JavaScript 04 – Operatory

Operatory arytmetyczne

Służą do wykonywania operacji arytmetycznych. W tej grupie znajdują się też operatory inkrementacji (zwiększania) i dekrementacji (zmniejszania). Operatory arytmetyczne są dwu argumentowe, natomiast operatory inkrementacji i dekrementacji są jedno argumentowe (tabela 3.1).

Tabela 3.1. Operatory arytmetyczne

Operator	Działanie	Przykład
+	dodawanie	a + b
-	odejmowanie	a - b
*	mnożenie	a * b
/	dzielenie	a / b
%	modulo (reszta z dzielenia)	a % b
++	inkrementacja	x++, ++x
--	dekrementacja	x--, --x

Operator inkrementacji powoduje zwiększenie wartości o jeden. Może występować w postaci przedrostkowej (++x) lub przyrostkowej (x++). Operacja x++ zwiększa wartość zmiennej po jej wykorzystaniu, natomiast ++x przed jej wykorzystaniem. Jak widać, oba operatory zwiększają wartość zmiennej, ale nie są równoważne.

Operator dekrementacji działa analogicznie, tylko zamiast zwiększać wartości zmiennych, zmniejsza je.

Przykład 3.9

```
<script type="text/javascript">
<!--
for (var i = 0; i < 10; i++) {
document.write("<br> Ile razy zostanie wykonana pętla? " + i);
}
// -->
</script>
```

Operatory porównania

Operatory porównania porównują argumenty. Ich wynikiem jest wartość logiczna true (prawda) lub false (fałsz) (tabela 3.2).

Tabela 3.2. Operatory porównania

Operator	Działanie	Przykład
==	Wynik `true`, gdy argumenty są równe.	`a == b`
!=	Wynik `true`, gdy argumenty są różne.	`a != b`
===	Wynik `true`, gdy argumenty są tego samego typu i są równe.	`a === b`
!==	Wynik `true`, gdy argumenty są różne lub są różnych typów.	`a != b`
>	Wynik `true`, gdy argument pierwszy jest większy od drugiego.	`a > b`
<	Wynik `true`, gdy argument pierwszy jest mniejszy od drugiego.	`a < b`
>=	Wynik `true`, gdy argument pierwszy jest większy od drugiego lub jest mu równy.	`a >= b`
<=	Wynik `true`, gdy argument pierwszy jest mniejszy od drugiego lub jest mu równy.	`a <= b`

Operatory bitowe

Operatory bitowe umożliwiają wykonanie operacji na poszczególnych bitach liczb (tabela 3.3).

Tabela 3.3. Operatory bitowe

Operator	Działanie	Przykład
&	iloczyn bitowy (AND)	`a & b`
|	suma bitowa (OR)	`a | b`
~	negacja bitowa (NOT)	`~a`
^	bitowa różnica symetryczna	`a ^ b`
>>	przesunięcie bitowe w prawo	`a >> n`
<<	przesunięcie bitowe w lewo	`a << n`
>>>	przesunięcie bitowe w prawo z wypełnieniem zerami	`a >>> n`

Operatory logiczne

Operatory logiczne wykonują operacje na argumentach, które posiadają wartość logiczną (true lub false) (tabela 3.4).

Tabela 3.4. Operatory logiczne

Operator	Działanie	Przykład
&&	iloczyn logiczny (AND)	a && b
||	suma logiczna (OR)	a || b
!	negacja logiczna (NOT)	!a

Wynik iloczynu logicznego przyjmuje wartość true tylko wtedy, gdy obydwa argumenty mają wartość true. W pozostałych przypadkach wynik przyjmuje wartość false.

Wynik sumy logicznej przyjmuje wartość false tylko wtedy, gdy obydwa argumenty mają wartość false. W pozostałych przypadkach wynik przyjmuje wartość true.

Negacja logiczna zmienia wartość argumentu na przeciwną.

Operatory przypisania

Operatory przypisania przypisują wartości argumentom znajdującym się po lewej stronie operatora. Oprócz prostej operacji przypisania pozwalają na połączenie operacji przypisania z inną operacją, np. dodawania.

Zapis i += 7 oznacza w praktyce to samo, co zapis i = i + 7. Stosowanie takich skróconych zapisów upraszcza tworzenie bardziej rozbudowanych skryptów. W języku JavaScript istnieje duża grupa operatorów tego typu (tabela 3.5).

Tabela 3.5. Niektóre operatory przypisania

Operator	Przykład	Znaczenie
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y