

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

Podczas pisania skryptów bardzo często zachodzi konieczność wykonywania różnych operacji w zależności od prawdziwości jakiegoś warunku. Najprostszym przykładem może być wyświetlanie różnych tekstów w zależności od tego, czy wybrana zmienna ma wartość mniejszą, czy większą od zera. Na takie rozwidlenie kodu programu pozwalają instrukcje warunkowe. W JavaScriptcie, tak jak w innych popularnych językach skryptowych oraz klasycznych językach programowania, do badania warunków służą różne postacie instrukcji if. **Instrukcja if**

Instrukcja if występuje w kilku wersjach, najprostsza z nich ma schematyczną postać:

```
if (warunek){  
  
    //instrukcje do wykonania  
  
}
```

Zapis ten należy rozumieć: jeśli warunek (umieszczony w nawiasie okrągłym, za słowem if) jest prawdziwy, wykonaj blok instrukcji znajdujący się pomiędzy znakami nawiasu klamrowego. Każda pojedyncza instrukcja w bloku pomiędzy znakami nawiasu klamrowego musi się kończyć znakiem średnika, natomiast za instrukcją if średnik może, ale nie musi występować (z reguły jest pomijany).

Znak rozpoczynający nawias klamrowy nie musi się znajdować w tym samym wierszu co rozpoczęcie instrukcji if. Niektórzy programiści preferują przeniesienie go do kolejnego wiersza kodu. Zapis wygląda wtedy następująco:

```
if (warunek)  
  
{  
  
    //instrukcje do wykonania  
  
}
```

Obie postacie mają takie samo znaczenie. Różnica dotyczy wyłącznie sposobu zapisu. Należy stosować ten, który jest dla nas czytelniejszy.

W sytuacji, kiedy w bloku instrukcji if miałyby się znaleźć tylko jedna instrukcja, czyli miałyby on postać:

```
if (warunek){  
  
    instrukcja;  
  
}
```

dopuszczalne jest pominięcie znaków nawiasu klamrowego, jednak obligatoryjny jest wtedy średnik kończący. Zatem prawidłowy jest zapis:

```
if (warunek)  
  
    instrukcja;  
  
lub  
  
if (warunek) instrukcja;
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

Do skonstruowania warunku wykorzystywane są operatory relacyjne przedstawione w podrozdziale dotyczącym operatorów. Zobaczmy, jak cała taka konstrukcja będzie wyglądała w praktyce. Przykładowy kod wykorzystujący instrukcję warunkową `if` do stwierdzenia, czy wartość zmiennej jest mniejsza od zera, czy też nie, został zaprezentowany na listingu 2.4.

Listing 2.4. Działanie instrukcji warunkowej `if`

```
<script type="text/javascript">
var liczba = -1;
if(liczba < 0){
    document.write("Wartość zmiennej liczba jest mniejsza od 0.");
}
if(liczba >= 0){
    document.write("Wartość zmiennej liczba nie jest mniejsza od 0.");
}
</script>
```

Powstała zmienna `liczba`, której została przypisana wartość `-1`. Pierwsza instrukcja warunkowa `if` sprawdza, czy wartość zapisana w zmiennej `liczba` jest mniejsza od zera (`liczba < 0`), i jeśli tak, wyświetla stosowną informację. Druga instrukcja `if` sprawdza, czy zmienna `liczba` jest większa od zera lub równa, i jeśli tak, również wyświetla informację o tym fakcie. Ponieważ wartość zmiennej jest mniejsza od zera, na ekranie pojawi się jedynie napis widoczny na rysunku 2.1.

Rysunek 2.1. Efekt działania skryptu z listingu 2.4

Oczywiście, gdyby pierwsza instrukcja została zmieniona np. na:

```
liczba = 10;
```

na ekranie pojawiłby się napis: `Wartość zmiennej liczba nie jest mniejsza od 0.` **Instrukcja `if...else`**

W poprzednio zaprezentowanym przykładzie zostały użyte dwie instrukcje `if` do sprawdzenia dwóch przeciwstawnych, wykluczających się wzajemnie warunków. Wykluczających się, ponieważ zmienna `liczba` mogła być albo mniejsza od zera, albo większa od zera lub równa mu (czyli nie mniejsza od zera). W takiej sytuacji można jednak również użyć rozbudowanej instrukcji `if`, nazywanej niekiedy instrukcją `if-else` (z reguły jednak mówi się po prostu o instrukcji `if`). Jej ogólna postać jest następująca:

```
if (warunek){
    //instrukcje do wykonania, kiedy warunek jest prawdziwy
}
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
else{  
  
    //instrukcje do wykonania, kiedy warunek jest fałszywy  
  
}
```

Jak widać, został dodany blok else, który jest wykonywany w sytuacji, kiedy warunek jest fałszywy. Zatem funkcjonalny odpowiednik poprzedniego skryptu wykorzystujący tę formę instrukcji if ma postać zaprezentowaną na listingu 2.5.

Listing 2.5. Działanie instrukcji warunkowej if...else

```
<script type="text/javascript">  
  
var liczba = -1;  
  
if(liczba < 0){  
  
    document.write("Wartość zmiennej liczba jest mniejsza od 0.");  
  
}  
  
else{  
  
    document.write("Wartość zmiennej liczba nie jest mniejsza od 0.");  
  
}  
  
</script> Instrukcja if...else if
```

Trzecia wersja instrukcji if pozwala na badanie wielu warunków. Otóż po bloku if może wystąpić wiele dodatkowych bloków else if. Schematyczna postać takiej konstrukcji to:

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

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
..instrukcjeN;  
  
}  
  
else{  
  
..instrukcjeM;  
  
}
```

Co oznacza: jeżeli warunek1 jest prawdziwy, to zostaną wykonane instrukcje1; w przeciwnym razie, jeżeli jest prawdziwy warunek2, to zostaną wykonane instrukcje2; w przeciwnym razie, jeśli jest prawdziwy warunek3, to zostaną wykonane instrukcje3, itd. Jeżeli żaden z warunków nie będzie prawdziwy, to zostaną wykonane instrukcjeM. Ostatni blok else jest jednak opcjonalny i nie musi być stosowany.

Taka konstrukcja może być wykorzystana np. w sytuacji, kiedy programista chce wykonać wiele różnych instrukcji w zależności od stanu zmiennej. W jaki sposób to zrobić, obrazuje skrypt widoczny na listingu 2.6.

Listing 2.6. Działanie złożonej instrukcji warunkowej if...else if

```
<script type="text/javascript">  
  
var liczba = 30;  
  
if(liczba == 0){  
  
    document.write("Wartość zmiennej liczba jest równa 0.");  
  
}  
  
else if(liczba == 10){  
  
    document.write("Wartość zmiennej liczba jest równa 10.");  
  
}  
  
else if(liczba == 20){  
  
    document.write("Wartość zmiennej liczba jest równa 20.");  
  
}  
  
else{  
  
    document.write("Wartość zmiennej liczba nie jest równa ani 0, ani 10, ani 20.");  
  
}  
  
</script> 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:

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
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. Zostało to zobrazowane w widocznej poniżej tabeli 2.15. Oczywiście nie trzeba się ograniczać do przedstawionych tu dwóch poziomów zagnieżdżenia — może być ich dużo więcej.

Tabela 2.15. Zależności między wykonywaniem instrukcji a prawdziwością warunków

Wykonywane instrukcje

warunek1

warunek2

warunek3

instrukcje1

prawdziwy

prawdziwy

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

–

instrukcje2

prawdziwy

falszywy

instrukcje3

falszywy

–

prawdziwy

instrukcje4

falszywy

–

falszywy

Spróbujmy wykorzystać taką zagnieżdżoną instrukcję `if` w pierwszym „poważnym” skrypcie. Będzie on obliczał rozwiązania równania kwadratowego o zadanych parametrach. Równanie takie ma ogólną postać: $A \cdot x^2 + B \cdot x + C = 0$, gdzie A , B i C to parametry równania. Równanie ma rozwiązanie w zbiorze liczb rzeczywistych, jeśli parametr D (delta) równy $B^2 - 4 \cdot A \cdot C$ jest większy od zera lub równy zero. Jeśli D równa jest 0, mamy jedno rozwiązanie wyliczane ze wzoru $-B / 2 \cdot A$; jeśli D jest większa od zera, mamy dwa rozwiązania: $x_1 = (-B + \sqrt{D}) / (2 \cdot A)$ i $x_2 = (-B - \sqrt{D}) / (2 \cdot A)$. Taka liczba warunków doskonale nadaje się do przećwiczenia działania instrukcji `if...else`. Jediną niedogodnością skryptu będzie to, że parametry A , B i C będą musiały być wprowadzone bezpośrednio w jego kodzie, nie został bowiem jeszcze przedstawiony sposób na przekazanie danych z formularzy HTML (temat ten będzie omówiony w dalszej części kursu). Kod skryptu został przedstawiony na listingu 2.7.

Listing 2.7. Skrypt obliczający pierwiastki równania kwadratowego

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

```
//deklaracje zmiennych określających parametry równania
```

```
var A = 1;
```

```
var B = 1;
```

```
var C = -2;
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
//wyświetlenie parametrów równania na stronie
```

```
document.write("&quot;Parametry równania: <br />&quot;);
```

```
document.write("&quot;A = &quot; + A + &quot;; B = &quot; + B + &quot;; C = &quot; + C + &quot;;<br />&quot;);
```

```
//sprawdzenie, czy jest to równanie kwadratowe
```

```
if (A == 0){
```

```
    //A jest równe zero, równania nie jest kwadratowe
```

```
    document.write("&quot;To nie jest równanie kwadratowe: A = 0!&quot;);
```

```
}
```

```
else{
```

```
    //A jest różne od zera, równanie jest kwadratowe
```

```
    //obliczenie delty
```

```
    var delta = B * B - 4 * A * C;
```

```
    //jeśli delta mniejsza od zera
```

```
    if (delta < 0){
```

```
        document.write (&quot;Delta < 0 <br/>&quot;);
```

```
        document.write (&quot;To równanie nie ma rozwiązania w zbiorze liczb rzeczywistych!&quot;);
```

```
    }
```

```
    //jeśli delta większa od zera lub równa zero
```

```
    else{
```

```
        //jeśli delta jest równa zero
```

```
        if (delta == 0){
```

```
            //obliczenie wyniku
```

```
            var wynik = - B / 2 * A;
```

```
            document.write("&quot;Rozwiązanie: x = &quot; + wynik);
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
}  
  
//jeśli delta jest większa od zera  
  
else{  
  
    //obliczenie wyników i wyświetlenie ich na stronie  
  
    wynik = (- B + Math.sqrt(delta)) / 2 * A;  
  
    document.write (&quot;Rozwiązanie: x1 = &quot; + wynik);  
  
    wynik = (- B - Math.sqrt(delta)) / 2 * A;  
  
    document.write(&quot;; x2 = &quot; + wynik);  
  
}  
  
}  
  
}  
  
</script>
```

Kod rozpoczyna się od zadeklarowania i zainicjowania trzech zmiennych: A, B i C, odzwierciedlających parametry równania, oraz wyświetlenia ich wartości na ekranie. Za pomocą instrukcji if jest sprawdzane, czy zmienna A jest równa 0. Jeśli tak, oznacza to, że równanie nie jest kwadratowe, jest więc wyświetlana stosowna informacja i... jest to koniec działania skryptu. Jeśli jednak A jest różne od zera, można przystąpić do obliczenia delty. Wynik obliczeń należy przypisać zmiennej delta. Kolejny krok to sprawdzenie, czy delta nie jest przypadkiem mniejsza od zera. Jeśli jest, oznacza to, że równanie nie ma rozwiązań w zbiorze liczb rzeczywistych, trzeba więc wyświetlić odpowiedni komunikat — po jego wyświetleniu skrypt kończy działanie.

Kiedy jednak delta nie jest mniejsza od zera, należy przystąpić do sprawdzenia kolejnych warunków. Jeśli jest równa 0, można od razu obliczyć rozwiązanie równania ze wzoru $-B / (2 * A)$. Wynik tych obliczeń należy przypisać zmiennej pomocniczej wynik i wyświetlić komunikat z rozwiązaniem na ekranie.

Gdy delta jest większa od 0, istnieją dwa pierwiastki (czyli rozwiązania) równania. Zostaną obliczone w wierszach:

```
wynik = (- B + sqrt(delta)) / 2 * A;
```

oraz:

```
wynik = (- B - sqrt(delta)) / 2 * A;
```

Rezultat obliczeń zostanie wyświetlony rzecz jasna na ekranie (rysunek 2.2).

Rysunek 2.2. Rozwiązanie równania kwadratowego

Nieznana nam instrukcja `Math.sqrt(delta)` powoduje obliczenie pierwiastka kwadratowego (drugiego stopnia) z wartości zawartej w zmiennej delta. Więcej informacji o operacjach matematycznych zostanie przedstawione w

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

dalszej części kursu.

Nic też nie stoi na przeszkodzie, aby w skrypcie obliczającym rozwiązania równań kwadratowych użyć instrukcji `if...else if`. Kod przyjąłby wtedy postać widoczną na listingu 2.8.

Listing 2.8. Użycie instrukcji `if...else if` w skrypcie obliczającym pierwiastki równania kwadratowego

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

//deklaracje zmiennych określających parametry równania

var A = 1;

var B = 1;

var C = -2;

//wyświetlenie parametrów równania na stronie

document.write("Parametry równania: <br />");

document.write("A = " + A + ", B = " + B + ", C = " + C + "<br />");

//sprawdzenie, czy jest to równanie kwadratowe

if (A == 0){

    //A jest równe zero, równanie nie jest kwadratowe

    document.write("To nie jest równanie kwadratowe: A = 0!");

}

else{

    //A jest różne od zera, równanie jest kwadratowe

    //obliczenie delty

    var delta = B * B - 4 * A * C;

    //jeśli delta mniejsza od zera

    if (delta < 0){

        document.write ("Delta < 0 <br/>");

    }

}

</script>
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
document.write (&quot;To równanie nie ma rozwiązania w zbiorze liczb rzeczywistych!&quot;);  
  
}  
  
//jeśli delta jest równa zero  
  
else if (delta == 0){  
  
    //obliczenie wyniku  
  
    var wynik = - B / 2 * A;  
  
    document.write(&quot;Rozwiązanie: x = &quot; + wynik);  
  
}  
  
//jeśli delta jest większa od zera  
  
else{  
  
    //obliczenie wyników i wyświetlenie ich na stronie  
  
    wynik = (- B + Math.sqrt(delta)) / 2 * A;  
  
    document.write (&quot;Rozwiązanie: x1 = &quot; + wynik);  
  
    wynik = (- B - Math.sqrt(delta)) / 2 * A;  
  
    document.write(&quot;; x2 = &quot; + wynik);  
  
}  
  
}  
  
</script>
```

Ogólna zasada działania pozostaje tu taka sama jak w poprzednim przykładzie, zmniejszyła się tylko o jedną liczbę instrukcji warunkowych, a także poziom zagnieżdżenia. W związku z tym kod wydaje się być bardziej czytelny. Funkcjonalnie obie wersje są jednak równoważne i należy stosować tę, która bardziej nam odpowiada. Ogólna zasada jest taka, że staramy się dążyć do jak najmniejszej liczby instrukcji w skrypcie, ale nie może się to odbywać kosztem czytelności kodu. **Wyrażenia warunkowe**

Warunki, a dokładniej wyrażenia warunkowe, stosowane w instrukcji if, mogą być bardziej złożone niż dotychczas prezentowane. Mogą się składać z wielu członów połączonych operatorami logicznymi. Załóżmy, że należy wykonać pewne czynności, np. wyświetlić napis, kiedy zmienna liczba jest większa od zera, ale jednocześnie mniejsza od 10. Istnieją dwa sposoby realizacji tego zadania. Pierwszy to wykorzystanie zagnieżdżonej instrukcji if, w postaci:

```
if(liczba > 0){  
  
    if(liczba < 10){  
  
        document.write(&quot;Liczba jest większa od 0 i mniejsza od 10.&quot;);  
  
    }  
  
}
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

```
}  
  
}
```

Drugi to wykorzystanie pojedynczej instrukcji if ze złożonym wyrażeniem warunkowym. Jak zbudować takie wyrażenie? Na takiej samej zasadzie, jak na szkolnych lekcjach matematyki. Skoro interesuje nas sytuacja, kiedy liczba jest większa od zera i mniejsza od dziesięciu, 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)){  
  
    document.write("Liczba jest większa od 0 i mniejsza od 10.");  
  
}
```

Z informacji zamieszczonych w prezentowanej wcześniej tablicy przedstawiającej priorytety operatorów wynika, że operatory relacyjne `>` i `<` mają większy priorytet (są silniejsze) niż operator iloczynu logicznego `&&`. Wynika z tego, że można pominąć nawiasy w wyrażeniu warunkowym, które równie dobrze może mieć postać:

```
liczba > 0 && liczba < 10
```

Jest to w pełni dopuszczalne, aczkolwiek niezalecane, gdyż zaciemnia kod i utrudnia jego analizę, szczególnie w przypadku bardziej złożonych wyrażeń.

I jeszcze jeden przykład. Tym razem 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, jak poniżej:

```
(liczba > 0) && (liczba != 5) && (liczba != 10) || (liczba == -8)
```

Pełna instrukcja if przyjmie zatem poniższą postać:

```
if(((liczba > 0) && (liczba != 5) && (liczba != 10)) || (liczba == -8)){  
  
    document.write("Zmienna liczba spełnia wszystkie warunki.");  
  
}
```

Instrukcje warunkowe

Dodał Administrator
niedziela, 14 marzec 2010 10:03

}