

# DOM (Document Object Model)

© Copyright by 3bird Projects 2021, <http://edukacja.3bird.pl>

## Czym jest DOM?

Platforma, która pozwala skryptom na dynamiczny dostęp do zawartości strony i jej aktualizację. Elementy można dodawać, zmieniać lub usuwać.

### Uwaga:

- Utworzony **obiekt** (element bądź tekst), znika po jego jednokrotnym użyciu, zazwyczaj za pomocą **appendChild()** (nie można użyć tego samego obiektu dwukrotnie lub w dwóch miejscach). Można za to wielokrotnie używać tego samego identyfikatora miejsca (*getElementById*) w strukturze DOM.
- Jeśli chcielibyśmy dodać jednocześnie dwa elementy, możemy użyć czysto javascriptowej metody **append(zmienna1, zmienna2, text)**.
- Nie trzeba (a nawet nie należy) tworzyć zamkniętych elementów / tagów, np. `</a>`. DOM tworzy zamknięcia od razu przy tworzeniu elementów.

## Specyfikacje

- Poziomy *DOM*: 1.0, 2.0, 3.0.
- Opis: [www.w3c.org](http://www.w3c.org)

## Struktura

Interfejs - metoda - atrybut

## Przykłady praktyczne

*/\* Piąte TD \*/*

```
mtd5 = document.createElement("td");
mtd5.className = "menu";
mtd5.title = "Odnośniki, ciekawe materiały...";
mtd5.id = "menutd5";
mkomorka5 = document.getElementById("menutr");
mkomorka5.appendChild(mtd5);
```

*/\* Piąte A \*/*

```
ma5 = document.createElement("a");
ma5.href = "linki.shtml";
ma5.className = "menu";
ma5.id = "menua5";
mkotwica5 = document.getElementById("menutd5");
mkotwica5.appendChild(ma5);
mtekstDoA5 = document.createTextNode("Linki");
mmiejsceTekstuDoA5 = document.getElementById("menua5");
mmiejsceTekstuDoA5.appendChild(mtekstDoA5);
```

## Adres e-mail

W sekcji <body>:

```
<span id="emailTekst"></span>
```

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

```
// ===== Zawartość pliku email.js =====
```

```
user = "ewa";
```

```
domena = "op.pl";
```

```
tematListu = "?Subject=Ze strony ABC";
```

```
// Poniżej nie zmieniaj NIC:
```

```
malpa = "@";
```

```
adres = user+malpa+domena;
```

```
ahref = "mailto:"+adres+tematListu;
```

```
// =====
```

```
// Tworzenie elementu A w pamięci
```

```
aEmailTekst = document.createElement("a");
```

```
aEmailTekst.href = ahref;
```

```
aEmailTekst.id = "mojEmail1";
```

```
// Miejsce wstawienia elementu A (zazwyczaj wewnątrz elementu <span>)
```

```
kotwicaEmailTekst = document.getElementById("emailTekst");
```

```
// Polecenie wstawiające element A z pamięci na stronę www
```

```
kotwicaEmailTekst.appendChild(aEmailTekst);
```

```
// =====
```

```
// Tworzenie w pamięci napisu wewnątrz elementu A
```

```
tekstAdres = document.createTextNode(adres);
```

```
// Wybranie miejsca wstawienia napisu do elementu A
```

```
miejsceNapisuDoAEmailTekst = document.getElementById("mojEmail1");
```

```
// Polecenie wstawiające napis z pamięci do wnętrza elementu A
```

```
miejsceNapisuDoAEmailTekst.appendChild(tekstAdres);
```

## Wyszukiwanie miejsc

**getElementById(„test”)** - wyszukuje tylko jeden element na podstawie „id”

**getElementsByTagName(„p”)[0]** - wyszuka wszystkie znaczniki <p> i ustawia fokus na pierwszym z nich (element tablicy [0] jest konieczny w tej metodzie);

**document.getElementById(„menu”).getElementsByTagName(„p”)[0]** - wyszuka wszystkie znaczniki <p> znajdujące się wewnątrz znacznika <div id=„menu”> i ustawia fokus na pierwszym z nich;

## Definiowanie stylów

```
mojaTabela = document.getElementById('tabela');
mojaTabela.style.borderWidth="1px";
// lub z wieloma parametrami
nazwaObiektu.style.cssText = "text-align: center; font-size: 16px; color: #ffffff";
// Generowanie stylów dla całej strony:
style = document.createElement("style");
miejsceNaStyle = document.getElementsByTagName("head")[0];
miejsceNaStyle.appendChild(style);
trescStylow = document.createTextNode('img.tapety {border-style: none; box-shadow: none;
border-radius: 0px; display: inline; margin: 5px;}');
miejsceNaTrescStylow = document.getElementsByTagName("style")[0];
miejsceNaTrescStylow.appendChild(trescStylow);
```

## Zamiana istniejącego tekstu

```
miejsceUsunieciaZawartosci = document.getElementById("moje_id");
miejsceUsunieciaZawartosci.replaceChild(nowyObiekt, staryObiekt.firstChild);
```

## Usuwanie znacznika wraz z zawartością

```
miejsceUsunieciaZawartosci = document.getElementById("moje_id");
konkretnaZawartosc = miejsceUsunieciaZawartosci.parentNode;
konkretnaZawartosc.removeChild(miejsceUsunieciaZawartosci);
```

## Usunięcie samej zawartości znacznika

```
miejsceUsunieciaZawartosci = document.getElementById("moje_id");
miejsceUsunieciaZawartosci.removeChild(miejsceUsunieciaZawartosci.firstChild);
```

## Odwołanie się do wartości znacznika

```
var miejsceZapisuEmail = document.getElementById('txtEmail');
var trescEmaila = miejsceZapisuEmail.value;
```

## Sprawdzanie czy znacznik ma atrybut

```
miejsceZnacznika = document.getElementById("idZnacznika");
if (miejsceZnacznika.hasAttribute("align")) {
    alert("Znacznik ma atrybut align");
}
```

## Sprawdzanie czy znacznik ma jakiegokolwiek atrybuty

```
miejsceZnacznika = document.getElementById("idZnacznika");
if (miejsceZnacznika.hasAttribute()) {
    alert("Znacznik ma jakieś atrybuty");
}
```

## Sprawdzanie czy opcja jest wybrana

```
miejsceZaznaczeniaOpcji = document.getElementById("idZnacznika");
if (miejsceZaznaczeniaOpcji.checked) {
    alert("Opcja jest zaznaczona.");
}
to samo standardową metodą JavaScripts:
if (document.forms[0].nazwaPola[1].checked) {
    alert("Opcja jest zaznaczona.");
}
```

## Tworzenie elementu wywołującego funkcje

```
function testowa() {
    alert(„Hello!");
}
kotwica = document.createElement(„a”);
kotwica.id = „CośTam”;
kotwica.onclick = testowa; // Nazwa funkcji bez nawiasów!!!
lub
kotwica.setAttribute(‘onclick’,‘testowa’);
```

## Włączanie / wyłączanie klasy

Klikając w element, raz włączamy mu jakąś klasę, po powtórny kliknięciu usuwamy ją (i tak na okrągło). Do poniższego kodu należy utworzyć element, w którego będziemy klikać i który będzie wywoływał „onclick”.

CSS:

```
.ukryty
{display: none;}
```

HTML:

```
<div id="Pudełko">Moje menu</div>
```

DOM:

```
ElementDoPokazania = document.getElementById(„Pudelko”);
ElementDoPokazania.classList.toggle("ukryty");
```

## Efekt Hover

```
<a id="link" href="http://"><b>aa</b></a><br>
<script>
function over(e) {
    textNode = document.createTextNode(':');
    this.insertBefore(textNode, this.childNodes[0]);
}
function out(e) {
    this.removeChild(this.childNodes[0]);
}
```

```
document.getElementById("link").onmouseover = over;
document.getElementById("link").onmouseout = out;
</script>
```

## Efekt Focus

Pseudo klasa `:focus` (w arkuszu stylów) działa tylko ze znacznikami edytowalnymi, np. `textarea`, `button`, `input`, a. Nie działa z `div`, `p`, `td`. Można za to rozwiązać w ten sposób:

```
<table>
<tr>
<td id="komorka1" style="background-color: green;" ><a href="#"
onclick="document.getElementById('komorka1').style.backgroundColor='red';
document.getElementById('komorka2').style.backgroundColor='green'">Komórka 1</a> jakiś
inny tekst</td>
<td id="komorka2" style="background-color: green;" ><a href="#"
onfocus="document.getElementById('komorka2').style.backgroundColor='red';
document.getElementById('komorka1').style.backgroundColor='green'">Komórka 1</a> jakiś
inny tekst</td>
</tr>
</table>
```

Można to zrobić także z użyciem funkcji:

```
znacznikA.onfocus = function () { document.getElementById("td1").style.backgroundColor =
'yellow'; };
```

W powyższym przypadku, po kliknięciu w odnośnik `<a>` zmieniane jest tło komórki `<td>` (jeśli akurat menu mamy w tabeli).

## Menu z efektem Focus

Jeśli menu stworzone jest w tabeli:

```
mtd = document.createElement("td");
/* Poniżej sprawdzamy pierwsze 5 liter w nazwie aktualnie załadowanego pliku */
var pelnaSciezka = window.location.pathname;
var nazwaPliku = pelnaSciezka.substring(pelnaSciezka.lastIndexOf('/')+1);
var krotkaNazwaPliku = nazwaPliku.slice(0, nazwaPliku.length-5);
var pierwsze5znakow = krotkaNazwaPliku.slice(0,5);
if (pierwsze5znakow == "infor") {
    var klasaPrzyciskuTD1 = "menuActive";
    var klasaPrzyciskuTD2 = "menu";
}
else if (pierwsze5znakow == "etyka") {
    var klasaPrzyciskuTD1 = "menu";
    var klasaPrzyciskuTD2 = "menuActive";
}
else {
    var klasaPrzyciskuTD1 = "menu";
    var klasaPrzyciskuTD2 = "menu";
}

mtd.className = klasaPrzyciskuTD1;
```

```
mtd.title = "Informatyka szkolna, zajęcia komputerowe...";  
mtd.id = "menutd1";  
mkomorka1 = document.getElementById("menutr");  
mkomorka1.appendChild(mtd);
```

## Jak wczytać zawartość innego pliku?

Za pomocą technologii *Ajax*, można wczytywać pliki do *<div>*. Ale mogą to być tylko pliki znajdujące się w tej samej domenie, co strona źródłowa (są to ograniczenia ze względu na bezpieczeństwo). Do wczytywania zewnętrznych plików można też użyć *<iframe>* lub *<object>*.