

# JavaScript – skryptowy język programowania

## *Umieszczanie skryptów w dokumencie*

### Zagadnienia

- Czym jest JavaScript?
- Jakie są sposoby na umieszczenie skryptu JavaScript?
- Jak umieszczać skrypt w dokumencie HTML?
- Jak korzystać ze skryptów zewnętrznych?

**JavaScript** jest językiem skryptowym pozwalającym na rozszerzenie standardowych dokumentów HTML m.in. o możliwość interakcji z użytkownikiem oraz na sprawdzanie poprawności informacji wprowadzanych przez użytkowników. Powstał w połowie lat 90. XX wieku w firmie Netscape jako język skryptowy LiveScript. W 1995 roku w wyniku współpracy z firmą Sun Microsystems ugruntował się jako język JavaScript. Jego głównym autorem jest Brendan Eich.

Program napisany w języku programowania wysokiego poziomu (C, C++, JAVA) przed zastosowaniem **kodu źródłowego**, którym jest uporządkowany ciąg instrukcji, wymaga skompilowania. Procesem tym zajmuje się **kompilator**, program którego zadaniem jest przetłumaczenie całego kodu źródłowego na **kod maszynowy**, ciąg liczb w systemie dwójkowym zrozumiały dla procesora. Podobne zadanie wykonuje **interpreter**. Jego sposób działania opiera się na pobraniu pojedynczej linii kodu, przetłumaczeniu jej, przekazaniu do procesora i przejściu do kolejnej linii kodu. Proces ten powtarza się do czasu przetłumaczenia całego programu.

JavaScript jest interpretowanym językiem programowania. Żeby zobaczyć efekty działania programu, nie trzeba go kompilować do kodu maszynowego. Wystarczy przeglądarka internetowa mająca włączoną obsługę języka JavaScript.

Jest to jeden z popularniejszych języków programowania działających po stronie klienta. To jedna z największych jego zalet, ponieważ wszystkie przeprowadzane operacje nie obciążają serwera. Zapewnia on również odpowiedni poziom bezpieczeństwa, ponieważ nie ma dostępu do systemu plików znajdujących się na komputerze użytkownika. Nieco trudności sprawia jedynie jego interpretacja przez różne przeglądarki. Może się zdarzyć, że prawidłowo napisany program będzie różnie działał w zależności od obsługującej go przeglądarki.

Sam język jest łatwy do opanowania. Do stworzenia dowolnego programu wystarczy zwykły edytor tekstu (np. Notatnik) lub dowolnie wybrane narzędzie wspomagające tworzenie strony WWW (Pajaczek, Zajaczek, Notepad++ i wiele innych). Wyniki działania programu obserwować można w dowolnie wybranej przeglądarce. Warto jednak pamiętać, żeby testować dany program w kilku dostępnych przeglądarkach (istnieją różnice w interpretacji). Wymagana jest również podstawowa wiedza z zakresu języka HTML.

Pierwszym sposobem na umieszczenie skryptu w dokumencie HTML jest wprowadzenie go bezpośrednio do kodu przez wykorzystanie znacznika `<script> ... </script>`. Można go wprowadzić w nagłówku **head** lub w głównej części dokumentu **body**.

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

Parametr **type** określa rodzaj języka skryptowego. Nie jest on jednak wymagany. Większość współczesnych przeglądarek zaakceptuje sam znacznik **script**, warto jednak stosować zapis **script** w całości. Natomiast całą treść skryptu (kodu) wprowadza się pomiędzy znaczniki.

Istnieje możliwość umieszczenia skryptu w osobnym pliku. Plik taki nazywamy skryptem zewnętrznym. Może on mieć dowolną nazwę oraz charakterystyczne rozszerzenie **.js**. Plik zewnętrzny zostaje powiązany z dokumentem HTML za pomocą znacznika **script** z dodatkowym oznaczeniem lokalizacji i nazwy pliku wraz z rozszerzeniem.

```
<script type="text/javascript" src="lokalizacja/nazwa.js">
</script>
```

W jednym dokumencie można umieścić kilka skryptów, zarówno osadzonych, jak i zewnętrznych. W przykładzie (list. 1.1) w nagłówku **head** wprowadzono skrypt zewnętrzny, a w głównej części dokumentu **body** – skrypt osadzony.

Listing 1.1

```
<html>
  <head>
    <title>
      Strona ze skryptem JavaScript
    </title>
    <meta http-equiv="Content-Type" content="text/html; charset=
iso-8859-2">
    <meta http-equiv="Content-Language" content="pl">
    <script type="text/javascript" src="lokalizacja/nazwa.js">
    </script>
  </head>
  <body>
<script type="text/javascript">
</script>
  </body>
</html>
```

Obecnie w internecie wiele stron oferuje gotowe przykłady skryptów, które można wykorzystać na stronie, nawet nie znając języka. Mogą to być elementy takie, jak: zegary, kalendarze, kalkulatory, efekty tła, przyciski i wiele innych.

## SPRAWDŹ SWOJĄ WIEDZĘ

1. Co to jest JavaScript?
2. Jakie znasz sposoby włączenia skryptu do dokumentu HTML?
3. Co jest potrzebne do uruchomienia skryptu napisanego w języku JavaScript?
4. Porównaj działanie kompilatora i interpretera.



# Instrukcja `document.write`

## Zagadnienia

- Za co odpowiada instrukcja `document.write`?
- Jak zmienić wygląd tekstu?
- Jak wyświetlać tekst na stronie za pomocą języka JavaScript?
- Jak stosować elementy języka HTML wewnątrz skryptu?

Kod języka JavaScript zbudowany jest z instrukcji. Instrukcją odpowiadającą za wyświetlenie tekstu na stronie jest `document.write`. Ciąg znaków, który ma zostać wyświetlony, należy umieścić w nawiasie okrągłym oraz w cudzysłowie. Na końcu każdej instrukcji znajduje się średnik:

```
document.write("Wyświetlany tekst");
```

Przed przystąpieniem do opisu instrukcji należy zapoznać się z zapisem ogólnym:

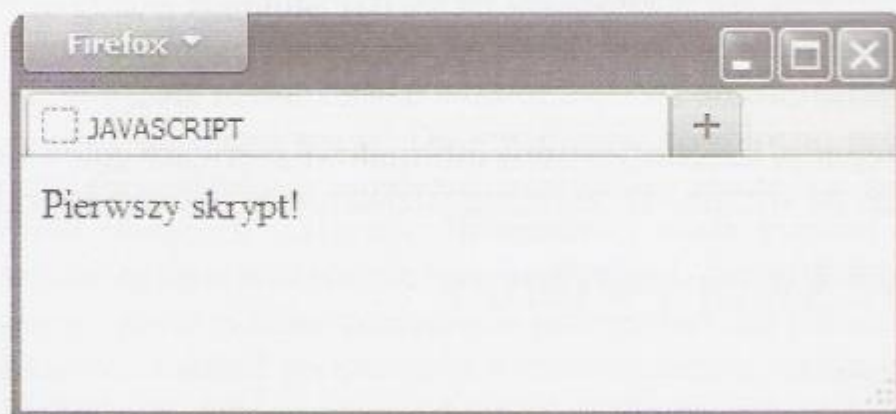
```
obiekt.metoda(argumenty metody);
```

Jak widać, `document` jest obiektem. Reprezentuje on aktualną stronę. Metoda `write` to funkcja działająca na obiekcie `document`, której zadaniem jest wyświetlenie argumentów (tekstu, wartości liczbowych) w oknie przeglądarki.

Pierwszy skrypt (list 2.1) ma za zadanie wyświetlić pojedynczą linijkę tekstu na stronie. Został on umieszczony w głównej części (`body`) dokumentu HTML.

Listing 2.1

```
<script type="text/javascript">
  document.write("Pierwszy skrypt!");
</script>
```



Rys. 2.1. Wynik działania skryptu z list. 2.1

Instrukcja `document.write` pozwala również na wyświetlanie wartości liczbowych. W takim przypadku wystarczy wprowadzić określoną wartość w nawiasie okrągłym bez cudzysłowu:

```
document.write(31);
```

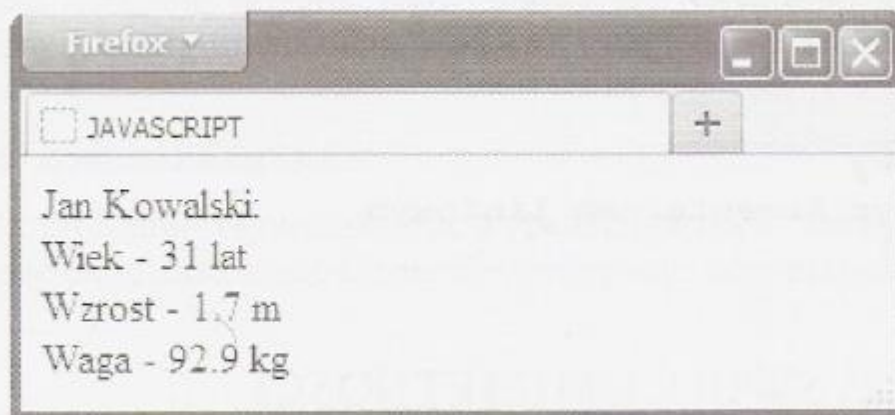
W jednej instrukcji można również umieścić bardziej rozbudowaną treść, składającą się zarówno z tekstu, jak i wartości liczbowych. Należy wówczas zastosować znak **+** do połączenia obu typów.

```
document.write("Wyświetlany tekst ma "+27+" znaków!");
```

Przykładowy skrypt (list. 2.2) prezentuje wyświetlanie tekstu oraz wartości liczbowych całkowitych i zmiennoprzecinkowych. W przypadku liczb ułamkowych wpisana w skrypcie wartość 1.70 po wyświetleniu na stronie została zaokrąglona do 1.7.

Listing 2.2

```
<script type="text/javascript">
document.write("Jan Kowalski: <br>");
document.write("Wiek - "+31+" lat <br>");
document.write("Wzrost - "+1.70+" m <br>");
document.write("Waga - "+92.9+" kg <br>");
</script>
```



Rys. 2.2. Wynik działania skryptu z list. 2.2

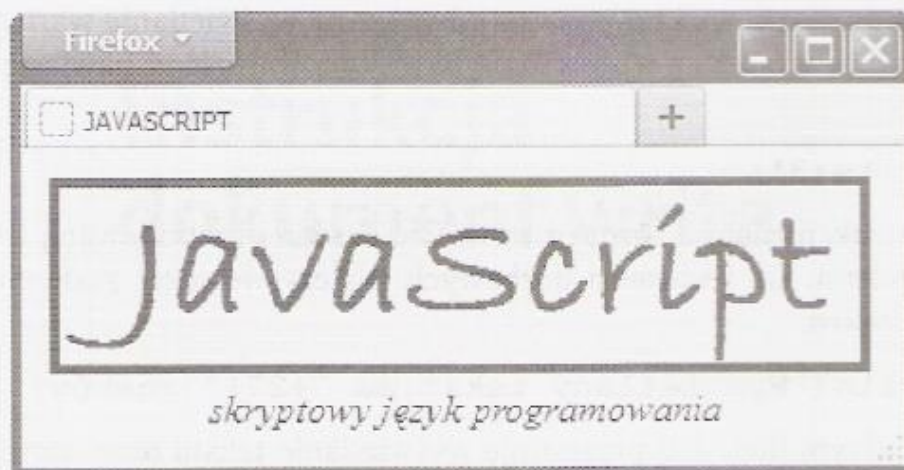
Wprowadzenie kilku instrukcji **document.write** nie spowoduje wyświetlenia ich zawartości w nowej linii. Ponieważ wyświetlana zawartość traktowana jest jak kod HTML, w list. 2.2 wprowadzono dodatkowo znacznik **<br>**, który odpowiada za przejście do nowej linii.

Wykorzystanie zagnieżdżonych znaczników HTML pozwala w łatwy sposób sformatować informacje wyświetlane za pomocą instrukcji **dokument.write**, a nawet wstawić obrazek (list. 2.3).

Listing 2.3

```
<script type="text/javascript">
document.write("<img src=\"js.png\"<br>");
document.write("<center><i>skryptowy język programowania</center></i>");
</script>
```





Rys. 2.3. Wynik działania skryptu z list. 2.3

Podczas tworzenia skryptów zdarza się, że chcemy ukryć fragment programu lub dodać opis. W tym celu należy wprowadzić **komentarz**. JavaScript daje możliwość zastosowania dwóch rodzajów komentarzy. Pierwszy jest jednowierszowy (liniowy). Rozpoczyna się od znaków `//` i działa do końca danej linii skryptu.

#### `// komentarz wierszowy`

☛ Komentarz wielowierszowy rozpoczyna się od znaków `/*` i kończy się `*/`. W takim komentarzu można umieścić komentarz liniowy, ale zabronione jest zagnieżdżanie w nim komentarzy wielowierszowych.

```
/*  
komentarz  
wielowierszowy  
//z dodatkowym komentarzem liniowym  
*/
```

## SPRAWDŹ SWOJĄ WIEDZĘ

1. Omów instrukcję `dokument.write`.
2. Jak można sformatować tekst wyświetlany przez skrypt?
3. Jak wprowadzić komentarz w skrypcie?
4. W jakim celu stosuje się komentarze w językach programowania?

# Okno dialogowe

## Zagadnienia

- Co to jest okno dialogowe?
- Jakie są rodzaje okien dialogowych?
- Jak wyświetlać okno informacyjne?
- Jak wyświetlać okno decyzyjne?
- Jak wyświetlać okno tekstowe?

**Okno dialogowe** jest narzędziem pozwalającym na nawiązanie interakcji z użytkownikiem. JavaScript umożliwia wykorzystanie trzech rodzajów okien dialogowych: informacyjnego, decyzyjnego i tekstowego.

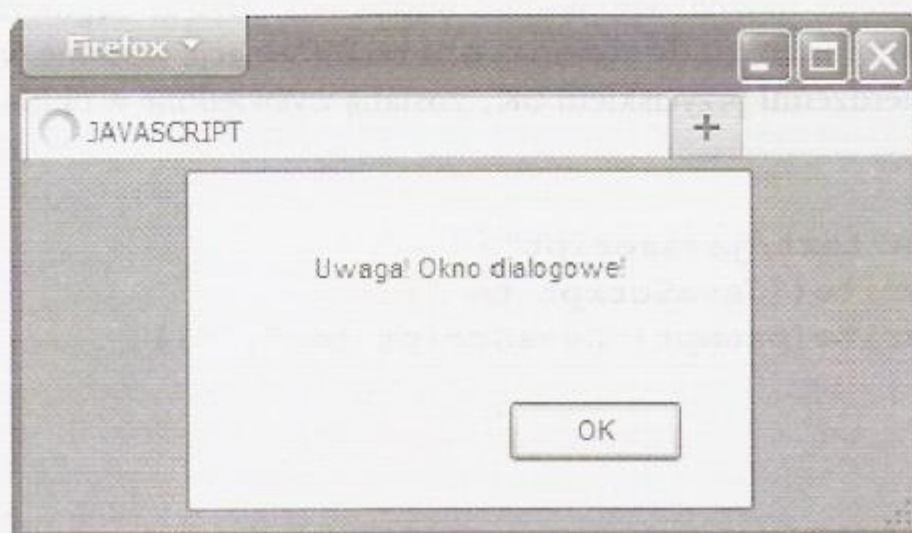
Zadaniem **okna informacyjnego** jest przekazanie określonej informacji. Nie ma ono wpływu na dalsze działanie skryptu. Jego budowa jest wyjątkowo prosta. Wyświetla ono tekst określony jako argument metody **alert** i ma jeden przycisk **OK**, powodujący zamknięcie okna.

```
alert("treść komunikatu");
```

Skrypt z list. 3.1 prezentuje okno dialogowe wyświetlające tekst "Uwaga! Okno dialogowe!". Wygląd okna może nieco różnić się w zależności od przeglądarki internetowej.

### Listing 3.1

```
<script type="text/javascript">  
alert("Uwaga! Okno dialogowe!");  
</script>
```



**Rys. 3.1.** Wynik działania skryptu z list. 3.1 – okno informacyjne



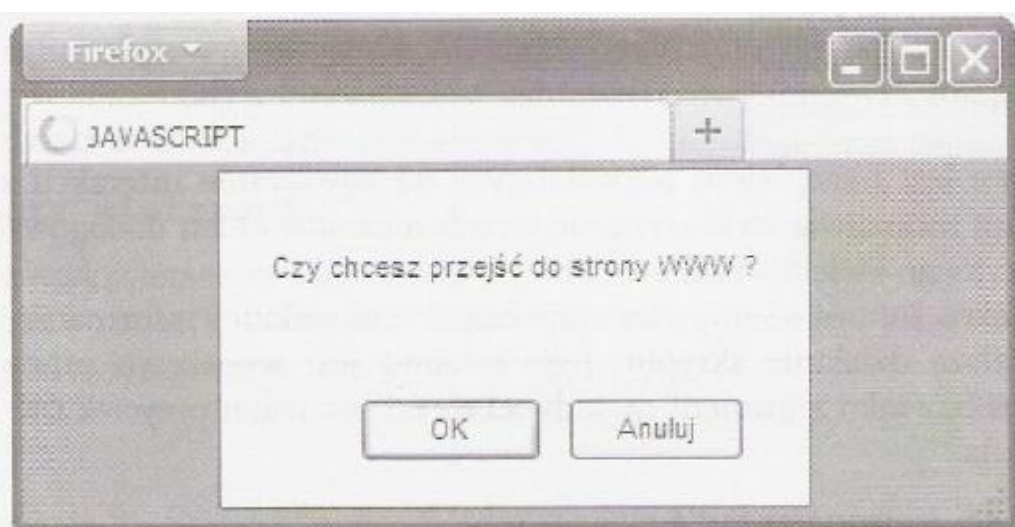
**Okno decyzyjne** odpowiada za wyświetlenie treści komunikatu stanowiącego argument metody `confirm`. Udostępnia dwa przyciski **OK** oraz **Anuluj**, które po wciśnięciu zwracają wartość logiczną `true` lub `false`.

```
confirm("treść komunikatu");
```

Skrypt z list. 3.2 prezentuje okno decyzyjne wyświetlające tekst "Czy chcesz przejść do strony WWW?". Ponieważ skrypt nie ma żadnej funkcji podpiętej do okna decyzyjnego, wciśnięcie dowolnego klawisza nie wywoła żadnej reakcji.

Listing 3.2

```
<script type="text/javascript">
  confirm("Czy chcesz przejść do strony WWW ?");
</script>
```



Rys. 3.2. Wynik działania skryptu z list. 3.2 – okno decyzyjne

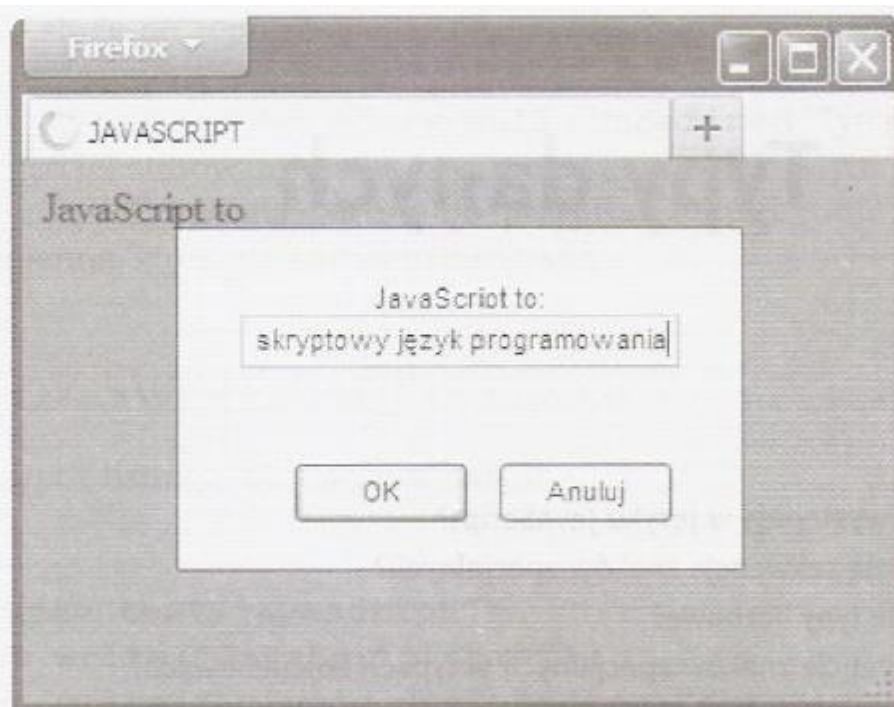
**Okno tekstowe** wyświetla treść komunikatu stanowiącego argument metody `prompt` oraz pole umożliwiające wprowadzenie danych przez użytkownika. W trakcie wywoływania okna tekstowego w polu może pojawić się tekst domyślny.

```
prompt("treść komunikatu", "tekst domyślny");
```

Skrypt list. 3.3 prezentuje okno tekstowe wyświetlające treść "JavaScript to:" oraz pole, do którego można wprowadzić odpowiedź. Dzięki zagnieżdżeniu metody `prompt` wewnątrz instrukcji `document.write` informacje wpisane w polu okna tekstowego, po zatwierdzeniu przyciskiem **OK**, zostaną wyświetlone w oknie przeglądarki.

Listing 3.3.

```
<script type="text/javascript">
  document.write("JavaScript to ");
  document.write(prompt("JavaScript to:", ""));
</script>
```



Rys. 3.3. Wynik działania skryptu z list. 3.3 – okno tekstowe

## SPRAWDŹ SWOJĄ WIEDZĘ

1. Scharakteryzuj różne okna dialogowe.
2. Jak wyświetlić okno dialogowe na stronie?
3. Podaj praktyczne przykłady zastosowania okien dialogowych.



# Typy danych

## Zagadnienia

- Jakie typy danych występują w języku JavaScript?
- Jakie znaczenie mają sekwencje znaków specjalnych?
- Jak stosować różne typy liczbowe?
- Jak stosować sekwencję znaków specjalnych w typach łańcuchowych?

Język JavaScript udostępnia kilka typów danych. Typ danych to zbiór wartości, jakie mogą przyjmować dane. Należą do nich: typ liczbowy, łańcuchowy, logiczny, obiektowy oraz typy specjalne.

**Typ liczbowy** reprezentuje różnego rodzaju liczby. W porównaniu z innymi językami programowania nie uwzględnia on podziału na liczby całkowite i zmiennoprzecinkowe. Umożliwia wprowadzanie liczb w postaci dziesiętnej (np. 12 lub 14), ósemkowej (np. 012) lub szesnastkowej (np. 0xBD). Dozwolona jest również notacja wykładnicza w postaci X.YeZ, gdzie X stanowi część całkowitą, Y – część dziesiętną, a Z jest wykładnikiem potęgi liczby 10 (np. 0.1e2).

**Typ łańcuchowy** to dowolne ciągi znaków. Należy umieścić je w cudzysłowie lub pomiędzy znakami apostrofów. Mogą dodatkowo zawierać sekwencje znaków specjalnych (tab. 4.1).

Tabela 4.1. Sekwencje znaków specjalnych

Sekwencja	Znaczenie
\b	backspace
\n	nowy wiersz
\r	powrót karetki
\f	nowa strona
\t	tabulacja pozioma
\"	cudzysłów
'	apostrof
\\	lewy ukośnik

**Typ logiczny** może przyjmować jedną z dwóch dostępnych wartości: **true** (prawda) oraz **false** (fałsz). Stosowany jest głównie przy budowaniu wyrażeń logicznych lub do porównywania danych.

**Typ obiektowy** służy do reprezentacji obiektów. Najczęściej wykorzystuje się obiekty wbudowane oraz udostępniane przez przeglądarkę.

**Typy specjalne** dzielą się na dwa rodzaje: **null** i **undefined**. Typ **null** określa wartość pustą. Najczęściej stosowany jest w programowaniu obiektowym. Typ **undefined** określa wartość niezdefiniowaną. Można go przypisać bezpośrednio do zmiennej lub przyjmuje go zmienna, która nie została zainicjowana.

## SPRAWDŹ SWOJE UMIEJĘTNOŚCI

1. Przetestuj skrypt z listingu 4.1. Co zauważyłeś?

Listing 4.1

```
<script type="text/javascript">
  document.write("JavaScript<br>");
  document.write("\Źycie jest piękne\"<br>");
  document.write(15+\"<br>");
  document.write(100.4+\"<br>");
  document.write(-26+\"<br>");
  document.write(0.1e2+\"<br>");
  document.write(0xAA+\"<br>");
  document.write(-0xCD);
</script>
```

## SPRAWDŹ SWOJĄ WIEDZĘ

1. Scharakteryzuj typy danych dostępne w języku JavaScript.
2. Jakie znasz sekwencje znaków specjalnych?