

Witold Wrotek

# JavaScript i jQuery

131 praktycznych skryptów



## JavaScript i jQuery — sprawdź je w praktyce!

- Pierwsze kroki, czyli JavaScript i jQuery dla początkujących
- JavaScript i jego możliwości, czyli skrypty do samodzielnej modyfikacji
- jQuery, czyli genialny framework pełen skryptowych gotowców

Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Michał Mrowiec

Projekt okładki: Studio Gravite / Olsztyn  
Obarek, Pokoński, Pazdrijowski, Zaprucki

Fotografia na okładce została wykorzystana za zgodą Shutterstock.com

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres  
<http://helion.pl/user/opinie/jjq131>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Kody źródłowe wybranych przykładów dostępne są pod adresem:  
<ftp://ftp.helion.pl/przyklady/jjq131.zip>

ISBN: 978-83-246-9997-1

Copyright © Helion 2015

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>Wstęp .....</b>	<b>7</b>
<b>Rozdział 1. Podstawy .....</b>	<b>17</b>
Gdy przeglądarka nie obsługuje JavaScript .....	17
Aby autor nie zapomniał, co miał na myśli .....	20
<b>Rozdział 2. Pisanie na stronie i działania na zmiennych .....</b>	<b>23</b>
Jak umieścić tekst na stronie? .....	23
Jak manipulować tekstem wyświetlanym na stronie? .....	27
Operatory .....	30
Operatory arytmetyczne .....	30
Operatory przypisania .....	35
Operatory porównania .....	41
Operatory logiczne .....	44
<b>Rozdział 3. Funkcje i obiekty .....</b>	<b>47</b>
Jak zdefiniować funkcję? .....	47
Funkcja zwracająca wartość .....	50
Zasięg zmiennych .....	51
Funkcje predefiniowane JavaScript .....	57
Obiekty .....	61
<b>Rozdział 4. Instrukcje warunkowe i pętle .....</b>	<b>71</b>
Instrukcja if .....	71
Instrukcja if... else .....	76
Instrukcja switch .....	78
Operator warunkowy .....	80
Pętla while .....	81
Pętla do-while .....	86
Pętla for .....	90
<b>Rozdział 5. Tablica .....</b>	<b>93</b>
<b>Rozdział 6. Obliczenia .....</b>	<b>111</b>
Obiekt Boolean .....	111
Obiekt Date .....	115
Metody .....	117
Obiekt Math .....	120

<b>Rozdział 7. Liczby i łańcuchy .....</b>	<b>123</b>
Obiekt Number .....	123
<b>Rozdział 8. Teksty .....</b>	<b>139</b>
Właściwości i metody .....	142
<b>Rozdział 9. Zdarzenia .....</b>	<b>153</b>
<b>Rozdział 10. JQuery .....</b>	<b>169</b>
Składnia .....	169
Dobra praktyka .....	170
Selektory jQuery .....	170
Akapit .....	170
Identyfikator .....	172
Gwiazdka .....	175
Element bieżący .....	177
Dwa selektory .....	179
Pierwsza pozycja listy .....	180
Selektor href .....	184
Nagłówki .....	186
Ciąg znaków .....	187
Hasło .....	188
Przycisk radiowy .....	190
Pole wyboru .....	191
Przycisk wysyłania .....	192
Obrazek .....	194
Plik .....	195
Elementy możliwe do zmiany .....	197
Elementy zablokowane .....	199
Pozycja wstępnie wybrana na liście rozwijanej .....	200
Wstępnie zaznaczone pole wyboru .....	201
<b>Rozdział 11. Efekty .....</b>	<b>205</b>
Zdarzenia .....	205
Składnia dla metod zdarzeń jQuery .....	205
Załadowanie dokumentu .....	206
Kliknięcie .....	207
Podwójne kliknięcie .....	209
Naprowadzenie wskaźnika myszy na element HTML .....	211
Usunięcie wskaźnika myszy z elementu HTML .....	213
Wciśnięcie klawisza myszy .....	214
Zwolnienie klawisza myszy .....	216
Naprowadzenie i zwolnienie wskaźnika myszy .....	218
Umieszczenie wskaźnika myszy w polu .....	220
Reakcja na wiele zdarzeń .....	222
Zmiana tekstu .....	224
Klonowanie akapitów .....	226
Identyfikacja elementu docelowego zdarzenia .....	229
Dane przekazywane do metody obsługi zdarzeń .....	230
<b>Rozdział 12. Manipulacja znacznikami HTML .....</b>	<b>233</b>
Tekst .....	233
HTML .....	235
Wartości pola formularza .....	237
Odczytywanie atrybutów .....	239

Zmiana atrybutów .....	240
Dodawanie nowego elementu HTML za istniejącym elementem .....	242
Dodawanie nowego elementu HTML przed istniejącym elementem .....	244
Dodawanie nowego elementu HTML we wskazanym miejscu .....	246
Usuwanie elementów .....	250
Usuwanie elementów podrzędnych .....	252
Filtrowanie elementów do ukrycia .....	254
Zmiana tła wyróżnionych elementów .....	257
Dodanie klasy do arkusza CSS .....	259
Usunięcie klasy z arkusza CSS .....	263
Przełączenie klasy z arkusza CSS .....	265
<b>Rozdział 13. Witalizacja strony .....</b>	<b>269</b>
Zmiana wymiarów obiektu .....	269
Zanikanie i pojawianie się elementów strony .....	273
Przesuwanie elementów strony .....	279
Animacje .....	282
Zakończenie operacji .....	297
Łańcuchy poleceń .....	300
Zmiana parametrów w pętli .....	302
Przesuwanie elementów po arkuszu .....	305
Przekierowywanie naabrany adres URL po upływie określonego czasu .....	308
Wcięcie tekstu .....	310
Odstęp między liniami .....	312
Rozmiar czcionki .....	313
Marginesy .....	317
<b>Skorowidz .....</b>	<b>321</b>



## Rozdział 9.

# Zdarzenia

Zdarzenia dzielą się na zdarzenia *HTML* i *HTML DOM*.

*Zdarzenia HTML* dotyczą elementów HTML. Zdarzeniami są np.:

- ◆ zakończenie ładowania strony HTML,
- ◆ zmiana zawartości pola do wpisywania danych,
- ◆ kliknięcie przycisku.

Zajście zdarzenia może być sygnałem do rozpoczęcia innej czynności, np. uruchomienia funkcji. Funkcja nie zostanie wykonana przed wystąpieniem zdarzenia; np. dopóki użytkownik nie kliknie przycisku, dane nie zostaną wczytane.

*HTML DOM* (ang. *Document Object Model* — model obiektów dokumentu) to standard określający sposoby zmiany zawartości, właściwości i parametrów poszczególnych elementów strony HTML.

Jakie jest zastosowanie obsługi zdarzeń? Oto kilka przykładów:

- ◆ Uruchomienie programów, które powinny być wykonywane za każdym razem, gdy strona się załaduje.
- ◆ Uruchomienie programów, które powinny być wykonywane za każdym razem, gdy strona jest zamykana.
- ◆ Działania, które powinny być wykonywane, gdy użytkownik kliknie przycisk.

W tabeli 9.1 zebrano zdarzenia HTML DOM związane z myszą.

### **PRZYKŁAD 50.**

Napisz program, który wyświetla na ekranie tekst *Kliknij tekst!*, zaś po wykonaniu polecenia wyświetla podziękowanie.

**Tabela 9.1.** Zdarzenia HTML DOM związane z myszą

Właściwość	Opis
onclick	Zdarzenie występuje, gdy użytkownik kliknie element.
ondblclick	Zdarzenie występuje, gdy użytkownik dwukrotnie kliknie element.
onmousedown	Zdarzenie występuje, gdy użytkownik naciśnie przycisk myszy, gdy kursor znajduje się nad elementem.
onmouseenter	Zdarzenie występuje, gdy kursor jest przesuwany na elemencie.
onmouseleave	Zdarzenie występuje, gdy użytkownik przesunie kursor z elementu.
onmousemove	Zdarzenie występuje, gdy kursor, będąc w ruchu, znajduje się nad elementem.
onmouseover	Zdarzenie występuje, gdy wskaźnik jest przesuwany na elemencie lub na jednym z jego dzieci.
onmouseout	Zdarzenie występuje, gdy użytkownik przesuwa wskaźnik myszy na element lub z jednego z jego dzieci.
onmouseup	Zdarzenie występuje, gdy użytkownik zwolni przycisk myszy na elemencie.

Program, który spełnia postawione założenia, wygląda następująco:

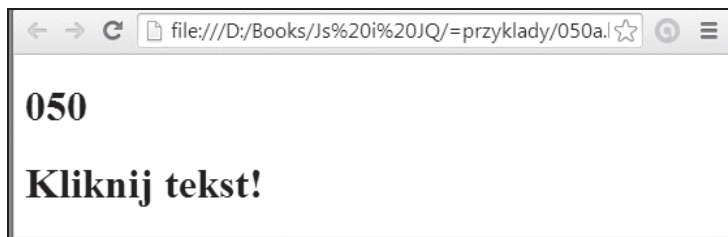
```

<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>050</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head><body><script>
document.write("<h1>050</h1>");
</script>
<h1 onclick="this.innerHTML='Dziękuję!'">Kliknij tekst!</h1>
</body></html>

```

Polecenie `document.write("<h1>050</h1>");` zadeklarowane jest jako skrypt. Powoduje ono wyświetlenie numeru przykładowo (rysunek 9.1).

**Rysunek 9.1.**  
Okno z poleceniem



Polecenie:

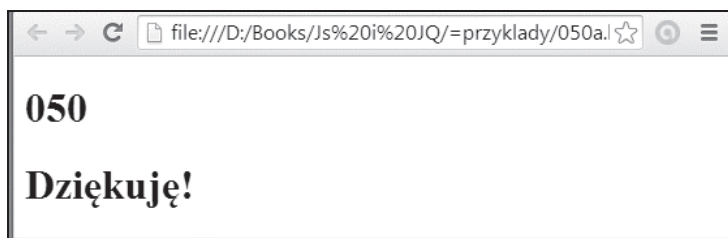
```
<h1 onclick="this.innerHTML='Dziękuję!'">Kliknij tekst!</h1>
```

sprawia, iż wyświetlane jest polecenie *Kliknij tekst!*. Jest ono jednocześnie „czułe na kliknięcie”. Kliknięcie go powoduje wyświetlenie okna z podziękowaniem (rysunek 9.2).



**Rysunek 9.2.**

Okno  
z podziękowaniem

**PRZYKŁAD 51.**

Napisz program, który wyświetla na ekranie tekst *Przesuń kursor nad napis!*, zaś po wykonaniu polecenia wyświetlone zostaje podziękowanie.

Program, który spełnia postawione założenia, wygląda następująco:

```

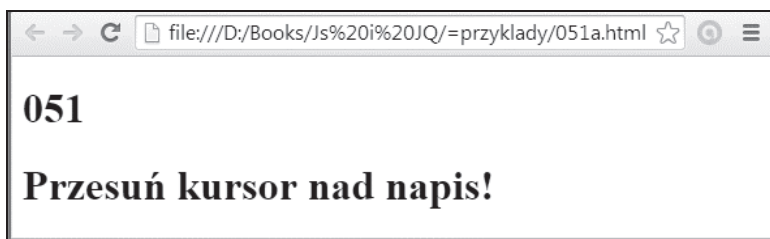
<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>051</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head><body><script>
document.write("<h1>051</h1>");
</script>
<h1 onmouseenter="this.innerHTML='Dziękuję!'">Przesuń kursor nad napis!</h1>
</body></html>

```

Polecenie `document.write("<h1>051</h1>");` zadeklarowane jest jako skrypt. Powoduje ono wyświetlenie numeru przykładu (rysunek 9.3).

**Rysunek 9.3.**

Okno z poleceniem



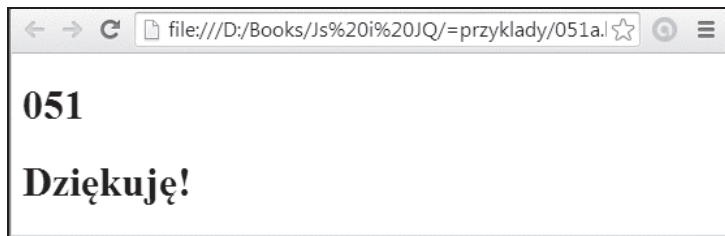
Polecenia:

```
<h1 onmouseenter="this.innerHTML='Dziękuję!'">Przesuń kursor na napis!</h1>
```

sprawiają, iż wyświetlane jest polecenie *Przesuń kursor na napis!*. Jest ono jednocześnie „czułe na położenie nad nim kursora”. Przesunięcie kursora nad napis powoduje, że polecenie zostaje zastąpione podziękowaniem (rysunek 9.4).

**Rysunek 9.4.**

*Okno  
z podziękowaniem.  
Odsunięcie kursora  
na puste pole nie  
powoduje ponownego  
wyświetlenia polecenia*

**PRZYKŁAD 52.**

Napisz program, który wyświetla na ekranie tekst *Przesuń kursor nad prostokąt!*. Umieszczony jest on w szarym prostokącie o wymiarach 200 (szerokość) na 20 pikseli (wysokość). Po wykonaniu polecenia wyświetlony zostaje inny tekst. Przesunięcie kursora z wnętrza prostokąta powoduje ponowne wyświetlenie prośby.

Program, który spełnia postawione założenia, wygląda następująco:

```

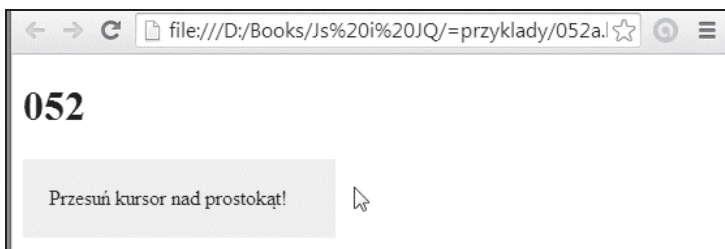
<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>052</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head><body><script>
document.write("<h1>052</h1>");
function mysz_nad(obj) {
    obj.innerHTML = "Widzisz, jakie czary!"
}
function mysz_na_zewn(obj) {
    obj.innerHTML = "Przesuń kursor nad prostokąt!"
}
</script>
<div onmouseover="mysz_nad(this)" onmouseout="mysz_na_zewn(this)"
style="background-color:#EEEEEE;width:200px;height:20px;padding:20px;">
Przesuń kursor nad prostokąt!</div>
</body></html>

```

Polecenie `document.write("<h1>052</h1>");` powoduje wyświetlenie numeru przykładu (rysunek 9.5).

**Rysunek 9.5.**

*Wskaźnik myszy  
znajduje się na prawo  
od prostokąta*

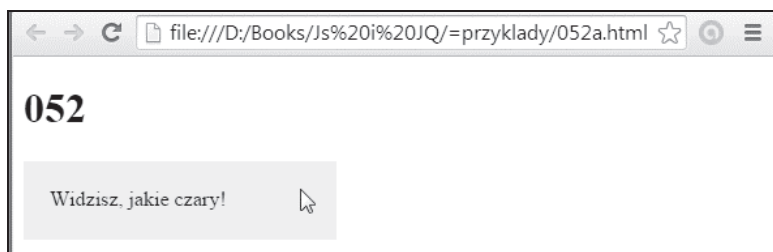


Polecenia:

```
<div onmouseover="mysz_nad(this)" onmouseout="mysz_na_zewn(this)"
  style="background-color:#EEEEEE;width:200px;height:20px;padding:20px;">
  Przesuń kursor nad prostokąt!</div>
```

sprawiają, iż wyświetlane jest polecenie *Przesuń kursor nad prostokąt!*. Jest ono jednocześnie „czułe na położenie nad nim kursora”. Przesunięcie kursora nad prostokąt powoduje, że polecenie zostaje zastąpione podziękowaniem (rysunek 9.6).

**Rysunek 9.6.**  
*Wskaźnik  
nad prostokątem*



Przesunięcie wskaźnika myszy na zewnątrz prostokąta ponownie powoduje wyświetlenie okna z prośbą (rysunek 9.5).

Gdy wskaźnik myszy znajduje się na zewnątrz prostokąta, wywoływana jest funkcja:

```
function mysz_na_zewn(obj) {
  obj.innerHTML = "Przesuń kursor nad prostokąt!"
}
```

Gdy wskaźnik myszy znajduje się wewnątrz prostokąta, wywoływana jest funkcja:

```
function mysz_nad(obj) {
  obj.innerHTML = "Widzisz, jakie czary!"
}
```

W tabeli 9.2 zebrano zdarzenia HTML DOM związane z klawiaturą.

**Tabela 9.2.** *Zdarzenia HTML DOM związane z klawiaturą*

Atrybut	Opis
onkeydown	Zdarzenie ma miejsce, gdy użytkownik trzyma wciśnięty klawisz.
onkeypress	Zdarzenie wystąpiło, gdy użytkownik nacisnął klawisz.
onkeyup	Zdarzenie wystąpiło, gdy użytkownik zwolnił klawisz.

### PRZYKŁAD 53.

Napisz program, który wyświetla na ekranie okno z polem do wpisania znaku. Wpisanie kolejnego znaku powoduje zamianę poprzednio wpisanej litery na wielką.

Program, który spełnia postawione założenia, wygląda następująco:

```
<!DOCTYPE HTML PUBLIC "
  -//W3C//DTD HTML 4.01 Transitional//EN"
```

```

"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>053</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head><body><script>
document.write("<h1>053</h1>");
function Funkcja()
{
var x=document.getElementById("znak");
x.value=x.value.toUpperCase();
}
</script>
<p>Wpisanie kolejnego znaku powoduje zamianę poprzednio wpisanej litery na
wielką.</p>
Wpisz numer rejestracyjny swojego samochodu: <input type="text" id="znak"
onkeypress="Funkcja()">
</body></html>

```

Program składa się ze skryptu, w którym zdefiniowana jest funkcja, oraz wywołania funkcji.

Definicja funkcji ma postać:

```

function Funkcja()
{
var x=document.getElementById("znak");
x.value=x.value.toUpperCase();
}

```

Znaki wpisane z klawiatury w polu formularza przypisywane są zmiennej *x*. Za konwersję wielkości liter odpowiedzialne jest polecenie `toUpperCase`.

Komentarz i miejsce, w którym pojawi się pole do wpisywania znaków, ustalone są przez miejsce wpisania kodu:

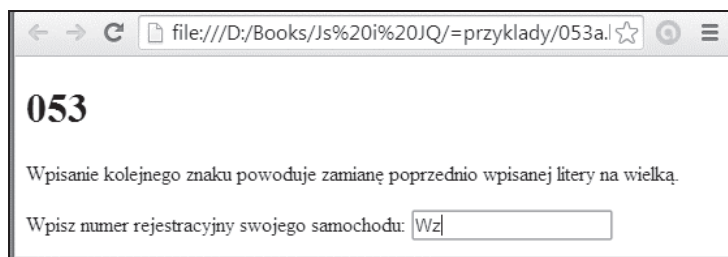
```

Wpisz numer rejestracyjny swojego samochodu: <input type="text" id="znak"
onkeypress="Funkcja()">

```

Na rysunku 9.7 pokazano rezultat wykonania programu.

**Rysunek 9.7.**  
Pierwszy znak  
został zamieniony  
na wielką literę



W tabeli 9.3 zebrano zdarzenia HTML DOM związane z obiektami (ramkami).

**Tabela 9.3.** Zdarzenia HTML DOM związane z obiektami (ramkami)

Atrybut	Opis
onabort	Ładowanie obrazu przerwano, zanim został załadowany w całości.
onerror	Obraz nie został załadowany prawidłowo (dla <object>, <body> i <frameset>).
onload	Dokument, frameset lub <object> został załadowany.
onresize	Zmieniono rozmiary widoku dokumentu.
onscroll	Przewinięto dokument.
onunload	Strona nie została załadowana (ma zastosowanie zarówno dla <body>, jak i <frameset>).

## PRZYKŁAD 54.

Napisz program, który powoduje wyświetlenie wyskakującego okienka z komunikatem powitalnym. Użytkownik powinien nacisnąć klawisz *OK*, aby potwierdzić zapoznanie się z treścią komunikatu.

Program, który spełnia postawione założenia, wygląda następująco:

```
<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>054</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head><body onload="Funkcja()"><script>
document.write("<h1>054</h1>");
function Funkcja()
{
alert("Witaj!");
}
</script>
</body></html>
```

W deklaracji sekcji body znajduje się polecenie `onload="Funkcja()`. Powoduje ono, że po załadowaniu dokumentu wykonywana jest funkcja o nazwie `Funkcja`.

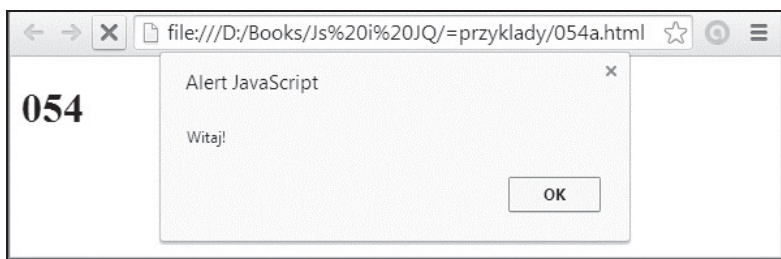
Co robi funkcja `Funkcja`? Jak można zorientować się z kodu:

```
function Funkcja()
{
alert("Witaj!");
}
```

powoduje ona wyświetlenie okna z komunikatem *Witaj!*. Okno można zamknąć, klikając przycisk *OK* lub krzyżyk widoczny w prawym górnym rogu okna.

Na rysunku 9.8 pokazano fragment ekranu po załadowaniu pliku *054a.html*.

**Rysunek 9.8.**  
Okno z komunikatem  
wyświetlanym  
po załadowaniu  
dokumentu HTML



W tabeli 9.4 zebrano zdarzenia HTML DOM związane z formularzami.

**Tabela 9.4.** Zdarzenia HTML DOM związane z formularzami

Atrybut	Opis
onblur	Zdarzenie występuje, gdy element formularza traci fokus.
onchange	Zdarzenie występuje, gdy zawartość elementu formularza, selekcja lub sprawdzony status uległy zmianie (ma zastosowanie do <input>, <select> oraz <textarea>).
onfocus	Zdarzenie występuje, gdy element uzyskuje fokus (ma zastosowanie dla <label>, <input>, <select>, <textarea> oraz <button>).
onreset	Zdarzenie występuje, gdy formularz jest resetowany.
onselect	Zdarzenie nastąpi, gdy użytkownik wybierze tekst (ma zastosowanie dla <input> and <textarea>).
onsubmit	Zdarzenie nastąpi, gdy formularz zostanie wysłany.

Człowiek posługuje się literami i liczbami. Komputer używa tylko liczb. Napišemy program, który pojedyncze znaki alfanumeryczne będzie zamieniał na odpowiedniki Unicode.

### PRZYKŁAD 55.

Napiš program, który wyświetli okno do wpisania znaku. Konwersja na odpowiednik *Unicode* zostanie wykonana, gdy kursor znajdzie się na zewnątrz pola.

Program, który spełnia postawione założenia, wygląda następująco:

```

<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>055</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head>
<body><script>
document.write("<h1>055</h1><br>1. Wpisz znak w polu.<br>2. Kliknij na zewnątrz
↳ pola.<br>3. Odczytaj odpowiednik Unicode.<br>");
function konwersja() {
    var kod = document.getElementById("znak");
    kod.value = kod.value.charCodeAt();

```

```

}
</script>
Wpisz znak: <input type="text" id="znak" onblur="konwersja()">
</body></html>

```

**Polecenia:**

```

document.write("<h1>055</h1><br>1. Wpisz znak w polu.<br>2. Kliknij na zewnątrz
↳ pola.<br>3. Odczytaj odpowiednik Unicode.<br>");

```

powodują wyświetlenie na ekranie tytułu strony oraz ujętej w trzech punktach informacji o działaniu programu.

**Przedostatnia linijka programu:**

```

Wpisz znak: <input type="text" id="znak" onblur="konwersja()">

```

powoduje wyświetlenie pola. Polecenie `onblur` powoduje, że gdy kursor znajdzie się na zewnątrz pola, wywołana zostaje funkcja `konwersja`. Ma ona następującą postać:

```

function konwersja() {
    var kod = document.getElementById("znak");
    kod.value = kod.value.charCodeAt();
}

```

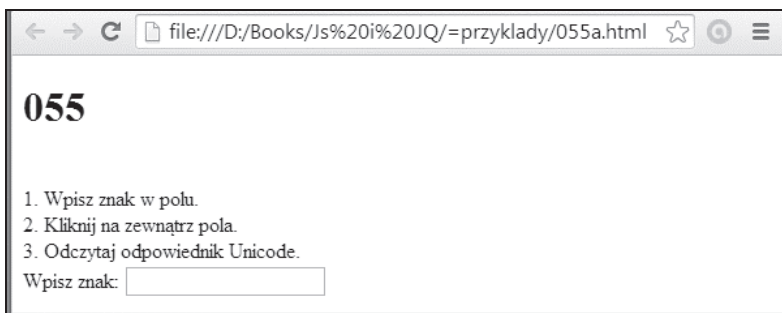
Jej zadaniem jest zamiana zmiennej `znak` na odpowiednik *Unicode*.

Po wykonaniu konwersji wpisany znak zostaje zastąpiony przez jego odpowiednik.

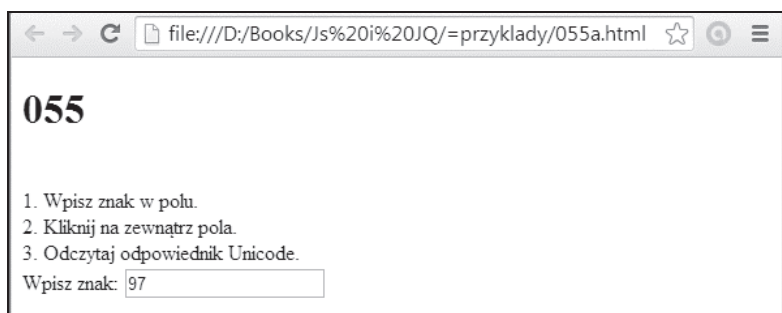
Na rysunku 9.9 pokazano okno przeglądarki po wczytaniu pliku *055a.html*. Na rysunku 9.10 pokazano okno przeglądarki po wpisaniu litery `a` i kliknięciu pola obok.

**Rysunek 9.9.**

*Okno  
po załadowaniu  
pliku*

**Rysunek 9.10.**

*Literze `a` odpowiada  
kod Unicode 97*



W tabeli 9.5 zebrano zdarzenia HTML DOM związane z właściwościami.

**Tabela 9.5.** Zdarzenia HTML DOM związane z właściwościami

Właściwość	Opis
bubbles	Zwraca informację, czy zdarzenie może przechodzić fazę propagacji (tzw. bąbelkowanie).
cancelable	Zwraca informację, czy zdarzenie może mieć zablokowane działanie w sposób domyślny.
currentTarget	Zwraca element, którego słuchacz zdarzeń wywołał zdarzenie.
eventPhase	Zwraca informację, która faza przepływu zdarzeń jest obecnie oceniana.
target	Zwraca element, który wywołał zdarzenie.
timeStamp	Zwraca czas mierzony w milisekundach w stosunku do momentu, w którym zdarzenie zostało utworzone.
type	Zwraca nazwę zdarzenia.



Uwaga

Bąbelkowanie to odpowiadanie na zdarzenia w trakcie ich propagacji od obiektu docelowego do obiektów zewnętrznych.

## PRZYKŁAD 56.

Napisz program, który po kliknięciu wyświetli w wyskakującym oknie liczony w milisekundach czas, który upłynął od zdefiniowania zdarzenia.

Program, który spełnia postawione założenia, wygląda następująco:

```
<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>056</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head>
<body onmousedown="getEventType(event)">
<script>
document.write("<h1>056</h1>");
function stoper(zdarzenie)
{
alert(zdarzenie.timeStamp);
}
</script>
<p onclick="stoper(event)">Kliknij akapit, a wyświetlone zostanie okno z mierzonym
↳w milisekundach czasem, który upłynął od utworzenia zdarzenia.</p>
</body></html>
```

Wyświetlanie informacji inicjowane jest przez wciśnięcie klawisza myszy. Zdarzenie zostało powiązane z pomiarem czasu w linii:

```
<body onmousedown="getEventType(event)">
```



Akapit reaguje na kliknięcie, ponieważ w jego definicji umieszczony jest kod:

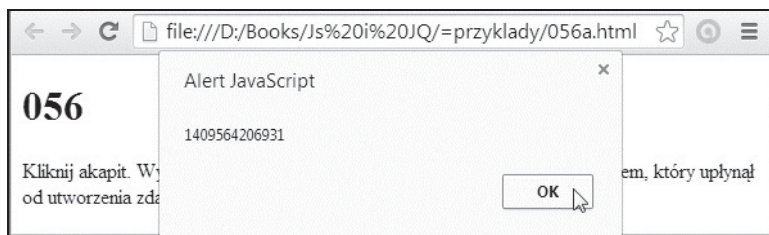
```
<p onClick="stoper(event)">
```

Funkcja, która powoduje wyświetlenie okna z wynikiem, ma postać:

```
function stoper(zdarzenie)
{
  alert(zdarzenie.timeStamp);
}
```

Na rysunku 9.11 pokazano okno przeglądarki z wynikiem pomiaru.

**Rysunek 9.11.**  
Wynik wyświetlony  
po kliknięciu akapitu



W tabeli 9.6 zebrano zdarzenia `MouseEvent/KeyboardEvent` HTML DOM związane z właściwościami.

**Tabela 9.6.** Zdarzenia `MouseEvent/KeyboardEvent` HTML DOM związane z właściwościami

Właściwość	Opis
<code>altKey</code>	Zwraca informację, czy klawisz <i>Alt</i> został wciśnięty, gdy zdarzenie zostało zarejestrowane.
<code>button</code>	Zwraca informację, który klawisz myszy został kliknięty, gdy zdarzenie zostało zarejestrowane.
<code>clientX</code>	Zwraca współrzędną poziomą wskaźnika myszy w stosunku do bieżącego okna, gdy zdarzenie zostało zarejestrowane.
<code>clientY</code>	Zwraca współrzędną pionową wskaźnika myszy w stosunku do bieżącego okna, gdy zdarzenie zostało zarejestrowane.
<code>ctrlKey</code>	Zwraca informację, czy klawisz <i>Ctrl</i> został wciśnięty, gdy zdarzenie zostało zarejestrowane.
<code>keyIdentifier</code>	Zwraca identyfikator klawisza.
<code>keyLocation</code>	Zwraca położenie klawisza w urządzeniu.
<code>metaKey</code>	Zwraca informację, czy klawisz <i>meta</i> został wciśnięty, gdy zdarzenie zostało zarejestrowane.
<code>relatedTarget</code>	Zwraca element związany z elementem, który spowodował zdarzenie.
<code>screenX</code>	Zwraca współrzędną poziomą wskaźnika myszy w stosunku do ekranu, gdy zdarzenie zostało zarejestrowane.
<code>screenY</code>	Zwraca współrzędną pionową wskaźnika myszy w stosunku do ekranu, gdy zdarzenie zostało zarejestrowane.
<code>shiftKey</code>	Zwraca informację, czy klawisz <i>Shift</i> został wciśnięty, gdy zdarzenie zostało zarejestrowane.

## PRZYKŁAD 57.

Napisz program, który po kliknięciu tekstu wyświetli współrzędne klikniętego miejsca w stosunku do bieżącego okna.

Program, który spełnia postawione założenia, wygląda następująco:

```
<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>057</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head>
<body >
<script>
document.write("<h1>057</h1>");
function wspolrzedne_kursora(event)
{
    var x=event.clientX
    var y=event.clientY
    alert("Współrzędna X: "+ x + ", współrzędna Y: "+ y)
</script>
<p onmousedown="wspolrzedne_kursora(event)">Kliknij akapit, a wyświetlone zostanie
↳okno ze współrzędnymi X i Y kursora. Kliknij akapit, a wyświetlone zostanie okno
↳ze współrzędnymi X i Y kursora. Kliknij akapit, a wyświetlone zostanie okno ze
↳współrzędnymi X i Y kursora. Kliknij akapit, a wyświetlone zostanie okno ze
↳współrzędnymi X i Y kursora. Kliknij akapit, a wyświetlone zostanie okno ze
↳współrzędnymi X i Y kursora. Kliknij akapit, a wyświetlone zostanie okno ze
↳współrzędnymi X i Y kursora.</p>
</body></html>
```

Funkcja, która odczytuje i wyświetla współrzędne kursora w chwili kliknięcia tekstu, ma postać:

```
function wspolrzedne_kursora(event)
{
    var x=event.clientX
    var y=event.clientY
    alert("Współrzędna X : " + x + ", Y Współrzędna : " + y)
}
```

Akapit, który jest „wrażliwy na kliknięcie”, rozpoczyna się od fragmentu:

```
<p onmousedown="wspolrzedne_kursora(event)">Kliknij akapit...
```

Na rysunku 9.12 pokazano okno przeglądarki z wynikiem pomiaru.

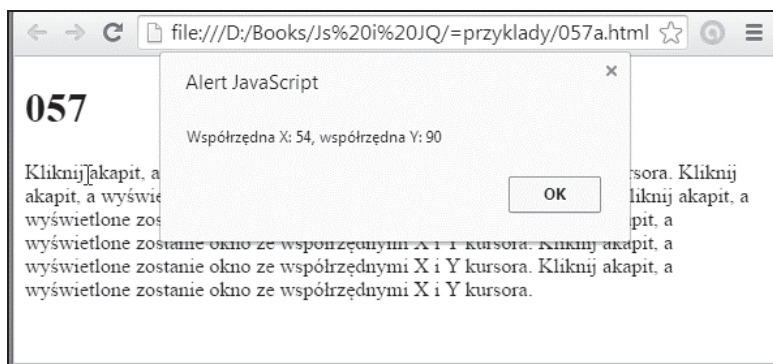


Uwaga

Obiekt Anchor reprezentuje element <a> języka HTML.

**Rysunek 9.12.**

Po kliknięciu akapitu wyświetlone zostały współrzędne kursora w stosunku do bieżącego okna



W tabeli 9.7 zebrano właściwości obiektu Anchor HTML DOM.

**Tabela 9.7.** Właściwości obiektu Anchor HTML DOM

Właściwość	Opis
charset	Nie jest obsługiwana w HTML 5. Ustawia lub zwraca wartość atrybutu charset linku.
download	Ustawia lub zwraca wartość atrybutu download linku.
hash	Ustawia lub zwraca wartość atrybutu href linku.
host	Ustawia lub zwraca nazwę hosta i portu atrybutu href.
hostname	Ustawia lub zwraca nazwę hosta atrybutu href.
href	Ustawia lub zwraca atrybut href linku.
hreflang	Ustawia lub zwraca atrybut hreflang linku.
origin	Zwraca protokół, nazwę hosta i portu atrybutu href.
id	Ustawia lub zwraca wartość atrybutu id linku.
name	Nie jest obsługiwana w HTML5. Zamiast niej należy używać id. Ustawia lub zwraca wartość atrybutu id linku.
password	Ustawia lub zwraca część password wartości atrybutu href.
pathname	Ustawia lub zwraca nazwę ścieżki dostępu wartości atrybutu href.
port	Ustawia lub zwraca nazwę portu z wartości atrybutu href.
protocol	Ustawia lub zwraca nazwę protokołu linku.
rel	Ustawia lub zwraca wartość atrybutu rel linku.
rev	Nie jest obsługiwana w HTML 5. Ustawia lub zwraca wartość atrybutu rev linku.
search	Ustawia lub zwraca część atrybutu href, która jest ciągiem kwerendy.
target	Ustawia lub zwraca wartość atrybutu target zawartą w linku.
text	Ustawia lub zwraca tekst zawarty w linku.
type	Ustawia lub zwraca wartość atrybutu type zawartą w linku.
username	Ustawia lub zwraca nazwę użytkownika zawartą w atrybucie href.

**PRZYKŁAD 58.**

Napisz program, który po kliknięciu przycisku wyświetli zapytanie zawarte w linku.

Program, który spełnia postawione założenia, wygląda następująco:

```
<!DOCTYPE HTML PUBLIC "
-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>058</title>
<meta http-equiv="CONTENT-LANGUAGE" CONTENT="PL">
<meta http-equiv="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
<meta http-equiv="Content-Script-Type" content="text/javascript">
</head>
<h1>058</h1>
<p><a id="yyy"
href="http://helion.pl/search?qa=&szukaj=wrotek&sortby=wd&serwisya1=1&wsprzed=
↳1&wprzyg=1&sent=1">Link</a></p>
<p>Kliknij przycisk, aby wyświetlić część ciągu kwerendy z linku powyżej.</p>
<button onclick="funkcja()">Kliknij</button>
<p id="xxx"></p>
<script>
function funkcja() {
    var x = document.getElementById("yyy").search;
    document.getElementById("xxx").innerHTML = x;
}
</script>
</body>
</html>
```

Link ma postać:

```
http://helion.pl/search?qa=&szukaj=wrotek&sortby=wd&serwisya1=1&wsprzed=1&wprzyg=
↳1&sent=1
```

Jest to zapytanie do bazy danych. Treść zapytania znajduje się po znaku zapytania.

Sygnałem do wydzielenia z linku treści zapytania jest kliknięcie przycisku. Przycisk i związana z nim funkcja są wyświetlane dzięki linijce:

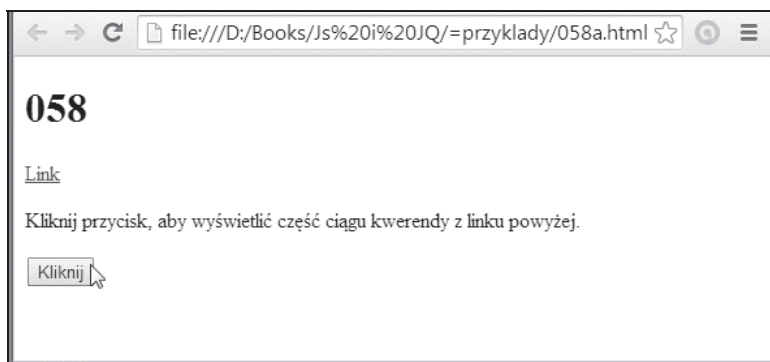
```
<button onclick="funkcja()">Kliknij</button>
```

Konieczne jest jeszcze zdefiniowanie funkcji uruchamianej kliknięciem przycisku:

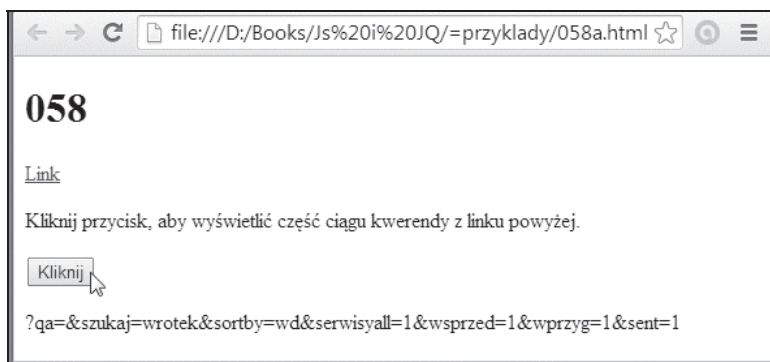
```
function funkcja() {
    var x = document.getElementById("yyy").search;
    document.getElementById("xxx").innerHTML = x;
}
```

Na rysunku 9.13 pokazano okno przeglądarki po załadowaniu pliku *058a.html*. Na rysunku 9.14 pokazano okno przeglądarki po kliknięciu przycisku.

**Rysunek 9.13.**  
*Dokument HTML  
po załadowaniu*



**Rysunek 9.14.**  
*Pod przyciskiem  
widoczna treść  
zapytania zawarta  
w linku*





# Skorowidz

-, 31  
--, 31  
' , 141  
!, 45  
!=, 41  
", 141  
\$, 169  
%, 31  
%=: 35  
&, 45  
\*, 31  
\*/, 20  
\*=, 35  
.attr, 239  
/, 31  
/\*, 20  
//, 20  
/=, 35  
\\, 141  
{, 72  
||, 45  
}, 72  
+, 31  
++, 31  
+=: 35  
<, 41  
<=: 41  
=, 35  
-=, 35  
==, 41  
===, 41, 42  
>, 41  
>=: 41  
[], 134  
0, 112  
0,1, 125  
0,2, 125  
0,3, 125  
0x, 126  
1, 112  
1em, 315

## A

abs, 121  
acos, 121  
adres  
  pliku, 72  
  strony WWW wyświetlony  
  w oknie komunikatu, 239  
after, 248  
akapitu usuwanie, 210  
alert, 57, 212  
Alt wciśnięty, 163  
altKey, 163  
Anchor, 165  
animacja, 284, 289, 303  
  w pętli, 302  
animate, 282, 284, 289, 293  
append, 242, 246  
arcus  
  cosinus, 121  
  sinus, 121  
  tangens, 121  
  tangens ilorazu, 121  
argument jest liczbą, 57  
arkusze stylów CSS, 25, 207, 259  
Array, 93  
ASCII, 86  
asin, 121  
atan, 121  
atan2, 121  
attr, 239

## B

b, 141  
backslash, 26, 141  
backspace, 141  
bąbelkowanie, 162  
before, 248  
białe spacje, 142

biblioteka, 8  
 binarna liczba, 127  
 bind, 222  
 blur, 220  
 Boolean, 111  
 bubbles, 162  
 button, 163

**C**

callback, 282  
 cancelable, 162  
 ceil, 121  
 Celsjusz, 39  
 charAt, 142  
 charCodeAt, 142  
 charset, 165  
 click, 207, 230, 241  
 clientX, 163  
 clientY, 163  
 concat, 107, 142  
 constructor, 142  
 content, 116  
 cos, 121  
 cosinus, 121  
 CSS, 259  
 Ctrl wciśnięty, 163  
 ctrlKey, 163  
 cudzysłów, 26, 140  
   podwójny, 141  
   pojedynczy, 141  
 currentTarget, 162  
 czas  
   jako ciąg znaków, 119  
   lokalny, 118  
   od momentu, w którym zdarzenie zostało utworzone, 162  
   pojawiania się, 273  
   zanikania, 273  
   życia zmiennej, 52  
 czcionka  
   kolor, 24  
 czcionka  
   pogrubiona, 24  
   rozmiar, 313  
   zmiana, 186

**D**

dane  
   predefiniowane, 197  
   z formularza wyświetlone w oknie komunikatu, 237  
 data  
   jako ciąg znaków  
     JSON, 119  
     ISO, 119  
   systemowa, 64, 115

Date, 115, 117, 118  
 dbclick, 209, 210  
 decodeURI(), 57  
   Component(), 57  
 definicja stylu, 25  
 deklaracja zmiennej, 27  
 dekodowanie, 58  
 dekrementacja, 31  
 delegate, 226  
 development version, 14  
 długości ciągu, 143  
 document.write, 19, 23  
 dodanie  
   przed elementem już istniejącym elementu nowego, 244  
   za elementem już istniejącym elementu nowego, 242  
 dodawanie, 31  
 Dojo, 9  
 dokładność, 125  
   obliczeń, 40  
 Dokument został załadowany, 159  
 DOM, 153, 233  
 domyślnie zaznaczone, 225  
 do-while, 86  
 download, 165  
 dwójkowa liczba, 127  
 dwukrotne kliknięcie, 154, 210  
 działania na zmiennych, 35  
 dzielenie, 31  
   przez 0, 37, 128  
 dzień  
   miesiąca, 118  
   tygodnia, 118  
 dziesiętna liczba, 127

**E**

E, 120  
 element  
   animowany, 284  
   bieżący, 177  
   który wywołał zdarzenie, 162  
   wstępnie wybrany, 200  
   związany z elementem, który spowodował zdarzenie, 163  
 empty, 252  
 encodeURI(), 57  
   Component(), 57  
 eval(), 57  
 event  
   .data, 231  
   Phase, 162  
 Ex, 121  
 exp, 121



**F**

f, 141  
 fade  
   In, 273  
   Out, 273  
   Toggle, 275  
 Fahrenheit, 39  
 false, 44, 112  
 fałsz, 44, 112  
 faza  
   propagacji, 162  
   przepływu zdarzeń, 162  
 floor, 121  
 focus, 160, 220  
 font-size, 315  
 for, 90  
 format  
   liczbowy, 41  
   tekstowy, 41  
 formatowanie arkuszami stylów CSS, 263  
 formularz, 160  
   dane predefiniowane, 197  
   wartość pola, 237  
 framework, 8  
 fromCharCode, 142  
 FullYear, 118  
 function, 47  
 funkcja, 47

**G**

get  
   Date, 118  
   Day, 118  
   FullYear, 118, 119  
   Hours, 118  
   Milliseconds, 118  
   Minutes, 118  
   Month, 118  
   Month(), 79  
   Seconds, 118  
   Time, 118  
   TimezoneOffset, 118, 120  
   UTCDate, 118  
   UTCDay, 118  
   UTCFullYear, 118  
   UTCHours, 118  
   UTCMilliseconds, 118  
   UTCMinutes, 118  
   UTCMonth, 118  
   UTCSeconds, 118  
   Year, 118  
 godzina, 118

**H**

hash, 165  
 head, 116  
 height, 269  
 heksadecymalna liczba, 127  
 hide, 184, 210  
 host, 165  
   name, 165  
 hover, 218  
 href, 165  
   lang, 165  
 HTML, 7, 153, 235  
   DOM, 153  
   szablon strony, 10

**I**

id, 165  
 idealnie równe, 41  
 identyczne, 42  
 identyfikator klawisza, 163  
 if, 72  
   ... else, 76  
 iloczyn logiczny, 45  
 indexOf, 142  
 infinity, 38, 59, 128  
 informacja o niewybraniu pliku, 195  
 inkrementacja, 31, 34  
 instrukcja wykonana co najmniej jeden raz, 86  
 interlinia, 312  
 is  
   Finite(), 57  
   NaN(), 57

**J**

jQuery, 169  
 JavaScript, 7  
   szablon strony, 10  
 język skryptowy, 8  
 jQuery, 9  
   Migrate, 239  
   szablon strony, 12  
   zalety, 9  
   zapisane na dysku, 14  
   zewnętrzne, 13

**K**

kapitaliki, 24  
 key  
   Identifier, 163  
   Location, 163  
 klawisz  
   meta wciśnięty, 163  
   myszy został kliknięty, 163

- kliknięcie, 154
  - lub dwukrotne kliknięcie, 222
- klonowanie akapitu, 226
- kod
  - wykonany, jeżeli warunek jest prawdziwy, 71
  - znaku znajdującego się na pozycji n w stringu, 142
- kodowanie, 58
- kolejka, 104
- kolejność
  - działań, 31
  - elementów, 107
- komentarz, 20
- komórki
  - zliczanie, 67
- konwersja
  - argumentu na liczbę, 57
  - czasu na ciąg, 119
  - na datę, 119
  - obiekt Date na ciąg, 119
  - Unicode na znaki, 142
  - wartości obiektu na ciąg, 57
  - znaku na odpowiednik Unicode, 160
- kroki zmiennej, 91
- cursor
  - nad elementem, 154
  - przesunięty z elementu, 154
  - przesuwany na elemencie, 154
  - przesuwany nad elementem lub jednym z jego dzieci, 154

## L, Ł

- lastIndexOf, 142
- length, 63, 142
- liczba
  - bardzo
    - duża, 124
    - mała, 124
  - całkowita, 57
  - dwójkowa, 127
  - dziesiętna, 127
  - Eulera, 120
  - jako ciąg, 132
  - losowa, 121
  - milisekund od
    - 1 stycznia 1970 00:00:00, 118
    - północy 1 stycznia 1970 roku według czasu UTC, 119
  - o określonej długości, 132
  - ósemkowa, 127
  - porównywanie, 113
  - szesnastkowa, 126
  - zaokrąglona i zapisana przy użyciu notacji wykładniczej, 132
  - określonej liczby miejsc po przecinku, 132
  - zapis, 123
  - zmiennoprzecinkowa podwójnej precyzji, 125

- link graficzny, 47
- lista rozwijana, 200
- LN10, 120
- LN2, 120
- localeCompare, 142
- log, 121
- LOG10E, 120
- LOG2E, 120
- logarytm
  - naturalny z 10, 120
  - naturalny z 2, 120
  - naturalny, 121
  - o podstawie 10 z E, 120
  - o podstawie 2 z E, 120
- losowa zmiana, 304
- ładowanie obrazu przerwano, 159
- łańcuch poleceń, 300
- łączenie
  - ciągów znaków, 142, 148
  - tablic, 107
  - tekstów, 32

## M

- marginesy wokół akapitów, 317
- maskowanie pól, 199
- match, 142
- Math, 120
  - .round, 40
- max, 121
- MAX\_VALUE, 129
- metaKey, 163
- metoda, 61, 117
- miesiąc, 118
  - z daty systemowej, 78
- milisekundy, 118
  - od północy 1 stycznia 1970, 118
- min, 121
- MIN\_VALUE, 129
- minus nieskończoność, 129
- minuty, 118
- mniejsze, 41
  - lub równe, 41
- mnożenie, 31
- modulo, 31, 32
- Moo Tools, 9
- mouse
  - down, 215
  - enter, 212, 218
  - leave, 214, 218
  - up, 217

## N

- n, 141
- naciśnięty klawisz, 157
- nadanie zmiennej wartości tekstowej, 37

najmniejsza liczba, 121, 129  
 największa liczba, 121, 129  
 name, 165  
 NaN, 129  
 nazwa  
   hosta, 165  
   użytkownika, 165  
   zdarzenia, 162  
   zmiennej, 28  
 NEGATIVE\_INFINITY, 129  
 new  
   Array(), 64  
   Boolean(), 64  
   Date(), 64  
   Function(), 64  
   Number(), 64  
   Object(), 64  
   RegExp(), 64  
   String(), 64  
 nie jest równe, 41  
 noscript, 17  
 notacja naukowa, 124  
 Not-a-Number, 129  
 nowa linia, 141  
 nowy obiekt, 64, 68  
   typu  
     ciąg, 64  
     data, 64  
     funkcja, 64  
     liczba, 64  
     logicznego, 64  
     tablica, 64  
 Number, 123, 129, 130, 132  
 number(), 57  
 numer pozycji, na której znajduje się  
   ostatni znak łańcucha, 142  
   pierwszy znak łańcucha, 142

## O, Ó

obiekt, 61  
   typu  
     ciąg, 64  
     data, 64  
     funkcja, 64  
     liczba, 64  
     logicznego, 64  
     tablica, 64  
 obiektowy model dokumentu, 233  
 obraz  
   nie został załadowany prawidłowo, 159  
   w powiększeniu, 269  
   zmiana wymiarów, 285  
 odejmowanie, 31  
 odstęp od krawędzi obrazka, 271  
 odsyłacz, 26  
 odświeżenie okna przeglądarki, 116

odwołanie do  
   funkcji, 49  
   komórki, 65  
 odwracanie kolejności elementów, 107  
 odwrotność liczby, 29  
 okno dialogowe, 57  
 oktalna liczba, 127  
 on  
   abort, 159  
   blur, 160  
   change, 160  
   click, 154  
   dblclick, 154  
   error, 159  
 focus, 160  
 keydown, 157  
 keypress, 157  
 load, 159  
 mouse  
   down, 154  
   enter, 154  
   leave, 154  
   move, 154  
   out, 154  
   over, 154  
   up, 154  
   reset, 160  
   resize, 159  
   scroll, 159  
   select, 160  
   submit, 160  
   unload, 159  
 opacity, 291, 303  
 operator  
   arytmetyczny, 30  
   identyczności, 42  
   logiczny, 44  
   porównania, 41  
   przypisania, 35  
   warunkowy, 80  
 operowanie datą, 117  
 origin, 165  
 ósemkowa liczba, 127

## P

p, 210, 214  
 params, 282  
 parse, 118  
   Float, 57, 130  
   Int, 57, 130  
 password, 165, 188  
 pathname, 165  
 pętla, 96,  
   w pętli, 103  
 Pi, 120  
 pierwiastek kwadratowy, 120, 121

pierwotna wartość obiektu, 132  
   String, 142  
 plus nieskończoność, 129  
 płynne  
   pojawiania się elementów, 273  
   ukrywanie elementu, 273  
 pobieranie znaków z ciągu, 149  
 podzielność liczby, 80  
 pole  
   do wpisywania hasła, 188  
   wyboru, 191  
   wyróżnione domyślnie, 202  
 położenie klawisza w urządzeniu, 163  
 porównanie  
   liczb, 113  
   stringów, 142  
 port, 165  
 position, 282  
 POSITIVE\_INFINITY, 129  
 potęgowanie, 121  
 pow, 121  
 powrót karetki, 141  
 powtarzanie operacji, 81  
 pozycja znaku w łańcuchu, 142  
 prawda, 44, 112  
 prepend, 244  
 production version, 14  
 prompt, 57, 114  
 prop, 239, 240, 241  
 protocol, 165  
 protokół, 165  
 prototype, 9, 142  
 przeglądarka nie obsługuje JavaScript, 17  
 przekazywanie wartości między funkcjami, 52  
 przekształcanie zmiennych w liczby, 130  
 przesunięcie  
   elementów w dół, 279  
   kursora  
     nad obszar, 156  
     z akapitu, 213  
     kursora z obszaru, 218  
   obrazka klikaniem przycisków, 305  
   strefy czasowej, 119  
 przeszukiwanie  
   ciągu, 144  
   stringów, 142  
 przewinięto dokument, 159  
 przezroczystość, 303  
   obrazka, 289

## R

r, 141  
 random, 121, 305  
 ready, 212  
 referencja, 134  
 refresh, 116  
 rekord, 62

rel, 165  
 relatedTarget, 163  
 remove, 250  
 replace, 142  
 reprezentacja zmiennoprzecinkowa, 125  
 rev, 165  
 reverse, 107  
 rok, 118  
 round, 121  
 rozmiar czcionki, 313  
 równe, 41, 42

## S, Ś

Screen  
   X, 163  
   Y, 163  
 search, 142, 165  
 sekundy, 118  
 selected, 201, 225  
 selektor, 179  
   globalny, 175  
 set  
   Date, 118  
   FullYear, 118  
   Hours, 118  
   Milliseconds, 118  
   Minutes, 118  
   Month, 118  
   Seconds, 118  
   Time, 118  
   UTC  
     Date, 118  
     FullYear, 118  
     Hours, 118  
     Milliseconds, 118  
     Minutes, 118  
     Month, 118  
     Seconds, 119  
   Year, 119  
 shift, 105  
   Key, 163  
   wciśnięty, 163  
 show, 184  
 sin, 121  
 sinus, 121  
 skrypt  
   osadzony, 11, 18  
   wyłączona obsługa, 17  
   zewnątrzny, 11, 18  
 slice, 142, 145  
 slide  
   Down, 279  
   Toggle, 280  
   Up, 284  
 słowa oddzielone spacjami, 58  
 słuchacz zdarzeń, 162  
 speed, 282

spełnienie warunku, 71  
 split, 142  
 sqrt, 120, 121  
 string, 57, 63, 139, 140, 142  
 Strona nie została załadowana, 159  
 substr, 142  
 substring, 142  
 suma logiczna, 45  
 sumowanie liczb, 50  
 switch, 78  
 szary obrazek tła, 194  
 szesnastkowa liczba, 126, 127  
 ścieżka dostępu, 165

## T

t, 141  
 tabela  
   sortowanie, 100  
 tablica, 64, 91, 93, 150  
   adresowanie  
     liczbowe, 98  
     przez nazwę, 98  
   dopisywanie danych, 66  
   dwuwymiarowa, 102  
   łączenie, 107  
   rozmiar, 96  
   usuwanie danych, 99  
 tabulator, 141  
 tan, 121  
 tangens, 121  
 target, 162, 165  
 tekst  
   akapitowy, 25  
   łączenie, 32  
   odstępów pomiędzy wyrazami, 294  
   powiększenie, 293  
   ukrywanie, 250  
   wraz ze znacznikami HTML, 235  
   wyświetlanie, 250  
   z podziałem na akapity, 23  
   zawarty w linku, 165  
 temperatura, zamiana skali, 39  
 text, 165, 188, 233  
 this, 177  
 timeStamp, 162  
 tło  
   dodawanie, 257  
   poła wyboru, 191  
   przycisków  
     radiowych, 190  
     do wysyłania danych, 192  
   usuwanie, 257  
 to  
   DateString, 119  
   Exponential, 132  
   Fixed, 132  
   GMTString, 119

ISOString, 119  
 JSON, 119  
 Locale  
   DateString, 119  
   LowerCase, 142  
   String, 119  
   TimeString, 119  
   UpperCase, 142  
 LowerCase, 142  
 Precision, 132  
 String, 119, 127, 132, 142  
 TimeString, 119  
 UpperCase, 142  
 UTCString, 119

trim, 142  
 true, 44, 112  
 typ zmiennej, 28, 139  
 type, 162, 165  
 typeof, 139

## U

ukrycie  
   elementu o określonym id, 172  
   fragmentów strony, 170  
   odsylaczy, 184  
   tekstu, 250  
 umieszczenie kursora w polu, 220  
 undefined, 36  
 Unicode, 161  
 unshift, 105  
 URI, 57, 59  
 URL, 59  
 URN, 59  
 uruchamianie  
   wielu poleceń jQuery, 300  
   skryptu po kliknięciu linku, 206  
 username, 165  
 ustalenie znaku, 31  
 ustawianie  
   daty liczbą milisekund  
     przed/po 1 stycznia 1970, 118  
   dnia miesiąca, 118  
   godziny, 118  
   miesiąca, 118  
   milisekund, 118  
   minut, 118  
   roku, 118  
   sekund, 118, 119  
 usuwanie elementów  
   podrzędnych., 252  
   z dokumentu HTML, 250  
   ze strony, 179  
 usuwanie pierwszej pozycji z listy, 180  
 UTC, 118, 119  
 utrata fokusu, 160  
 uzyskanie fokusu, 160

**V**

val, 237  
 value, 238  
   Of, 119, 132, 142  
 Verdana, 24

**W**

wartość  
   bezwzględna, 121  
   logiczna, 112  
   szesnastkowa, 126  
   zmiennej, 29, 36  
   zmiennoprzecinkowa, 57, 60  
 warunek, 74  
   nie zostanie spełniony, 76  
   sprawdzany na końcu., 86  
 wcięcie tekstu, 310  
 wciśnięcie  
   klawisza, 157  
   przycisku myszy, 214  
 wczytanie daty z systemu operacyjnego, 29  
 while, 81, 84  
 width, 269  
 większe, 41  
   lub równe, 41  
 właściwość, 61  
 włączanie i wyłączanie formatowania, 265  
 wpisanie wartości zwracanej do programu, 57  
 wrap, 191  
 wskazanie  
   akapitu kursorem, 211  
   obszaru kursorem, 218  
 wskaźnik myszy usunięty z monitorowanego obszaru, 220  
 współrzędna  
   pionowa wskaźnika myszy w stosunku do bieżącego okna, gdy zdarzenie zostało zarejestrowane, 163  
   ekranu, gdy zdarzenie zostało zarejestrowane, 163  
   klikniętego miejsca w stosunku do bieżącego okna, 164  
 wstawianie elementu we wskazanym miejscu, 246  
 wybranie tekstu, 160  
 wydzielenie  
   część ciągu, 142  
   liczby z tekstu, 131  
   tekstu bez znaczników HTML, 233  
 wykonanie  
   gałęzi programu, 42  
   kodu określoną liczbę razy, 90  
 wynik porównywania, 41  
 wyróżnienie akapitu, który zawierają ciąg znaków, 187  
 wyskakujące okno, 80, 206  
 wysłanie formularza, 160  
 wysunięcie papieru, 141

wyszarzenie, 194  
 wyświetlanie  
   fragmentu tekstu w oknie komunikatu, 229  
   komunikatu po kliknięciu, 208  
   odsyłaczy, 184  
   okna  
     komunikatu, 58  
     po załadowaniu strony głównej, 206  
   tekstu, 250  
 wzorzec dopasowania, 64

**X**

x do potęgi y, 121

**Z**

zakończenie operacji, 297  
 zamiana  
   ciągu na tablicę, 142  
   liter, 146  
     na małe, 142  
     na wielkie, 142, 157  
   znaków w stringu, 142  
 zaokrąglenie  
   liczb, 135  
   w dół do najbliższej liczby całkowitej, 121  
   w górę do najbliższej liczby całkowitej, 121  
   do najbliższej liczby całkowitej, 121  
 zapis wykładniczy, 39, 124  
 zaprzeczenie, 45  
 zasięg zmiennej, 52  
 zdarzenie, 153, 205  
   zablokowane, 162  
 zgodność wartości i typów, 41  
 zmiana  
   adresu po kliknięciu przycisku, 240  
   atrybutów w kodzie HTML, 240  
   czcionki, 186  
   elementu formularza, 160, 224  
   przezroczystości, 276  
   rozmiarów widoku dokumentu, 159  
 zmienna, 27, 134  
   czas życia, 52  
   globalna, 50  
   lokalna, 55  
   typ, 139  
   wartość, 36  
 zmiennoprzecinkowa liczba, 125  
 znak  
   na pozycji n w stringu, 142  
   znak specjalny, 141  
 zwolnienie  
   przycisku myszy, 216  
   klawisza, 157

# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

# JavaScript i jQuery

JavaScript jest tak popularnym językiem programowania, że właściwie nie trzeba go przedstawiać. To on pozwala na umieszczanie w kodzie HTML lub XHTML skryptów umożliwiających reagowanie na zdarzenia, sprawdzanie poprawności danych i wyświetlanie okien dialogowych. Dzięki tym oraz wielu innym zaletom JavaScript jest nieodzownym elementem prawie każdej współczesnej strony WWW. Z kolei biblioteka jQuery znacząco rozszerza jego możliwości, ponieważ zawiera gotowe fragmenty kodu, co zwalnia nas z konieczności pisania każdego skryptu z osobna. Ten duet jest naprawdę fantastyczny!

Jeśli chcesz się przekonać, jakie cuda JavaScript i jego framework potrafią wspólnie zdziałać, koniecznie sięgnij po ten podręcznik. Znajdziesz tu grubo ponad setkę gotowych skryptów, które wydatnie przyczynią się do poprawy jakości Twoich stron. Dowiesz się, jak używać funkcji i obiektów, do czego służą instrukcje warunkowe i po co istnieją tablice. Nauczysz się operować zmiennymi do wykonywania obliczeń matematycznych, wstawiać odpowiednio sformatowane teksty i poprawnie obsługiwać zdarzenia. Każdy kolejny krok został dokładnie opisany, a skrypty tylko czekają, by zacząć działać na Twoich stronach WWW...

- Podstawy
- Pisanie na stronie i działania na zmiennych
- Funkcje i obiekty
- Instrukcje warunkowe i pętle
- Tablica
- Obliczenia
- Liczby i łańcuchy
- Teksty
- Zdarzenia

**JavaScript i jQuery — szybka nauka na dobrych przykładach!**

**Helion**

26747 numer katalogowy

księgarnia internetowa



<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

Sprawdź najnowsze promocje:

🔗 <http://helion.pl/promocje>

Książki najchętniej czytane:

🔗 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

🔗 <http://helion.pl/nowosci>

Helion SA  
ul. Kościuszki 1c, 44-100 Gliwice  
tel.: 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-246-9997-1



9 788324 699971

Informatyka w najlepszym wydaniu

cena: 59,00 zł