

JavaScript dla dzieci

— ściągawka

Kiedy programujesz w JavaScriptcie, powinieneś wiedzieć, jakie są w nim odpowiedniki nazw właściwości CSS. Możliwość reagowania na zdarzenia, w tym te opisane tutaj, przez podpinanie użytecznych funkcji do wykonania w przeglądarce jest ważną częścią możliwości JavaScriptu. Na koniec przedstawimy również listę zarezerwowanych słów kluczowych, które nie mogą być użyte jako nazwy zmiennych, metod czy obiektów.

Konwersja nazw właściwości CSS w JavaScriptcie

Możesz używać właściwości CSS w JavaScriptcie, jest jednak kilka rzeczy, o których musisz pamiętać. Przy zmianach stylów w JavaScriptcie istnieje kilka prostych reguł:

- ▶▶ Jeśli nazwa właściwości CSS składa się z jednego słowa (np. `height`), to pozostaje bez zmian.
- ▶▶ Jeśli nazwa właściwości CSS składa się z kilku słów oddzielonych myślnikami, konwertujemy ją na zapis w stylu *camelCase*.

Oto kompletna lista nazw właściwości CSS i ich odpowiedników w JavaScriptcie.

Właściwość CSS	Odpowiednik w JavaScriptcie
<code>background</code>	<code>background</code>
<code>background-attachment</code>	<code>backgroundAttachment</code>
<code>background-color</code>	<code>backgroundColor</code>
<code>background-image</code>	<code>backgroundImage</code>
<code>background-position</code>	<code>backgroundPosition</code>
<code>background-repeat</code>	<code>backgroundRepeat</code>
<code>border</code>	<code>border</code>
<code>border-bottom</code>	<code>borderBottom</code>
<code>border-bottom-color</code>	<code>borderBottomColor</code>
<code>border-bottom-style</code>	<code>borderBottomStyle</code>
<code>border-bottom-width</code>	<code>borderBottomWidth</code>
<code>border-color</code>	<code>borderColor</code>
<code>border-left</code>	<code>borderLeft</code>
<code>border-left-color</code>	<code>borderLeftColor</code>
<code>border-left-style</code>	<code>borderLeftStyle</code>
<code>border-left-width</code>	<code>borderLeftWidth</code>
<code>border-right</code>	<code>borderRight</code>
<code>border-right-color</code>	<code>borderRightColor</code>

Właściwość CSS	Odpowiednik w JavaScriptcie
border-right-style	borderRightStyle
border-right-width	borderRightWidth
border-style	borderStyle
border-top	borderTop
border-top-color	borderTopColor
border-top-style	borderTopStyle
border-top-width	borderTopWidth
border-width	borderWidth
clear	clear
clip	clip
color	color
cursor	cursor
display	display
filter	filter
float	cssFloat
font	font
font-family	fontFamily
font-size	fontSize
font-variant	fontVariant
font-weight	fontWeight
height	height
left	left
letter-spacing	letterSpacing
line-height	lineHeight
list-style	listStyle
list-style-image	listStyleImage
list-style-position	listStylePosition
list-style-type	listStyleType
margin	margin
margin-bottom	marginBottom

Właściwość CSS	Odpowiednik w JavaScriptcie
margin-left	marginLeft
margin-right	marginRight
margin-top	marginTop
overflow	overflow
padding	padding
padding-bottom	paddingBottom
padding-left	paddingLeft
padding-right	paddingRight
padding-top	paddingTop
page-break-after	pageBreakAfter
page-break-before	pageBreakBefore
position	position
text-align	textAlign
text-decoration	textDecoration
text-decoration: blink	textDecorationBlink
text-decoration: line-through	textDecorationLineThrough
text-decoration: none	textDecorationNone
text-decoration: overline	textDecorationOverline
text-decoration: underline	textDecorationUnderline
text-indent	textIndent
text-transform	textTransform
top	top
vertical-align	verticalAlign
visibility	visibility
width	width
z-index	zIndex

Zdarzenia przeglądarki w JavaScriptcie

Strony internetowe to już obecnie bardzo rzadko tylko statycznie wyświetlany tekst i grafika. JavaScript daje spore możliwości interakcji i wykonywania użytecznej pracy. Ważną częścią możliwości JavaScriptu w tej kwestii jest reagowanie na zdarzenia zachodzące w przeglądarce.

Oto lista częstych zdarzeń, które zachodzą w przeglądarce, a na które może reagować JavaScript.

Zdarzenie	Kiedy zachodzi
abort	Ładowanie pliku zostało anulowane
blur	Element został dezaktywowany. Innymi słowy, element nie jest już tym, nad czym użytkownik aktywnie pracuje
change	Wartość elementu uległa zmianie od czasu dezaktywacji i ponownej aktywacji
click	Element został kliknięty myszką
dblclick	Element został podwójnie kliknięty myszką
error	Wystąpił błąd przy ładowaniu pliku
focus	Element został aktywowany. Innymi słowy, element jest teraz tym, nad czym użytkownik aktywnie pracuje
input	Wartość elementu <code><input></code> lub <code><textarea></code> została zmieniona
keydown	Klawisz został naciśnięty
keyup	Klawisz został zwolniony po naciśnięciu
load	Plik i powiązane z nim pliki zostały załadowane
mousedown	Przycisk myszy został naciśnięty na elemencie
mouseenter	Wskaźnik myszy wszedł w obręb elementu
mouseleave	Wskaźnik myszy opuścił obręb elementu
mouseout	Wskaźnik myszy opuścił obręb elementu lub jednego z jego elementów wewnętrznych
mouseover	Wskaźnik myszy wszedł w obręb elementu lub jednego z jego elementów wewnętrznych
mouseup	Przycisk myszy został zwolniony nad elementem
mousewheel	Kółko myszy zostało przekręcone
reset	Formularz został zresetowany
resize	Rozmiar dokumentu został zmieniony
scroll	Dokument lub element został przewinięty
select	Tekst został zaznaczony
submit	Formularz został wysłany

Czym są zarezerwowane słowa w JavaScriptcie?

W JavaScriptcie istnieją zarezerwowane słowa, które powinieneś znać przed rozpoczęciem kodowania. Poniższa tabela zawiera listę wszystkich zarezerwowanych słów kluczowych, które nie mogą być użyte jako nazwy zmiennych, funkcji, metod czy obiektów.

<code>abstract</code>	<code>boolean</code>	<code>break</code>
<code>byte</code>	<code>case</code>	<code>catch</code>
<code>char</code>	<code>class</code>	<code>const</code>
<code>continue</code>	<code>debugger</code>	<code>default</code>
<code>delete</code>	<code>do</code>	<code>double</code>
<code>else</code>	<code>enum</code>	<code>export</code>
<code>extends</code>	<code>false</code>	<code>final</code>
<code>finally</code>	<code>float</code>	<code>for</code>
<code>function</code>	<code>goto</code>	<code>if</code>
<code>implements</code>	<code>import</code>	<code>in</code>
<code>instanceof</code>	<code>int</code>	<code>interface</code>
<code>long</code>	<code>native</code>	<code>new</code>
<code>null</code>	<code>package</code>	<code>private</code>
<code>protected</code>	<code>public</code>	<code>return</code>
<code>short</code>	<code>static</code>	<code>super</code>
<code>switch</code>	<code>synchronized</code>	<code>this</code>
<code>throw</code>	<code>throws</code>	<code>transient</code>
<code>true</code>	<code>try</code>	<code>typeof</code>
<code>var</code>	<code>void</code>	<code>volatile</code>
<code>while</code>	<code>with</code>	