

Ajax pochodzi od Asynchronous JavaScript and XML, co oznacza „asynchroniczny JavaScript i XML”.

Jest techniką pozwalającą na komunikację i wymianę danych między stroną WWW a serwerem. Operacje te wykonywane są w tle-bez konieczności przeładowania witryny, tak jak ma to miejsce w klasycznych technikach. Dzięki temu witryna zachowuje się podobnie jak zwykły program uruchomiony pod kontrolą systemu operacyjnego.

Obiekt **XMLHttpRequest** jest niezbędny do wymiany danych między serwerem a stroną WWW. Tworzenie go różni się w zależności od użytej przeglądarki. Obiekt XMLHttpRequest posiada właściwości i metody, które różnią się nieco w zależności od wybranej przeglądarki, jednak zestaw podstawowych cech wykorzystywanych w skryptach Ajax-a jest wspólny.

Przykład.

Tworzenie obiektu XMLHttpRequest w zależności od przeglądarki

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

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}

else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

if (XMLHttpRequestObject)
{
    document.write("obiekt został utworzony");
}

else
{
    document.write("obiekt nie został utworzony");
}

</script>

</body>
</html>
```

Przykład 1.

Napisz skrypt wysyłający do serwera za pomocą obiektu XMLHttpRequest żądanie typu GET. Wyświetl statusy operacji oraz numery kolejnych wywołań funkcji obsługujących transmisję.

```
<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function pobierzDane()
{
    if (XMLHttpRequestObject)
    {
        var calls=0;
        var div=document.getElementById("div1");
        div.innerHTML="";
        XMLHttpRequestObject.open("GET","dane.txt");

        XMLHttpRequestObject.onreadystatechange = function()
        {
            calls ++;
            var tekst="wywołanie nr" + calls + "<br/>";
            tekst+="readyState=" + XMLHttpRequestObject.readyState+"<br/>";

            if(XMLHttpRequestObject.readyState!=1)
            {
                tekst+="status=" + XMLHttpRequestObject.status + "<br />";
            }
            else
            {
                tekst+="status=niedostepny<br />";
            }
            tekst += "<br />";
            div.innerHTML=div.innerHTML + tekst;

            XMLHttpRequestObject.send(null);
        }

    }

}

</script>
</head>

<body>

<div>
<input type="button" value="kliknij" onclick="pobierzDane();" />
</div>

<div id="div1">

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

Zainicjowanie żądania, które ma być wysłane na serwer za pomocą metody open. Więcej o metodzie open po przykładzie.

Funkcja, która służy do odbierania danych.

Odpowiedź otrzymana z serwera

Wysłanie żądania za pomocą metody send. Jej wywołanie ma postać send(null)

Tu będzie wpisywana odpowiedź z serwera

Metoda `open` ma następujący nagłówek:

`open(metoda_transmisji,url,async,uzytkownik,haslo)`

metoda_transmisji-metoda transmisji danych – GET,POST,PUT

url-adres pliku lub skryptu umieszczony na serwerze (jeśli w tym samym katalogu, co pliku html, to wystarczy sama nazwa)

async-określenie, czy transmisja ma być asynchroniczna (domyślnie jest true-dzięki temu po wysłaniu żądania do serwera skrypt nie będzie oczekiwał na transmisję danych, tylko będzie kontynuował działanie)

uzytkownicy, *haslo* – używamy tylko w sytuacji, kiedy dostęp do serwera wymaga autoryzacji.

np. `open(„GET”,„dane.txt”)` oznacza przygotuj obiekt `XMLHttpRequest` do pobrania za pomocą metody GET z serwera `dane.txt`.

Nie oznacza to jednak wysyłania żądania do serwera, a jedynie przygotowanie żądania.

Przykład2.

Napisz funkcję anonimową odbierającą dane przesłane z serwera w odpowiedzi na żądanie wysłane za pomocą obiektu `XMLHttpRequest`.

Utwórz na serwerze plik `dane.txt`, a w nim swoje imię i nazwisko.

```
<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function pobierzDane()
{
    if (XMLHttpRequestObject)
    {
        var div=document.getElementById("div1");
        XMLHttpRequestObject.open("GET", "dane.txt", "true", "tu login", "tu haslo");

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {
                div.innerHTML=XMLHttpRequestObject.responseText;
            }
        }

        XMLHttpRequestObject.send(null);
    }
}

</script>

</head>

<body>
```

```

<div>
<input type="button" value="kliknij" onclick="pobierzDane();" />
</div>

<div id="div1">
ten test zostanie zmieniony
</div>

</body>
</html>

```

Przykład 3

Umieść na witrynie 3 przyciski. Kliknięcie każdego z nich powinno powodować pobranie z serwera zawartości przypisanego mu pliku i wyświetlanie jej na stronie. Utwórz na serwerze trzy pliki dane1.txt, dane2.txt, dane3.txt o różnej zawartości.

```

<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function pobierzDane(tekstId)
{

    if (XMLHttpRequestObject)
    {

        var div=document.getElementById("div1");

        switch(tekstId)
        {
            case 1:
                XMLHttpRequestObject.open("GET","dane1.txt","true","tu login","tu haslo");
                break;
            case 2:
                XMLHttpRequestObject.open("GET","dane2.txt","true","tu login","tu haslo");
                break;
            case 3:
                XMLHttpRequestObject.open("GET","dane3.txt","true","tu login","tu haslo");
                break;
            default:
                return;
        }

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {

                div.innerHTML=XMLHttpRequestObject.responseText;
            }
        }

        XMLHttpRequestObject.send(null);
    }
}

```

```

}

</script>

</head>

<body>

<div>
<input type="button" value="pierwszy" onclick="pobierzDane(1);" />
<input type="button" value="drugi" onclick="pobierzDane(2)">
<input type="button" value="trzeci" onclick="pobierzDane(3)">
</div>

<div id="div1">
ten test zostanie zmieniony
</div>

</body>
</html>

```

Przykład 4.

Napisz skrypt wyświetlający na witrynie przycisk. Po kliknięciu przycisku z serwera ma zostać pobrana nazwa pliku z obrazem, a sam obraz powinien zostać wyświetlony na stronie. Wstawiamy na serwer plik z obrazem oraz plik obraz.txt, którego zawartością będzie nazwa pliku z obrazem (wraz z rozszerzeniem).

```

<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function pobierzDane()
{
    if (XMLHttpRequestObject)
    {
        var div=document.getElementById("div1");
        XMLHttpRequestObject.open("GET","obraz.txt","true","tu login","tu haslo");

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {
                var obraz=XMLHttpRequestObject.responseText;
                var tag="<img src=\"\" + obraz + \"\" alt=\"opis obrazu\">";
                div.innerHTML=tag;
            }
        }

        XMLHttpRequestObject.send(null);
    }
}

```

```

</script>

</head>

<body>

<div>
<input type="button" value="kliknij" onclick="pobierzDane();" />
</div>

<p>
<div id="div1">

<p>
tu bedzie obraz

</div>

</body>
</html>

```

Przykład 5.

Umieść na witrynie przycisk tekstowy. Każde kliknięcie przycisku powinno powodować pobranie danych z kolejnego pliku tekstowego i wyświetlenie tego obrazu w warstwie HTML. Wstawiamy na serwer pliki z obrazami oraz plik obraz1.txt, obraz2.txt, obraz3.txt, którego zawartością będzie nazwa pliku z obrazem (wraz z rozszerzeniem).

```

<html>
<head>

<script type="text/javascript">
var XMLHttpRequestObject=false;
var ile=3;
var aktualny=1;

if (window.XMLHttpRequest)
{
    XMLHttpRequestObject=new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
    XMLHttpRequestObject=new ActiveXObject("Microsoft.XMLHTTP");
}

function pobierzDane(nazwa)
{

    if (XMLHttpRequestObject)
    {

        var div=document.getElementById("div1");
        XMLHttpRequestObject.open("GET",nazwa,"true","tu login","tu haslo");

        XMLHttpRequestObject.onreadystatechange = function()
        {
            if(XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200)
            {

                var obraz=XMLHttpRequestObject.responseText;
                var tag="<img src=\"\" + obraz + \"\" alt=\"opis obrazu\">";
                div.innerHTML=tag;
            }
        }

        XMLHttpRequestObject.send(null);
    }
}

```

```

}

function kliknij()
{
    if (aktualny>ile)
    {
        aktualny=1;
    }

    var nazwa="obraz"+aktualny+".txt";
    aktualny++;
    pobierzDane(nazwa);
}

</script>

</head>

<body>

<div>
<input type="button" value="kliknij" onclick="kliknij();" />
</div>

<p>
<div id="div1">

<p>
tu bedzie obraz

</div>

</body>
</html>

```

Zadania

Zadanie1.

Napisz skrypt, który będzie pobierał z serwera dane zapisane w trzech różnych plikach. Wybór ma być dokonywany za pomocą listy rozwijalnej.

Zadanie2.

Napisz skrypt, w którym za pomocą listy rozwijalnej będzie można wybrać wyświetlany na stronie obraz. Nazwy obrazów mają być zawarte w plikach tekstowych umieszczonych na serwerze tak, aby była możliwa ich zamiana bez ingerencji w kod witryny.

Zadanie3.

Umieść na stronie trzy przyciski. Najechnie kursorem myszy na dowolny przycisk powinno powodować wczytanie danych z przypisanego mu pliku tekstowego. (podpowiedź : zdarzenie na ruch myszki onmouseover).

Zadanie4

Umieść na stronie WWW trzy przyciski. Najechnie kursorem myszy na przycisk ma spowodować wyświetlanie obrazu określonego przez przypisany przyciskowi plik tekstowy znajdujący się na serwerze.

Źródło : „Ajax – ćwiczenia”