

Instrukcje warunkowe

If

Instrukcja warunkowa służy do ograniczenia wykonania jakiejś czynności do momentu, kiedy spełnione zostaną podane warunki.

Podstawowa budowa instrukcji warunkowej:

```
if (warunek) {  
    instrukcje wykonywane, jeśli warunek zostanie spełniony;  
}
```

Pełna budowa instrukcji warunkowej języka JavaScript:

```
if (warunek) {  
    instrukcje wykonywane, gdy warunek zostanie spełniony;  
} else if (warunek2) {  
    instrukcje wykonywane, gdy warunek2 zostanie spełniony;  
} else if (warunek...) {  
    instrukcje wykonywane, gdy warunek... zostanie spełniony;  
} else {  
    instrukcje wykonywane, gdy żaden z powyższych warunków nie był spełniony;  
}
```

Co oznacza stwierdzenie, że warunek jest spełniony? Nic innego, jak to, iż jego wartość logiczna jest prawdą.

Jeśli zapytamy użytkownika, czy ma więcej niż 18 lat, to instrukcje zawarte wewnątrz bloku **if** zostaną wykonane tylko wtedy, gdy to będzie prawda.

Gdy mamy więcej niż jeden warunek, sprawdzane są one po kolei. Gdy którykolwiek jest spełniony, kolejne nie są już sprawdzane.

Zastanówmy się przez chwilę, kiedy warunek nie jest spełniony?

Dzieje się tak, gdy wartość logiczna warunku jest fałszem (false) czyli logicznym 0. Każda inna wartość jest prawdziwa.

Sprawdźmy działanie instrukcji warunkowej na prostym przykładzie:

```
1.   var wiek=18;
2.
3.   if(wiek >= 18){
4.       console.log("Mam do czynienia z osobą pełnoletnią");
5.   }else{
6.       console.log("Nie dla psa kiełbasa.");
7.   }
```

Wynikiem oczywiście będzie zawartość bloku if, czyli wypisanie na ekran potwierdzenia pełnoletności.

Przeanalizujmy kolejny przykład:

```
1.   var a = 1, b=1, c=1;
2.   if(a>5)
3.       b++;
4.       c++;
5.
6.   document.write(b+'<br>'+c);
```

```
1
2
```

Wypisujemy na ekran wartości zmiennych b oraz c. Początkowo obie te zmienne miały wartość 1, przy końcowym wypisaniu jedna z nich zmieniła wartość na 2. Dlaczego tylko jedna?

Ano dlatego, że bloku instrukcji if nie zamknęliśmy w klamry. Bez oznaczenia bloku instrukcji zasięgiem działania instrukcji warunkowej zostanie objęta tylko pierwsza instrukcja.

Wynika stąd, że zależna od wielkości a była tylko zmiana wartości zmiennej b. Jako, że a nie jest >5, instrukcja zwiększająca b o 1 nie wykonała się. Zwiększenie C jest już poza blokiem instrukcji warunkowej i wykona się zawsze.

Zmodyfikujmy powyższy przykład dodając klamry.

```

1.  var a = 1, b=1, c=1;
2.  if(a>5) {
3.    b++;
4.    c++;
5.  }
6.  document.write(b+'<br />' + c);

```

```

1
1

```

Tym razem blok instrukcji obejmował zarówno zwiększenie zmiennej b oraz c. Warunek nie został spełniony. Wartości zmiennych pozostały bez zmian.

Prosty przykład pobrania od wartości wartości zmiennej całkowitoliczbowej i sprawdzenie czy jest ona parzysta czy nieparzysta.

```

1.  var a = parseInt(prompt());
2.  if(a%2==0) document.write('Liczba parzysta');
3.  if(a%2!=0) document.write('Liczba nieparzysta');

```

prompt() – wyskakujące okienko z prośbą o wpisanie wartości przez użytkownika

parseInt() – zamiana wprowadzonej wartości na liczbę całkowitą

a%2 == 0 – sprawdzenie, czy reszta z dzielenia wartości zmiennej a przez 2 jest równe 0, jeśli tak, to mamy do czynienia z liczbą parzystą

Zawarte w powyższym przykładzie dwa bloki instrukcji warunkowych możemy zastąpić blokiem else (w przeciwnym razie).

```

1.  var a = parseInt(prompt());
2.  if(a%2==0) document.write('Liczba parzysta');
3.  else document.write('Liczba nieparzysta');

```

Pozbyliśmy się jednego warunku, poprawiliśmy czytelność kodu, a else wykona się zawsze wtedy, kiedy warunek instrukcji warunkowej nie zostanie spełniony.

Przeanalizuj kolejny przykład. Tym razem sprawdzenie, czy liczba jest mniejsza, większa czy równa 0.

```
1. var a = parseInt(prompt('Podaj liczbę'));
2. if(a>0) document.write('Liczba większa od 0');
3. else if(a<0) document.write('Liczba mniejsza od 0');
4. else document.write('Liczba równa 0');
```

Kolejny przykład pokaże możliwość zagnieżdżania instrukcji warunkowych wewnątrz siebie:

```
1. var dzien = "wtorek";
2. var pora = "rano";
3.
4. if(dzien == "wtorek"){
5.     if(pora == "rano"){
6.         console.log("Nareszcie!");
7.     }
8. }
```

Aby w konsoli wypisało się słowo nareszcie, muszą zostać spełnione jednocześnie dwa warunki: zmienna dzień musi mieć wartość wtorek oraz zmienna pora wartość rano.

Moglibyśmy to samo zawrzeć w jednej instrukcji warunkowej:

```
1. var dzien = "wtorek";
2. var pora = "rano";
3.
4. if(dzien == "wtorek" && pora == "rano"){
5.     console.log("Nareszcie!");
6. }
```

Łączenia warunków możemy dokonać za pomocą dwóch symboli:

- warunek1 && warunek2 (i, logiczne mnożenie, and) oba warunki muszą zostać spełnione
- warunek1 || warunek2 (lub, suma logiczna, or) przynajmniej jeden warunek musi zostać spełniony

Instrukcja switch jest kolejnym sposobem testowania warunków:

```

1.  var a = parseInt(prompt('Podaj liczbę'));
2.
3.  switch(a) {
4.      case 1:
5.          document.write('Wybrano opcję 1');
6.          break;
7.      case 2:
8.          document.write('Wybrano opcję 2');
9.          break;
10.     case 3:
11.         document.write('Wybrano opcję 3');
12.         break;
13.     default:
14.         document.write('Po co naciskasz nie to co powinienes?!');
15.         break;
16. }

```

Wartość zmiennej a, będącej sprawdzaną przez instrukcję switch musi być liczba całkowita. Na powyższym przykładzie każdy z przypadków (case) sprawdza czy zmienna a przyjmuje kolejno wartości 1, 2, 3. Jeśli nie, zostanie wykonany blok domyślny (default). Instrukcje zamknięte w danym bloku case muszą zostać zakończone słówkiem break (przerwij).

Poniżej przedstawiam przykład wykorzystania instrukcji switch do samodzielnej analizy:

```

1.  switch (new Date().getDay()) {
2.      case 0:
3.          dzien = "Niedziela";
4.          break;
5.      case 1:
6.          dzien = "Poniedziałek";
7.          break;
8.      case 2:
9.          dzien = "Wtorek";
10.         break;
11.     case 3:
12.         dzien = "Środa";
13.         break;
14.     case 4:
15.         dzien = "Czwartek";
16.         break;
17.     case 5:
18.         dzien = "Piątek";
19.         break;
20.     case 6:
21.         dzien = "Sobota";
22.     }
23.  document.write(dzien);

```

Instrukcja switch

Struktura:

```
switch (wyrażenie) {  
    case przypadek1:  
        instrukcjaGdyPrzypadek1  
        break;  
    case przypadek2:  
        instrukcjaGdyPrzypadek2  
        break;  
    default:  
        instrukcjaGdyBrakDopasowanegoPrzypadku  
}
```

Instrukcja switch przyjmuje dowolne wyrażenie, najczęściej jest to po prostu zmienna. Na podstawie wartości zmiennej następuje dopasowanie do konkretnego przypadku (**case**). Jeśli wartość zmiennej pasuje do któregoś z przypadków, wtedy wykonywany jest blok kodu zawarty w tym przypadku, a jeżeli nie pasuje do żadnego przypadku, wtedy wywoływany jest kod z instrukcji **default** (**odpowiednik else w instrukcji if**). Przypadków (case'ów) może być nieskończenie wiele. Ważne, aby w **każdym bloku kodu case (przypadku) umieszczać break'a**, dzięki czemu jeżeli wykona się kod z danego przypadku to nastąpi zakończenie instrukcji. Jeżeli zabraknie słowa kluczowego break, to program nie "wyskoczy" z danej instrukcji case i będzie się wykonywał tak długo, aż trafi na innego break'a lub jeśli nie będzie go wcale, to wykona wszystkie kolejne przypadki. Możesz zobaczyć to na przykładzie:

```
let number = 1;
switch (number) {
  case 1:
    console.log('1');
  case 2:
    console.log('2');
  case 3:
    console.log('3');
  default:
    console.log('default');
}
```

Wynikiem powyższego programu będzie:

```
//1
//2
//3
//default
```

Poprawny kod powinien wyglądać w taki sposób:

```
let number = 1;
switch (number) {
  case 1:
    console.log('1');
    break;
  case 2:
    console.log('2');
    break;
  case 3:
    console.log('3');
    break;
  default:
    console.log('default');
}
```

Kod, który będzie robił to samo co powyższy switch, można zapisać również za pomocą instrukcji warunkowej if:

```
let number = 1;
if (number === 1)
  console.log('1');
else if (number === 2)
  console.log('2');
else if (number === 3)
  console.log('3');
else
  console.log('default');
```

Kiedy użyć instrukcji switch zamiast if? Instrukcji switch najlepiej użyć zawsze wtedy, gdy mamy więcej możliwych przypadków. Zamiast rozpisywać np. 5 razy if else, to lepszym pomysłem w takiej sytuacji będzie użycie po prostu instrukcji switch.

Co warto zapamiętać

- do wielu warunków mamy instrukcję switch zamiast if
- za pomocą instrukcji switch możemy sprawdzać pojedyncze warunki lub grupę warunków
- słowo kluczowe break pozwala wyjść z instrukcji po wykonaniu warunku, ale nie jest obowiązkowe
- klauzula default wykonywana jest, gdy żaden inny warunek się nie wykonał, nie jest obowiązkowa

Operator warunkowy