

JavaScript 07 – Pętle

Pętle

Pętle są używane do wykonywania powtarzających się czynności. W języku JavaScript występują następujące rodzaje pętli: for, while, do ... while.

Pętla for

Pętla typu for służy do budowania pętli, gdy został podany licznik wykonań pętli oraz warunek, który musi być spełniony, aby kolejny raz wykonać pętlę. Składnia instrukcji jest następująca:

```
for (wyrażenie początkowe; wyrażenie warunkowe; wyrażenie modyfikujące) {  
    blok instrukcji;  
}
```

- wyrażenie początkowe — inicjuje zmienną, która jest używana jako licznik pętli,
- wyrażenie warunkowe — określa warunek, który musi być spełniony, aby pętla została wykonana kolejny raz,
- wyrażenie modyfikujące — modyfikuje zmienną, która jest licznikiem.

Pętlę for wykorzystuje się zwykle wtedy, gdy znamy liczbę wykonywanych powtórzeń.

Przykład 3.12

```
<script type="text/javascript">  
    <!--  
    for (var i = 0; i < 5; i++) {  
        document.write("Pętla wykonana " + i + " raz/y<br>");  
    }  
    // -->  
</script>
```

W uproszczonej postaci pętla może zostać pozbawiona wyrażenia modyfikującego.

Przykład 3.13

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

```
    for (var i = 0; i < 5;) {  
        document.write("Pętla wykonana " + i + " raz/y<br>");  
        i++;  
    }  
    // -->  
</script>
```

W podanym przykładzie zwiększanie licznika zostało przeniesione z pętli do bloku instrukcji. Należy pamiętać o tym, aby średnik występujący po wyrażeniu `i < 5` pozostał w pętli, ponieważ jest on niezbędny do jej prawidłowego działania.

W podobny sposób można postąpić z wyrażeniem początkowym, przenosząc je do bloku przed pętlą.

Przykład 3.14

```
<script type="text/javascript">
<!--
var i = 0;
for (;i < 5;) {
document.write("Pętla wykonana " + i + " raz/y<br>");
i++;
}
// -->
</script>
```

W tym przypadku również średnik występujący przed wyrażeniem `i < 5` powinien pozostać w pętli. Podobnie jak poprzednio, jest on niezbędny do prawidłowego jej działania.

Pętla while

Pętla `while` jest zwykle wykorzystywana wtedy, gdy liczba wykonywanych powtórzeń nie jest znana. Składnia instrukcji jest następująca:

```
while (wyrażenie warunkowe) {
    blok instrukcji;
}
```

Blok instrukcji jest wykonywany w pętli, dopóki wyrażenie warunkowe jest prawdziwe. Konstrukcja ta znaczy: „Dopóki wyrażenie warunkowe jest prawdziwe, wykonuj instrukcje”.

Przykład 3.15

```
<script type="text/javascript">
<!--
i = 0;
while (i++ < 5) {
document.write("Pętla wykonana " + i + " raz/y<br>");
}
// -->
</script>
```

Pętla do ... while

Pętla `do ... while` jest odmianą pętli `while`. Jej składnia jest następująca:

```
do {
    blok instrukcji;
}
while (wyrażenie warunkowe);
```

Konstrukcja ta oznacza: „Wykonuj instrukcje, dopóki wyrażenie warunkowe jest prawdziwe”. W pętli `do ... while` blok instrukcji jest wykonywany co najmniej raz, nawet jeżeli warunek zapisany w wyrażeniu warunkowym jest fałszywy — ponieważ najpierw wykonywany jest ciąg instrukcji, a dopiero potem sprawdzany jest warunek.

Przykład 3.16

```
<script type="text/javascript">
<!--
var i = 1;
do {
document.write("Pętla wykonana" + i + "raz/y</br>");
}
while (i++ <= 5);
// -->
</script>
```

Instrukcja break

Instrukcja break jest instrukcją modyfikującą zachowanie się pętli. Służy do przerywania jej wykonywania.

Przykład 3.17

```
<script type="text/javascript">
<!--
var i = 0; k = 5;
do {
j = k * i;
document.write("Wynik " + j + "</br>");
if (j > 30) break;
}
while (i++ < 10);
// -->
</script>
```

Pętla do ... while będzie wykonywana, dopóki $i < 10$, ale jeżeli wartość zmiennej j obliczanej w pętli przekroczy 30, nastąpi przerwanie wykonywania powtarzających się instrukcji i wyjście z pętli.

Instrukcja continue

Instrukcja continue, podobnie jak break, służy do modyfikowania zachowania się pętli. Po jej napotkaniu następuje przerwanie wykonywania bieżącej iteracji i przejście na jej początek.

Przykład 3.18

```
<script type="text/javascript">
<!--
for (var i = 0; i < 30; i++) {
if ((i%3) != 0)
continue;
document.write(i + "; ");
}
// -->
</script>
```

Podany kod wyświetli liczby z zakresu od 0 do 30 podzielne przez 3. Jeżeli wynik dzielenia przez trzy nie jest liczbą całkowitą, następuje przerwanie wykonywania pętli i powrót na jej początek (liczby niepodzielne przez 3 nie będą wyświetlane).