

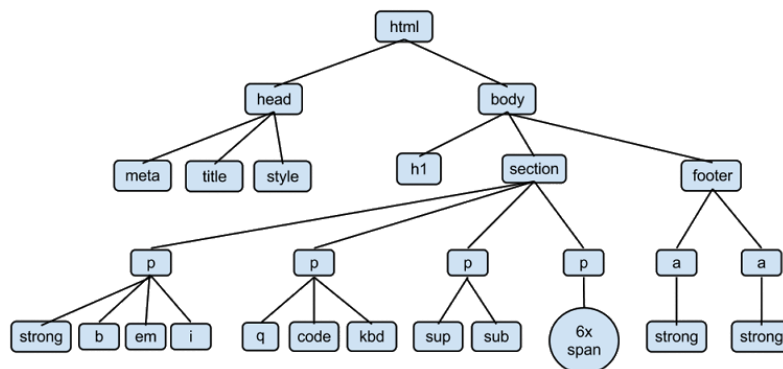
Temat: DOM. Charakterystyka JavaScript. Osadzanie kodu źródłowego JS w HTML oraz umieszczonego w osobnym pliku.
Profil: Technik Informatyk (Technikum Informatyczne)
Klasa: 1
Przedmiot: E14.3

Spis treści

1	DOM. Charakterystyka JavaScript.....	2
1.1	Osadzanie kodu źródłowego JS w HTML.....	4
1.2	Osadzanie kodu źródłowego JS umieszczonego w osobnym pliku.	5
1.3	Zdarzenia oraz adresy url	6

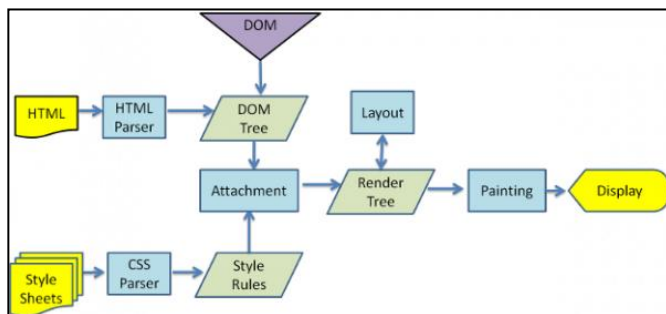
1 DOM. Charakterystyka JavaScript

Drzewo **DOM** (Document Object Model) to struktura drzewiasta symbolizująca układ elementów strony. Elementem głównym (root) jest `<html>`, wszystkie kolejne elementy tworzą zależności dzieci-rodzic (np. element `<table>` jest rodzicem dla kilku wierszy tabeli, czyli elementów `<tr>`). Z kolei w każdym wierszu istnieje wiele dzieci nadrzędnych elementów `<tr>`, czyli komórki tabeli – elementy `<td>`.

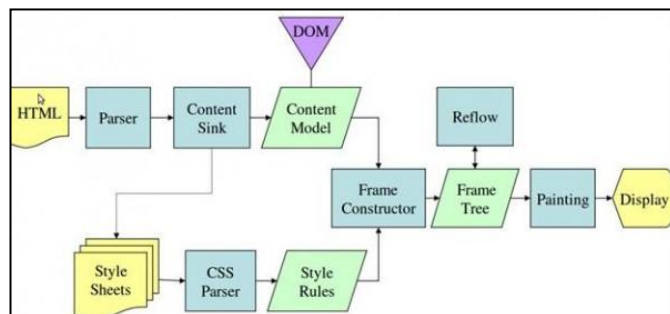


Przykład struktury drzewiastej DOM strony WWW w HTML.

Po analizie i parsowaniu kodu CSS elementom drzewa DOM jest nadawany wygląd, zgodnie z regułami zawartymi w arkuszach stylów, oraz są one rozmieszczane względem siebie. Całość tworzy tzw. „render tree” (drzewo renderingu), które ostatecznie jest wyświetlane w oknie przeglądarki (tzw. faza „painting”, czyli „odmalowywania”, bądź po prostu rysowania strony w oknie przeglądarki).



Proces generowania dokumentu w przeglądarce opartej na silniku WebKit (Google Chrome, Chromium, Safari).



Proces generowania dokumentu w przeglądarce opartej na silniku Gecko (Mozilla Firefox, Iceweasel).

DOM dostarcza informacje niezbędne językowi JavaScript do operowania na elementach strony www. Oferuje także narzędzia pozwalające na poruszanie się po zawartości strony, modyfikowanie istniejących elementów HTML oraz dodawanie nowych. Sam DOM nie jest związany z JavaScriptem, stanowi odrębny standard opracowany przez konsorcjum W3C.

Przykładowo, możesz pobrać wartość pola tekstowego jako łańcuch znaków używając DOM. Następnie możesz użyć operatora JavaScript `+`, by połączyć go z innym łańcuchem znaków i stworzyć z tego sensowną treść. Gdy to zrobisz, możesz użyć metody DOM `alert()`, aby wyświetlić użytkownikowi nowy łańcuch znaków w oknie dialogowym.

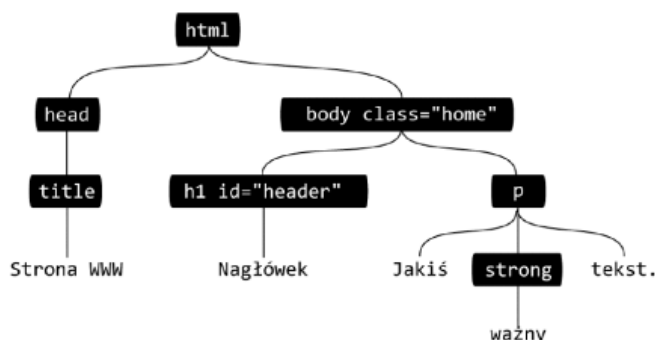
Jeśli strona internetowa byłaby meblem, DOM byłby ilustracjami jego elementów — półek, uchwyty, gwoździ oraz śrubokrętów. Możliwe jest napisanie instrukcji jak połączyć je w całość i używać tych elementów, instrukcji w wielu językach, ale Ty użyjesz tylko tego, który rozumiesz. Podręcznik ułatwia Ci złożenie tego mebla przy użyciu napisanych instrukcji (JavaScript) odwołujących się do przedstawionych ilustracji elementów (DOM), które reprezentują faktyczne obiekty (silnik renderujący przeglądarki).

Przykład działania DOM

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Strona WWW</title>
  </head>

  <body class="home">
    <h1 id="header">Nagłówek</h1>
    <p>Jakiś <strong>ważny</strong> tekst.</p>
  </body>
</html>
```

Wzajemne relacje pomiędzy poszczególnymi znacznikami można przedstawić za pomocą struktury przypominającej drzewo genealogiczne. Korzeniem tego drzewa jest <html>, pozostałe znaczniki to gałęzie <head>, <body>. Przeglądarki oprócz znaczników HTML, zapamiętują teksty wyświetlane wewnątrz nich oraz atrybuty przypisywane poszczególnym znacznikom (class, id).



DOM traktuje wszystkie te znaczniki, atrybuty, teksty jako odrębne jednostki nazywane węzłami (ang. „node”).

JavaScript udostępnia kilka sposobów wybierania elementów strony, tak by później można było coś z nimi zrobić.

Zastosowanie JS

Najczęściej spotykanym zastosowaniem języka JavaScript są strony WWW. Skrypty służą najczęściej do zapewnienia interaktywności poprzez reagowanie na zdarzenia, sprawdzania poprawności formularzy lub budowania elementów nawigacyjnych. Skrypty JavaScriptu uruchamiane przez strony internetowe mają znacznie ograniczony dostęp do komputera użytkownika. Po stronie serwera może także działać JavaScript w postaci node.js lub Ringo.

W języku JavaScript można także pisać pełnoprawne aplikacje. Fundacja Mozilla udostępnia środowisko złożone z technologii takich jak XUL, XBL, XPCOM oraz JSLib. Umożliwiają one tworzenie korzystających z zasobów systemowych aplikacji o graficznym interfejsie użytkownika dopasowującym się do danej platformy. Przykładem aplikacji napisanych z użyciem JS i XUL może być klient IRC o nazwie ChatZilla, domyślnie dołączony do pakietu Mozilla. Microsoft udostępnia biblioteki umożliwiające tworzenie aplikacji JScript jako część środowiska Windows Scripting Host. Ponadto JScript.NET jest jednym z podstawowych języków środowiska .NET. Istnieje także stworzone przez IBM środowisko SashXB dla systemu Linux, które umożliwia tworzenie w języku JavaScript aplikacji korzystających z GTK+, GNOME i OpenLDAP.

Interakcja

Wszystkie implementacje JavaScriptu dostępne w przeglądarkach internetowych dostarczają obiektów reprezentujących drzewo dokumentu. Mogą także umożliwiać tworzenie ciasteczek, manipulowanie oknami przeglądarki, wyświetlanie prostych okien dialogowych, pobieranie informacji o przeglądarce, zarządzanie jej pluginami oraz arkuszami stylów. Reagują także na zdarzenia wywoływane w interfejsie.

1.1 Osadzanie kodu źródłowego JS w HTML

Skrypty JavaScript mogą być umieszczane wewnątrz kodu (X)HTML — mówimy wtedy o **skryptach osadzonych**. Aby osadzić kod JavaScript na stronie internetowej, należy posłużyć się znacznikiem `script`.

W języku HTML5 można to zrobić banalnie prosto:

```
<script>
... kod skryptu
</script>
```

Kod w JS możemy umieszczać w dowolnym miejscu, ale trzeba go umieścić między `script`, może być w sekcji `head` albo `body`.

Przykład: Wyświetlenie dzisiejszej daty.

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="utf-8" />
    <link href="styl.css" rel="stylesheet">

    <script>
      var data = new Date();
      var rok = data.getFullYear();
      var mies = data.getMonth()+1;
      var dzien = data.getDate();

      alert(rok + "-" + mies + "-" + dzien);
    </script>
  </head>
</body>
</html>
```

Kod zostanie załadowany najpierw zanim zacznie się generowanie strony.

Przykład: Wyświetlenie dzisiejszej daty.

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="utf-8" />
    <link href="styl.css" rel="stylesheet">
  </head>
</body>

<script>
var data = new Date();
var rok = data.getFullYear();
var mies = data.getMonth()+1;
var dzien = data.getDate();

alert(rok + "-" + mies + "-" + dzien);
</script>

</body>
</html>
```

Do metody `getMonth()` dodaliśmy 1, ponieważ Javascript numeruje miesiące od 0.

1.2 Osadzanie kodu źródłowego JS umieszczonego w osobnym pliku.

Jeżeli ten sam skrypt chcemy osadzić np. na kilku podstronach, możemy zapisać go w pliku z rozszerzeniem *.js (rozszerzenie jest dowolne, ale przyjęło się stosować .js) i potem wstawić w dowolnym miejscu między body do niego odwołanie:

```
<script src="plik.js"></script>
```

Parameter src oznacza lokalizację naszego skryptu względem głównego korzenia katalogów na serwerze. Oczywiście można podać ścieżkę bezwzględną w stylu `http://nazwa.domeny/skrypty/plik.js`

Ćwiczenie: Uczniowie tworzą w/w skrypt w osobnym pliku.

Przyjęły się jeszcze dwa warianty z tego gatunku wstawiania skryptu:

a) Między znacznikiem head

Jeżeli tutaj umieścimy wstawienie kodu JS to kod ten zostanie załadowany najpierw.

Plik: **index.html**

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="utf-8" />
    <link href="styl.css" rel="stylesheet">
    <script src="skrypt.js"></script>
  </head>
  <body>
    Zakazana treść.
  </body>
</html>
```

Plik: **skrypt.js**

```
var liczba = 5;
alert ("witaj w html"+liczba);
```

Najpierw zostanie wyświetlone okienko alert, po kliknięciu OK przeglądarka wygeneruje stronę.

b) Poza znacznikiem html

Jeżeli tutaj umieścimy wstawienie kodu JS to najpierw zostanie wygenerowana strona a potem uruchomiony plik ze skryptem w JS.

Plik: **index.html**

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="utf-8" />
    <link href="styl.css" rel="stylesheet">
  </head>
  <body>
    Zakazana treść.
  </body>
</html>

<script src="skrypt.js"></script>
```

Plik: **skrypt.js**

```
var liczba = 5;  
alert ("witaj w html"+liczba);
```

Najpierw przeglądarka wygeneruje stronę a następnie zostanie wyświetlone okienko alert.

Ćwiczenie: Uczniowie sprawdzają oba warianty na przykładzie, oraz podają przykłady zastosowań dla obu wariantów.

W jednym dokumencie można naraz umieścić kilka skryptów, zarówno osadzonych, jak i zewnętrznych.

1.3 Zdarzenia oraz adresy url

Dwa kolejne sposoby umieszczania skryptów to:

a) zdarzenia

Zdarzenia w JavaScript jest to ogólny mechanizm pozwalający na uruchomienie dowolnego fragmentu skryptu JavaScript w momencie gdy wystąpi określone zdarzenie, np. przycisk zostanie kliknięty.

Przykład:

```
<p onclick="alert('Cześć!')">Kliknij tutaj</p>
```

onclick po kliknięciu

b) adresy url

Linki na stronie można wykorzystać nie tylko do przechodzenia do innych stron, ale także do inicjowania określonych akcji na tej samej stronie. Jedną z metod którą można do tego użyć jest opisane wyżej zdarzenie onclick. Drugą metodą jest natomiast wstawienie kodu JavaScript bezpośrednio do atrybutu href znacznika <a>, poprzedziwszy go wcześniej przedrostkiem

javascript:

Przykład:

```
<p><a href="javascript:alert('Hello!')">Kliknij tutaj</a></p>
```