym razem wynikiem będzie ciąg liczb: 2000-2-3. Oto jego działanie. W linii 1. występuje deklaracja zmiennej `liczba1` i przypisanie jej wartości 2, dokładnie tak, jak w poprzednim przykładzie. W linii 2. została zastosowana przyrostkowa forma operatora `--`, zatem najpierw zostanie wyświetlona wartość zmiennej `liczba1` (`liczba1 = 2`) na ekranie, a dopiero potem będzie zmniejszona jej wartość o 1 (`liczba1 = 1`). W linii 3. będzie dokładnie odwrotnie, to znaczy przez zastosowanie formy przedrostkowej najpierw zostanie zmniejszona wartość zmiennej `liczba1` o jeden (`liczba1 = 0`), a dopiero potem wartość ta zostanie wyświetlona na ekranie. W linii 4. jedyną operacją jest ponowne wyświetlenie wartości `liczba1` (`liczba1 = 0`).

W linii 5. najpierw ma miejsce przypisanie aktualnej wartości `liczba1` (`liczba1 = 0`) zmiennej `liczba2` (`liczba2 = 0`) i dopiero potem następuje zmniejszenie `liczba1` o jeden (`liczba1 = -1`). W linii 6. zostaje wyświetlona wartość `liczba2` (`liczba2 = 0`). W linii 7. najpierw ma miejsce zmniejszenie `liczba1` o jeden (`liczba1 = -2`), a następnie przypisanie tej wartości `liczba2` (`liczba2 = -2`). W linii 8. zostaje wyświetlona wartość `liczba2` (`liczba2 = -2`). W linii ostatniej, 9., najpierw zostaje zmniejszona `liczba2` o jeden (`liczba2 = -3`), a dopiero potem ta wartość zostaje wyświetlona na ekranie.