

# Technologie Internetu. DOM

Aleksander Denisiuk (denisjuk@pja.edu.pl)  
Polsko-Japońska Akademia Technik Komputerowych  
Wydział Informatyki w Gdańsku  
ul. Brzegi 55, 80-045 Gdańsk

27 maja 2020

# DOM

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

Najnowsza wersja tego dokumentu dostępna jest pod adresem  
<http://users.pja.edu.pl/~denisjuk/>

## BOM

Obiekty przeglądarki

navigator

screen

location

frames

history

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

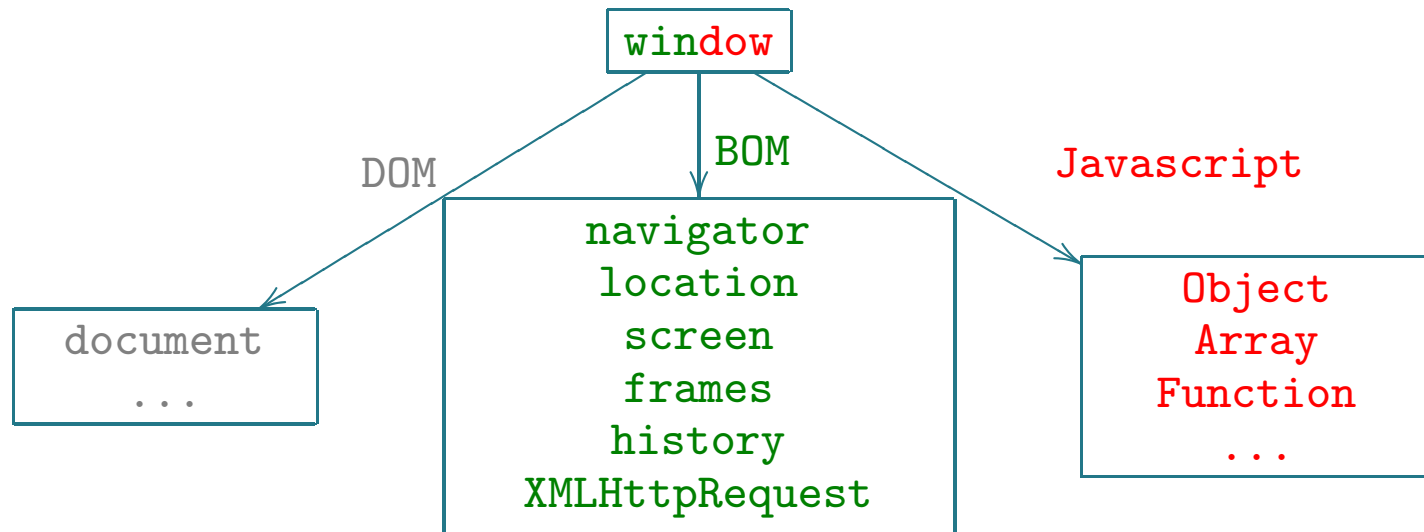
Wymiary okna

Współrzędne

# BOM

# Obiekty przeglądarki

- ✓ Przeglądarka pozwala na dostęp do hierarchii obiektów
- ✓ Trzy typy obiektów globalnych



# Obiekt `navigator`

- BOM
- Obiekty przeglądarki
  - navigator**
  - screen
  - location
  - frames
  - history
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

- ✓ ogólna informacja o przeglądarce i systemie operacyjnym
  - ✗ `navigator.userAgent` informacja o przeglądarce
  - ✗ `navigator.platform` informacja o systemie operacyjnym
    - ✓ niektóre wersje przeglądarek na niektórych platformach mają błędy
- ✓ `navigator.geolocation`
- ✓ **Zobacz**

# Przykład navigator

BOM
Obiekty przeglądarki
<b>navigator</b>
screen
location
frames
history
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

```
<!DOCTYPE html>
<html>
  <body>
    <script>
      alert(navigator.userAgent);
      alert(navigator.platform);
    </script>
  </body>
</html>
```

✓ DEMO

# Obiekt **screen**

- ✓ informacja o ekranie (rozdzielczość, kolorowość, etc)

```
<script>
document.write(screen.width + \
    ' x ' + screen.height)
</script>
```

- ✓ DEMO
- ✓ Statystyka W3C

[BOM](#)

[Obiekty przeglądarki](#)  
[navigator](#)

**[screen](#)**

[location](#)

[frames](#)

[history](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

# location

- ✓ informacja o bieżącym URL
- ✓ pozwala przekierować na inny URL
- ✓ wartością tej właściwości jest obiekt typu **Location**

BOM
Obiekty przeglądarki
navigator
screen
location
frames
history
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne



# Obiekt Location

- BOM
- Obiekty przeglądarki
  - navigator
  - screen
  - location
  - frames
  - history
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

- ✓ `toString()`
  - ✗ `alert(window.location)` [Demo](#)
  - ✗ [Pomyłka](#): `window.location.indexOf('/://')`
  - ✗ [Poprawnie](#):  
`window.location.toString.indexOf('/://')`

# Właściwości obiektu `Location`

BOM

Obiekty przeglądarki

`navigator`

`screen`

`location`

`frames`

`history`

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ przykładowy URL `http://www.com:80/search?q=js#test`
  - ✗ `href` — całość:  
`http://www.com:80/search?q=js#test`
  - ✗ `hash` — część URL po `#` włącznie: `#test`
  - ✗ `host` — host i port: `www.com:80`
  - ✗ `hostname` — host: `www.com`
  - ✗ `pathname` — ścieżka na serwerze: `/search`
  - ✗ `port` — port: `80`
  - ✗ `protocol` — protokół (z dwukropkiem): `http:`
  - ✗ `search` — część adresu po `?` (razem z tym znakiem): `?q=js`
- ✓ wszystkie są typu `String`

# Metody obiektu **Location**

BOM

Obiekty przeglądarki

navigator

screen

**location**

frames

history

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ `assign(url)` załadować dokument. Można również `window.location.href = url`.
- ✓ `reload([forceget])` przeładować dokument. Jeżeli `forceget==true`, to z serwera. Jeżeli `forceget==false` (domyślnie), to może być z cache'a.
- ✓ `replace(url)` zastąpić dokument na `url`, strona nie zapisuje się w historii
- ✓ `toString()`

# Przykłady z Location

BOM

Obiekty przeglądarki

navigator

screen

location

frames

history

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne



```
window.location = "http://denisiuk.com";
```

Demo



```
for (var prop in location){  
    alert(prop + ': ' + location[prop]);  
}
```

Demo



```
function refreshSearch(search) {  
    window.location.search = search;  
}
```

Demo

# frames

BOM
Obiekty przeglądarki
navigator
screen
location
<b>frames</b>
history
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ Kolekcja, zawierająca obiekty `window` podrzędnych `frame`ów oraz `iframe`ów

- ✓ Przykład

```
<iframe name="google" src="http://google.pl">
</iframe>
<script>
    window.frames.google.location = \
        'http://duckduckgo.com/';
</script>
```

## Demo

- ✓ okno ma dostęp do danych frame'a, jeżeli jest z nim na tej samej domenie (łącznie z portem).
- ✓ wyjątek: można zmienić `location`

# history

BOM
Obiekty przeglądarki
navigator
screen
location
frames
history
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ pozwala zmieniać URL bez przeładowania strony (w granicach tej samej domeny) za pomocą **History API**
- ✓ przewijać historię stron do tyłu **history.back()** oraz do przodu **history.forward()**
- ✓ adresy URL z historii nie są dostępne (bezpieczeństwo)
- ✓ HTML5: nowe funkcje: **pushState()**, **replaceState()**

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Wprowadzenie do DOM</a>
<a href="#">Autokorekta</a>
<a href="#">Inne węzły</a>
<a href="#">Normalizacja</a>
<a href="#">Możliwości DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

# DOM

# Wprowadzenie do DOM

[BOM](#)

[DOM](#)

[Wprowadzenie do DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ Document Object Model — model dokumentu XML/HTML
- ✓ level1, level2, level3
- ✓ level4
- ✓ level0
- ✓ Dokument jest hierarchią elementów.
- ✓ Każdy znacznik tworzy węzeł
- ✓ Każdy fragment tekstu także tworzy węzeł
- ✓ Włożoność elementów nadaje DOMowi strukturę drzewa



# Przykład DOMu

- [BOM](#)
- [DOM](#)
- [Wprowadzenie do DOM](#)
- [Autokorekta](#)
- [Inne węzły](#)
- [Normalizacja](#)
- [Możliwości DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

```
<html>
  <head>
    <title>Title</title>
  </head>
  <body>
    Fajny dokument
  </body>
</html>
```

# Przykład DOMu, cd

BOM

DOM

Wprowadzenie do  
DOM

Autokorekta

Inne węzły

Normalizacja

Możliwości DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

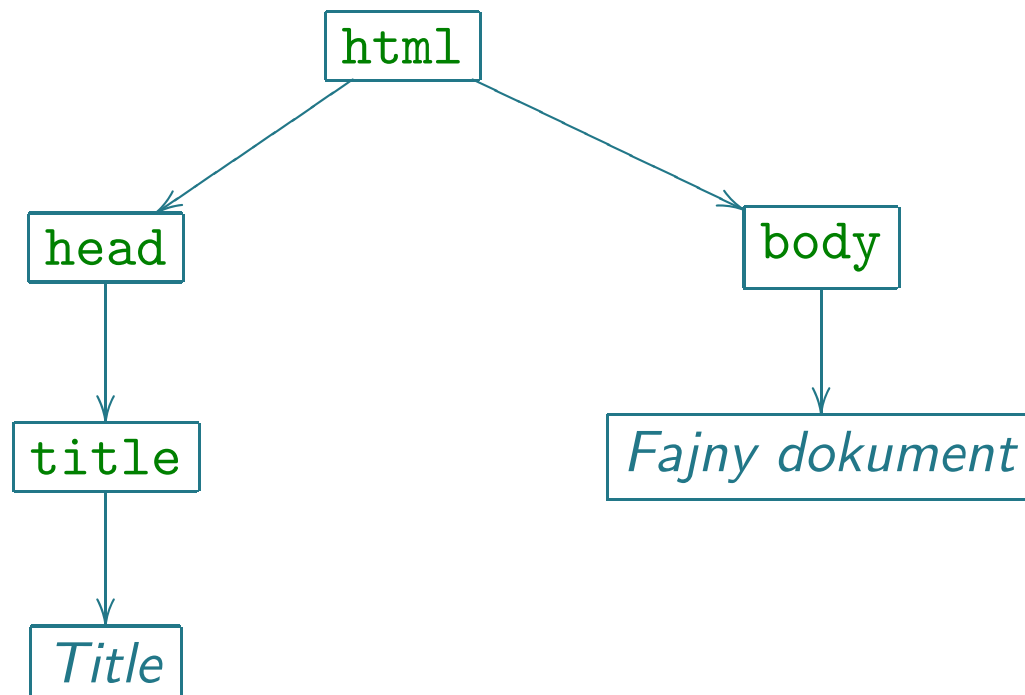
Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

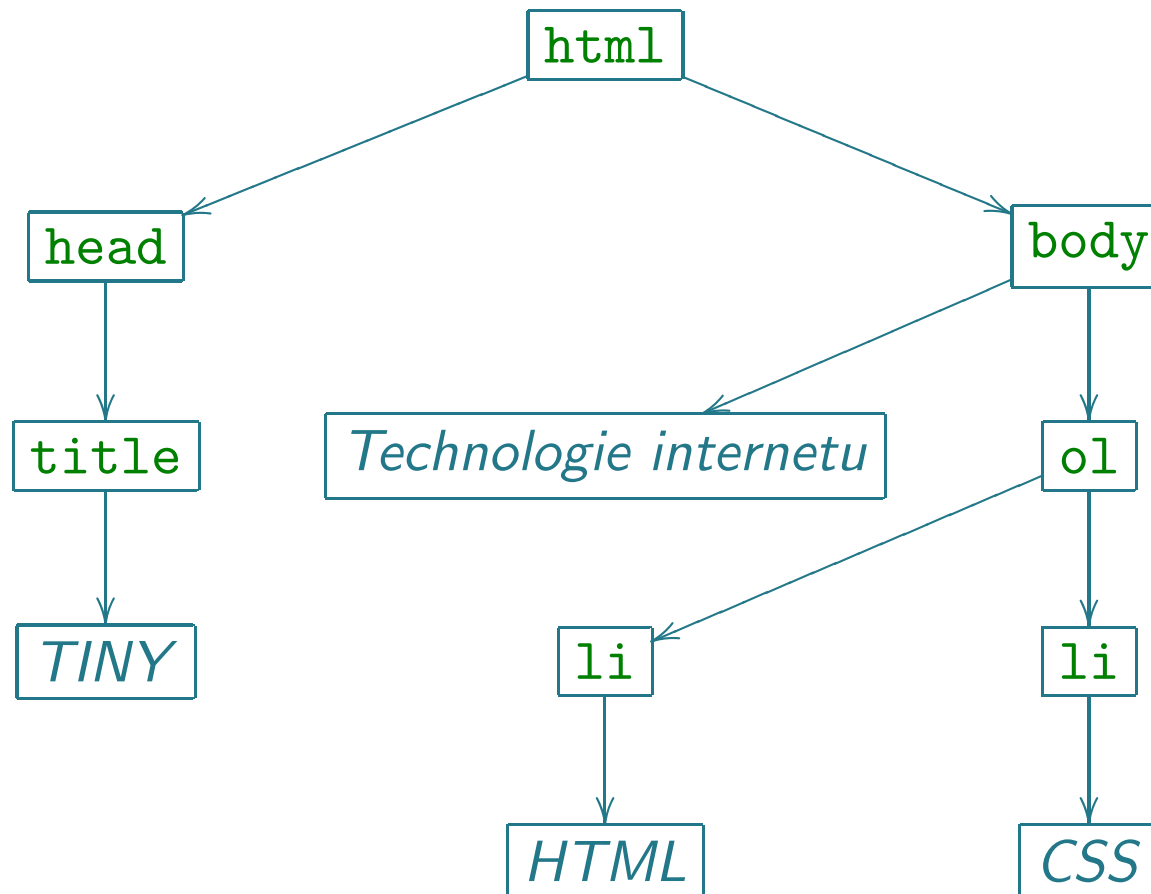


## Drugi przykład

BOM
DOM
Wprowadzenie do DOM
Autokorekta
Inne węzły
Normalizacja
Możliwości DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

```
<html>
  <head>
    <title>
      TINY
    </title>
  </head>
  <body>
    Technologie internetu.
    <ol>
      <li>HTML</li>
      <li>CSS</li>
    </ol>
  </body>
</html>
```

## Drugi przykład, cd



BOM

DOM

Wprowadzenie do  
DOM

Autokorekta

Inne węzły

Normalizacja

Możliwości DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# DOM. Firebug i inne

[BOM](#)

[DOM](#)

[Wprowadzenie do DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)



DEMO

# Autokorekta

[BOM](#)

[DOM](#)

[Wprowadzenie do DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

✓ Wczytując niepoprawny HTML, przeglądarka automatycznie go koryguje

✗ zawsze będzie element `html`

✗ zawsze będzie element `body`

✓ DEMO

# Autokorekta, uzupełnienie

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Wprowadzenie do DOM</a>
<a href="#">Autokorekta</a>
<a href="#">Inne węzły</a>
<a href="#">Normalizacja</a>
<a href="#">Możliwości DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

- ✓ uzupełnia się i domyka się znaczniki

```
<p>Me,  
<li>Myself  
<li>and  
<li>Eye
```

- ✓ DEMO

# Autokorekta a tabele

- [BOM](#)
- [DOM](#)
- [Wprowadzenie do DOM](#)
- [Autokorekta](#)
- [Inne węzły](#)
- [Normalizacja](#)
- [Możliwości DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

- ✓ tabele zawsze mają `tbody`

```
<table id="table">  
  <tr><td>1</td></tr>  
</table>
```

- ✓ DEMO



# Doctype i atrybuty

[BOM](#)

[DOM](#)

[Wprowadzenie do DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ Doctype też jest DOM-węzłem. Znajduję się na drzewie po lewej od `html`
- ✓ Atrybuty w modelu DOM także są węzłami, potomnymi do elementów, w których zostały zdefiniowane.
- ✓ W programowaniu w JavaScript uważamy, że atrybuty są właściwościami obiektów-węzłów, które można odczytać i/lub ustawić.

# Doctype i atrybuty. Przykład

BOM

DOM

Wprowadzenie do  
DOM

Autokorekta

Inne węzły

Normalizacja

Możliwości DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

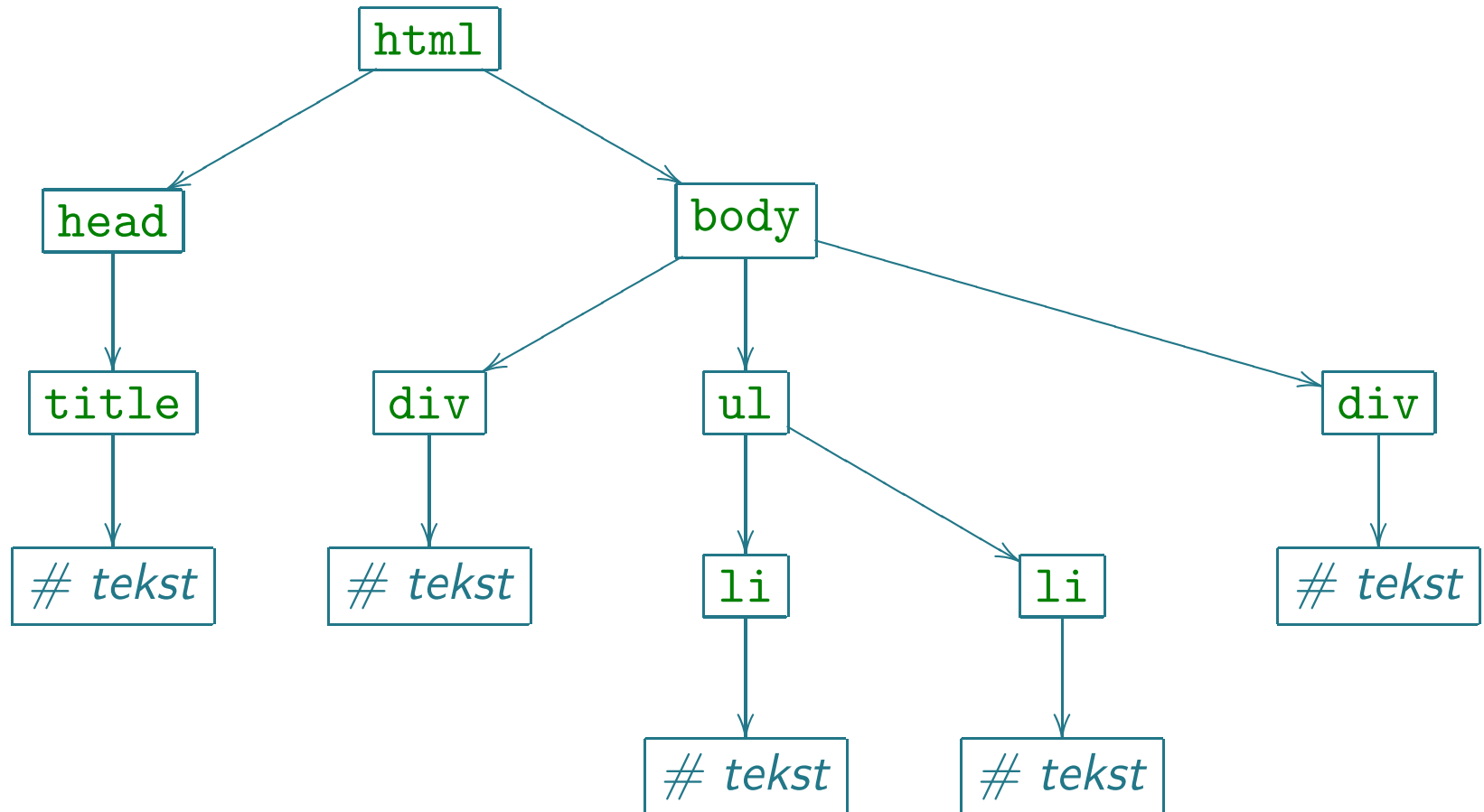
Wymiary okna

Współrzędne

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional  
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>  
  <head>  
    <title>Tiny</title>  
  </head>  
  <body>  
    <div id="dataKeeper">Data</div>  
    <ul>  
      <li style="background-color:red">  
        Look out!</li>  
      <li class="info">Information</li>  
    </ul>  
    <div id="footer">&copy; 2012</div>  
  </body>  
</html>
```

# Doctype i atrybuty. Drzewo



# Komentarze

[BOM](#)

[DOM](#)

[Wprowadzenie do DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ Komentarz też jest DOM-węzłem (comment node).

```
<html>
```

```
...
```

```
<!-- comment -->
```

```
...
```

```
</html>
```

- ✓ JavaScript ma dostęp do tego węzła jak i do jego zawartości

# Normalizacja

BOM

DOM

Wprowadzenie do  
DOM

Autokorekta

Inne węzły

Normalizacja

Możliwości DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

✓ Drzewo DOM nazywa się *normalizowanym*, jeżeli w nim został zignorowany węzły tekstowe z samych „białych” znaków

✓ Czy należy normalizować DOM?

**Standard XML:** Nie

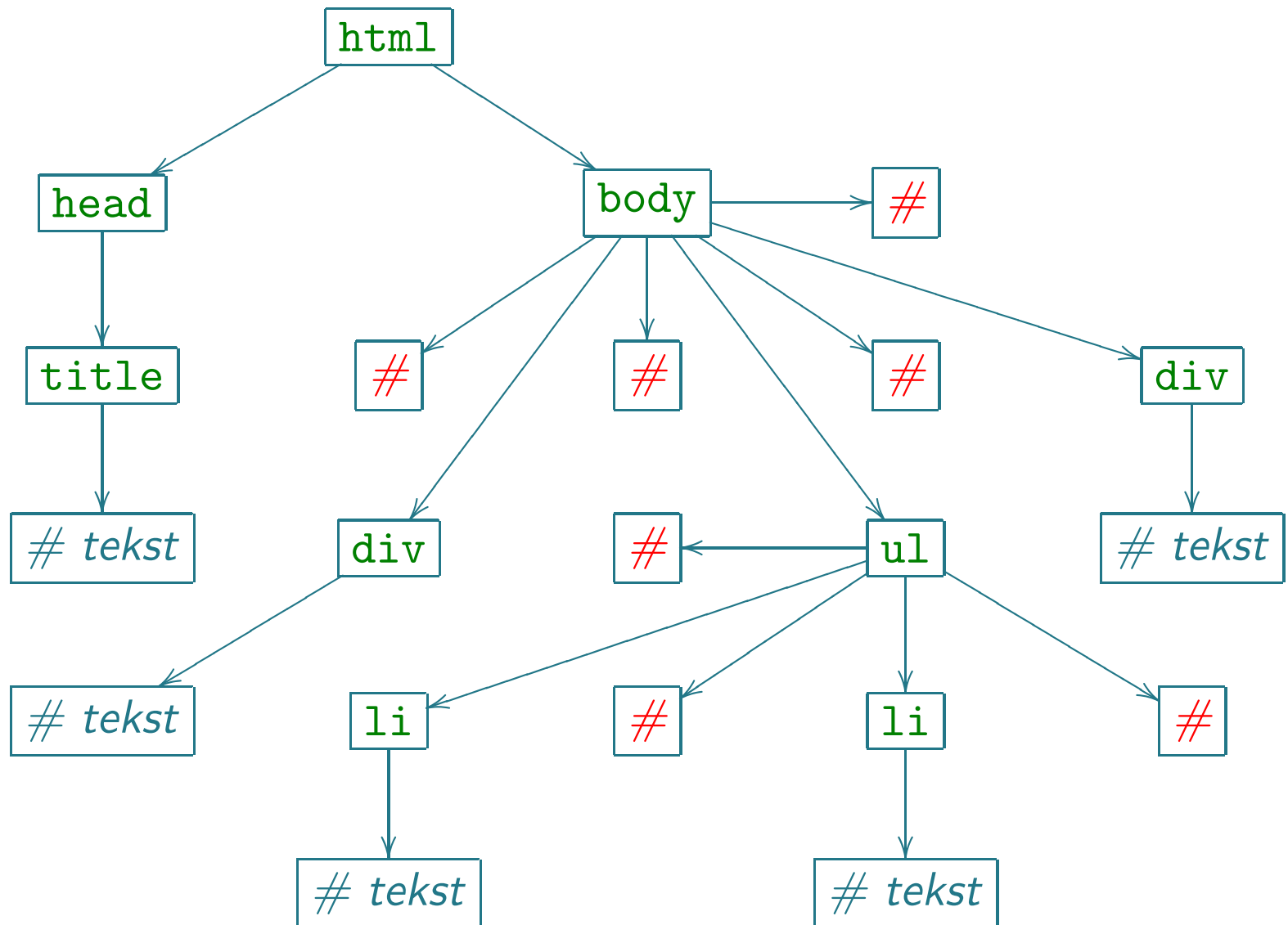
**Gecko:** Nie

**Webkit:** Nie

**Internet Explorer (< 9):** Tak

✓ Firebug i inne nie pokazują „białych” węzłów

# Nienormalizowany DOM



# Możliwości DOM

BOM

DOM

Wprowadzenie do  
DOM

Autokorekta

Inne węzły

Normalizacja

Możliwości DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ Każdy element DOM jest obiektem, który pozwala na manipulacje atrybutami, zawartością (w tym dodawanie i usuwanie węzłów)
- ✓ A także na dostęp do dziecięcych, siostrzanych oraz rodzicielskich elementów
- ✓ Dostęp do drzewa DOM możliwy jest przez obiekt `document`
- ✓ Za pomocą `document` otrzymamy dostęp do wszystkich elementów drzewa
- ✓ Funkcja `document.write()` jest przestarzała

# Przykład DOM. HTML

[BOM](#)

[DOM](#)

[Wprowadzenie do DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

```
<body>
  Formula 1:
  <ol>
    <li>
      Jensen Button
    </li>
    <li>
      Sebastian Vettel
    </li>
  </ol>
</body>
```



# Przykład DOM. Javascript

BOM

DOM

Wprowadzenie do  
DOM

Autokorekta

Inne węzły

Normalizacja

Możliwości DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
function go() {  
    var ol =  
        document.getElementsByTagName('ol')[0]  
    var pierwszy =  
        ol.removeChild(ol.firstChild)  
    var drugi = ol.removeChild(ol.firstChild)  
    ol.appendChild(drugi)  
    ol.appendChild(pierwszy)  
}  
window.onload = function(){  
    document.body.onclick = go  
}
```

# Przykład DOM. Demo

[BOM](#)

[DOM](#)

[Wprowadzenie do  
DOM](#)

[Autokorekta](#)

[Inne węzły](#)

[Normalizacja](#)

[Możliwości DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)



DEMO

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Wierzchołek</a>
<a href="#">Węzły dziecięce</a>
<a href="#">Nawigacja w pionie</a>
<a href="#">Tabele</a>
<a href="#">Tylko elementy</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

# Nawigacja

# Wierzchołek drzewa

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Wierzchołek](#)

[Węzły dziecięce](#)

[Nawigacja w pionie](#)

[Tabele](#)

[Tylko elementy](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ `document.documentElement` — wierzchołek drzewa (`html`)
- ✓ `document.body` — element `body`
- ✓ `document.head` — element `head`
- ✓ `document.doctype` — `DOCTYPE`
- ✓ `body` może nie istnieć na moment wykonania skryptu: [DEMO](#)
- ✓ Uwaga: w DOM zawsze, gdy nie ma elementu, wykorzystywany jest `null`

# Węzły dziecięce `childNodes`

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Wierzchołek](#)

[Węzły dziecięce](#)

[Nawigacja w pionie](#)

[Tabele](#)

[Tylko elementy](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

```
var childNodes = document.body.childNodes;  
for(var i=0; i<childNodes.length; i++) {  
    alert(childNodes[i]);  
}
```

- ✓ DEMO
- ✓ IE < 9 nie tworzy „białych” tekstowych węzłów
- ✓ czemu nie wyświetla się `TextNode` po `script`?

# Elementy dziecięce `children`

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Wierzchołek](#)

[Węzły dziecięce](#)

[Nawigacja w pionie](#)

[Tabele](#)

[Tylko elementy](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

```
var childNodes = document.body.children;
for(var i=0; i<childNodes.length; i++) {
    alert(childNodes[i]);
}
```

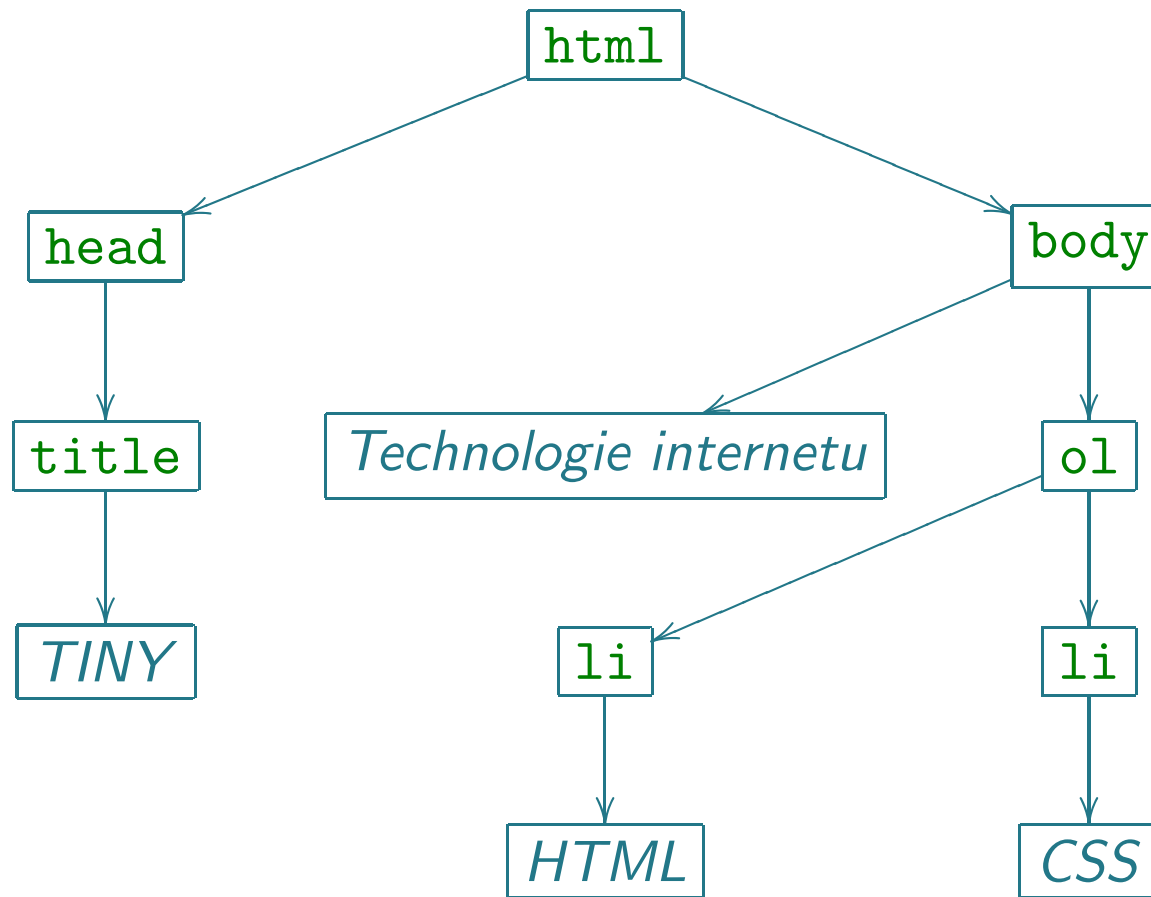
- ✓ DEMO
- ✓ IE < 9 dodaje do `children` węzły-komentarze (nie zgodnie ze standardem)
- ✓ tylko `children`

# Nawigacja w pionie

- BOM
- DOM
- Nawigacja
- Wierzchołek
- Węzły dziecięce
- Nawigacja w pionie
- Tabele
- Tylko elementy
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

- ✓ każdy element ma właściwości:
  - ✗ `firstChild` — pierwsze „dziecko”
  - ✗ `lastChild` — ostatnie „dziecko”
  - ✗ `parentNode` — „rodzic”
  - ✗ `previousSibling`, `nextSibling`, — wierzchołki siostrzane
- ✓ jeżeli odpowiedniego wierzchołka nie ma, właściwość jest ustawiona w `null`

# Nawigacja. Przykład



BOM

DOM

Nawigacja

Wierzchołek

Węzły dziecięce

Nawigacja w pionie

Tabele

Tylko elementy

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne



# Nawigacja. Uwagi

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Wierzchołek](#)
- [Węzły dziecięce](#)
- [Nawigacja w pionie](#)
- [Tabele](#)
- [Tylko elementy](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

- ✓ `firstChild`, `lastChild` mogą być węzłami tekstowymi
- ✓ węzła może nie być na moment wykonania skryptu
- ✓ wszystkie referencje są tylko do odczytu
- ✓ przy zmianie DOM referencje są odnawiane automatycznie

# Nawigacja. Tabele

BOM

DOM

Nawigacja

Wierzchołek

Węzły dziecięce

Nawigacja w pionie

Tabele

Tylko elementy

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

✓ `table`

✗ `table.rows` — lista wierszy tabeli

✗ `table.caption/tHead/tFoot` — referencje na elementy tabeli `caption`, `thead`, `tfoot`

✗ `table.tBodies` — lista elementów tabeli `tbody`

✓ `thead/tfoot/tbody`

✗ `tbody.rows` — lista wierszy (`tr`)

✓ `tr`

✗ `tr.cells` — lista komórek (`td/th`)

✗ `tr.sectionRowIndex` — numer wiersza w bieżącym rozdziale (`thead/tbody`)

✗ `tr.rowIndex` — numer wiersza w tabeli

# Nawigacja. Tabele, `td/th`

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Wierzchołek](#)
- [Węzły dziecięce](#)
- [Nawigacja w pionie](#)
- [Tabele](#)
- [Tylko elementy](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

✓ `td/th`

✗ `td.cellIndex` — numer komórki w wierszu

# Tabele. Przykład

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Wierzchołek](#)

[Węzły dziecięce](#)

[Nawigacja w pionie](#)

[Tabele](#)

[Tylko elementy](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

```
<table>
  <tr> <td>raz</td> <td>dwa</td>      </tr>
  <tr> <td>trzy</td> <td>cztery</td> </tr>
</table>
<script>
var table = document.body.children[0];
alert( table.rows[0].cells[0].innerHTML )
</script>
```

✓ DEMO

# Nawigacja. Tylko elementy

BOM

DOM

Nawigacja

Wierzchołek

Węzły dziecięce

Nawigacja w pionie

Tabele

Tylko elementy

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ `childElementCount` — liczba dziecięcych elementów (`==children.length`)
- ✓ `firstElementChild` — pierwszy dziecięcy element (`==children[0]`)
- ✓ `lastElementChild` — ostatni dziecięcy element (`==children[children.length-1]`)
- ✓ `nextElementSibling` — prawy siostrzany element (nie tekstowy węzeł)
- ✓ `previousElementSibling` — lewy siostrzany element
- ✓ `parentElement` — element rodzicielski (dla `document.documentElement` jest `null`)

# Przykład

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Wierzchołek](#)

[Węzły dziecięce](#)

[Nawigacja w pionie](#)

[Tabele](#)

[Tylko elementy](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

```
<body>
firstElementChild: <div>...</div>
    <!-- komentarz -->
lastElementChild: <span>...</span>
    <script>
alert(document.body.\
firstElementChild.nextElementSibling);
    </script>
</body>
```

✓ DEMO

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
po id
po css
po znaczniku
po name
po className
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

# Dostęp

# Dostęp do elementu po id

- BOM
- DOM
- Nawigacja
- Dostęp
  - po id**
  - po css
  - po znaczniku
  - po name
  - po className
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

✓ `getElementById(id)` element o (unikatowym) `id`

✓ Przykład

```
var b = document.getElementById('b52')  
b.style.color = 'navy'
```

✓ Demo

✓ metoda działa bardzo szybko, przeglądarki mają *zahashowane* wszystkie elementy, mające `id`



# Id a niejawnne zmienne

- BOM
- DOM
- Nawigacja
- Dostęp
  - po id
  - po css
  - po znaczniku
  - po name
  - po className
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

- ✓ przeglądarki tworzą niejawnne elementy o nazwie `id`  

```
<div id="a">Test</div>  
<script>  
    alert(a)  
</script>
```
- ✓ Demo
- ✓ metoda jest przestarzała, mimo że zgodna ze standardem
- ✓ należy unikać

# Dostęp po `elem.querySelectorAll(css)`

- BOM
- DOM
- Nawigacja
- Dostęp
  - po id
  - po css
  - po znaczniku
  - po name
  - po className
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

✓ wybiera elementy za pomocą selektora **CSS3**



```
var elements = document.  
querySelectorAll('UL > LI:last-child')
```



Demo



Odmiana: `elem.querySelector(css)` — tylko pierwszy element, nie przeszukuje całego drzewa

# Dostęp do elementów po znaczniku

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[po id](#)

[po css](#)

[po znaczniku](#)

[po name](#)

[po className](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ `elem.getElementsByTagName(tag)` tablica elementów o znaczniku `tag`

- ✓ Przykład

```
div0 = document.getElementsByTagName('div')[0];  
li2 = div0.getElementsByTagName('li')[1];
```

- ✓ Demo

# Przykład. Lista potomków

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
  - [po id](#)
  - [po css](#)
  - [po znaczniku](#)
  - [po name](#)
  - [po className](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

```
<div id="d1">  
  <ol id="ol1">  
    <li id="li1">1</li>  
    <li id="li2">2</li>  
  </ol>  
</div>
```

```
var div = document.getElementById('d1')  
var elems = div.getElementsByTagName('*')
```

# Dostęp do elementów po **name**

BOM

DOM

Nawigacja

Dostęp

po id

po css

po znaczniku

po name

po className

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ `getElementsByName(nam)` — lista wszystkich elementów o atrybucie **name** warsości **nam**
- ✓ HTML< 5, IE9- działa tylko dla tych elementów, które zgodnie ze specyfikacją mogą mieć atrybut **name** (**form**, **input**, **a**, **select**, **textarea** i kilka innych)
- ✓ we współczesnym standardzie znacznik nie ma znaczenia
- ✓ Demo

# Dostęp do elementów po `className`

BOM

DOM

Nawigacja

Dostęp

po id

po css

po znaczniku

po name

po className

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ `getElementsByClassName(nam)` — lista wszystkich elementów o atrybucie `class` warsości `nam` — nie działa w `IE < 9`
- ✓ działa poprawnie nawet gby element ma kilka klas
- ✓ może być wywołany z poziomu dokumentu jak i elementu
- ✓ Demo

## O wynikach `getElements*`

- ✓ Wyniki wszystkich DOM-zapytań `getElements*` — dynamiczne kolekcje
  - ✗ zmieniają się na bieżąco
  - ✗ mają indeksy, `length`
  - ✗ nie mają `pop`, `push` i innych właściwości tablic
  - ✗ działają wolniej, niż statyczne tablice
- ✓ wyniki innych zapytań — tablice statyczne

- BOM
- DOM
- Nawigacja
- Dostęp
  - po id
  - po css
  - po znaczniku
  - po name
  - po className
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

## Przykład z `getElements*`

BOM
DOM
Nawigacja
Dostęp
po id
po css
po znaczniku
po name
po className
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

```
<ul id="menu">
  <li>Home</li>
  <li>Forum</li>
  <li>Sklep</li>
</ul>
```

✓ jaki wynik będzie miało zapytanie

```
var lis
  = document.body.getElementsByTagName('li');
```

```
document.body.innerHTML = '';
```

```
alert(lis.length);
```

✓ Rozwiązanie



## Drugi przykład z `getElements*`

BOM

DOM

Nawigacja

Dostęp

po id

po css

po znaczniku

po name

po className

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<ul id="menu">
  <li>Home</li>
  <li>Forum</li>
  <li>Sklep</li>
</ul>
```

✓ jaki wynik będzie miało zapytanie

```
var lis =
  document.getElementById('menu')
    .getElementsByTagName('li');
```

```
document.body.innerHTML = '';
```

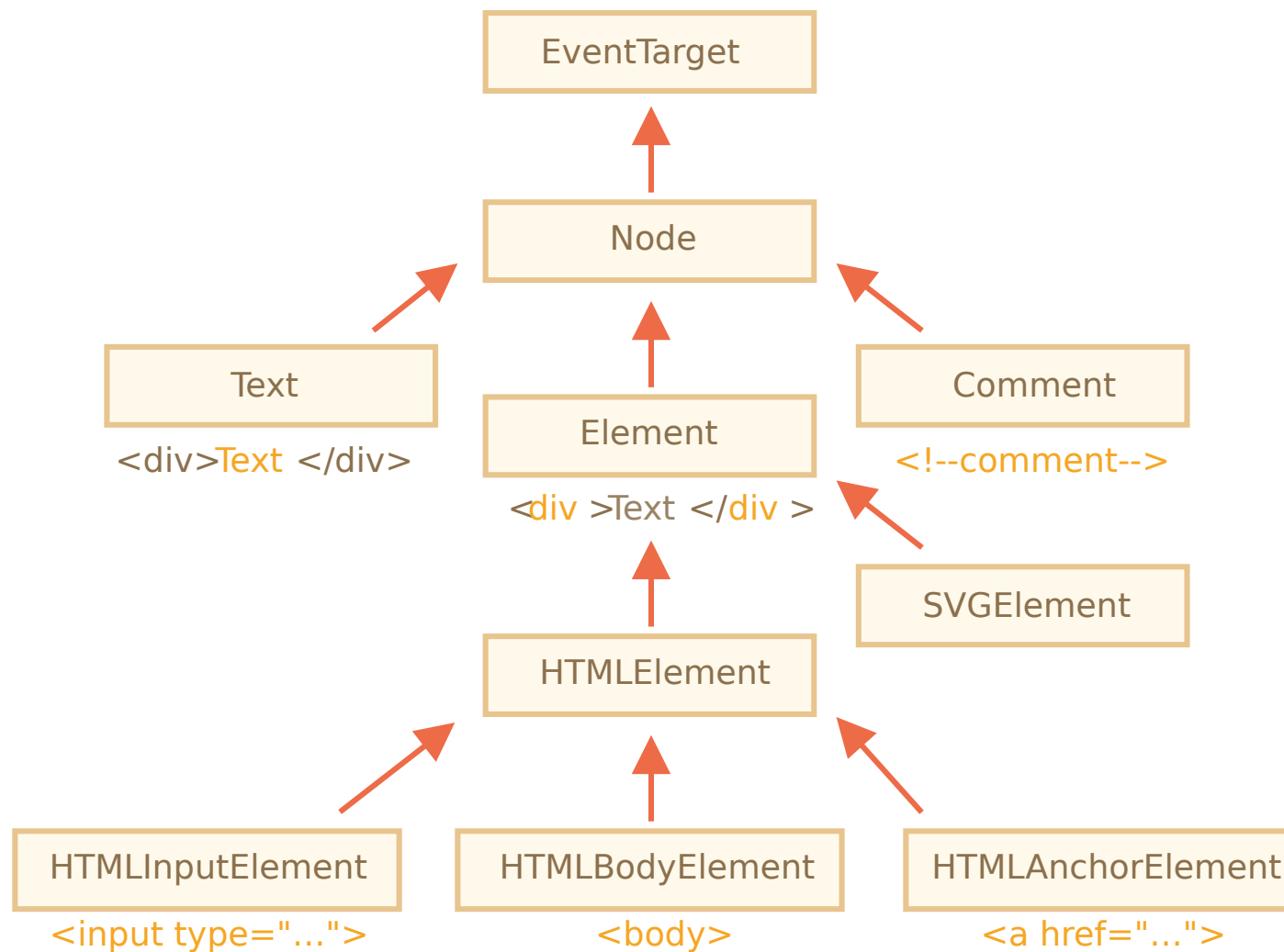
```
alert(lis.length);
```

✓ Rozwiązanie

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)**
  - [Typy węzłów](#)
  - [nodeType](#)
  - [nodeName tagName](#)
  - [innerHTML](#)
  - [outerHTML](#)
  - [nodeValue/data](#)
  - [textContent](#)
  - [hidden](#)
  - [Właściwości użytkownika](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

# Właściwości

# Typy węzłów



# Sprawdzanie typu węzła.

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Typy węzłów</a>
<a href="#">nodeType</a>
<a href="#">nodeName tagName</a>
<a href="#">innerHTML</a>
<a href="#">outerHTML</a>
<a href="#">nodeValue/data</a>
<a href="#">textContent</a>
<a href="#">hidden</a>
<a href="#">Właściwości użytkownika</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

```
alert( document.body.constructor.name ); // HTMLBodyElement
alert( document.body ); // [object HTMLBodyElement]
alert( document.body instanceof HTMLBodyElement ); // true
alert( document.body instanceof HTMLElement ); // true
alert( document.body instanceof Element ); // true
alert( document.body instanceof Node ); // true
alert( document.body instanceof EventTarget ); // true
```

# nodeType

- ✓ numer typu zapisany jest w polu **nodeType**
- ✓ najważniejsze są 1 i 3

```
Node.ELEMENT_NODE == 1
Node.ATTRIBUTE_NODE == 2
Node.TEXT_NODE == 3
Node.CDATA_SECTION_NODE == 4
Node.ENTITY_REFERENCE_NODE == 5
Node.ENTITY_NODE == 6
Node.PROCESSING_INSTRUCTION_NODE == 7
Node.COMMENT_NODE == 8
Node.DOCUMENT_NODE == 9
Node.DOCUMENT_TYPE_NODE == 10
Node.DOCUMENT_FRAGMENT_NODE == 11
Node.NOTATION_NODE == 12
```

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

**nodeType**

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# Typy węzłów. Przykład

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<body>
  <div>Userzy:</div>
  <ul>
    <li>John</li>
    <li>Bob</li>
  </ul>  <!-- komentarz -->
  <script>
    var childNodes = document.body.childNodes;
    for (var i=0; i<childNodes.length; i++) {
      if (childNodes[i].nodeType == 1){
        alert(childNodes[i]);
      }
    }
  </script>
</body>
```

✓ DEMO

# Typy węzłów. Zadanie

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
  <script>
```

```
    alert(document.body.firstChild.nodeType);
```

```
  </script>
```

```
</body>
```

```
</html>
```

✓ Jaki będzie wynik?

✓ DEMO

# Typy węzłów. Zadanie prim

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
  <script>
```

```
    alert(document.body.lastChild.nodeType);
```

```
  </script>
```

```
</body>
```

```
</html>
```

✓ Jaki będzie wynik?

✓ DEMO



## nodeName, tagName

- ✓ nodeName oraz tagName zawiera nazwę elementu (DUŻE LITERY) `alert(document.body.tagName)` // BODY
- ✗ wyjątek: jeżeli przeglądarka jest w trybie XML
  - ✓ przez XMLHttpRequest (AJAX)
  - ✓ przy nagłówku HTTP  
`Content-Type: application/xml+xhtml`
- ✗ w trybie XML będzie dokładnie tak, jak jest w dokumencie

BOM
DOM
Nawigacja
Dostęp
Właściwości
Typy węzłów
nodeName
tagName
innerHTML
outerHTML
nodeValue/data
textContent
hidden
Właściwości użytkownika
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

# Różnica `nodeName` a `tagName`

- ✓ `nodeName` określone dla węzłów
- ✓ `tagName` tylko dla elementów
- ✓ jeżeli węzeł nie jest elementem, to `nodeName` ma specjalną wartość, zaś  
`tagName == undefined`  
(oprócz komentarzy w IE < 9)
- ✓ przy pracy tylko z elementami należy korzystać z `tagName`

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Typy węzłów</a>
<a href="#">nodeType</a>
<a href="#">nodeName tagName</a>
<a href="#">innerHTML</a>
<a href="#">outerHTML</a>
<a href="#">nodeValue/data</a>
<a href="#">textContent</a>
<a href="#">hidden</a>
<a href="#">Właściwości użytkownika</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

## Przykład nodeName, tagName

```
<body><!-- comment -->
<script>
    // #comment
    alert(document.body.firstChild.nodeName);
    // undefined
    alert(document.body.firstChild.tagName);

    // #document
    alert(document.nodeName);
    // undefined
    alert(document.tagName);
</script>
</body>
```

✓ DEMO

BOM
DOM
Nawigacja
Dostęp
Właściwości
Typy węzłów
nodeType
nodeName tagName
innerHTML
outerHTML
nodeValue/data
textContent
hidden
Właściwości użytkownika
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

# innerHTML

- ✓ `innerHTML` opisane w specyfikacji HTML 5—embedded content
- ✓ pozwala na otrzymanie zawartości elementu HTML jako obiektu `String`
- ✓ `innerHTML` jest dostępne do odczytu i do modyfikacji

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Typy węzłów](#)

[nodeType](#)

[nodeName tagName](#)

[innerHTML](#)

[outerHTML](#)

[nodeValue/data](#)

[textContent](#)

[hidden](#)

[Właściwości](#)

[użytkownika](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

# Przykład z `innerHTML`

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Typy węzłów
- nodeType
- nodeName tagName
- innerHTML**
- outerHTML
- nodeValue/data
- textContent
- hidden
- Właściwości użytkownika
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

```
<body>
  <p>Akapit</p>
  <div>Div</div>
  <script>
    alert( document.body.innerHTML );
    document.body.innerHTML = 'Nowe BODY!';
  </script>
</body>
```

✓ DEMO

# Dodawanie do `innerHTML`

- ✓ dodawanie `innerHTML+=...` zastępuje zawartość
- ✗ usuwa się stara wartość, wstawia się nowa  
`chatDiv.innerHTML +=`  
`"<div>Hi<img src='smile.gif' />!</div>";`  
`chatDiv.innerHTML += "How are you?";`
- ✗ obrazek zostanie przeładowany

BOM
DOM
Nawigacja
Dostęp
Właściwości
Typy węzłów
nodeType
nodeName tagName
<b>innerHTML</b>
outerHTML
nodeValue/data
textContent
hidden
Właściwości użytkownika
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

# innerHTML a skrypty

- ✓ skrypt, osadzony w `innerHTML` nie będzie wykonany

```
<div id="my"></div>
<script>
    var elem = document.getElementById('my');
    elem.innerHTML = 'TEST<script>alert(1);
                    </scr'+ 'ipt>';
</script>
```

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# outerHTML

- ✓ Cały HTML elementu

```
<div>Hello, <b>World!</b></div>
```

```
<script>  
    var div = document.body.children[0];  
    alert( div.outerHTML );  
    // <div>Hello, <b>World!</b></div>  
</script>
```

- ✓ DEMO

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne



## outerHTML. Uwaga

- ✓ `outerHTML` wykorzystywany jest tylko do odczytu
- ✓ Zmiana powoduje zmianę elementu i zastępowanie przez nowy
  - ✗ w zmiennej zostaje stary element

```
<div>Hello, World!</div>
```

```
<script>
```

```
    var div = document.body.children[0];
```

```
    div.outerHTML = '<p>Nowy element!</p>';
```

```
    alert( div.outerHTML ); // <div>Hello, World!</div>
```

```
</script>
```

- ✓ DEMO

BOM
DOM
Nawigacja
Dostęp
Właściwości
Typy węzłów
nodeType
nodeName tagName
innerHTML
outerHTML
nodeValue/data
textContent
hidden
Właściwości użytkownika
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

## nodeValue/data

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ innerHTML mają tylko elementy
- ✓ pozostałe węzły mają **nodeValue** i **data**

```
<body>
  Hi  <!-- Comment -->
<script>
for (var i=0;
      i<document.body.childNodes.length; i++) {
  alert(document.body.childNodes[i].nodeValue);
  alert(document.body.childNodes[i].data);
}
</script>
</body>
```

- ✓ DEMO

# textContent

- ✓ `textContent` zawiera sam tekst elementu (bez znaczników)

```
<div>
  <h1>Niedziela</h1>
  <p>Przelotny deszcz</p>
</div>
```

```
<script>
  var news = document.body.children[0];
  alert( news.textContent );
</script>
```

- ✓ DEMO

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

**textContent**

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# Zapis do `textContent`

- ✓ tekst wstawia się jako tekst, bez tworzenie elementów ze znaczników

```
<div></div>
```

```
<div></div>
```

```
<script>
```

```
    var name = "<b>Aleksander Kowalski</b>";
```

```
    document.body.children[0].innerHTML = name;
```

```
    document.body.children[1].textContent = name;
```

```
</script>
```

- ✓ DEMO

BOM
DOM
Nawigacja
Dostęp
Właściwości
Typy węzłów
nodeType
nodeName tagName
innerHTML
outerHTML
nodeValue/data
<b>textContent</b>
hidden
Właściwości użytkownika
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

# hidden

- ✓ Właściwość `hidden` określa widoczność elementu
- ✓ HTML5, IE11+

```
<div>Tekst</div>
```

```
<div hidden>Z atrybutem <code>hidden</code></div>
```

```
<div>Z właściwością hidden</div>
```

```
<script>
```

```
    var lastDiv = document.body.children[2];
```

```
    lastDiv.hidden = true;
```

```
</script>
```

- ✓ DEMO

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości

użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# Właściwości użytkownika

- ✓ Każdy węzeł DOM jest obiektem JavaScript
- ✓ Można dodawać dowolne właściwości

```
var elem = document.getElementById('Element')
elem.ocena = 6
elem.autor = {
  name: 'Aleksander',
  surname: 'Denisiuk'
}
alert(elem.ocena)
alert(elem.autor.name)
```

- ✓ Demo

BOM

DOM

Nawigacja

Dostęp

Właściwości

Typy węzłów

nodeType

nodeName tagName

innerHTML

outerHTML

nodeValue/data

textContent

hidden

Właściwości  
użytkownika

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# Nowe funkcje

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Typy węzłów](#)
- [nodeType](#)
- [nodeName tagName](#)
- [innerHTML](#)
- [outerHTML](#)
- [nodeValue/data](#)
- [textContent](#)
- [hidden](#)
- [Właściwości użytkownika](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

```
document.body.sayHi = function() {  
    alert(this.nodeName);  
}
```

```
document.body.sayHi();
```

✓ Demo

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)**
  - [DOM-atrybuty](#)
  - [Synchronizacja](#)
  - [Atrybuty niestandardowe](#)
  - [dataset](#)
  - [Polyfill dla hidden](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

# Atrybuty



# DOM-atrybuty

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atrybuty

Synchronizacja

Atrybuty

niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ Każdy element DOM może mieć dowolne (HTML i niestandardowe) atrybuty

```
<div id="MyElement" class="big" alpha="omega">
```

- ✓ Funkcje do atrybutów:
  - ✗ `setAttribute(name, value)`
  - ✗ `getAttribute(name)`
  - ✗ `hasAttribute(name)`
  - ✗ `removeAttribute(name)`

# DOM-atrybuty

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
DOM-atrybuty
Synchronizacja
Atrybuty niestandardowe
dataset
Polyfill dla hidden
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ mogą być tylko **String**
- ✓ nazwa atrybutu jest *case insensitive*
- ✓ otworzone w HTML, w tym w **innerHTML**
- ✓ wszystkie atrybuty można uzyskać przez właściwość **attributes**, która zawiera listę obiektów klasy **Attr** o właściwościach **name** i **value**

# Atrybuty. Przykład

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atorybuty

Synchronizacja

Atrybuty

niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<div about="Elephant" class="smiling"></div>
<script>
  var div = document.body.children[0];
  alert( div.getAttribute('ABOUT') ); // (1)

  div.setAttribute('Test', 123); // (2)
  alert( document.body.innerHTML ); // (3)

  var attrs = div.attributes; // (4)
  for (var i=0; i<attrs.length; i++) {
    alert(attrs[i].name + " = " + attrs[i].value);
  }
</script>
```

✓ Demo

# Synchronizacja

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">DOM-atributy</a>
<a href="#">Synchronizacja</a>
<a href="#">Atrybuty niestandardowe</a>
<a href="#">dataset</a>
<a href="#">Polyfill dla hidden</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

- ✓ Każdy element oprócz *podstawowych standardowych* atrybutów ma identyczne właściwości
- ✓ Każda przeglądarka synchronizuje wartości *standardowych* atrybutów i właściwości: zmiana jednego powoduje zmianę drugiego

# Synchronizacja id

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [DOM-atrybuty](#)
- [Synchronizacja](#)
- [Atrybuty niestandardowe](#)
- [dataset](#)
- [Polyfill dla hidden](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

```
<script>  
    document.body.setAttribute('id', 'la-la-la');  
    alert(document.body.id); // la-la-la  
</script>
```

✓ Demo

# Synchronizacja href

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atrybuty

Synchronizacja

Atrybuty

niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<a href="#"></a>
```

```
<script>
```

```
var a = document.body.children[0];
```

```
a.href = '/';
```

```
alert('attribute:' + a.getAttribute('href'));
```

```
alert('property:' + a.href);
```

```
</script>
```

✓ właściwość — pełny URL

✓ Demo

# Synchronizacja checked

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- DOM-atrybuty
- Synchronizacja**
- Atrybuty niestandardowe
- dataset
- Polyfill dla hidden
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

```
<input type="checkbox" checked>
<script>
    var input = document.body.children[0];

    alert(input.checked);

    alert(input.getAttribute('checked'));
</script>
```

✓ [Demo](#)

# Synchronizacja value

- ✓ z atrybutu we właściwość:

```
<body>
  <input type="text" value="markup">
  <script>
    var input = document.body.children[0];

    input.setAttribute('value', 'new');

    alert( input.value ); // 'new'
  </script>
</body>
```

- ✓ Demo

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- DOM-atrybuty
- Synchronizacja
- Atrybuty niestandardowe
- dataset
- Polyfill dla hidden
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne



# Niesynchronizacja value

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atrybuty

Synchronizacja

Atrybuty

niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ z właściwości w atrybut:

```
<input type="text" value="markup">
<script>
  var input = document.body.children[0];

  input.value = 'new';
  alert(
    input.getAttribute('value')); // 'markup'
</script>
```

- ✓ Demo
- ✓ atrybut **value** przechowuje pierwotną wartość nawet jeżeli użytkownik ją zmieni z klawiatury
- ✓ zawsze można sprawdzić, czy wartość się zmieniła!

# Synchronizacja class

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atrybuty

Synchronizacja

Atrybuty

niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ atrybutowi **class** odpowiada właściwość **className**

```
<body>
  <script>
    document.body.setAttribute('class',
                                'big red bloom');

    alert( document.body.className );
  </script>
</body>
```

- ✓ Demo
- ✓ zawsze korzystać z właściwości, nie z atrybutów

# Atrybuty niestandardowe

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atrybuty

Synchronizacja

Atrybuty  
niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ każdy element DOM ma atrybuty standardowe, którym odpowiadają właściwości
- ✓ dla atrybutów niestandardowych właściwości się nie tworzy

```
<div id='elem; about="Elephant"></div>
```

```
<script>  
    var div = document.body.children[0];  
    alert( div.id ); // elem  
    alert( div.about ); // undefined  
</script>
```

- ✓ Demo

# Atrybuty niestandardowe

- ✓ standardowe — to są tylko te atrybuty, które są opisane w standardzie właśnie dla tego elementu

```
<img href="test">  
<a alt="test"></a>
```

```
<script>  
    var img = document.body.children[0];  
    var link = document.body.children[1];  
    alert( img.href ); // undefined  
    alert( link.alt ); // undefined  
</script>
```

- ✓ Demo

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atributy

Synchronizacja

Atrybuty  
niestandardowe

dataset

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

# Przykład. HTML

- ✓ Korzystamy z niestandardowych atrybutów, żeby pokazać status zamówienia

```
<div class="order" order-state="new">
```

Nowe zamówienie.

```
</div>
```

```
<div class="order" order-state="pending">
```

Zamówienie oczekujące.

```
</div>
```

```
<div class="order" order-state="canceled">
```

Zamówienie wycofane.

```
</div>
```

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
DOM-atributy
Synchronizacja
Atrybuty niestandardowe
dataset
Polyfill dla hidden
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne

# Przykład. CSS

- ✓ Korzystamy z niestandardowych atrybutów, żeby pokazać status zamówienia

```
.order[order-state="new"] {  
    color: green;  
}
```

```
.order[order-state="pending"] {  
    color: blue;  
}
```

```
.order[order-state="canceled"] {  
    color: red;  
}
```

- ✓ Demo

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- DOM-atrybuty
- Synchronizacja
- Atrybuty niestandardowe**
- dataset
- Polyfill dla hidden
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne

# Przykład. Zmiana statusu

```
div.setAttribute('order-state', 'canceled');
```

- ✓ zamieniamy wszystkie na **canceled**
- ✓ Demo

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [DOM-atrybuty](#)
- [Synchronizacja](#)
- [Atrybuty niestandardowe](#)
- [dataset](#)
- [Polyfill dla hidden](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

# Atrybuty `data-*`

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atrybuty

Synchronizacja

Atrybuty

niestandardowe

**dataset**

Polyfill dla hidden

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ Standard HTML5 dla wszystkich elementów deklaruje jako standardowe, atrybuty, zaczynające się od `data-`
- ✓ Z poziomu JavaScript, można odwołać się do tych atrybutów poprzez właściwość `dataset` (IE11+)

```
<div data-about="Elephant" data-user-location="zoo">
```

Kierownik ogrodu zoologicznego okazał się karierowiczem

```
</div>
```

```
<script>
```

```
var elem=document.body.children[0];
```

```
alert( elem.dataset.about ); // Elephant
```

```
alert( elem.dataset.About ); // undefined
```

```
alert( elem.dataset.userLocation ); // zoo
```

```
</script>
```

- ✓ Demo



# Przykład. Zmiana statusu

```
div.dataset.orderState = 'canceled'
```

✓ Demo

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [DOM-atrybuty](#)
- [Synchronizacja](#)
- [Atrybuty niestandardowe](#)
- [dataset](#)
- [Polyfill dla hidden](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

# Polyfill dla atrybutu `hidden`

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

DOM-atorybuty

Synchronizacja

Atrybuty  
niestandardowe

dataset

Polyfill dla `hidden`

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ W IE11 nie jest obsługiwany atrybut `hidden`
- ✓ Można poprawić przez css

```
[hidden] { display: none }
```
- ✓ ale nie będzie działała synchronizacja, w szczególności

```
elem.hidden = true;
```

nie da porządanego efektu

# Rozwiązanie

[BOM](#)[DOM](#)[Nawigacja](#)[Dostęp](#)[Właściwości](#)[Atrybuty](#)[DOM-atrybuty](#)[Synchronizacja](#)[Atrybuty](#)[niestandardowe](#)[dataset](#)[Polyfill dla hidden](#)[Modyfikacja](#)[Style](#)[Wymiary elementów](#)[Wymiary okna](#)[Współrzędne](#)

```
if (document.documentElement.hidden === undefined) {  
  Object.defineProperty(Element.prototype, "hidden", {  
    set: function(value) {  
      this.setAttribute('hidden', value);  
    },  
    get: function() {  
      return this.getAttribute('hidden');  
    }  
  });  
}
```



Demo

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)**
  - [Zmiana DOM](#)
  - [Tworzenie](#)
  - [Dodawanie](#)
  - [Usuwanie](#)
  - [Przemieszczenie](#)
  - [Klonowanie](#)
  - [DocumentFragment](#)
  - [Stare metody](#)
  - [document.write](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)

# Modyfikacja

# Zmiana DOM

- ✓ Zmiana DOM — sposób na tworzenie dynamicznych stron

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Zmiana DOM](#)

[Tworzenie](#)

[Dodawanie](#)

[Usuwanie](#)

[Przemieszczenie](#)

[Klonowanie](#)

[DocumentFragment](#)

[Stare metody](#)

[document.write](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

# Tworzenie nowych węzłów

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

**Tworzenie**

Dodawanie

Usuwanie

Przemieszczenie

Klonowanie

DocumentFragment

Stare metody

document.write

Style

Wymiary elementów

Wymiary okna

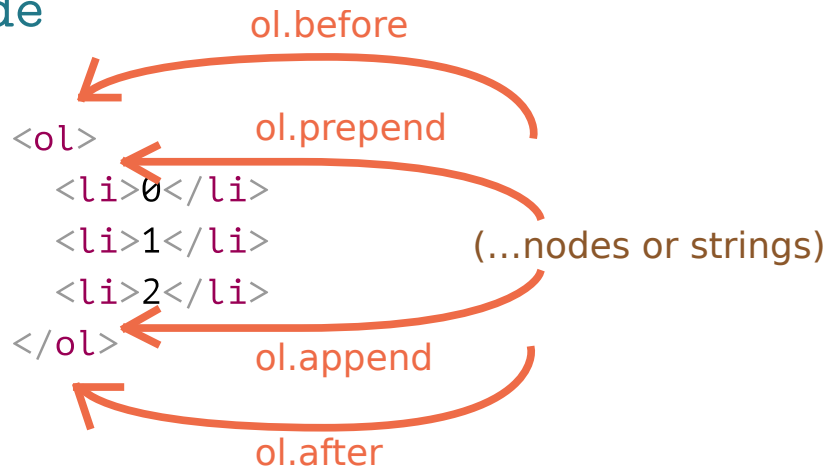
Współrzędne

- ✓ `document.createElement(tag)`
- ✓ `document.createTextNode(text)`

```
let div = document.createElement('div');  
div.className = 'alert';  
div.id = 'alert22';  
div.innerHTML = `Uwaga!</strong>  
                To jest bardzo ważny komunikat.`;
```

# Dodawanie węzłów

- ✓ `node.append(...nodes or strings)` — dodać na końcu node
- ✓ `node.prepend(...nodes or strings)` — dodać na początku node
- ✓ `node.before(...nodes or strings)` — wstawić przed node
- ✓ `node.after(...nodes or strings)` — wstawić po node
- ✓ `node.replaceWith(...nodes or strings)` — zastąpić node



# Przykład

```
<div id="div">World!</div>
<script>
  div.before('<p>Hello</p>',
            document.createElement('hr'));
</script>
```

✓ Demo

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

**Dodawanie**

Usuwanie

Przemieszczenie

Klonowanie

DocumentFragment

Stare metody

document.write

Style

Wymiary elementów

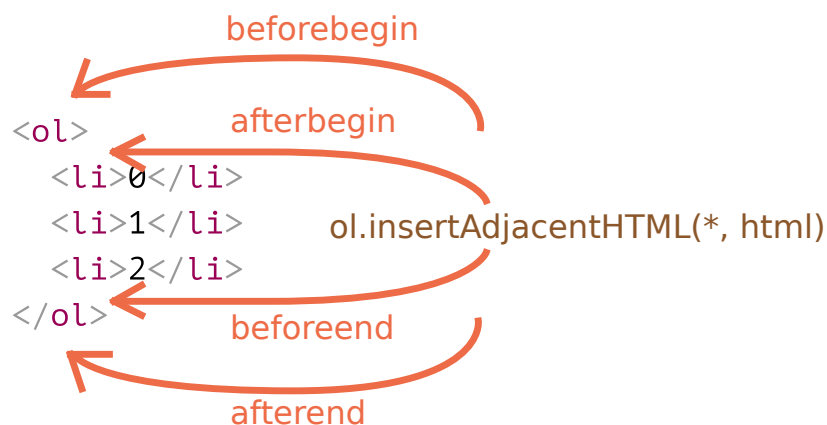
Wymiary okna

Współrzędne



# insertAdjacentHTML

- ✓ `elem.insertAdjacentHTML(where, text);` — wstawka HTML
- ✓ **where:**
  - ✗ `beforeBegin` — przed element
  - ✗ `afterBegin` — wewnątrz jako pierwszy
  - ✗ `beforeEnd` — wewnątrz jako ostatni
  - ✗ `afterEnd` — po elemencie



## Przykład insertAdjacentHTML

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Zmiana DOM
Tworzenie
<b>Dodawanie</b>
Usuwanie
Przemieszczenie
Klonowanie
DocumentFragment
Stare metody
document.write
Style
Wymiary elementów
Wymiary okna
Współrzędne

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>5</li>
</ul>

<script>
let ul = document.body.children[0];
let li5 = ul.children[2];

li5.insertAdjacentHTML("beforeBegin",
  "<li>3</li><li>4</li>");
</script>
```

✓ Demo

# Odmiany `insertAdjacentHTML`

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Zmiana DOM</a>
<a href="#">Tworzenie</a>
<a href="#">Dodawanie</a>
<a href="#">Usuwanie</a>
<a href="#">Przemieszczenie</a>
<a href="#">Klonowanie</a>
<a href="#">DocumentFragment</a>
<a href="#">Stare metody</a>
<a href="#">document.write</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

- ✓ `elem.insertAdjacentText(where, text);` — wstawka tekstu
- ✓ `elem.insertAdjacentElement(where, element);` — wstawka elementu

# Usuwanie węzłów

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

Dodawanie

Usuwanie

Przemieszczenie

Klonowanie

DocumentFragment

Stare metody

document.write

Style

Wymiary elementów

Wymiary okna

Współrzędne

✓ `node.remove();`

`<script>`

```
let div = document.createElement('div');
div.className = "message";
div.innerHTML = `

# Uwaga!</h1> <p class='content'>Ważna informacja</p>`;


```

```
document.body.append(div);
setTimeout(() => div.remove(), 3000);
```

`</script>`

✓ Demo

# Przemieszczanie węzłów

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

Dodawanie

Usuwanie

Przemieszczanie

Klonowanie

DocumentFragment

Stare metody

document.write

Style

Wymiary elementów

Wymiary okna

Współrzędne

```
<div>First div</div>
```

```
<div>Last div</div>
```

```
<script>
```

```
  let first = document.body.children[0];
```

```
  let last = document.body.children[1];
```

```
  last.after(first);
```

```
</script>
```

✓ nie ma potrzeby na `remove()`, ponieważ wszystkie metody wstawki usuwają element ze starego miejsca

✓ Demo

# Klonowanie węzłów

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

Dodawanie

Usuwanie

Przemieszczenie

Klonowanie

DocumentFragment

Stare metody

document.write

Style

Wymiary elementów

Wymiary okna

Współrzędne

✓ `elem.cloneNode(true)` — klonowanie elementu razem z atrybutami oraz z podrzędnymi elementami

✓ `elem.cloneNode(false)` — bez podrzędnych

```
<div class="alert" id="div">
```

```
  <strong>Uwaga!</strong>
```

```
  Ważny komunikat.
```

```
</div>
```

```
<script>
```

```
  let div2 = div.cloneNode(true);
```

```
  div2.querySelector('strong').textContent = 'Ciao!';
```

```
  div.after(div2);
```

```
</script>
```

✓ Demo

# DocumentFragment

- ✓ Jest DOM-obiektem, podobnym do DOM-węzła
- ✓ Wstawka dużej ilości elementów
- ✓ `document.createDocumentFragment()` bądź `new DocumentFragment()`
- ✓ jeżeli wstawiamy `DocumentFragment` do DOM, to wstawia się tylko jego dziecięce węzły

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

Dodawanie

Usuwanie

Przemieszczenie

Klonowanie

DocumentFragment

Stare metody

`document.write`

Style

Wymiary elementów

Wymiary okna

Współrzędne

## Przykład `DocumentFragment`

```
<ul id="ul"></ul>
<script>
function getListContent() {
    let fragment = new DocumentFragment();
    for(let i=1; i<=3; i++) {
        let li = document.createElement('li');
        li.append(i);
        fragment.append(li);
    }
    return fragment;
}
ul.append(getListContent());
</script>
```

✓ [Zobacz](#)

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

Dodawanie

Usuwanie

Przemieszczenie

Klonowanie

**DocumentFragment**

Stare metody

document.write

Style

Wymiary elementów

Wymiary okna

Współrzędne



## Przykład bez DocumentFragment

```
<ul id="ul"></ul>
<script>
function getListContent() {
  let result = [];
  for(let i=1; i<=3; i++) {
    let li = document.createElement('li');
    li.append(i);
    result.push(li);
  }
  return result;
}
ul.append(...getListContent());
</script>
```

✓ [Zobacz](#)

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Zmiana DOM

Tworzenie

Dodawanie

Usuwanie

Przemieszczenie

Klonowanie

DocumentFragment

Stare metody

document.write

Style

Wymiary elementów

Wymiary okna

Współrzędne

# Stare metody modyfikacji

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Zmiana DOM](#)

[Tworzenie](#)

[Dodawanie](#)

[Usuwanie](#)

[Przemieszczenie](#)

[Klonowanie](#)

[DocumentFragment](#)

[Stare metody](#)

[document.write](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ `parentElem.appendChild(newNode)`
- ✓ `parentElem.insertBefore(newNode, oldNode)`
- ✓ `parentElem.replaceChild(node, oldChild)`
- ✓ `parentElem.removeChild(node)`

# document.write

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Zmiana DOM
Tworzenie
Dodawanie
Usuwanie
Przemieszczenie
Klonowanie
DocumentFragment
Stare metody
<b>document.write</b>
Style
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ bardzo stara metoda
- ✓ `document.write(str)` działa poprawnie tylko póki dokument nie jest załadowany
- ✓ `str` dodaje się do dokumentu, jak by było zapisane w kodzie HTML
- ✓ bardzo szybko
- ✓ nie ma ograniczeń na `str`, można wprowadzać elementy częściowo, nie przestrzegając reguł HTML
- ✓ istnieje odmiana, `document.writeln(str)`

# Przykłady `document.write`

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Zmiana DOM](#)

[Tworzenie](#)

[Dodawanie](#)

[Usuwanie](#)

[Przemieszczenie](#)

[Klonowanie](#)

[DocumentFragment](#)

[Stare metody](#)

[document.write](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Współrzędne](#)

```
<script>
    document.write('<em>Hello, World!</i>');
</script>

<script>
    document.write('<style> td { color: #F40 } \
</style>');
</script>
<table>
    <tr>
        <script> document.write('<td>') </script>
        tekst TD.
        <script> document.write('</td>') </script>
    </tr>
</table>
```

## document.write po załadowaniu strony

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Zmiana DOM
Tworzenie
Dodawanie
Usuwanie
Przemieszczenie
Klonowanie
DocumentFragment
Stare metody
<b>document.write</b>
Style
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ po załadowaniu strony dokument staje się *zamkniętym*
- ✓ próba zapisywania przez `document.write` otwiera dokument na nowo
- ✓ cała zawartość zostaje zniszczona
- ✓ Demo
- ✓ do modyfikacji zamkniętego dokumentu należy korzystać z metod DOM

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<b><a href="#">Style</a></b>
<a href="#">className</a>
<a href="#">classList</a>
<a href="#">style</a>
<a href="#">Informacja o stylach</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Współrzędne</a>

# Style

## className

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

className

classList

style

Informacja o stylach

Wymiary elementów

Wymiary okna

Współrzędne

✓ className odpowiada atrybutowi html **class**

✓ Przykład

```
<body class="class1 class2">
<script>
    alert(document.body.className);
    document.body.className += ' class3';
    alert(document.body.className);
</script>
</body>
```

✓ Zobaczć

# Właściwość `classList`

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

className

**classList**

style

Informacja o stylach

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ Obiekt `element.classList`
- ✓ `elem.classList.contains(cls)`
- ✓ `elem.classList.add/remove(cls)`
- ✓ `elem.classList.toggle(cls)`
- ✓ jest iterowalny

```
<body class="main page">
  <script>
    for (let name of document.body.classList) {
      alert(name);
    }
  </script>
</body>
```



# Właściwość `style`

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

className

classList

**style**

Informacja o stylach

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ `element.style` dostępne jest do odczytu jak i do zapisu
- ✓ można zmienić większość właściwości css
  - ✗ `elem.style.width='100px'`
- ✓ właściwości wielocłonowe
  - ✗ `background-color` ⇒ `backgroundColor`
  - ✗ `z-index` ⇒ `zIndex`
  - ✗ `border-left-width` ⇒ `borderLeftWidth`
  - ✗ `-moz-border-radius` ⇒ `MozBorderRadius`
  - ✗ `-webkit-border-radius` ⇒ `WebkitBorderRadius`
- ✓ `float` ⇒ `cssFloat`

# Style

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

className

classList

style

Informacja o stylach

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ Wartość `elem.style` zastępuje wartość z css
- ✓ Żeby przywrócić wartość z css, wystarczy nadać odpowiedniej właściwości `elem.style` wartość *pustą*

✗ `elem.style.width=' '`

- ✓ Demo

```
document.body.style.backgroundColor  
    = prompt('background color?', 'green')
```

# style.cssText

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
className
classList
style
Informacja o stylach
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ `style.cssText` pozwala na czytanie/modyfikację całej deklaracji
- ✓ brak ograniczeń na właściwości: przeglądarka zmienia tylko znane jej  

```
div.style.cssText="color: red !important; \
background-color: yellow; \
width: 100px; text-align: center; \
blabla: 5; ";
```
- ✓ to jest jedyny sposób dodać `!important`
- ✓ przy ustawieniu `style.cssText` zastępuje się *wszystkie* istniejące właściwości
- ✓ Demo

# Informacja o stylach

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
className
classList
style
Informacja o stylach
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ jeżeli styl został zdefiniowany w atrybucie **style**

```
<body style="color:red">
```

Czerwony

```
<script>
```

```
    alert(document.body.style.color); // red
```

```
</script>
```

```
</body>
```

- ✓ Demo

# Informacja o stylach z css

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
className
classList
style
Informacja o stylach
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ jeżeli styl został zdefiniowany w arkuszu css, to javascript o nim nie wie

- ✓ zwłaszcza o właściwościach złożonych

```
<style>
  body { margin: 10px }
</style>
<body>
  <script>
    alert(document.body.style.marginTop)
  </script>
</body>
```

- ✓ Demo

# Złożone właściwości, określone javascriptem

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

className

classList

style

Informacja o stylach

Wymiary elementów

Wymiary okna

Współrzędne

- ✓ właściwości złożone, określone javascriptem, zostają *rospakowywane*

```
<body>
```

```
<script>
```

```
    document.body.style.margin = '20px';
```

```
    alert(document.body.style.marginTop); // 20px!
```

```
</script>
```

```
</body>
```

- ✓ Demo

# Interpretacja właściwości

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
className
classList
style
Informacja o stylach
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ przeglądarka interpretuje właściwości stylu
- ✓ format przedstawienia stylu może zostać zmieniony

```
<body style="color:#abc">
  <script>
    alert(document.body.style.color);
  </script>
</body>
```

- ✓ Demo

# Bieżące właściwości stylów

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

className

classList

style

Informacja o stylach

Wymiary elementów

Wymiary okna

Współrzędne

✓ `window.getComputedStyle(element, pseudo)`

✗ `element` — element

✗ `pseudo` — pseudoselektor (lub `null`)

```
<style>body{ margin: 10px}</style>
```

```
<body>  <script>
```

```
  alert(document.body.style.marginTop)
```

```
  let computedStyle =
```

```
    getComputedStyle(document.body, null)
```

```
  alert(computedStyle.marginTop)
```

```
  </script>
```

```
</body>
```

✓ Demo



# Uwagi

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
className
classList
style
Informacja o stylach
Wymiary elementów
Wymiary okna
Współrzędne

- ✓ `getComputedStyle` wymaga dokładnej właściwości
- ✗ zachowanie niezdefiniowane:

```
<style>body{ margin: 10px}</style>
<body>  <script>
let computedStyle =
    getComputedStyle(document.body, null)
alert(computedStyle.margin)
</script>
</body>
```
- ✗ Demo
- ✓ JavaScript nie ma dostępu do `:visited`

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)**
- [CSS box model](#)
- [Metryki](#)
- [Odstęp zewnętrzny](#)
- [Rozmiar zewnętrzny](#)
- [Odstęp wewnętrzny](#)
- [Rozmiar wewnętrzny](#)
- [Rozmiar całkowity](#)
- [Część przewinięta](#)
- [Rozmiar z css](#)
- [Wymiary okna](#)
- [Współrzędne](#)

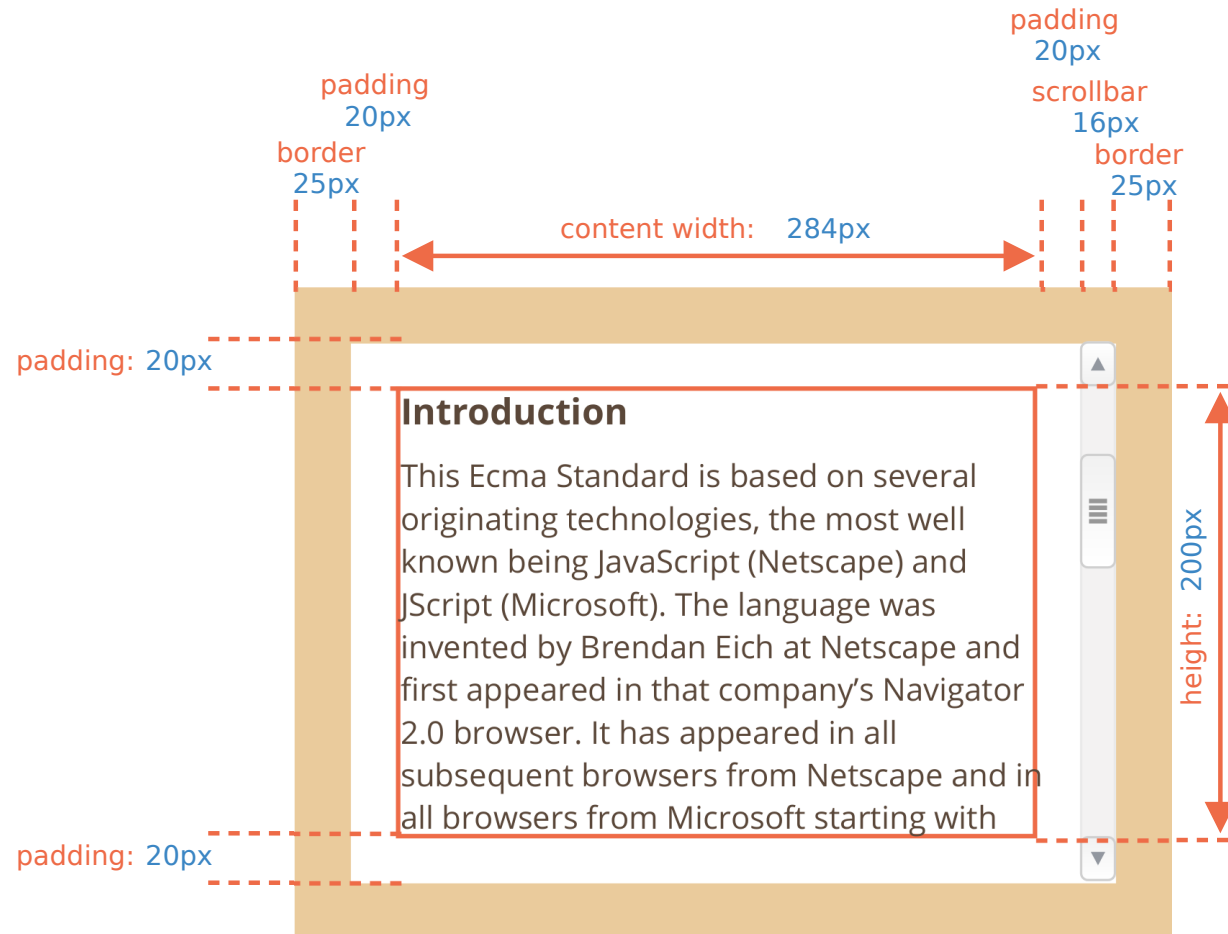
# Wymiary elementów

# CSS box model, przykład

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
<b>CSS box model</b>
Metryki
Odstęp zewnętrzny
Rozmiar zewnętrzny
Odstęp wewnętrzny
Rozmiar wewnętrzny
Rozmiar całkowity
Część przewinięta
Rozmiar z css
Wymiary okna
Współrzędne

```
<div id="example">
    ...Tekst...
</div>
<style>
    #example {
        width: 300px;
        height: 200px;
        border: 25px solid #E8C48F;
        padding: 20px;
        overflow: auto;
    }
</style>
```

# CSS box model



✓ **padding-bottom** może być wypełnione tekstem

# Pasek przewijania

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

Odstęp wewnętrzny

Rozmiar wewnętrzny

Rozmiar całkowity

Część przewinięta

Rozmiar z css

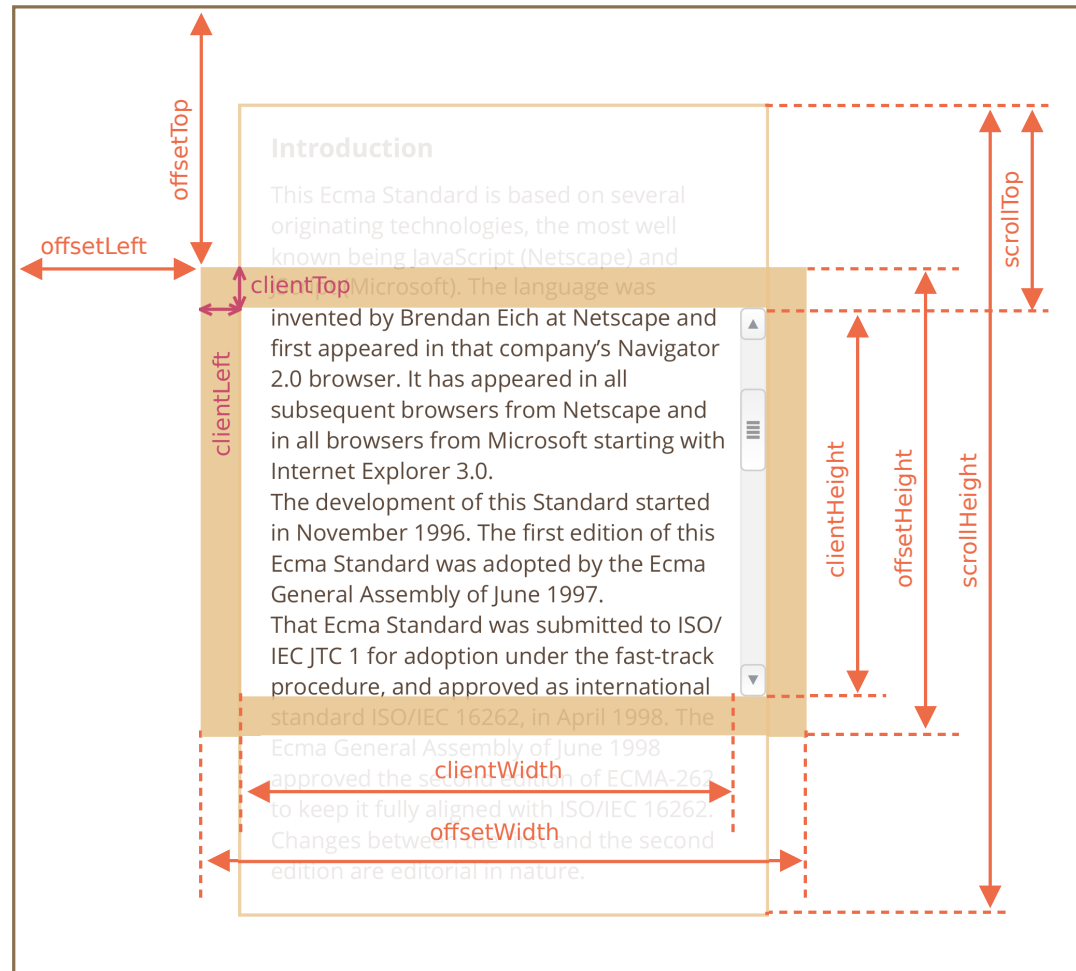
Wymiary okna

Współrzędne

- ✓ zazwyczaj ma szerokość 16px
- ✓ zabiera miejsce u wnętrza elementu (szerokość się zmniejsza o 16px)
- ✓ `getComputedStyle().width`
- ✗ nie wszystkie przeglądarki zmniejszają
- ✓ Zobaczć FF/Chrome

# Metryki Javascript

- ✓ Zawierają liczby (w pikselach (bez **px**))



# offsetParent, offsetLeft/Top

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

Odstęp wewnętrzny

Rozmiar wewnętrzny

Rozmiar całkowity

Część przewinięta

Rozmiar z css

Wymiary okna

Współrzędne

- ✓ offsetParent — element rodzicielski, względem którego mierzy się współrzędne:
  - ✗ position: relative, absolute, fixed bądź sticky
  - ✗ <td>, <th>, <table>
  - ✗ <body>
- ✓ offsetParent === null:
  - ✗ dla elementów niewidocznych
    - ✓ display: none
    - ✓ nie umieszczony na drzewie DOM
  - ✗ dla <body> oraz <html>
  - ✗ dla elementów z position: fixed
- ✓ offsetLeft/offsetTop — współrzędne względem lewego górnego rogu offsetParent

# Przykład

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

Odstęp wewnętrzny

Rozmiar wewnętrzny

Rozmiar całkowity

Część przewinięta

Rozmiar z css

Wymiary okna

Współrzędne

```
<main style="position: relative" id="main">
  <article>
    <div id="example"
      style="position: absolute;
      left: 180px;
      top: 180px">...</div>
  </article>
</main>
<script>
  alert(example.offsetParent.id); // main
  alert(example.offsetLeft); // 180 (bez „px”)
  alert(example.offsetTop); // 180
</script>
```



# Przykład

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

Odstęp wewnętrzny

Rozmiar wewnętrzny

Rozmiar całkowity

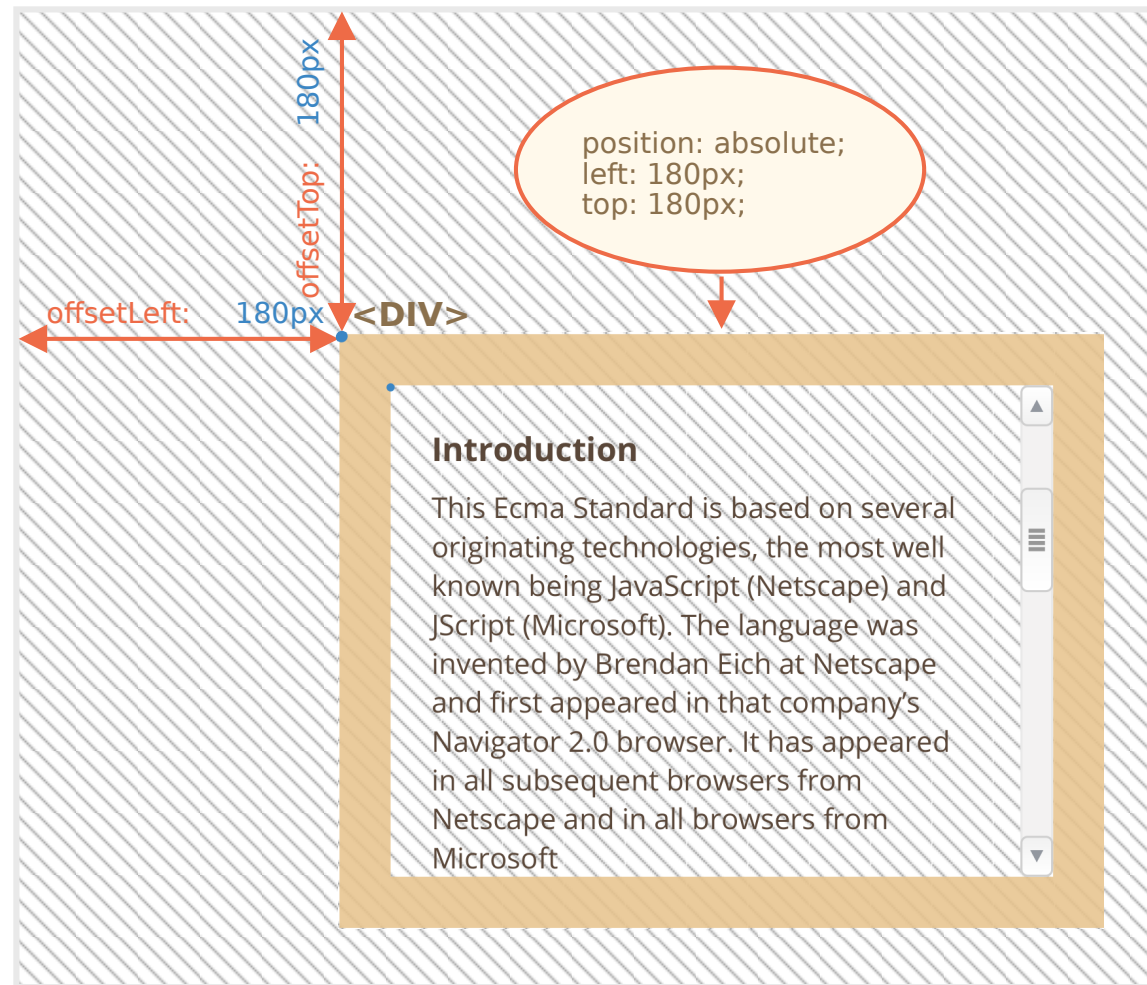
Część przewinięta

Rozmiar z css

Wymiary okna

Współrzędne

offsetParent <MAIN>



# offsetWidth/offsetHeight

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

Odstęp wewnętrzny

Rozmiar wewnętrzny

Rozmiar całkowity

Część przewinięta

Rozmiar z css

Wymiary okna

Współrzędne

- ✓ offsetWidth, offsetHeight całkowity rozmiar:
  - ✗ łącznie z obramowaniem
- ✓ Dla elementów niewidocznych wszystkie rozmiary równe są zeru:
  - ✗ `display: none` (również na rodzicielskim elemencie)
  - ✗ nie umieszczony na drzewie DOM
- ✓ można sprawdzić widoczność elementu:

```
function isHidden(elem) {  
    return !elem.offsetWidth && !elem.offsetHeight;  
}
```

  - ✗ zwraca `true` także dla pustych elementów

# Przykład

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

Odstęp wewnętrzny

Rozmiar wewnętrzny

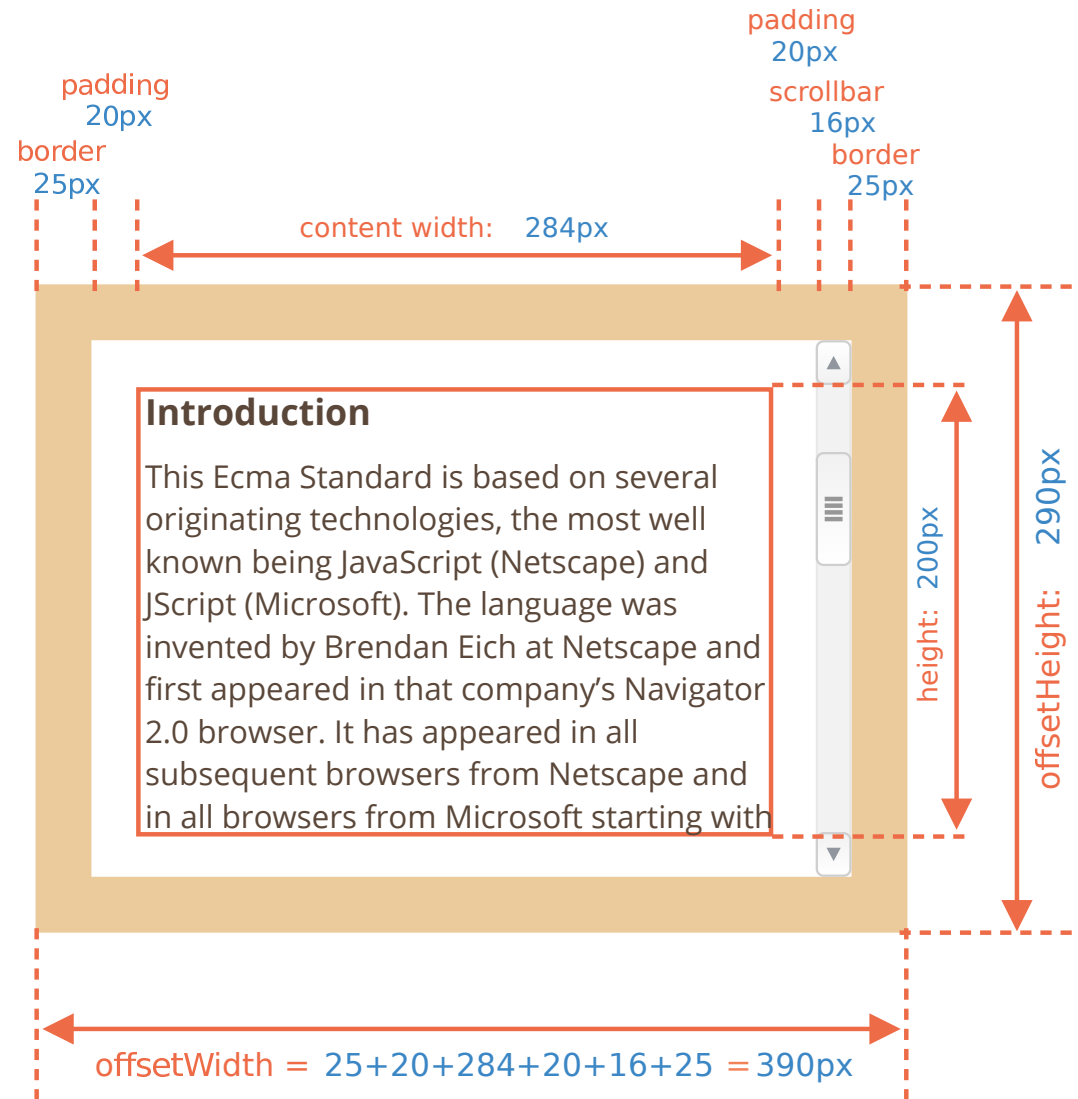
Rozmiar całkowity

Część przewinięta

Rozmiar z css

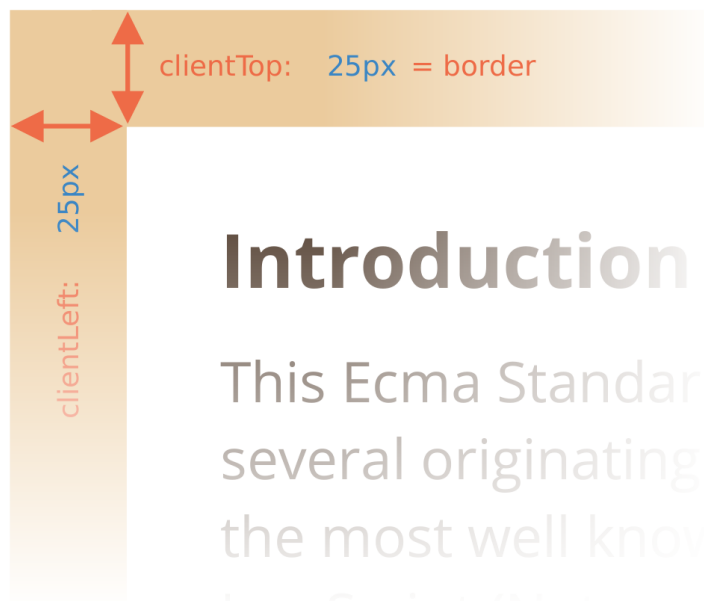
Wymiary okna

Współrzędne



# clientTop/Left

- ✓ clientTop, clientLeft — odstęp zawartości elementu od granicy:



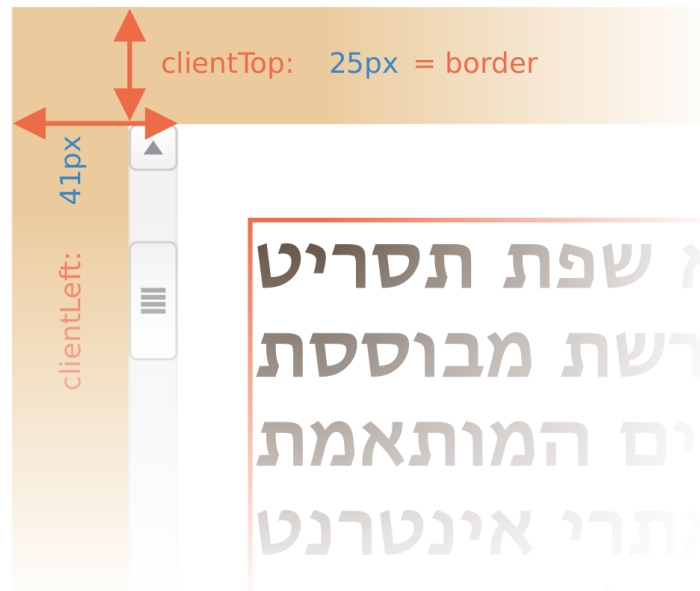
## Introduction

This Ecma Standard defines several originating from the most well known JavaScript (Mozila

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- CSS box model
- Metryki
- Odstęp zewnętrzny
- Rozmiar zewnętrzny
- Odstęp wewnętrzny**
- Rozmiar wewnętrzny
- Rozmiar całkowity
- Część przewinięta
- Rozmiar z css
- Wymiary okna
- Współrzędne

# clientTop/Left

- ✓ Prawie zawsze zgadza się z rozmiarem ramki



BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

CSS box model

Metryki

Odstęp zewnętrzny

Rozmiar zewnętrzny

**Odstęp wewnętrzny**

Rozmiar wewnętrzny

Rozmiar całkowity

Część przewinięta

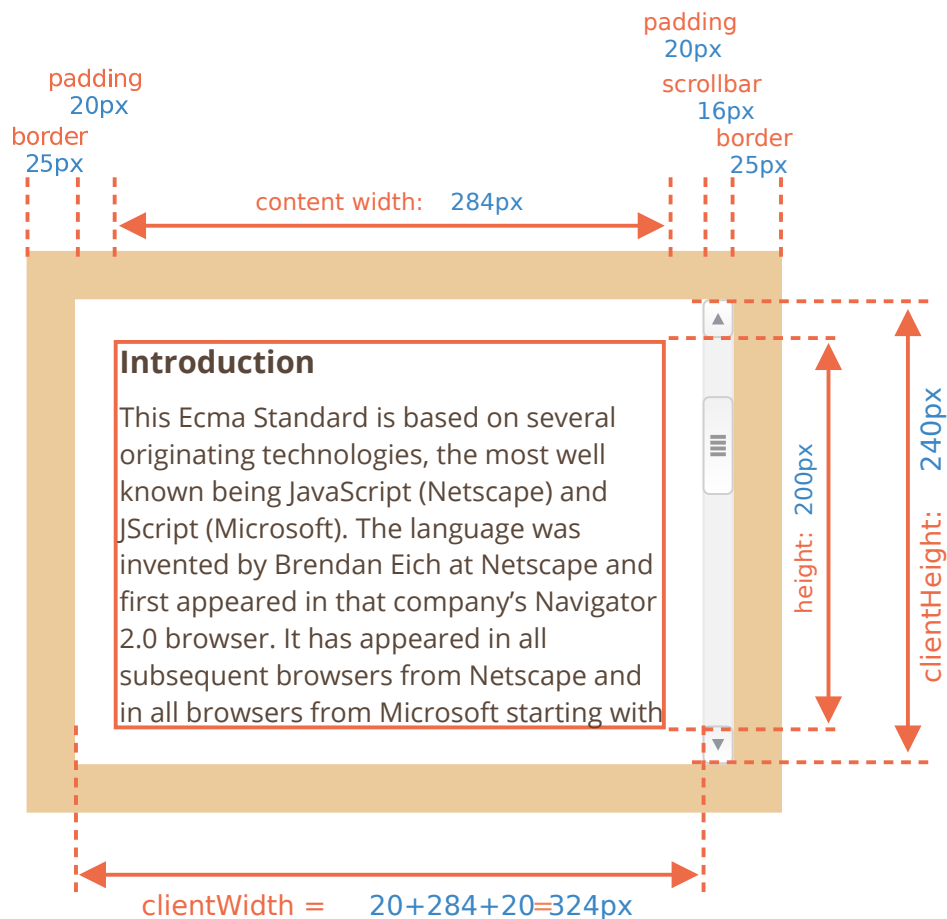
Rozmiar z css

Wymiary okna

Współrzędne

# clientWidth/Height

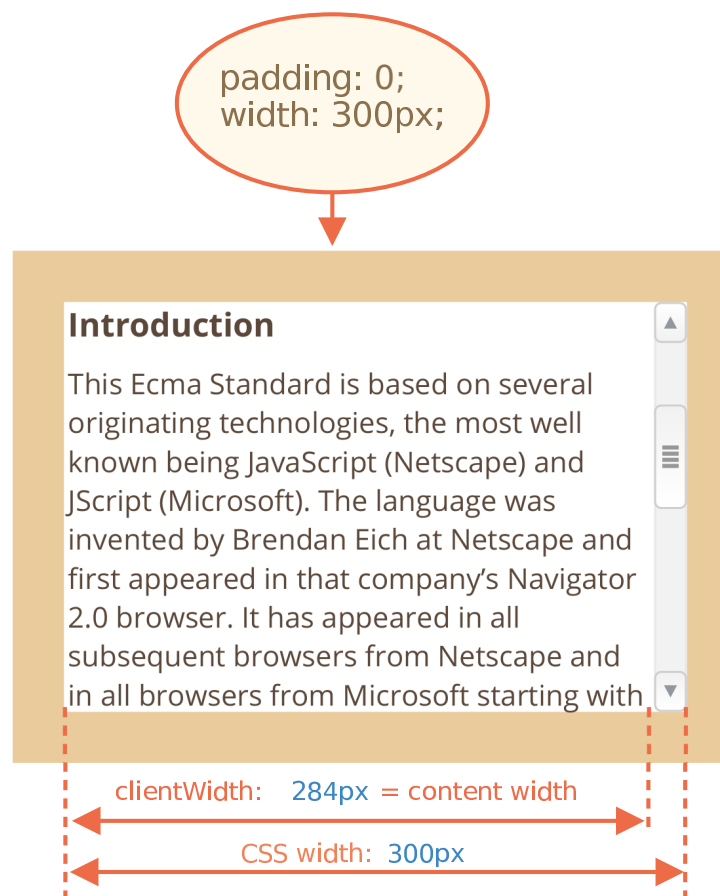
- ✓ `clientWidth`, `clientHeight` — rozmiar zawartości elementu razem z **padding**, ale bez paska przewijania:



- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- CSS box model
- Metryki
- Odstęp zewnętrzny
- Rozmiar zewnętrzny
- Odstęp wewnętrzny
- Rozmiar wewnętrzny**
- Rozmiar całkowity
- Część przewinięta
- Rozmiar z css
- Wymiary okna
- Współrzędne

# clientWidth/Height

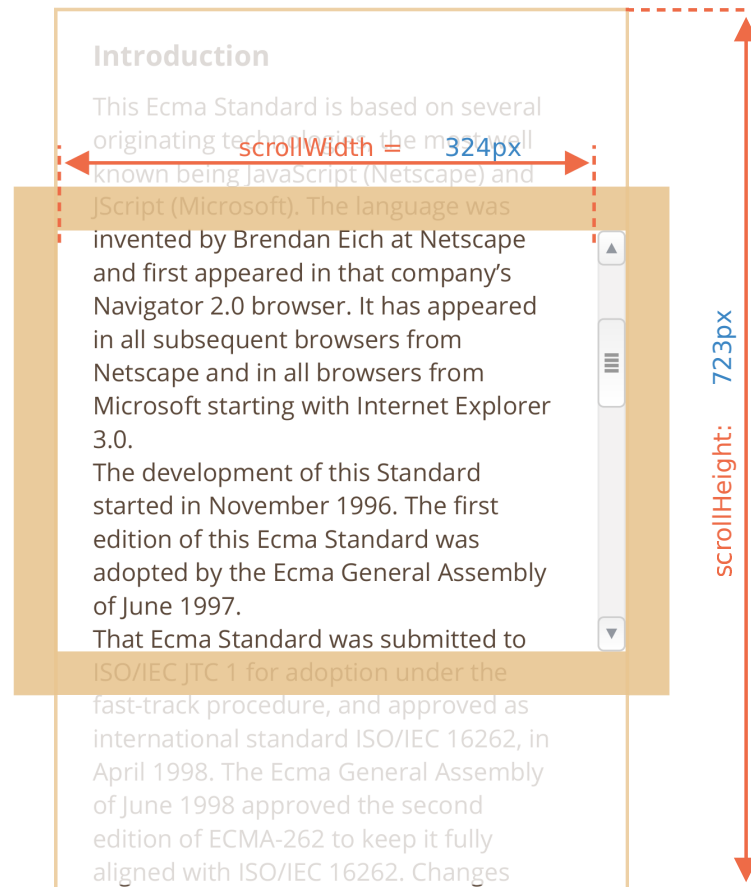
- ✓ Jeżeli `padding==0`, `clientWidth/clientHeight` zgadza się z rozmiarem zawartości:



- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [CSS box model](#)
- [Metryki](#)
- [Odstęp zewnętrzny](#)
- [Rozmiar zewnętrzny](#)
- [Odstęp wewnętrzny](#)
- [Rozmiar wewnętrzny](#)
- [Rozmiar całkowity](#)
- [Część przewinięta](#)
- [Rozmiar z css](#)
- [Wymiary okna](#)
- [Współrzędne](#)

# scrollWidth/Height

- ✓ `scrollWidth`, `scrollHeight` — rozmiar zawartości elementu razem z ukrytą (przewiniętą) częścią



- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- CSS box model
- Metryki
- Odstęp zewnętrzny
- Rozmiar zewnętrzny
- Odstęp wewnętrzny
- Rozmiar wewnętrzny
- Rozmiar całkowity**
- Część przewinięta
- Rozmiar z css
- Wymiary okna
- Współrzędne



# Rozwinąć

- ✓ Rozwinąć element na całą wysokość

```
element.style.height  
    = element.scrollHeight + 'px';
```

- ✓ [Zobaczyć](#)

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[CSS box model](#)

[Metryki](#)

[Odstęp zewnętrzny](#)

[Rozmiar zewnętrzny](#)

[Odstęp wewnętrzny](#)

[Rozmiar wewnętrzny](#)

[Rozmiar całkowity](#)

[Część przewinięta](#)

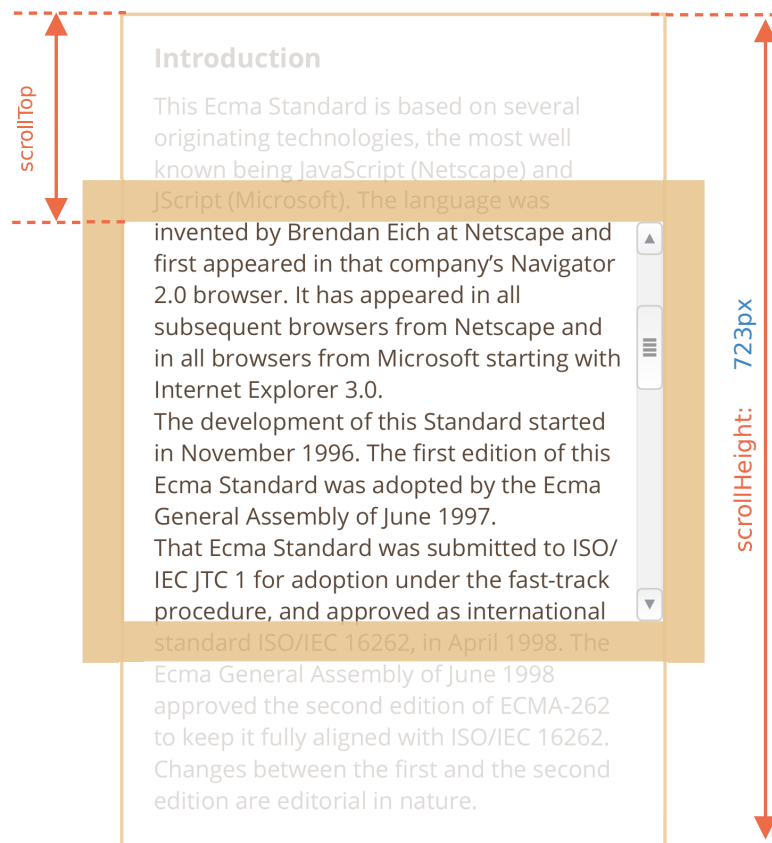
[Rozmiar z css](#)

[Wymiary okna](#)

[Współrzędne](#)

# scrollLeft/scrollTop

- ✓ scrollLeft, scrollTop — rozmiar przewiniętej części elementu (z lewej i z góry):



- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- CSS box model
- Metryki
- Odstęp zewnętrzny
- Rozmiar zewnętrzny
- Odstęp wewnętrzny
- Rozmiar wewnętrzny
- Rozmiar całkowity
- Część przewinięta**
- Rozmiar z css
- Wymiary okna
- Współrzędne

# Można zmieniać

- ✓ `scrollTop/scrollLeft` (i tylko `scrollTop/scrollLeft`) można zmieniać

`onclick = "this.scrollTop+=10"`

- ✓ [Zobaczyć](#)

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[CSS box model](#)

[Metryki](#)

[Odstęp zewnętrzny](#)

[Rozmiar zewnętrzny](#)

[Odstęp wewnętrzny](#)

[Rozmiar wewnętrzny](#)

[Rozmiar całkowity](#)

[Część przewinięta](#)

[Rozmiar z css](#)

[Wymiary okna](#)

[Współrzędne](#)

# Rozmiar elementu z css

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[CSS box model](#)

[Metryki](#)

[Odstęp zewnętrzny](#)

[Rozmiar zewnętrzny](#)

[Odstęp wewnętrzny](#)

[Rozmiar wewnętrzny](#)

[Rozmiar całkowity](#)

[Część przewinięta](#)

[Rozmiar z css](#)

[Wymiary okna](#)

[Współrzędne](#)

- ✓ `elem.style`, `getComputedStyle()` nie są wiarygodne:
  - ✗ zależy od `box-sizing`
  - ✗ może być `auto`
  - ✗ pasek przewijania jest włączany w `width/height` przez jedne przeglądarki i nie jest przez inne

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<b><a href="#">Wymiary okna</a></b>
<a href="#">Widoczna część</a>
<a href="#">Rozmiar dokumentu</a>
<a href="#">Przewijanie strony</a>
<a href="#">Współrzędne</a>

# Wymiary okna

# Szerokość/wysokość widocznej części

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Widoczna część](#)

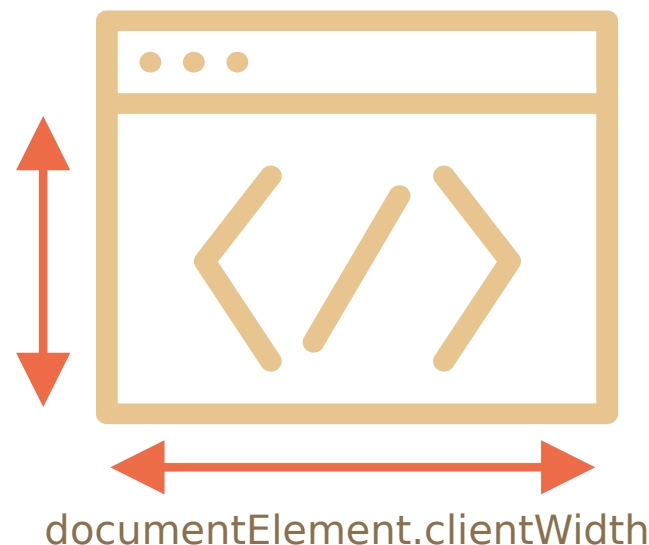
[Rozmiar dokumentu](#)

[Przewijanie strony](#)

[Współrzędne](#)

- ✓ właściwości `clientWidth/Height` elementu `document.documentElement` pozwalają otrzymać rozmiary widocznej części okna
- ✓ [Zobaczyć](#)

`documentElement.clientHeight`



# Wymiary dokumentu z przewijaniem

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Widoczna część
Rozmiar dokumentu
Przewijanie strony
Współrzędne

- ✓ właściwości `scrollWidth/Height` elementu `document.documentElement` pozwalają otrzymać rozmiary okna razem z paskiem przewijania
- ✓ jeżeli paska nie ma, to `scrollWidth/Height` na `documentElement` działa niepoprawnie
  - ✗ problem tylko dla `document.documentElement`, zwykłe elementy są w porządku
- ✓ należy brać maksymalną wartość

# Szerokość/wysokość z przewijaniem

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Widoczna część](#)

[Rozmiar dokumentu](#)

[Przewijanie strony](#)

[Współrzędne](#)

```
let scrollHeight = Math.max(  
    document.body.scrollHeight,  
    document.documentElement.scrollHeight,  
    document.body.offsetHeight,  
    document.documentElement.offsetHeight,  
    document.body.clientHeight,  
    document.documentElement.clientHeight  
);
```

```
alert(scrollHeight);
```

✓ [Zobaczyć](#)



# Współrzędne przewijania strony

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Widoczna część</a>
<a href="#">Rozmiar dokumentu</a>
<a href="#">Przewijanie strony</a>
<a href="#">Współrzędne</a>

- ✓ `window.pageXOffset/pageYOffset` zawiera bieżące współrzędne przewijania strony
  - ✗ dostępne tylko do odczytu
- ✓ [Zobaczyć](#)

# Zmiana przewijania

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Widoczna część</a>
<a href="#">Rozmiar dokumentu</a>
<a href="#">Przewijanie strony</a>
<a href="#">Współrzędne</a>

- ✓ `scrollBy(x,y)`
- ✓ `scrollTo(x,y)`
- ✓ [Zobaczyć](#)

## elem.scrollToView(top)

- ✓ `true` — element do góry
- ✓ `false` — element na dół
- ✓ [Zobaczyć](#)

[BOM](#)

[DOM](#)

[Nawigacja](#)

[Dostęp](#)

[Właściwości](#)

[Atrybuty](#)

[Modyfikacja](#)

[Style](#)

[Wymiary elementów](#)

[Wymiary okna](#)

[Widoczna część](#)

[Rozmiar dokumentu](#)

[Przewijanie strony](#)

[Współrzędne](#)

# Zatrzymanie przewijania

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<a href="#">Widoczna część</a>
<a href="#">Rozmiar dokumentu</a>
<a href="#">Przewijanie strony</a>
<a href="#">Współrzędne</a>

```
document.body.style.overflow = "hidden"
```

✓ [Zobaczyć](#)

<a href="#">BOM</a>
<a href="#">DOM</a>
<a href="#">Nawigacja</a>
<a href="#">Dostęp</a>
<a href="#">Właściwości</a>
<a href="#">Atrybuty</a>
<a href="#">Modyfikacja</a>
<a href="#">Style</a>
<a href="#">Wymiary elementów</a>
<a href="#">Wymiary okna</a>
<b><a href="#">Współrzędne</a></b>
<a href="#">Układy współrzędnych</a>
<a href="#">Względem okna</a>
<a href="#">elementFromPoint</a>
<a href="#">Względem dokumentu</a>

# Współrzędne

# Dwa układy współrzędnych

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

Współrzędne

Układy  
współrzędnych

Względem okna  
elementFromPoint  
Względem  
dokumentu

- ✓ względem dokumentu: `pageX/pageY`
  - ✗ Początek układu współrzędnych — lewy górny róg strony
- ✓ względem okna: `clientX/clientY`
  - ✗ Początek układu współrzędnych — lewy górny róg widocznego obszaru
- ✓ Współrzędne elementu == współrzędne jego lewego górnego rogu

# Dwa układy współrzędnych

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

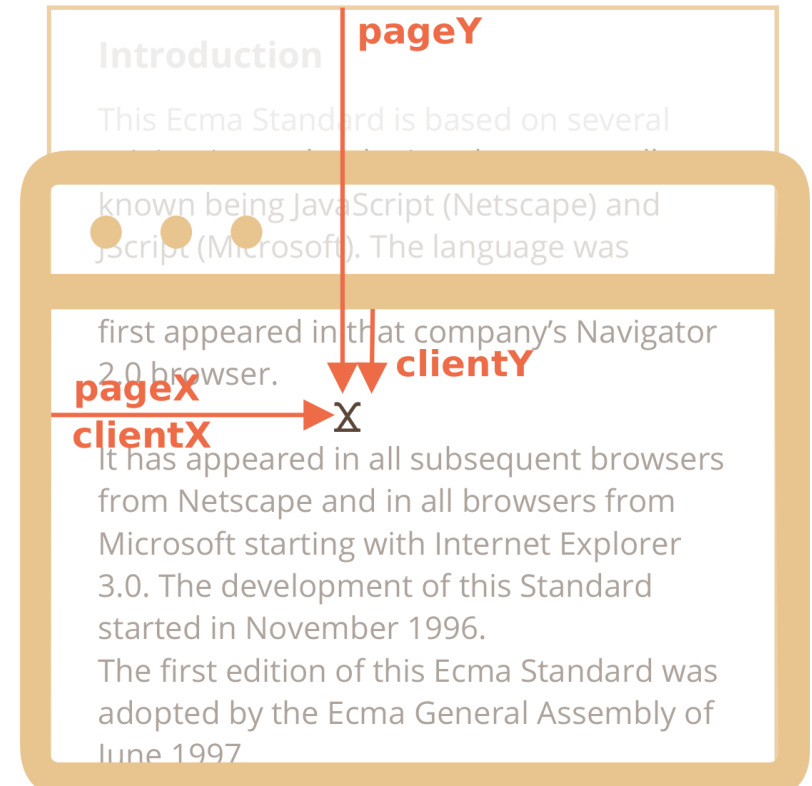
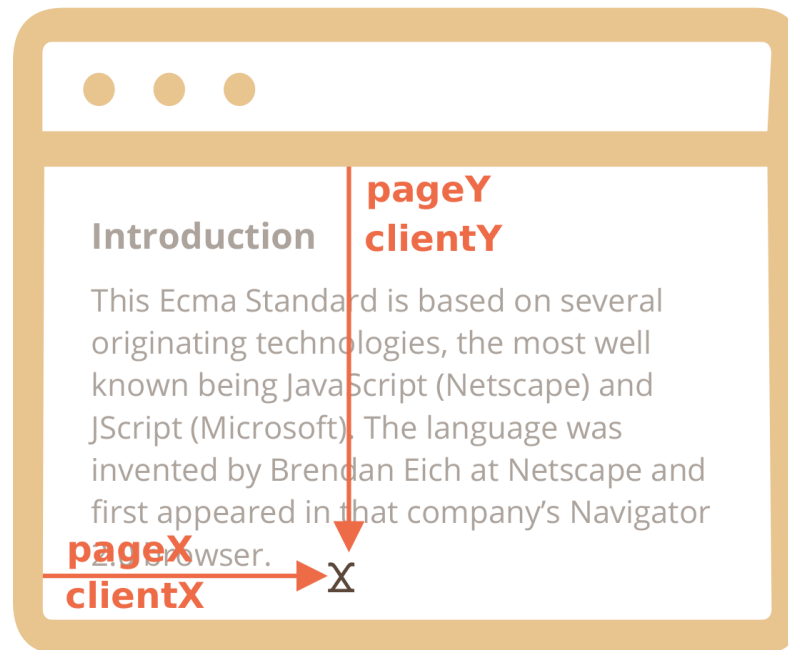
Wymiary elementów

Wymiary okna

Współrzędne

Układy współrzędnych

Względem okna  
elementFromPoint  
Względem dokumentu



# Współrzędne względem okna

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne
- Układy współrzędnych
- Względem okna
- elementFromPoint
- Względem dokumentu

- ✓ `elem.getBoundingClientRect()`
- ✓ wynik: obiekt `DOMRect`, zawierający współrzędne prostokąta:  
`top, left, right, bottom`  
`x, y, width, height`
- ✓ jeżeli obiekt składa się z kilka prostokątów (na przykład, `p`), to `elem.getBoundingClientRect()` daje prostokąt obejmujący wszystkie
- ✓ [Zobaczyć](#)



# Współrzędne względem okna

BOM

DOM

Nawigacja

Dostęp

Właściwości

Atrybuty

Modyfikacja

Style

Wymiary elementów

Wymiary okna

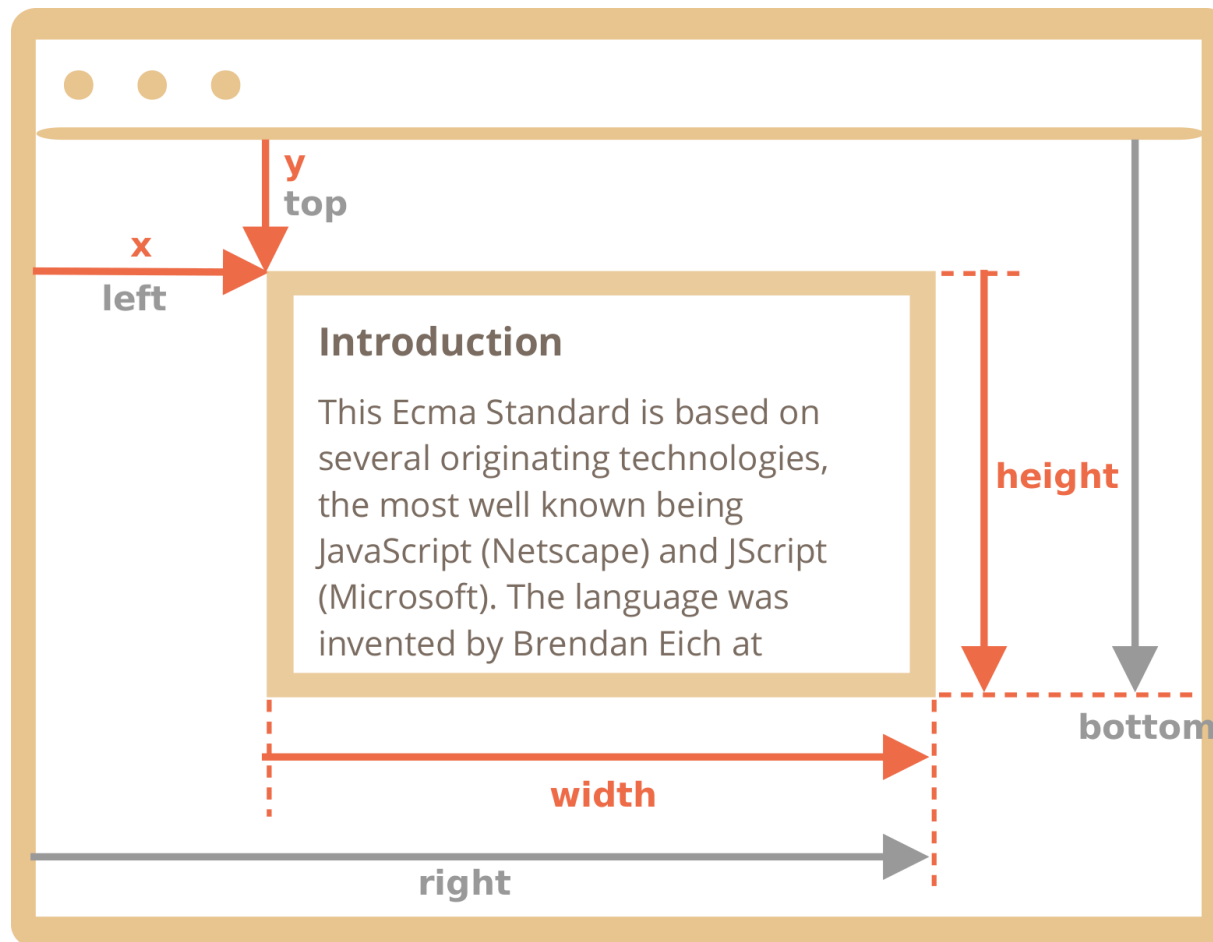
Współrzędne

Układy  
współrzędnych

Względem okna

elementFromPoint

Względem  
dokumentu



# Uwagi

- BOM
- DOM
- Nawigacja
- Dostęp
- Właściwości
- Atrybuty
- Modyfikacja
- Style
- Wymiary elementów
- Wymiary okna
- Współrzędne
- Układy współrzędnych
- Względem okna
- elementFromPoint
- Względem dokumentu

- ✓ `left == x, top == y,`
- ✓ mogą być ujemne (przy przewijaniu)
- ✓ Internet Explorer oraz Edge: nie wspierają `x/y`
  - ✗ `polifill`
  - ✗ korzystać tylko z `top, left, right, bottom`
- ✓ współrzędne `right/bottom` różnią się od właściwości `css right/bottom`

## elementFromPoint(x, y)

- ✓ `document.elementFromPoint(x, y)` — element pod punktem o współrzędnych  $(x, y)$ .
- ✗ jeżeli punkt  $(x, y)$  jest poza oknem, funkcja zwraca `null`
- ✓ [Zobacz](#)

- [BOM](#)
- [DOM](#)
- [Nawigacja](#)
- [Dostęp](#)
- [Właściwości](#)
- [Atrybuty](#)
- [Modyfikacja](#)
- [Style](#)
- [Wymiary elementów](#)
- [Wymiary okna](#)
- [Współrzędne](#)
- [Układy współrzędnych](#)
- [Względem okna](#)
- [elementFromPoint](#)**
- [Względem dokumentu](#)

# Dodanie elementu obok

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne
Układy współrzędnych
Względem okna
<b>elementFromPoint</b>
Względem dokumentu

```
function createMessageUnder(elem, html) {  
  let message = document.createElement('div');  
  message.style.cssText = "position:fixed; color: red";  
  let coords = elem.getBoundingClientRect();  
  message.style.left = coords.left + "px";  
  message.style.top = coords.bottom + "px";  
  message.innerHTML = html;  
  return message;  
}
```

```
let message = createMessageUnder(elem, 'Hello, world!')  
document.body.append(message);  
setTimeout(() => message.remove(), 5000);
```

✓ [Zobaczyć](#)

# Przy przewijaniu

- ✓ Przy przewijaniu komunikat zostaje na miejscu
- ✗ zamienić `position: fixed` na `position: absolut`
- ✓ Potrzebne współrzędne względem dokumentu

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne
Układy
współrzędnych
Względem okna
elementFromPoint
Względem
dokumentu

# Współrzędne względem dokumentu

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne
Układy współrzędnych
Względem okna
elementFromPoint
Względem dokumentu

- ✓ Brak standardowej metody
- ✓  $\text{pageY} = \text{clientY} + \text{wysokość przewiniętej części}$
- ✓  $\text{pageX} = \text{clientX} + \text{szerokość przewiniętej części}$

```
function getCoords(elem) {  
  let box = elem.getBoundingClientRect();  
  return {  
    top: box.top + pageYOffset,  
    bottom: box.bottom + pageYOffset,  
    left: box.left + pageXOffset,  
    right: box.right + pageXOffset  
  };  
}
```

# Dodanie elementu obok, nowa wersja

BOM
DOM
Nawigacja
Dostęp
Właściwości
Atrybuty
Modyfikacja
Style
Wymiary elementów
Wymiary okna
Współrzędne
Układy współrzędnych
Względem okna
elementFromPoint
Względem dokumentu

```
function createMessageUnder(elem, html) {  
  let message = document.createElement('div');  
  message.style.cssText = "position:absolute";  
  let coords = elem.getBoundingClientRect();  
  message.style.left = coords.left + "px";  
  message.style.top = coords.bottom + "px";  
  message.innerHTML = html;  
  return message;  
}
```

```
let message = createMessageUnder(elem, 'Hello, world!')  
document.body.append(message);  
setTimeout(() => message.remove(), 5000);
```

✓ [Zobaczyć](#)