

Temat: Instrukcja warunkowa w JavaScript - cd

Zagnieżdżanie instrukcji warunkowych

Zarówno w bloku if, jak i w bloku else mogą wystąpić dowolne instrukcje JavaScript. Oznacza to, że można tam umieścić kolejne instrukcje if, a zatem, że mogą być one **zagnieżdżane**. Schematycznie taka konstrukcja ma postać:

```
if (warunek1) {  
    if (warunek2) {  
        instrukcje1;  
    }  
    else {  
        instrukcje2;  
    }  
}  
else {  
    if (warunek3) {  
        instrukcje3;  
    }  
    else {  
        instrukcje4;  
    }  
}
```

Taka struktura ma znaczenie następujące: instrukcje1 zostaną wykonane, kiedy prawdziwe są warunki warunek1 i warunek2; instrukcje2 — kiedy prawdziwy jest warunek warunek1 i fałszywy jest warunek warunek2; instrukcje3 — kiedy fałszywy jest warunek warunek1 i prawdziwy jest warunek warunek3, a instrukcje instrukcje4, kiedy są fałszywe warunki warunek1 i warunek3.

Badanie złożonych warunków

Warunki, a dokładniej wyrażenia warunkowe, stosowane w instrukcji if mogą się składać z wielu członów połączonych operatorami logicznymi. Załóżmy, że pewne czynności (np. wyświetlenie napisu) mają być wykonane, gdy dana zmienna jest większa od 0, ale jednocześnie mniejsza od 10. Jednym ze sposobów realizacji tego zadania jest **użycie zagnieżdżonej instrukcji if** w postaci:

```
if(liczba > 0) {  
    if(liczba < 10) {  
        console.log("Liczba jest większa od 0 i mniejsza od  
10.");  
    }  
}
```

Kod umieszczony w bloku zewnętrznej instrukcji if zostanie wykonany, kiedy zmienna liczba jest większa od 0. W tym kodzie znajduje się druga instrukcja if, która sprawdza, czy zmienna liczba jest mniejsza od 10. Tym samym napis zostanie wyświetlony na ekranie tylko wtedy, gdy oba użyte warunki będą prawdziwe.

W takim przypadku nie trzeba jednak stosować zagnieżdżonych instrukcji warunkowych. Można skorzystać z **pojedynczej instrukcji if ze złożonym wyrażeniem warunkowym**. Takie wyrażenie tworzy się na takiej samej zasadzie jak na lekcjach matematyki. Skoro ma być wykryta sytuacja, kiedy liczba jest większa od 0 i mniejsza od 10, trzeba skorzystać z iloczynu logicznego, który w JavaScriptcie jest wyrażany za pomocą operatora &&. A zatem wyrażenie warunkowe będzie miało postać:

```
(liczba > 0) && (liczba < 10)
```

Cała instrukcja if będzie więc wyglądała następująco:

```
if((liczba > 0) && (liczba < 10)){  
    console.log("Liczba jest większa od 0 i mniejsza od  
10.");  
}
```

Ze względu na to, że operatory relacyjne > i < mają większy priorytet (są silniejsze) niż operator iloczynu logicznego &&, nawiasy w wyrażeniu warunkowym mogą być pominięte:

```
if(liczba > 0 && liczba < 10) ){  
    console.log("Liczba jest większa od 0 i mniejsza od  
10.");  
}
```

Konwersja ciągów na liczby

UWAGA: Wczytanie liczby z okna dialogowego do zmiennej s daje liczbę zapamiętaną jako napis.

```
var s = prompt("Podaj liczbę:");
```

Często należy wykonać **zamianę liczby zapisanej jako napis na typ liczbowy**. Potrzebne jest to m. in. do prawidłowego wykonania obliczeń matematycznych.

```
var liczba = parseInt(s);
```

albo (2 w jednym):

```
var liczba = parseInt(prompt("Podaj liczbę:"));
```

Aby mieć pewność, że zmienna ma określony typ, możemy dokonać konwersji poprzez rzutowanie za pomocą funkcji: `parseInt()`, `parseFloat()`, `Number()`, `String()`

Jest to przydatne, gdy pobierzemy od użytkownika dane jako string, a potem będziemy chcieli te dane dodać do siebie. Bez konwersji możliwe jest, że dojdzie do sklejenia napisów, a nie dodania liczb.

```
var s = "100";  
// rzutowanie na typ liczbowy z obcięciem do liczby całkowitej:  
var liczba = parseInt(s);  
// rzutowanie na typ liczbowy (z częścią ułamkową):  
var liczba = parseFloat(s);
```

```
var liczba = 52;  
// rzutowanie na typ string:  
var s = String(liczba);
```

Przykład 1. Przygotuj stronę html wykorzystującą poniższy kod.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Instrukcja if, złożone warunki- T5p1 </title>  
  </head>  
  <body>  
    <h3>1 sposób</h3>  
    <p id="output1"></p>  
    <script>  
      let liczba = parseInt(prompt("Podaj liczbę całkowitą"));  
      if(liczba > 0){  
        if(liczba < 10){  
          document.getElementById("output1").innerHTML = "Liczba jest większa od 0 i  
          mniejsza od 10.";  
        }  
      }  
    </script>  
  
    <h3>2 sposób</h3>  
    <p id="output2"></p>  
    <script>  
      if(liczba > 0 && liczba < 10){  
        if(liczba < 10){  
          document.getElementById("output2").innerHTML = "Liczba jest większa od 0 i  
          mniejsza od 10.";  
        }  
      }  
    </script>  
  </body>  
</html>
```

Przetestuj działanie skryptu

Przykład 2. Przygotuj stronę html wykorzystującą poniższy kod.

```
1  <! DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Instrukcja if, złożone warunki- T5p2 </title>
6  </head>
7  <body>
8      <h3>Instrukcja if, złożone warunki- T5p2</h3>
9      <script>
10         const day=prompt("Podaj dzień tygodnia");
11         if ((day == "sobota") || (day == "niedziela")) {
12             alert("Uwielbiam weekendy!");
13
14             confirm("Czy Ty też?");
15         }
16         else if (day == "poniedziałek") {
17             alert("Nie cierpię poniedziałków");
18         }
19         else {
20             document.write("Środek tygodnia");
21         }
22     </script>
23 </body>
</html>
```

Przetestuj działanie skryptu

Przykład 3. Przygotuj stronę html wykorzystującą poniższy kod.

Odpowiedni napis powinien zostać wyświetlony, kiedy wartość zmiennej liczba jest większa od 0, ale różna od 5 i różna od 10, lub też, gdy jest równa –8. Można tu wyróżnić 4 wyrażenia składowe:

- liczba > 0,
- liczba != 5,
- liczba != 10,
- liczba == -8,

które należy połączyć za pomocą operatorów logicznych.

```

<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Instrukcja if, złożone warunki- T5p3 </title>
  </head>
  <body>
    <h3>T5p3</h3>
    <p id="output"></p>
    <script>
      const liczba = parseInt(prompt("Podaj liczbę całkowitą"));
      if(((liczba > 0) && (liczba != 5) && (liczba != 10)) || (liczba == -8)){
        document.getElementById("output").innerHTML=
          "Zmienna liczba spełnia warunki zadania.";
      }
      else {
        document.getElementById("output").innerHTML=
          "Liczba nie spełnia wszystkich warunków.";
      }
    </script>
  </body>
</html>

```

Przetestuj działanie skryptu dla różnych wartości zmiennej liczba.

Operator warunkowy

Operator warunkowy pozwala w niektórych przypadkach na wygodne zastąpienie bloku if...else. Konstrukcja taka wygląda następująco:

```
(wyrażenie warunkowe) ? wartość1 : wartość2
```

Należy rozumieć to w sposób następujący: jeżeli wyrażenie warunkowe jest prawdziwe, czyli ma wartość true — całość przyjmuje wartość wartość1 (w miejscu wyrażenia podstawiana jest wartość1), w przeciwnym przypadku —wartość2.

Mówiąc inaczej: przypisanie

```
let zmienna = warunek ? wyrażenie1 : wyrażenie2
```

jest równoważne:

```

let zmienna;
if(warunek){
  zmienna = wyrażenie1;
}
else{
  zmienna = wyrażenie2;
}

```

}

Przykład 4. Przygotuj stronę html wykorzystującą poniższy kod.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Operator warunkowy- T5p4 </title>
6   </head>
7   <body>
8     <p id="output"></p>
9     <script>
10      let x = liczba = parseInt(prompt("Podaj
11      liczbę całkowitą"));
12      let y;
13      y = (x < 0) ? 10 : 20;
14      document.getElementById("output").innerHTML=
15      `Wartość y to ${y}.`;
16    </script>
17  </body>
18 </html>
```

Przetestuj działanie skryptu dla różnych wartości, wpisz zarówno liczby dodatnie jak i ujemne.

Najważniejsza jest tu linia `y = (x < 0) ? 10 : 20;`. Po lewej stronie operatora przypisania `=` znajduje się zmienna (`y`), natomiast po stronie prawej — wyrażenie warunkowe, czyli linia ta oznacza: przypisz zmiennej `y` wartość wyrażenia warunkowego. Wyrażenie: `(x < 0) ? 10 : 20.` oznacza: jeżeli wartość zmiennej `x` jest mniejsza od zera, przypisz wyrażeniu wartość 10, w przeciwnym przypadku (zmienna `x` większa lub równa zero) przypisz wyrażeniu wartość 20.

Instrukcja wyboru switch

Instrukcja wyboru **switch** (nazywana również instrukcją **switch...case**) pozwala w wygodny sposób sprawdzić ciąg warunków i wykonać różne instrukcje w zależności od wyników porównywania. Ma ona ogólną postać:

```
switch(wyrażenie){
  case wartość1 :
    instrukcje1;
    break;
  case wartość2 :
    instrukcje2;
    break;
  case wartość3 :
    instrukcje3;
    break;
  default :
    instrukcje4;
}
```

którą należy rozumieć następująco: sprawdź wartość wyrażenia *wyrażenie*; jeśli wynikiem jest *wartość1*, wykonaj *instrukcje1* i przerwij wykonywanie bloku switch (instrukcja **break**). Jeśli wynikiem jest *wartość2*, wykonaj *instrukcje2* i przerwij wykonywanie bloku switch, jeśli wynikiem jest *wartość3*, wykonaj *instrukcje3* i przerwij wykonywanie bloku switch. Jeśli nie zachodzi żaden z wymienionych przypadków, wykonaj *instrukcje4* i zakończ blok switch. **Blok default** jest opcjonalny i może zostać pominięty.

Jest to odpowiednik złożonej instrukcji **if...else if** w postaci:

```
if(wyrażenie == wartość1) {
    instrukcje1;
}
else if(wyrażenie == wartość2) {
    instrukcje2;
}
else if(wyrażenie == wartość3) {
    instrukcje3;
}
else{
    instrukcje4;
}
```

Przykład 5. Przygotuj stronę html wykorzystującą poniższy kod.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Instrukcja wyboru switch- T5p5 </title>
  </head>
  <body>
    <script>
      let liczba = parseFloat(prompt("Podaj liczbę
dwucyfrową podzielną przez 10:"));
      switch(liczba){
        case 10 :
          console.log("Zmienna liczba jest równa 10.");
          break;
        case 20:
          console.log("Zmienna liczba jest równa 20.");
          break;
        case 30:
          console.log("Zmienna liczba jest równa 30.");
          break;
        default:
          console.log("Zmienna liczba nie jest równa
ani 10, ani 20, ani 30.");
      }
    </script>
  </body>
</html>
```

Przetestuj działanie skryptu dla różnych liczb. Usuń funkcję `parseFloat` i sprawdź działanie skryptu w tej sytuacji.

Na początku skryptu została utworzona zmienna `liczba` o wartości podanej przez użytkownika. Dalej znajduje się instrukcja `switch`, w której najpierw obliczana jest wartość wyrażenia umieszczonego w nawiasie okrągłym. Ponieważ w tym przypadku jako wyrażenie występuje nazwa zmiennej, wartością wyrażenia staje się wartość tej zmiennej (czyli np. 20). Wartość ta jest porównywana do wartości występujących po słowach `case`, czyli 10, 20 i 30. Jeśli zgodność zostanie stwierdzona, zostaną wykonane instrukcje występujące w danym bloku `case`. Jeśli nie uda się dopasować wartości wyrażenia do żadnej z wartości występujących po słowach `case`, jest wykonywany blok `default`. Występująca po `console.log` instrukcja `break` przerywa wykonywanie instrukcji `switch`.

Ten sam efekt uzyskasz wykorzystując instrukcje `if else`:

Przykład 5a. Przygotuj stronę html wykorzystującą poniższy kod.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Instrukcja wyboru switch a if else- T5p5a
    </title>
  </head>
  <body>
    <script>
      let liczba = parseFloat(prompt("Podaj liczbę
      dwucyfrową podzielną przez 10:"));
      if(liczba==10){
        console.log("Zmienna liczba jest równa 10.");
      }
      else if(liczba==20){
        console.log("Zmienna liczba jest równa 20.");
      }
      else if(liczba==30){
        console.log("Zmienna liczba jest równa 30.");
      }
      else {
        console.log("Zmienna liczba nie jest równa
        ani 10, ani 20, ani 30.");
      }
    </script>
  </body>
</html>
```

Na instrukcję **break** występującą w instrukcji `switch` należy zwrócić szczególną uwagę. Jej przypadkowe pominięcie może doprowadzić do nieoczekiwanych wyników i błędów w skrypcie. Aby przekonać się, w jaki sposób działa instrukcja `switch` bez instrukcji `break`, wykonaj kolejny przykład.

Przykład 6. Zmodyfikuj kod z przykładu 5, usuwając z niego wszystkie instrukcje break i podaj wartość 10 w oknie dialogowym. Zaobserwuj działanie skryptu.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Instrukcja wyboru switch- T5p5 </title>
  </head>
  <body>
    <script>
      let liczba = parseFloat(prompt("Podaj liczbę
      dwucyfrową podzieloną przez 10:"));
      switch(liczba){
        case 10 :
          console.log("Zmienna liczba jest równa 10.");

        case 20:
          console.log("Zmienna liczba jest równa 20.");

        case 30:
          console.log("Zmienna liczba jest równa 30.");

        default:
          console.log("Zmienna liczba nie jest równa
          ani 10, ani 20, ani 30.");

      }
    </script>
  </body>
</html>
```

Po uruchomieniu skryptu w przeglądarce pojawi się obraz, który nie spełnia swojego zadania. Wartość zmiennej nie może przecież jednocześnie spełniać kilku przeciwstawnych warunków. Kod w tym przykładzie działa następująco: jeśli w którymś z bloków (przypadków) case zostanie wykryta zgodność z wyrażeniem występującym za switch, zostaną wykonane wszystkie dalsze instrukcje, aż do napotkania instrukcji break lub dotarcia do końca instrukcji switch. W kodzie przykładu 6 zgodność jest stwierdzana już w pierwszym bloku case, jest więc wykonywana znajdująca się w nim instrukcja console.log. Ponieważ jednak w bloku tym nie ma instrukcji break, są wykonywane instrukcje znajdujące się w kolejnych blokach case. W tym bloku również brakuje break, a zatem są wykonywane instrukcje znajdujące się po słowie default. Tym samym wykonane zostaną wszystkie znajdujące się w kodzie instrukcje console.log.

Ćwiczenie 1

Rozbuduj przykład 1 tak, aby skrypt wyświetlał także komunikaty: „Podałś liczbę większą lub równą 10” lub „Podałś liczbę mniejszą lub równą 0”. Wykonaj zadanie w obu wersjach instrukcji warunkowej.

Zapisz skrypt pod nazwą T5cw1.html

Ćwiczenie 2

Napisz skrypt, który zapyta o podanie koloru w języku angielskim, a następnie wyświetli tłumaczenie koloru na język polski (red = czerwony, blue = niebieski, green = zielony, yellow = żółty, purple = fioletowy, white = biały, black = czarny, brown = brązowy, pink = różowy, orange = pomarańczowy). Jeżeli użytkownik poda inny kolor, to skrypt wyświetli komunikat „Nie znam takiego koloru”.

- a. Wykorzystaj instrukcję if-else if-else
- b. Wykorzystaj switch

Zapisz skrypt pod nazwami T5cw2a.html i T5cw2b.html

Ćwiczenie 3

Napisz skrypt, który pobierze liczbę od użytkownika i wyświetli jeden z komunikatów „Podałeś liczbę parzystą”, „Podałeś liczbę nieparzystą”, wykorzystując operator warunkowy.

Wskazówka:

Liczba parzysta daje resztę 0 z dzielenia przez 2, a liczba nieparzysta daje resztę 1. Do obliczenia reszty wykorzystaj operator %.

Zapisz skrypt pod nazwą T5cw3.html