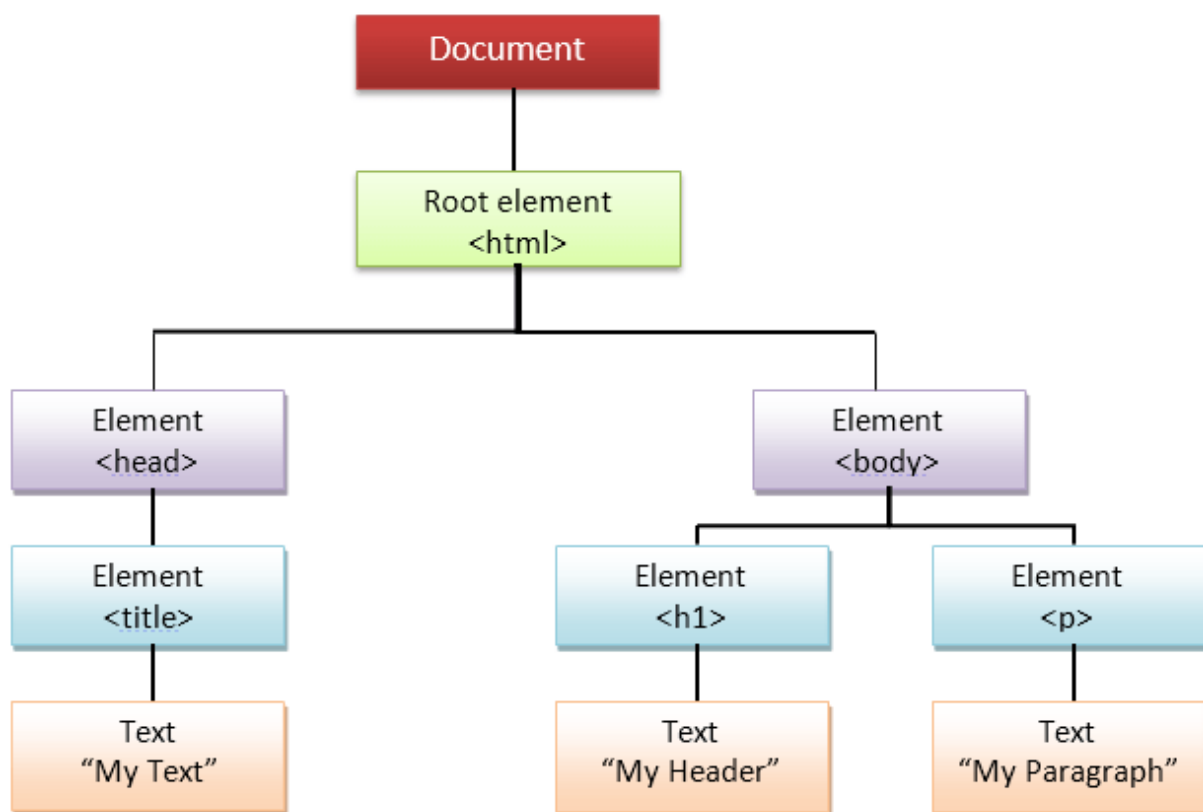


Zastanówmy się jak można by było wykorzystać wiedzę którą nabyłeś na temat JavaScript do tworzenia nowoczesnych stron internetowych. Najpierw przyjrzyjmy się skryptom wykorzystującym struktury DOM (ang. Document Object Model), co pozwala na manipulowanie elementami strony, ich atrybutami oraz tekstem prezentowanym na stronie. Najczęściej wykorzystujemy go do pracy z dokumentami HTML, ale może być też używany w połączeniu z dowolnym innym wariantem języka XML. Ze struktur DOM możemy korzystać również w takich językach programowania jak PHP, Ruby, Python, C++, Java i wielu innych.

Model dom jest interfejsem programistycznym (API) do obsługi stron HTML i XML. Tworzy on strukturalną mapę dokumentu i udostępnia zestaw metod pozwalających na interakcję z elementami tej struktury. Oznacza to, że kod strony jest przekształcany do formatu zrozumiałego np. dla języka JavaScript. Można wykorzystać go do wyszukiwania elementów po ich nazwach lub atrybutach, a następnie do dodawania, zmieniania bądź usuwania elementów i ich zawartości.



<https://www.guru99.com/how-to-use-dom-and-events-in-javascript.html>

Każdy element na stronie jest opisywany za pomocą tzw. węzła (ang. node). Jeżeli na strukturę DOM będziemy patrzeć jak na drzewo, to każdy węzeł będzie w nim osobną gałęzią, z której mogą wyrastać kolejne gałęzie.

Obiekt window

W przeglądarce korzeniem hierarchii jest obiekt window zawierający wiele właściwości i metod, które ułatwiają pracę z przeglądarką (patrz pierwszy rysunek na następnej stronie). Za pomocą obiektu window można m.in. uzyskiwać dostęp do bieżącego adresu URL, uzyskiwać informacje o wszystkich klatkach na stronie, wykorzystywać pamięć lokalną, sprawdzać informacje o ekranie, bawić się paskiem przewijania, ustawiać tekst na pasku stanu oraz robić wszelkiego rodzaju rzeczy mające zastosowanie do kontenera, w którym wyświetlana jest dana strona internetowa.

Obiekt document

Obiekt document jest bramą do wszystkich elementów HTML, które składają się na to, co jest pokazywane.

Wszelkie zmiany wprowadzane w DOM za pomocą JavaScriptu znajdują odzwierciedlenie w tym, co jest pokazywane w przeglądarce. Oznacza to, że można dynamicznie dodawać elementy, usuwać je, przenosić, zmieniać ich atrybuty, ustawiać lokalne style CSS.

Przykład:

Instrukcja nakazuje przeglądarce wyszukanie na stronie (document) elementu o identyfikatorze (id) pierwszy, odnalezienie treści HTML tego elementu (innerHTML) i przypisanie jej zawartości do zmiennej.

```
var foo = document
    .getElementById( "pierwszy" )
    .innerHTML;
```

querySelector i querySelectorAll

```
<div id="main">
  <div class="pictureContainer">
    
  </div>
  <div class="pictureContainer">
    
  </div>
  <div class="pictureContainer">
    
  </div>
  <div class="pictureContainer">
    
  </div>
</div>
```

W tym przykładzie mamy jeden element div z id o wartości **main**, a następnie mamy cztery elementy div i cztery elementy **img**, z których każdy ma wartości klas odpowiednio **pictureContainer** i **theImage**.

Funkcja `querySelector` zasadniczo działa w następujący sposób:

```
var element = document.querySelector("selektor_CSS");
```

Jeśli chcielibyśmy uzyskać dostęp do div, którego id ma wartość **main**, napisalibyśmy:

```
var element = document.querySelector("#main");
```

Podobnie możemy określić selektor dla klasy `pictureContainer`:

```
var element = document.querySelector(".pictureContainer");
```

Funkcja `querySelectorAll` zwraca wszystkie znalezione elementy, które pasują do podanego selektora:

```
var element = document.querySelectorAll("selektor_CSS");
```

```
var images = document.querySelectorAll(".theImage");
for (var i = 0; i < images.length; i++) {
    var image = images[i];
    alert(image.getAttribute("src"));
}
```

Jeśli chcielibyśmy wyznaczyć wszystkie elementy `img` bez określania wartości klasy, wywołanie `querySelectorAll` mogłoby wyglądać tak:

```
var images = document.querySelectorAll("img");
```

Jeśli chcielibyśmy znaleźć tylko ten obraz, którego atrybut `src` jest ustawiony na `meh.png`, można by zrobić to następująco:

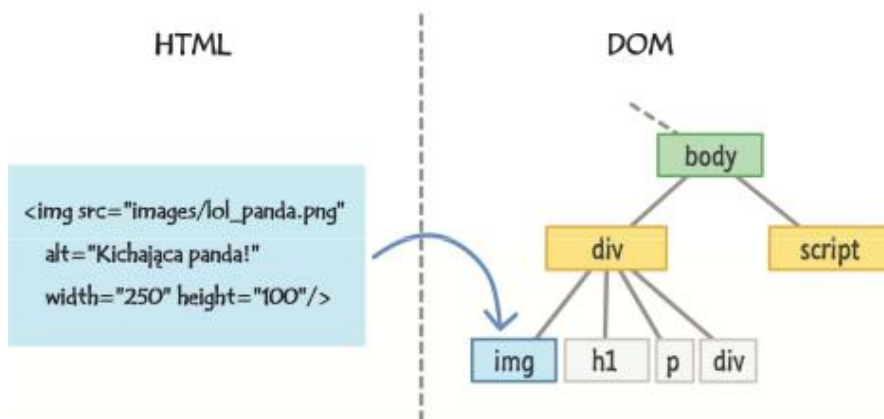
```
var images = document.querySelectorAll("img[src='meh.png']");
```

MODYFIKOWANIE ELEMENTÓW DOM

```

```

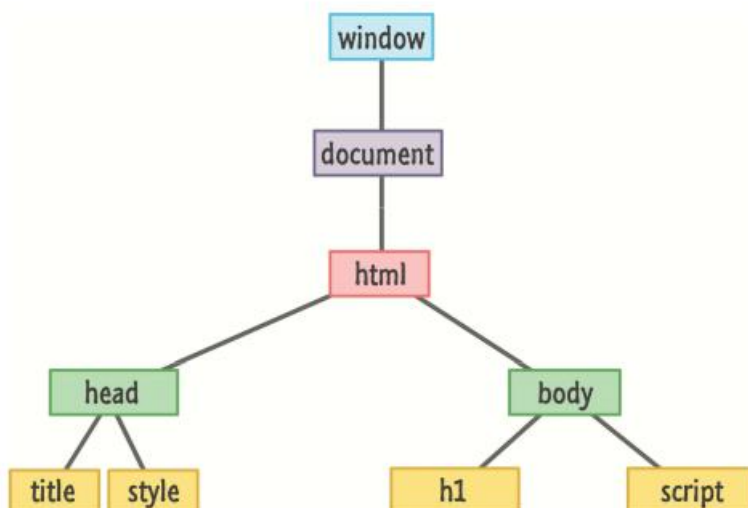
Gdy przeglądarka parsuje dokument i dociera do tego elementu obrazu, tworzy w strukturze DOM reprezentujący go węzeł.



DOM przechowuje wpis dla tego elementu obrazu w odpowiedniej lokalizacji.

```
<!DOCTYPE html>
<html>
<head>
  <title>Witaj...</title>
  <style>
    .highlight {
      font-family: "Arial";
      padding: 30px;
    }
    .summer {
      font-size: 64px;
      color: #0099FF;
    }
  </style>
</head>
<body>
  <h1 id="theTitle" class="highlight summer">Co się dzieje?</h1>
  <script>
  </script>
</body>
</html>
```

Gdy przełożymy to teraz na strukturę DOM



- Zmiana wartości tekstowej elementu

```
<body>
  <h1 id="theTitle" class="highlight summer">Co się dzieje?</h1>
  <script>
    var title = document.querySelector("#theTitle");
    title.textContent = "Oppa Gangnam Style!";
  </script>
</body>
```

- pobieranie wartości atrybutu

```
<body>
  <h1 id="theTitle" class="highlight summer">Co się dzieje?</h1>
  <script>
    var title = document.querySelector("h1");
    alert(title.getAttribute("id"));
  </script>
</body>
```

- ustawianie wartości atrybutu

```
<body>
  <h1 id="theTitle" class="highlight summer">Co się dzieje?</h1>
  <script>
    document.body.setAttribute("class", "bar foo");
  </script>
</body>
```

Ustawiamy atrybut class w elemencie body i przypisujemy mu wartość foo bar.

Innym sposobem ustawiania wartości atrybutów jest wykorzystanie id i className

```
<body>
  <h1 id="theTitle" class="highlight summer">Co się dzieje?</h1>
  <script>
    var title = document.querySelector("h1");
    alert(title.id);
    document.body.className = "bar foo";
  </script>
</body>
```

Źródła:

- <https://www.guru99.com/how-to-use-dom-and-events-in-javascript.html>
- Projektowanie stron internetowych. Przewodnik dla początkujących webmasterów po HTML5, CSS3 i grafice. Jennifer Niederst Robbins
- JavaScript. Przewodnik dla absolutnie początkujących Chinnathambi Kirupa