

# OBIEKTOWY MODEL DOKUMENTU DOM

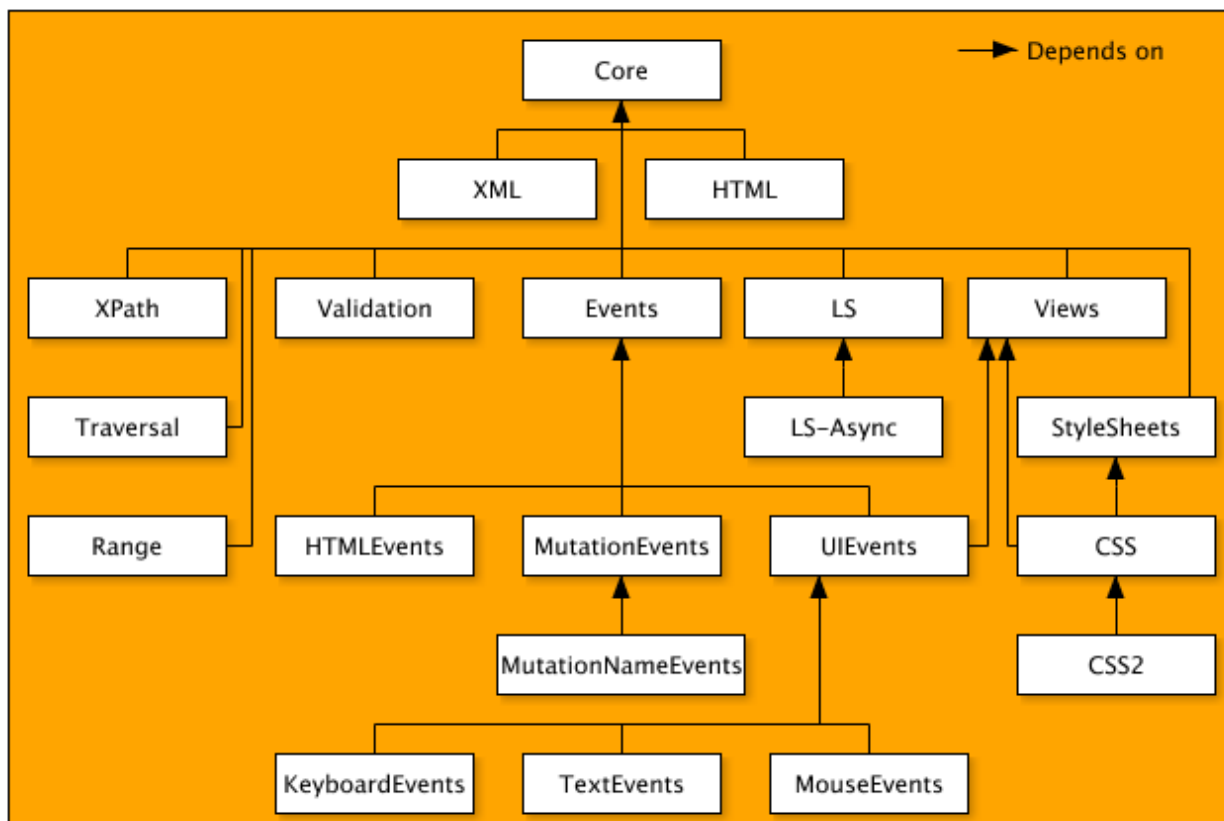
DOM = Document Object  
Model

# DOM – CO TO ?

- ▶ API (interfejs programistyczny) do interakcji z dokumentami HTML i XML
- ▶ Reprezentacja dokumentów HTML i XML w postaci modelu obiektowego
- ▶ Model uniwersalny - nie zależy od języka programowania
  - ▶ Sposób dostępu do elementów dokumentu jest taki sam w różnych językach programowania
- ▶ Standard stworzony przez W3C
- ▶ W3C definiuje interfejs dla Javascript i Java
- ▶ Istnieją biblioteki dla innych języków

- ▶ DOM level 1 (poziom 1) – 1998
  - ▶ Pełna obsługa dla modyfikacji dowolnej części dokumentów HTML/XML
- ▶ DOM level 2 (poziom 2) – 2000
  - ▶ Wprowadzenie funkcji *getElementById*, modelu zdarzeń, obsługi CSS, przestrzeni nazw XML
- ▶ DOM level 3 (poziom 3) – 2004
  - ▶ Wsparcie dla XPath, obsługa zdarzeń klawiatury, serializacja dokumentów jako XML
- ▶ DOM jest obszernie wspierany przez większość współczesnych przeglądarek

- Specyfikacja DOM 3 jest zbudowana z modułów.
- Można sprawdzić czy dany moduł jest obsługiwany przez przeglądarkę używając *DOMImplementation.hasFeature*



- ▶ DOM traktuje dokumenty jako struktury logiczne przypominające drzewo
- ▶ Drzewo składa się z węzłów
- ▶ Każdy dokument zawiera:
  - ▶ 0 lub 1 węzeł typu dokumentu
  - ▶ 1 węzeł dokumentu który służy jako korzeń drzewa
- ▶ Każdy element HTML jest węzłem
- ▶ Tekst wewnątrz elementu HTML też jest węzłem (tekstowym)
- ▶ Tekst poza tagami HTML (w tym przejście do następnej linii) też jest węzłem
- ▶ Każdy atrybut HTML jest węzłem (węzłem atrybutu)
- ▶ Każdy komentarz jest węzłem (węzłem komentarza)

- ▶ Węzły są powiązane relacjami
  - ▶ *Rodzice, dzieci, rodzeństwo*
- ▶ Każdy węzeł ma 1 rodzica (z wyjątkiem korzenia – html)
- ▶ Węzeł może mieć dowolną ilość dzieci
- ▶ Mogą być węzły bez żadnych dzieci (nazywane liśćmi)
- ▶ Węzły są dostępne jako obiekty dla języków programowania (np. JS)
- ▶ Węzły (obiekty) mają swoje właściwości i metody

- ▶ Węzły reprezentujące elementy HTML które zawierają tekst nie zawierają tego tekstu jako swojej wartości (*value*)
- ▶ Tekst jest zawarty jako wartość (*value*) w kolejnym węźle – dziecku elementu HTML
  - ▶ `<title>Tekst </title>` - *Tekst* nie jest wartością w `<title>` a osobnym węzłem – dzieckiem *title*
- ▶ Dostęp do tekstu można uzyskać albo poprzez *value* węzła tekstowego, albo poprzez właściwość *innerHTML* węzła elementu HTML

- ▶ Przykładowe właściwości węzłów (x jest obiektem węzła)
  - ▶ *x.innerHTML* – HTML (tekst) zawarty w węźle
  - ▶ *x.nodeName* – nazwa
  - ▶ *x.nodeValue* – wartość
  - ▶ *x.parentNode* – rodzic
  - ▶ *x.childNodes* – dzieci
  - ▶ *x.attributes* - atrybuty



- ▶ Przykładowe metody węzłów (x jest obiektem węzła)
  - ▶ ***x.getElementById(id)*** – zwraca element z podanym id
  - ▶ ***x.getElementsByTagName(tag)*** – zwraca wszystkie węzły które zostały stworzone przy użyciu podanego tagu
  - ▶ ***x.appendChild(węzeł)*** – dodaje węzeł podany jako parametr jako dziecko x
  - ▶ ***x.removeChild(węzeł)*** – usuwa węzeł podany jako parametr z węzła x

# DOM – PRZYKŁAD W JS

*strona.html*

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

```
<html>
```

```
<head>
```

```
<title>Strona internetowa</title>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tbody>
```

```
<tr>
```

```
<td>zawartość 1</td>
```

```
<td>zawartość 2</td>
```

```
</tr>
```

```
<tr>
```

```
<td>zawartość 3</td>
```

```
<td>zawartość 4</td>
```

```
</tr>
```

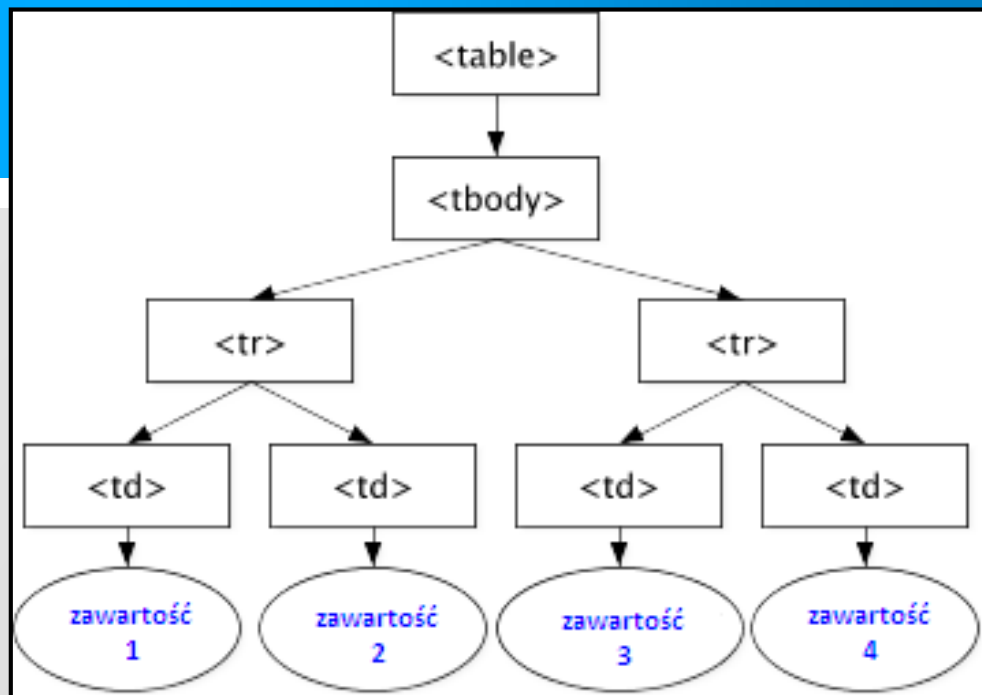
```
</tbody>
```

```
</table>
```

```
</body>
```

```
</html>
```

Skrypt zadziała  
o ile nie będzie  
przejsć do  
kolejnej linii



skrypt.js

```
var tabela=document.getElementById("tabela")
```

```
// odwołanie do elementu tbody
```

```
var elementTbody = tabela.firstChild;
```

```
// odwołanie do drugiego elementu tr zawartego w tbody
```

```
// jako dziecka - listy w DOM zaczynają się od 0
```

```
var drugiElementTr = elementTbody.childNodes[1];
```

```
// usunięcie pierwszego elementu td zawartego w drugim tr
```

```
drugiElementTr.removeChild(drugiElementTr.firstChild);
```

```
// zmiana zawartości (tekstu) elementu td który pozostał
```

```
drugiElementTr = elementTbody.childNodes[1];
```

```
drugiElementTr.firstChild.firstChild.data = "Nowa treść";
```

▶ Dostęp do tekstu poprzez *innerHTML*

▶ W XHTML **nie należy zapisywać** do *innerHTML*!

*strona.html*

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

```
<html><head>
```

```
</head><body>
```

```
<p id="akapit">To jest tekst</p>
```

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

```
//metoda getElementById węzła/obiektu document zwraca
```

```
//referencję do elementu o podanym id
```

```
//innerHTML zawiera treść danego węzła
```

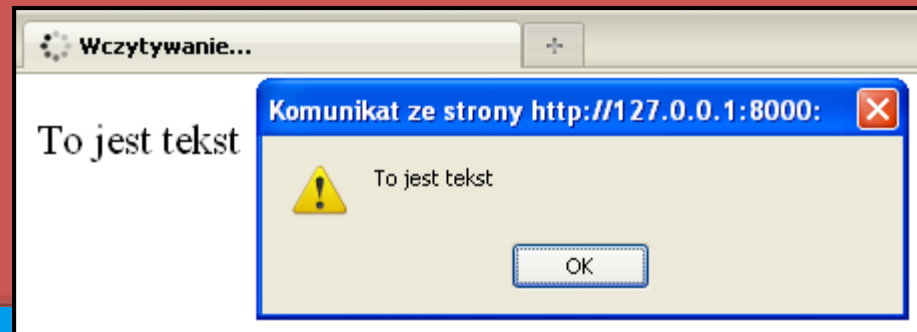
```
tekst=document.getElementById("akapit").innerHTML;
```

```
alert(tekst)
```

```
</script>
```

```
</body>
```

```
</html>
```



# DOM – DOSTĘP DO TEKSTU POPRAZEC *NODEVALUE*

Wyższe Szkoły Bankowe

## ▶ Dostęp do tekstu poprzez wartość

*strona.html*

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

```
<html><head>
```

```
</head><body>
```

```
  <p id="akapit">To jest tekst</p>
```

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

```
  //metoda getElementById węzła/obiektu document zwraca
```

```
  //referencję do elementu o podanym id
```

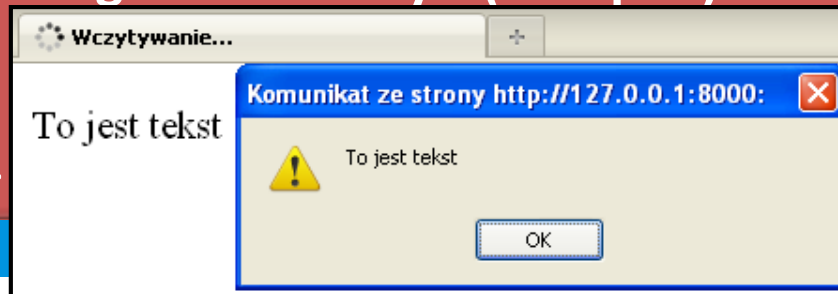
```
  //childNodes to tablica zawierająca dzieci węzła
```

```
  //pod indeks. 0 jest pierwszy element. nodeValue zawiera treść węzła
```

```
  tekst=document.getElementById("akapit").childNodes[0].nodeValue;  
  alert(tekst)
```

```
</script>
```

```
</body></html>
```



- ▶ Dostęp / wyszukiwanie węzłów są możliwe poprzez
  - ▶ ***x.getElementById(id)*** – zwraca element z podanym id
  - ▶ ***x.getElementsByTagName(tag)*** – zwraca (w formie tablicy) wszystkie węzły które zostały stworzone przy użyciu podanego tagu
  - ▶ Nawigacja przy pomocy relacji (dzieci, rodzice, rodzeństwo)
    - ▶ przy rozbudowanych dokumentach zapis może być skomplikowany
    - ▶ Przypadkowe spacje czy znaki przejścia do następnej linii między tagami są traktowane jako osobne węzły!

- getElementById i getElementsByTagName

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "  
<html><head> </head><body>  
<p id="akapit">akapit 1</p>  
<p id="akapit2">akapit 2</p>  
<script type="text/javascript">
```

```
//obiekt zawiera referencję do elementu o id akapit
```

```
var obiekt = document.getElementById("akapit")
```

```
//zwraca tablicę z wszystkimi elementami p
```

```
var lista = document.getElementsByTagName("p")
```

```
//.length zawiera rozmiar tabeli
```

```
var liczbaAkapitow = document.getElementsByTagName("p").length
```

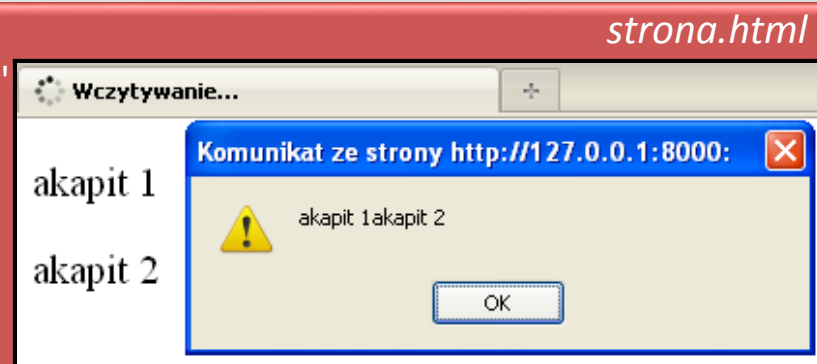
```
var tekst=""
```

```
for(i=0;i<liczbaAkapitow;i++) { tekst+= lista[i].innerHTML }
```

```
alert(tekst)
```

```
/script>
```

```
/body></html>
```



- ▶ Popularne względne obiekty DOM
  - ▶ ***parentNode*** – rodzic
  - ▶ ***firstChild*** – pierwsze dziecko
  - ▶ ***lastChild*** – ostatnie dziecko
- ▶ Obiekty nadrzędne DOM
  - ▶ ***document.documentElement*** – korzeń dokumentu
  - ▶ ***document.body*** – tag body

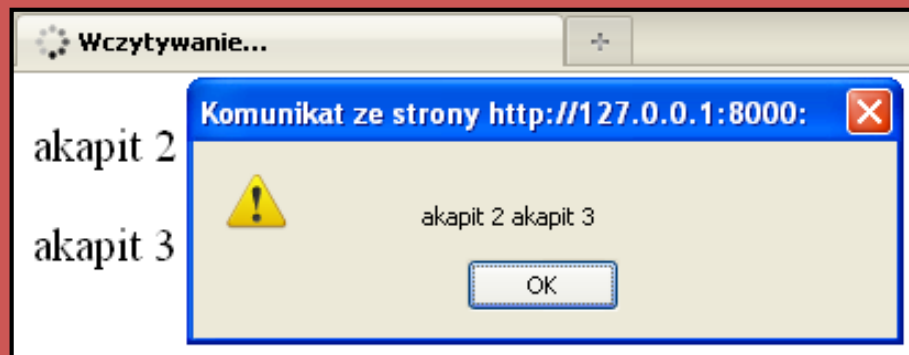
# DOM – NAWIGACJA POPRZECZ

## RELACJE: PRZYKŁAD

Wyższe Szkoły Bankowe

*strona.html*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html><head>
</head><body>
<p>akapit2</p>
<div id="identyfikator"><p>
    akapit 2</p>
    <p id="akapit3">akapit 3
</p></div>
<script type="text/javascript">
    //div zawiera referencję do elementu div (id identyfikator)
    var div = document.getElementById("identyfikator")
    //tekst wewnątrz elementu to kolejny węzeł stąd firstChild.firstChild
    //nodeValue przechowuje zawartość węzła (w tym wypadku tekst)
    var tekst=div.firstChild.firstChild.nodeValue //akapit 1
    tekst+=" " + div.lastChild.firstChild.nodeValue //akapit 2
    alert(tekst)
</script></body></html>
```





- ▶ Każdy węzeł jest obiektem
- ▶ Węzły/Obiekty mają właściwości, Niektóre właściwości:
  - ▶ *nodeName* nazwa węzła (tylko do odczytu): jest identyczna jak nazwa tagu lub atrybutu (w zależności od rodzaju węzła)
    - ▶ Dla węzłów tekstowych wartość to *#text* a dla węzła dokumentu - *#document*
    - ▶ Dla węzłów elementów HTML wartość jest pisana dużymi literami dla XHTML małymi
  - ▶ *nodeType* – typ węzła, najważniejsze typy:

▶ Typ węzła	wartość <i>NodeType</i>
▶ element	1
▶ atrybut	2
▶ tekstowy	3
▶ komentarz	8
dokument	9

- ▶ *nodeValue* – wartość/zawartość węzła
  - ▶ Dla węzłów elementów jest „*undefined*”
  - ▶ Dla węzłów tekstowych zawiera tekst węzła
  - ▶ Dla atrybutów zawiera wartość atrybutu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01...>
<html><head>
</head><body>
```

```
<p id="akapit">to jest akapit</p>
```

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

```
    //elementP zawiera referencje do p (id=akapit)
```

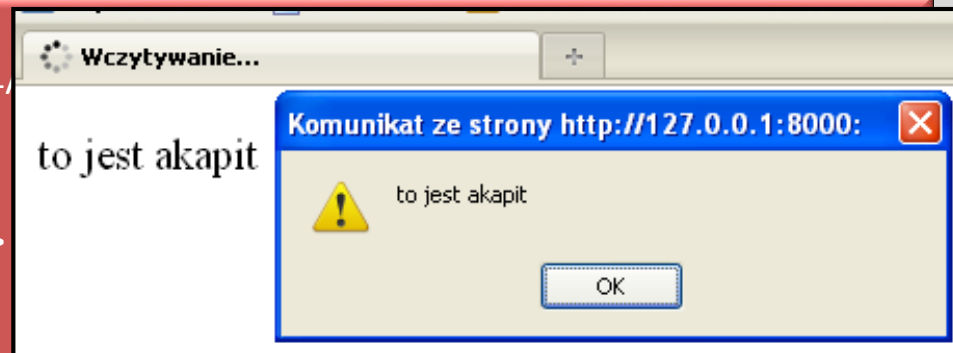
```
    elementP=document.getElementById("akapit");
```

```
    //pierwszym dzieckiem P jest wezel tekstowy
```

```
    alert(elementP.firstChild.nodeValue);
```

```
</script>
```

```
</body></html>
```



# DOM – ZMIANA ELEMENTÓW HTML: PRZYKŁAD

Wyższe Szkoły Bankowe

http://127.0.0.1:8000/dom/005.html

Tekst z JavaScript

zmień kolor tła na niebieski

Zmien tekst

http://127.0.0.1:8000/dom/005.

Tekst z JavaScript

zmień kolor tła na niebieski

http://127.0.0.1:8000/dom/005.html

Tekst z funkcji

zmień kolor tła na niebieski

Zmien tekst

```
<html><head> </head><body>
```

```
<p id="akapit">to jest akapit</p>
```

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

```
//właściwość bgColor dla body odpowiada bgColor w HTML
```

```
document.body.bgColor="green";
```

```
//nadpisanie tekstu „to jest akapit” nowym tekstem
```

```
document.getElementById("akapit").firstChild.nodeValue="Tekst z JavaScript";
```

```
function zmienTekst(){ //nadpisanie tekstu w p o id=akapit nowym tekstem
```

```
document.getElementById("akapit").firstChild.nodeValue = "Tekst z funkcji";
```

```
}
```

```
</script>
```

```
<input type="button" onclick="document.body.bgColor='blue';"
```

```
value="zmień kolor tła na niebieski" /> <!-- zmiana koloru po kliknięciu -->
```

```
<!-- zmiana tekstu po kliknięciu -->
```

```
<input type="button" onclick="zmienTekst()" value="Zmien tekst" />
```

```
</body></html>
```



# DOM – PRZYKŁAD MODYFIKACJI STYLÓW CSS

Wyższe Szkoły Bankowe

```
<!DOCTYPE HTML PUBLIC
```

```
<html><head>
```

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

```
function zmienTlo() {
```

```
//ustawia właściwość backgroundColor stylów CSS
```

```
document.body.style.backgroundColor="yellow"; }
```

```
function zmienAkapit() {
```

```
//stylAkapitu zawiera referencję do obiektu stylów elementu o id=akapit
```

```
var stylAkapitu=document.getElementById("akapit").style
```

```
stylAkapitu.color="blue"; //ustawienie właściwości CSS: koloru i czcionki
```

```
stylAkapitu.fontFamily="Arial"; }
```

```
</script>
```

```
</head><body>
```

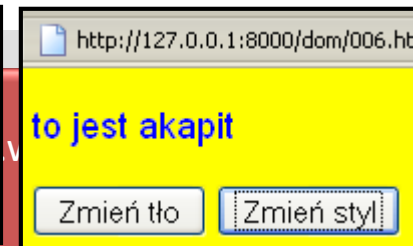
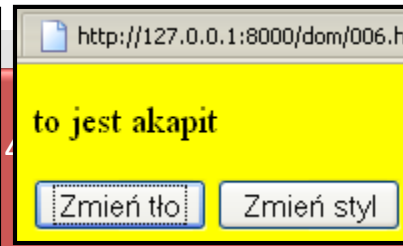
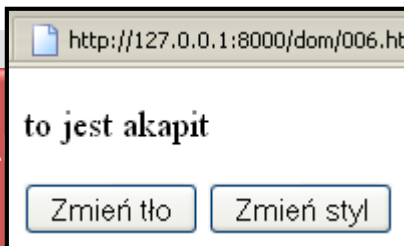
```
<p id="akapit">to jest akapit</p>
```

```
<!-- wciśnięcie przycisków wywołuje odpowiednie funkcje JS -->
```

```
<input type="button" onclick="zmienTlo()" value="Zmień tło" />
```

```
<input type="button" onclick="zmienAkapit()" value="Zmień styl" />
```

```
</body></html>
```



strona.html  
dtd">



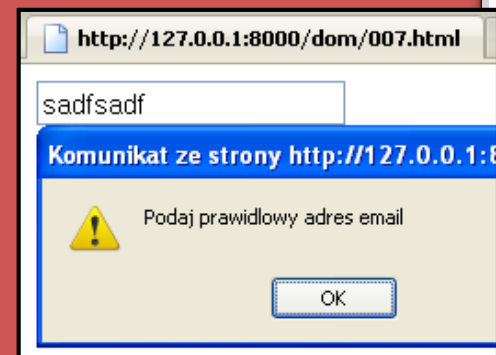
▶ Zdarzenia mogą rozpocząć działanie skryptu

▶ Często wykorzystywane

▶ **onLoad, onUnload** – wykorzystywane do uruchamiania skryptów przy otwieraniu/opuszczaniu strony

▶ **onFocus, onBlur, onChange**: ich kombinacja często jest wykorzystywana przy wypełnianiu formularzy

```
strona.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html><head>
  <script type="text/javascript">
    function sprawdzEmail() {
      wartosc=document.getElementById('email').value
      if (wartosc.search("/@/")<1)
        alert("Podaj prawidłowy adres email") }
  </script>
</head><body>
  <form action="skrypt.php">
    <input type="text" id="email" onchange="sprawdzEmail()" />
  </form></body></html>
```



## ► Specyfikacje DOM

► <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>

## ► Czy dana metoda/właściwość jest wspierana przez daną przeglądarkę można min. sprawdzić tutaj:

► <http://www.w3.org/2003/02/06-dom-support.html>

► <http://www.webdevout.net/browser-support-dom>

► [http://www.w3schools.com/jsref/dom\\_obj\\_document.asp](http://www.w3schools.com/jsref/dom_obj_document.asp)