

XML + Ajax + HTML + CSS + JS + DOM

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

Ogólne

Projekt składa się z trzech plików (strony www, bazy XML, skryptu JS). Wymagane jest konto hostingowe, aby przetestować jego działanie (nie działa lokalnie). Kod należy przepisać ze zrozumieniem.

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępy. Kod należy przepisać ze zrozumieniem.

Plik książki.html

```
<!DOCTYPE html>
```

```
<html lang="pl">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Baza XML</title>
```

```
  <style>
```

```
    th.xml
```

```
    {border-style: solid;
    border-width: 2px;
    border-color: #c0c0c0;
    background-color: #000000;
    color: #ffffff;
    padding: 5px;}
```

```
    td.xml1kolumna
```

```
    {border-style: solid;
    border-width: 2px;
    border-color: #c0c0c0;
    padding: 7px;}
```

```
    td.xml2kolumna
```

```
    {border-style: solid;
    border-width: 2px;
    border-color: #c0c0c0;
    font-style: italic;
    padding: 7px;}
```

```
    td.xml3kolumna
```

```
    {border-style: solid;
    border-width: 2px;
    border-color: #c0c0c0;
    text-align: center;
    padding: 7px;}
```

td.xml4kolumna

```
{border-style: solid;
border-width: 2px;
border-color: #c0c0c0;
text-align: center;
padding: 7px;
vertical-align: bottom;}
```

td.xml4kolumna > img

```
{margin-bottom: 0px;
border-radius: 0;}
```

</style>**</head>****<body>**

```
<table style="border-style: solid; border-width: 2px; width: 100%; border-collapse: collapse;">
```

```
<colgroup span="4"></colgroup>
```

```
<thead>
```

```
<tr><th class="xml">Autor</th><th class="xml">Tytuł</th><th class="xml">Cena</th><th class="xml">Okładka</th></tr>
```

```
</thead>
```

```
<tbody id="tbody-xml"> </tbody>
```

</table>

```
<script src="wczytanie-xml.js" async></script>
```

</body>**</html>**

Plik książki.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<katalog>
```

<ksiazka>

```
<autor>Janina Brzostowska</autor>
```

```
<tytul>"Lubuskie dziewczyny"</tytul>
```

```
<wydawnictwo>Ludowa Spółdzielnia Wydawnicza</wydawnictwo>
```

```
<rok_wydania>1987</rok_wydania>
```

```
<cena>10zł</cena>
```

```
<okladka>dziejanny.jpg</okladka>
```

</ksiazka>**<ksiazka>**

```
<autor>Janina Katz</autor>
```

```
<tytul>"Pucka"</tytul>
```

```
<wydawnictwo>Jacek Santorski Co</wydawnictwo>
```

```
<rok_wydania>2008</rok_wydania>
```

```

        <cena>38zł</cena>
        <okladka>pucka.jpg</okladka>
    </ksiazka>

    <ksiazka>
        <autor>Wilhelm Dilthey</autor>
        <tytul>"O istocie filozofii"</tytul>
        <wydawnictwo>PWN</wydawnictwo>
        <rok_wydania>1987</rok_wydania>
        <cena>12zł</cena>
        <okladka>filozofia.jpg</okladka>
    </ksiazka>
</katalog>

```

Plik wczytanie-xml.js

```
// Skrypt wczytujący dane z pliku XML na stronę HTML.
```

```
// Tworzymy nowy obiekt XMLHttpRequest i zapisujemy go pod zmienną "zapytanieXML"
var zapytanieXML = new XMLHttpRequest();
```

```
/* Poniżej, parametr "true" oznacza asynchroniczność (dokument jako całość jest wczytywany pomimo, że jego poszczególne części, tj. sam plik XML, nie zostały wczytane). Jest to wyraźny zwrot w stronę filozofii AJAX. Większość przeglądarek internetowych, nie obsługuje już parametru "false". Poniższe dwie linie można również umieścić za sekcją onreadystatechange, ale ładowanie dokumentu będzie wtedy o wiele dłużej trwać. */
```

```
zapytanieXML.open('GET', 'ksiazki.xml', true);
zapytanieXML.send();
```

```
/* Przypisujemy obiektowi właściwość onreadystatechange, która spowoduje ponowne uruchomienie funkcji za każdym razem, gdy coś się zmieni, np. gdy w końcu zostanie wczytany plik XML, co zajmie jakieś 5 milisekund po wczytaniu całej strony: */
```

```
zapytanieXML.onreadystatechange = function() {
    // Sprawdzamy czy dane (odpowieź) są gotowe i pełne:
    if (zapytanieXML.readyState == 4 && zapytanieXML.status == 200) {
        // Powyższa linijka może być także zapisana w postaci:
        // if (zapytanieXML.readyState === XMLHttpRequest.DONE) {
        // lub w postaci:
        // if (this.readyState === XMLHttpRequest.DONE) {
        // Dla celów czysto diagnostycznych, zapisujemy na konsoli komunikat,
        // iż funkcja została wykonana o określonym czasie:
        // console.log("zapytanieXML.readyState zostało w końcu
        // zrealizowane - "+Date.now());

        // Gdy mamy już gotowe w pamięci wszystkie dane:
        var dokumentXML=zapytanieXML.responseXML;

        // Wyszukuje wszystkie znaczniki <ksiazka>
    }
}
```

```

var x=dokumentXML.getElementsByTagName("ksiazka");
for (i=0;i<x.length;i++) {
    // Potrzebne zmienne DOM. Po każdej pętli dodawany jest kolejny numer:
    /* Tworzymy TR */
    elementTR = document.createElement("tr");
    elementTR.id = "xmlTR"+[i]; //ID jest powiększany w każdej pętli o 1
    miejsceNaTR = document.getElementById("tbody-xml");
    miejsceNaTR.appendChild(elementTR);

    /* Tworzymy TD 1 */
    element1TD = document.createElement("td");
    element1TD.id = "xml1TD"+[i];
    element1TD.className = "xml1kolumna";
    miejsceNa1TD = document.getElementById("xmlTR"+[i]);
    miejsceNa1TD.appendChild(element1TD);

    // Zawartość bazy XML wyciągana jest jako pole tekstowe
    // (uwaga: jedna długa pomarańczowa linia!)
    wartosc1XML = dokumentXML.createTextNode(x[i].
getElementsByTagName("autor")[0].childNodes[0].nodeValue);
    miejsceNaWartosc1XML = document.getElementById("xml1TD"+[i]);
    miejsceNaWartosc1XML.appendChild(wartosc1XML);

    /* Tworzymy TD 2 */
    element2TD = document.createElement("td");
    element2TD.id = "xml2TD"+[i];
    element2TD.className = "xml2kolumna";
    miejsceNa2TD = document.getElementById("xmlTR"+[i]);
    miejsceNa2TD.appendChild(element2TD);

    wartosc2XML = dokumentXML.createTextNode(x[i].
getElementsByTagName("tytul")[0].childNodes[0].nodeValue);
    miejsceNaWartosc2XML = document.getElementById("xml2TD"+[i]);
    miejsceNaWartosc2XML.appendChild(wartosc2XML);

    /* Tworzymy TD 3 */
    element3TD = document.createElement("td");
    element3TD.id = "xml3TD"+[i];
    element3TD.className = "xml3kolumna";
    miejsceNa3TD = document.getElementById("xmlTR"+[i]);
    miejsceNa3TD.appendChild(element3TD);

    wartosc3XML = dokumentXML.createTextNode(x[i].
getElementsByTagName("cena")[0].childNodes[0].nodeValue);
    miejsceNaWartosc3XML = document.getElementById("xml3TD"+[i]);
    miejsceNaWartosc3XML.appendChild(wartosc3XML);

    /* Tworzymy TD 4 */
    element4TD = document.createElement("td");

```

```

element4TD.id = "xml4TD"+[i];
element4TD.className = "xml4kolumna";
miejsceNa4TD = document.getElementById("xmlTR"+[i]);
miejsceNa4TD.appendChild(element4TD);

wartosc4XML = document.createElement("img");
wartosc4XML.src = x[i].getElementsByTagName("okladka")
[0].childNodes[0].nodeValue;
wartosc4XML.setAttribute("width", "60");
wartosc4XML.setAttribute("height", "auto");
miejsceNaWartosc4XML = document.getElementById("xml4TD"+[i]);
miejsceNaWartosc4XML.appendChild(wartosc4XML);
    }
}
else {
    // Dla celów czysto diagnostycznych, wysyłamy na konsolę komunikat:
    // console.log("zapytanieXML.ReadyState nadal nie zostało zrealizowane -
"+Date.now());
}
}

```

===== INFORMACJE KOŃCOWE =====

1. W plikach XML można także umieszczać nazwy plików graficznych, a tym samym wczytywać na stronę fotki.
2. Obiekt *XMLHttpRequest* umożliwia wysyłanie zapytań do serwera po tym, gdy strona zostanie załadowana; umożliwia także otrzymywanie danych ze serwera po tym, jak strona zostanie załadowana.

Ostatnia aktualizacja: 17 grudnia 2022.