# ORW Ćwiczenie 2 – Technologia AJAX

Autor: Łukasz Pawłowski

Grupa: 125NCI\_B

Link do projektu: <https://github.com/mojzesz/orw-www-cw2>

Język programowania: JavaScript

**Zadania do zrealizowania:**

**ORW ćw. 2**

1. Za pomocą wybranego przez siebie języka programowania zademonstruj działanie technologii AJAX.
2. Bądź gotowy do prezentacji możliwości i modyfikacji napisanego programu.
3. Możliwe jest zrealizowanie załączonych zadań lub samodzielny projekt.

**OOR ćw. 3**

1. Zademonstruj działanie obiektów Web Workers w celu rozwiązania wybranego problemu obliczeniowego.
2. Bądź gotowy do prezentacji możliwości i modyfikacji napisanego programu.

**Przykładowe pytania ORW ćw. 2**

* Co to jest AJAX i po co się go stosuje?

Asynchroniczny Javascript i XML – sposób tworzenia aplikacji internetowych, w których przy każdym wywołaniu Javascirpt asynchronicznie wywoływana jest część dokumentu. Stosuje się w celu przyspieszenia ładowania strony oraz usunięciu konieczności przeładowywania całego dokumentu za każdym razem.

* Czym jest obiekt XMLHttpRequest?

Klasa pozwalająca na wykonywanie zapytań HTTP do serwera nawet już po załadowaniu strony, umożliwiająca asynchroniczne przesyłanie danych.

* Co to jest model DOM? Narysuj i omów na prostym przykładzie.

Obiektowy model dokumentu (tutaj: dokumentu HTML najczęściej w standardzie W3C) określający sposób przechowywania i reprezentacji danych na stronach WWW.

* Scharakteryzuj zmienną readyState.

Przetrzymuje ona status XMLHttpRequest – przyjmuje wartości 0-4 – nie zainicjowany, jest połączenie, jest odebrane, processing, response ready.

* Scharakteryzuj metodę onreadystatechange.

Jest to metoda określająca wydarzenie, które następuje przy każdej zmianie statusu readyState.

**Przykładowe pytania OOR ćw. 3**

* W jaki sposób tworzony jest nowy Worker?

Nowy Worker tworzony jest przy pomocy utworzenia nowego obiektu klasy Worker zawierającygo skrypt JavaScript jako parametr, na przykład: *var w = new Worker(‘skrypt.js’);*

* Podaj rolę właściwości *onmessage* i *onerror*.

*Onmessage* jest wydarzeniem, które wywoływane jest, gdy worker zwróci dane przy pomocy funkcji *postMessage*. W ten sposób skrypt główny może komunikować się ze workerem. Wydarzenie *onerror* wywoływane jest, gdy worker trafi na błąd.

* Podaj rolę metod *postMessage* i *terminate*.

*postMessage* może przekazywać dane z workera do wątku głównego oraz od wątku głównego do workera. Metoda *terminate* służy do zakończenia działania workera.

* Co mogą, a czego nie potrafią zrealizować Web Workers?

Web Workers mogą wykonywać przeróżne obliczenia w tle bez konieczności zmniejszenia responsywności UI. Nie mogą natomiast manipulować drzewem DOM, co jest wymuszone ze względów bezpieczeństwa – aby przykładowo dwa wątki nie próbowały w różny sposób zmienić drzewa DOM jednocześnie. Komunikacja odbywa się tylko przy pomocy *postMessage()*.

* Scharakteryzuj na przykładzie komunikację asynchroniczną i synchroniczną.

Komunikacja asynchroniczna pozwala na nieregularne przesyłanie i odbieranie danych. Pozwala to na zaoszczędzenie czasu i transferu. Wątki asynchroniczne są od siebie niezależne – na przykład można jednocześnie ładować kilka części strony web, co inaczej musiałoby być przesłane w jednym dokumencie, w jednym żądaniu. W komunikacji synchronicznej dane są przesyłane zgodnie z zegarem w wyznaczonych momentach czasu. Taka transmisja może być konieczna, gdy kolejne dane transmitowane w różnych wątkach są od siebie zależne – na przykad dwie części działania są wykonywane przez dwa różne wątki, ale sumarycznie muszą być połączone w wątku głównym.

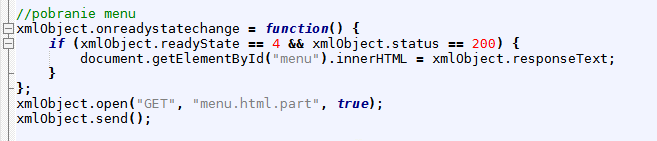
# Realizacja zadań

**Technologia AJAX – ORW ćw. 2**



W pierwszym boxie od góry przy pomocy technologii AJAX ładowane jest menu. Jest ono ładowane zaraz przy otwarciu strony z pliku menu.html.part – zawartość tego pliku wyświetlana jest bezpośrednio w pierwszym boksie od góry. Znajdują się tam cztery linki, których kliknięcie powoduje asynchroniczne pobranie odpowiedniego pliku \*.part do drugiego boksa.

Menu pobierane jest przy pomocy kodu:



Oznacza on, że treść pliku menu.html.part jest wyświetlana dopiero, kiedy zostanie poprawnie pobrana – a więc spełnione zostaną warunki, że obiekt xmlObject klasy XMLHttpRequest zwróci readyState o wartości 4 (co oznacza, że pobrane dane są gotowe) oraz stasusie 200 (pobieranie dokumentu zakończone sukcesem).

Każdy artykuł pobierany jest przy pomocy podobnego kodu, który przyjmuje jako parametr nazwę pliku, który ma otworzyć wybrany plik zawierający fragment dokumentu HTML i umieszczający go w drugim boksie.

**Web Workers – OOR ćw. 3**

Tworzone są dwa workers. Jeden odpowiedzialny jest za dodawanie kolejnych liczb, a po osiągnięciu liczby 4 zwraca błąd do wątku głównego. Drugi wątek losuje kolor, który w głównym wątku wykorzystywany jest do kolorowania tła. Ponadto przy wpisaniu danych do obiektu *input* do tekstu wyświetlanego w boksie wyników dodawany jest tekst z tymi danymi.

Obiekt input wykorzystuje wydarzenie *oninput*, co pozwala na wywołanie metody i przesłanie danych do wątku przez postMessage. W ten sposób, przy pomocy postMessage, została przedstawiona transmisja zarówno z workera jak i do workera.