

Uniwersytet Morski w Gdyni

przedmiot:

Narzędzia Informatyczne

Ćw. 9 Zastosowanie przeglądarek internetowych

1. Cel ćwiczenia

Celem ćwiczenia jest przedstawienie działania przeglądarek internetowych jako narzędzia wykorzystywanego przez informatyków, np. programistów aplikacji webowych czy administratorów sieci.

2. Wprowadzenie

Przeglądarki internetowe nie służą jedynie do odwiedzania stron internetowych. Są to rozbudowane narzędzia pozwalające na interakcję z treściami pochodzącymi z internetu, jak i plikami lokalnymi. Z zagadnieniem przeglądarek wiążą się następujące definicje:

- HTML – język znaczników służący do tworzenia struktury stron internetowych. Pozwala na umieszczanie elementów w formie znaczników (np. akapit `<p></p>`). Stanowi szkielet strony.
- CSS (Kaskadowy Arkusz Stylów) – pozwala na określanie wyglądu poszczególnych elementów strony internetowej. Może operować na znacznikach, klasach i identyfikatorach. Jest dołączany do strony HTML.
- DTD – (ang. *document type definition*) określa formalną strukturę dokumentów HTML, XML, XHTML i innych. Służy do określenia jakie znaczniki mogą zostać wykorzystane w danym dokumencie. Określa także właściwości oraz dostępne atrybuty poszczególnych znaczników.
- JavaScript – skryptowy język programowania pozwalający na interakcję z użytkownikiem. Wprowadza do stron internetowych dynamikę, opcję zmiany wyświetlanej zawartości na podstawie działań użytkownika (np. walidacja formularzy).
- Responsive Web Design – idea tworzenia stron internetowych w taki sposób by można było je prawidłowo wyświetlać na urządzeniach o różnych rozmiarach ekranu. Dzięki projektowaniu stron internetowych zgodnie z zasadami Responsive User Interface, strona internetowa dostosowuje swój layout do rozmiarów ekranu urządzenia użytkownika.

2.1 Struktura strony HTML

Każda strona internetowa bazująca na HTML musi spełniać zasady struktury wytyczone przez swoje DTD. Oznacza to, że język HTML jest zbiorem znaczników, których może użyć web developer według ściśle określonych warunków. Większość znaczników posiada właściwości oraz atrybuty. Przykładem atrybutu może być *src*, który służy określeniu źródła używanego obiektu. Natomiast właściwości określają w jaki sposób zachowa się treść znacznika oraz w jaki sposób ta treść zostanie sformatowana i wyświetlona. Właściwości edytować można stosując kaskadowe arkusze stylów.

HTML wraz z CSSem nadają się doskonale do stworzenia szablonu strony i umieszczeniu na niej danych, lecz

oprócz kilku zdefiniowanych znaczników, nie umożliwiają zbyt na interakcję z użytkownikiem. Aby wprowadzić dynamikę działania i rozszerzyć nieco funkcjonalność stron internetowych wprowadzono JavaScript. JavaScript pozwala nie tylko na edycję właściwości znaczników w czasie rzeczywistym, lecz także na dodawanie nowych elementów na stronie (np. ostrzeżeń lub odpowiedzi) kiedy użytkownik będzie ich potrzebował. Bardzo dobrym przykładem jest walidacja formularzy. Formularze są elementem języka HTML pozwalającym na wprowadzanie użytkownikowi danych na stronie w specjalnie przygotowane pola formularza. Dane następnie są przetwarzane przez stronę internetową na podstawie atrybutu *action* formularza (np. wprowadzony tekst przesyłany jest na określony w *action* adres mailowy). Problemem formularzy był niewielki zakres możliwości przetwarzania danych oraz brak możliwości interakcji z użytkownikiem podczas wprowadzania informacji do pól. Problem ten rozwiązuje JavaScript. Dzięki JS w czasie rzeczywistym można sprawdzać poprawność informacji wprowadzanej do pola oraz wyświetlać dotyczące tej danej komunikaty (np. hasło zbyt krótkie, nazwa użytkownika zajęta, itp.). Umieszczając funkcję JavaScript jako właściwość argumentu formularza *onsubmit*, możemy użyć jej do przetworzenia danych – nie jesteśmy już ograniczeni przez możliwości argumentu *action*.

Na każdej stronie internetowej umieszczona musi być informacja o typie dokumentu oraz trzy znaczniki: `<html>`, `<head>` oraz `<body>`.

Typ dokumentu deklaruje się w pierwszej linii kodu HTML. W przypadku HTML5 ta deklaracja przyjmuje postać: `<!DOCTYPE html>`.

Deklaracja ma za zadanie poinformować przeglądarkę internetową o zastosowanej wersji języka HTML. Dzięki deklaracji typu dokumentu przeglądarka internetowa wie, którym plikiem DTD się posłużyć do przetworzenia znaczników.

W kolejnej linii powinien pojawić się znacznik `<html>`. Jest to korzeń dokumentu HTML, który zawierać musi znaczniki `<head>` oraz `<body>`. Znacznik `<html>` może posiadać atrybuty, m.in. warto umieścić atrybut `lang`, by zdefiniować główny język strony (np. `<html lang="pl">`).

Jako pierwszy element w znaczniku `<html>` powinien pojawić się znacznik `<head>`. Znacznik ten służy definicji meta danych strony. `<head>` nie przyjmuje argumentów, lecz umieszcza się w nim znaczniki służące do: nadania tytułu strony internetowej (znacznik `<title>`), określenia strony kodowej (znacznik `<meta>`), dołączenia zewnętrznego kaskadowego arkusza stylów (znacznik `<link>`) oraz do zdefiniowania wewnętrznego arkusza stylów (znacznik `<style>`). W `<head>` można także umieścić skrypt (znacznik `<script>`), lecz uznane za dobrą praktykę jest umieszczanie skryptów jako ostatnie znaczniki w `<body>` - w ten sposób skrypt nie opóźnia załadowania strony internetowej.

Znacznikiem, który należy umieścić po `<head>` jest znacznik `<body>`. Jest to znacznik, w którym zostają umieszczone znaczniki umieszczające na stronie elementy (np. `<p>`, ``, itp.). Znacznik przyjmuje atrybuty dotyczące stanu strony (m.in. `onload`, `onresize`).

Każdy znacznik posiada określoną strukturę w pliku DTD. Przyjrzyjmy się definicji znacznika ``

```
<!ELEMENT IMG - O EMPTY          -- Embedded image -->
<!ATTLIST IMG
  %attrs;                -- %coreattrs, %i18n, %events --
  src    %URI;           #REQUIRED -- URI of image to embed --
  alt    %Text;          #REQUIRED -- short description --
  longdesc %URI;         #IMPLIED -- link to long description
                                (complements alt) --
  name    CDATA          #IMPLIED -- name of image for scripting --
  height  %Length;       #IMPLIED -- override height --
  width   %Length;       #IMPLIED -- override width --
  usemap  %URI;          #IMPLIED -- use client-side image map --
  ismap   (ismap)        #IMPLIED -- use server-side image map --
>
```

`` określone jest jako element pusty (EMPTY) – oznacza to, że nie ma on odpowiadającego znacznika zamykającego ``

DTD zawiera listę argumentów znacznika oraz ich opis. Określa także, które argumenty są wymagane (powinny zostać umieszczone, aby element prawidłowo funkcjonował), a które są opcjonalne. Jak widzimy wymaganymi argumentami są `src`, którego wartość określa lokalizację obrazu do umieszczenia oraz `alt`, którego wartość to krótki opis obrazka (tekst ten jest wyświetlany, jeśli obraz nie może zostać załadowany).

Nie musimy jednak czytać DTD, aby dowiedzieć się w jaki sposób działają poszczególne znaczniki – w tym celu lepiej posłużyć się dokumentacją HTML dostępną na stronie: <https://dev.w3.org/html5/html-author/>. Kaskadowe Arkusze Stylów służą do opisu sposobu wyświetlania się poszczególnych elementów. Dzięki CSS możemy zmieniać np. kolor tekstu umieszczonego w elemencie. Dużą zaletą CSS jest to, że pozwala oddzielić szablon i zawartość strony internetowej od sposobu jej wyświetlania. Pozwala to na tworzenie czystego kodu HTML. Wyróżniamy trzy typy stylów CSS ze względu na sposób ich umieszczenia na stronie HTML:

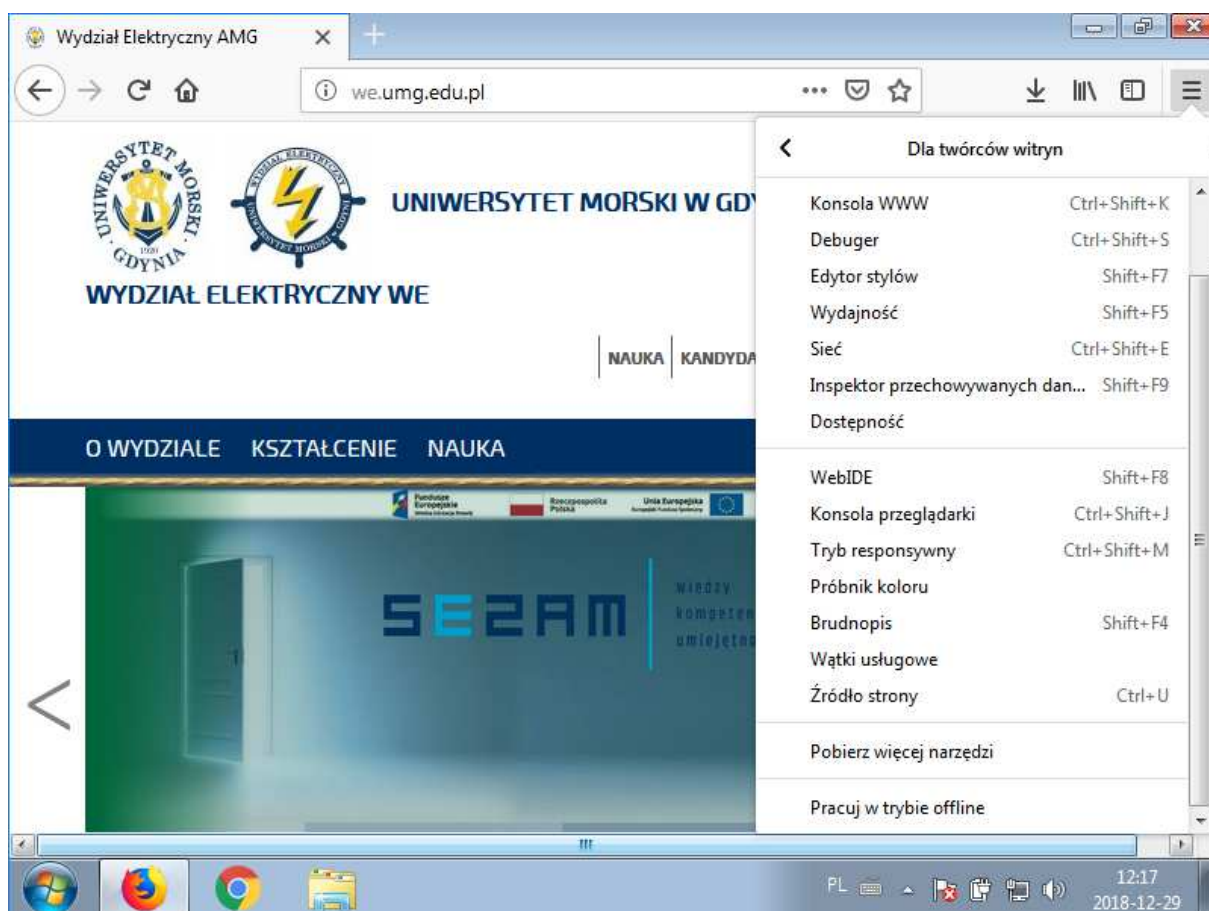
- Styl lokalny – styl lokalny występuje wtedy kiedy formatowanie stosujemy w kodzie HTML, jako argument znacznika (np. `<p style="color: red">Zawartość akapitu</p>`). Ten typ stylu ma najwyższą pozycję w hierarchii (pozostałe formatowania koloru czcionki tego akapitu są ignorowane). Pozwala to na zmianę sposobu formatowania np. pojedynczego akapitu na inny niż pozostałe akapity, nawet jeśli stosujemy też inny typ CSS np. zewnętrzny plik. W takim przypadku formatowanie stylu lokalnego zostanie scalone z formatowaniem stylu niższego w hierarchii, i jeśli wystąpią kolizje zastosowane zostanie formatowanie stylu lokalnego.
- Styl wbudowany – styl wbudowany to definicja stylu umieszczona w znaczniku `<style></style>`, w nagłówkowej części kodu HTML (w znaczniku `<head>`). Styl ten zastosowany jest dla całego pliku HTML. Jest wyżej w hierarchii niż styl zewnętrzny, lecz niżej niż styl lokalny. Jako, że styl wbudowany działa tylko w pliku HTML w którym zostaje umieszczony, stosowany jest głównie do zmiany formatowania podstron, kiedy w użyciu jest styl zewnętrzny dla wszystkich plików HTML strony internetowej.
- Styl zewnętrzny – styl zewnętrzny to definicja stylu umieszczona w osobnym pliku o rozszerzeniu `.css`. Styl następnie dołączany jest do pliku HTML poprzez znacznik `<link>` w nagłówkowej części kodu HTML (w znaczniku `<head>`). Jako, iż styl ten znajduje się w osobnym pliku, można go podłączyć do wielu plików HTML, np. do strony głównej i wszystkich podstron, tak aby otrzymać jednakowy wygląd całej strony internetowej. Styl ten jest najniżej w hierarchii.

3. Przykłady

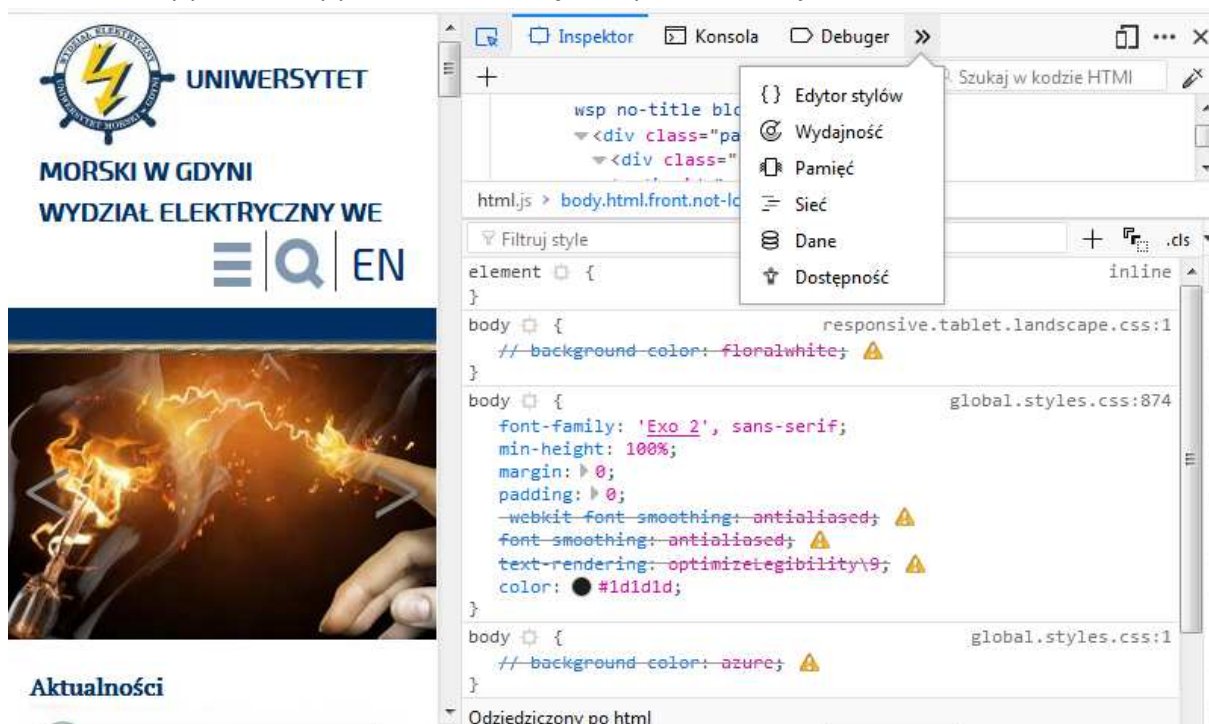
Najpopularniejszymi przeglądarkami są Mozilla Firefox oraz Google Chrome. Służą one użytkownikom głównie do przeglądania stron internetowych. Nie są to jednak ich jedyne opcje. Przeglądarki internetowe posiadają wbudowane narzędzia, które pozwalają nie tylko przeglądać strony, ale także je analizować, a nawet zmieniać. Lecz jak działa przeglądarka internetowa? W jaki sposób otwiera strony internetowe? Zadaniem przeglądarki internetowej jest rozwiązać wpisany adres internetowy dzięki usłudze DNS, a następnie pobrać stronę internetową z serwera WWW i ją wyświetlić. Najczęściej odbywa się to na portach 80, 443, 8080.

Uruchommy maszynę wirtualną ze skonfigurowaną kartą sieciową tak, aby miała dostęp do Internetu.

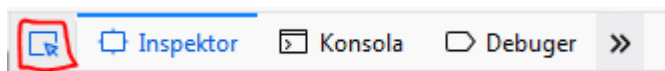
Wejdźmy na dowolną stronę internetową (w tym przypadku `we.umg.edu.pl`) i skorzystajmy z narzędzi „Dla twórców witryn” *Firefox*a. Aby je uruchomić należy nacisnąć przycisk *Menu*, następnie z menu kontekstowego wybrać *Dla twórców witryn*. Wybierzmy narzędzie np. *Konsola*.



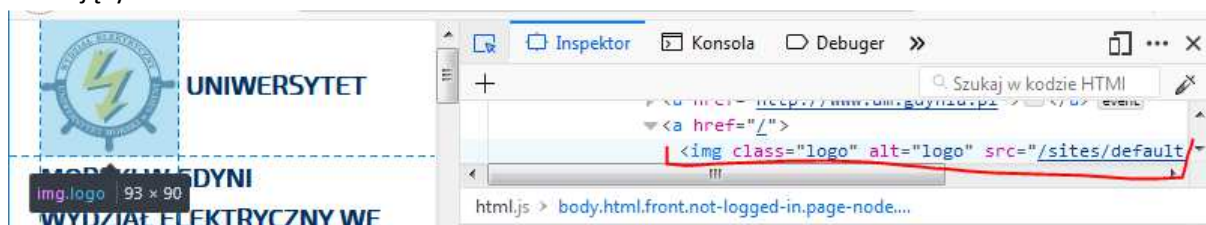
Zostanie otworzony panel, który pozwala na analizę strony internetowej.



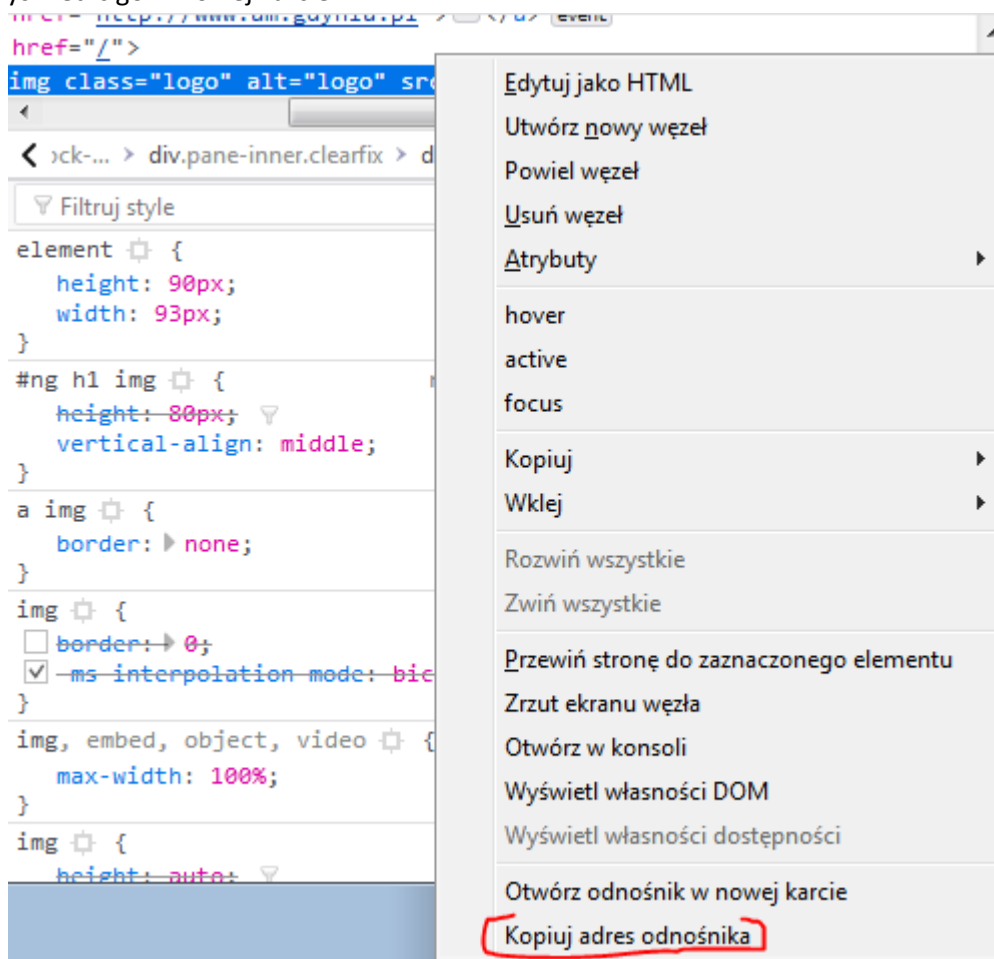
Wybermy zakładkę *Inspektor* i zaznaczmy opcję *Wskaż element na stronie*.

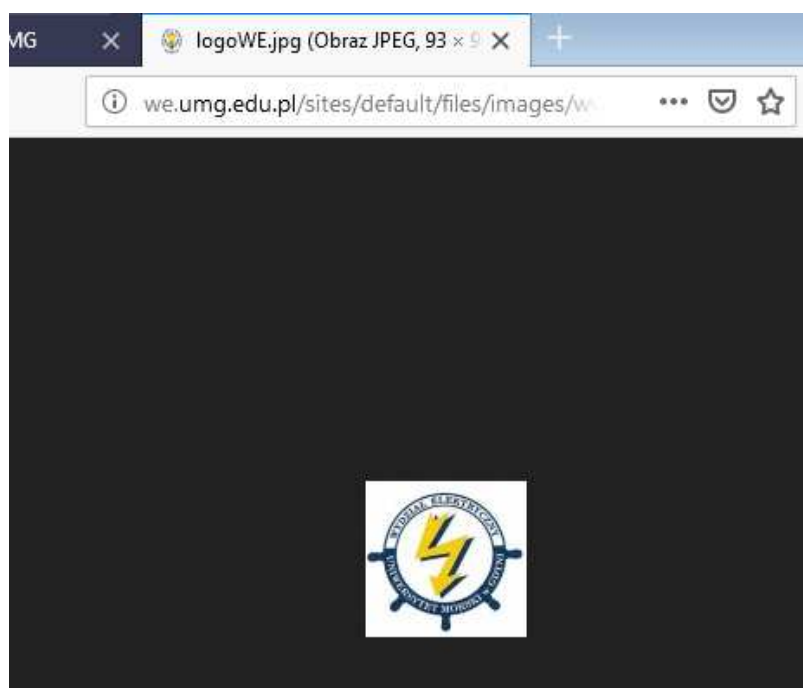


Inspektor pozwala na wyświetlenie kodu HTML oraz zawartości Kaskadowego Arkusza Stylów strony. Dzięki temu możemy przeanalizować jaki fragment kodu odpowiada za umieszczony na stronie element. Opcja *Wskazaj element na stronie* pozwala nam wskazać kursorem obiekt na stronie, a *Inspektor* sam zaznaczy nam odpowiadający za ten element fragment kodu. Najedźmy kursorem na dowolny element strony, aby wyświetlić odpowiadający za ten element kod.

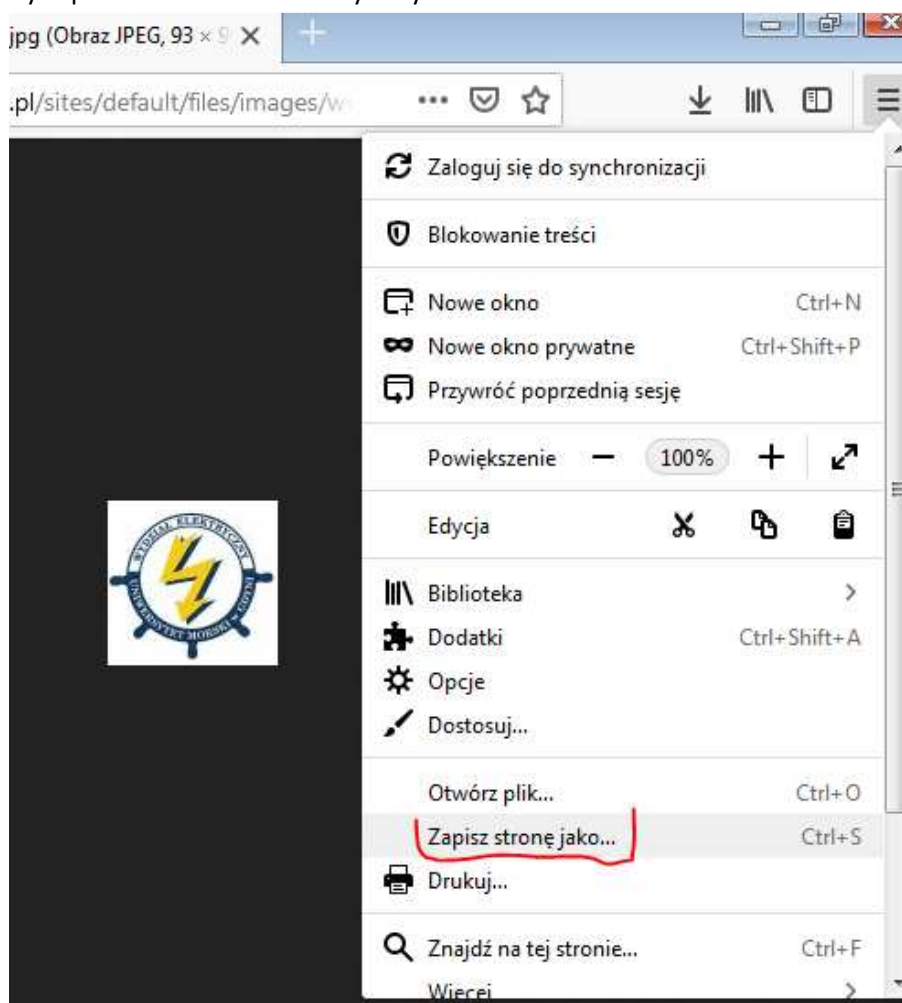


Jak widać element został umieszczony za pomocą znacznika *img*, klasą CSS odpowiadającą za formatowanie obiektu jest klasa *logo*, a atrybut *alt* (odpowiadający za zastąpienie obrazka wybranym tekstem, jeśli obrazek nie może zostać załadowany) posiada wartość „logo”, atrybut *src* to źródło obrazka. Możemy skopiować adres odnośnika i wyświetlić go w nowej karcie.



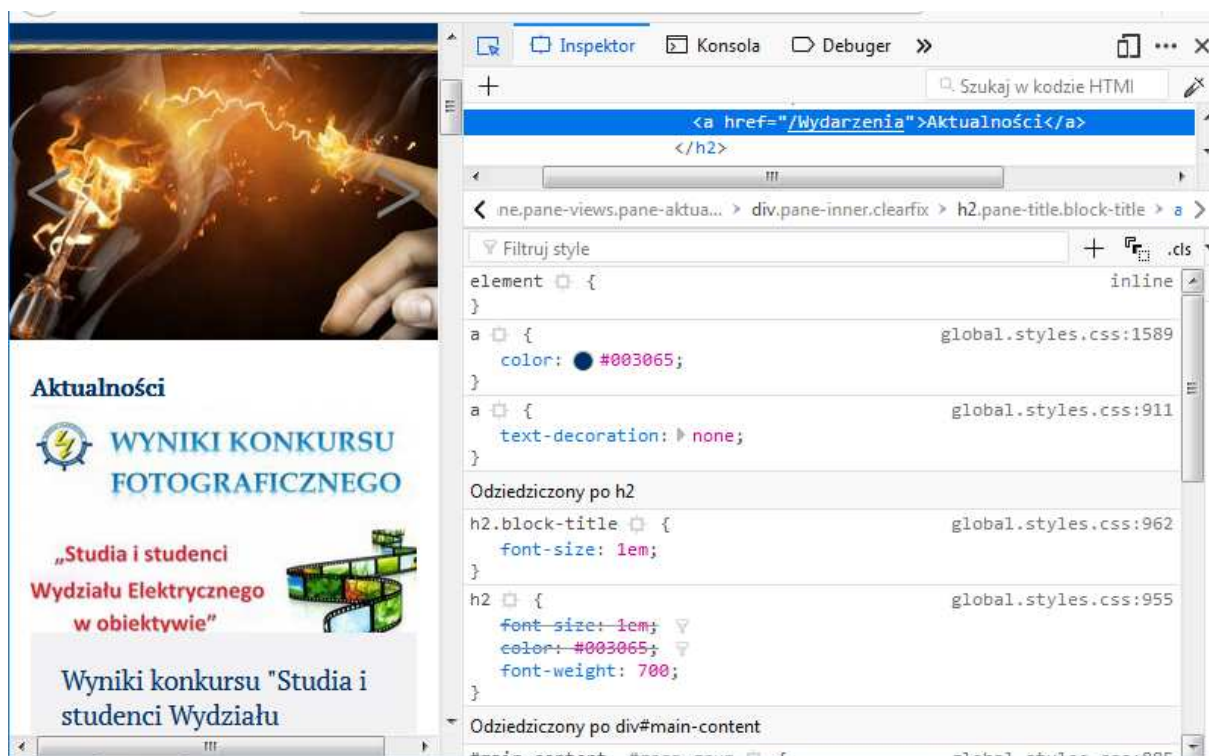


W ten sposób otrzymujemy oryginalny wygląd elementu, nie sformatowany przez CSS. Wybierając opcję *Zapisz stronę jako* możemy zapisać element na lokalnym dysku.



Możemy zaznaczyć fragment kodu HTML, aby wyświetlić związane z nim klasy CSS.

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój



Klasy CSS możemy wyłączać.

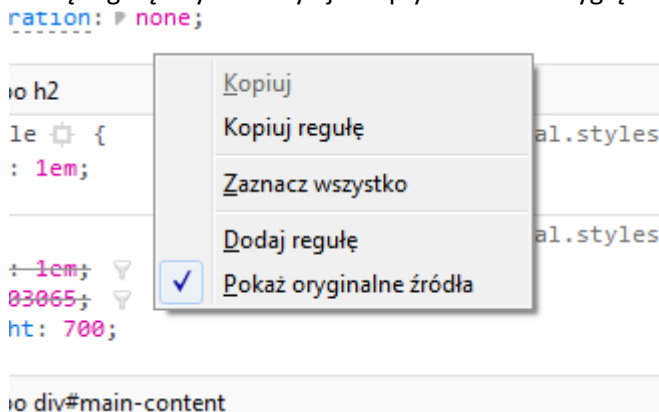
```
#main-content, #gora-grup {  
  font-family: 'PT Serif', serif;  
}
```

W ten sposób dokonujemy zmiany wyglądu elementu.

Przed **Po**
Aktualności **Aktualności**

Jak widać zmienił się krój czcionki.

Możemy także dodać tymczasową regułę aby zobaczyć jak wpłynie ona na wygląd strony.



Dodajmy regułę *font-style: italic* do znacznika h2.

```
h2 {  
  font-style: italic;  
}
```

W rezultacie napis „Aktualności” wyświetlany będzie czcionką pochyłą.



Fundusze Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny

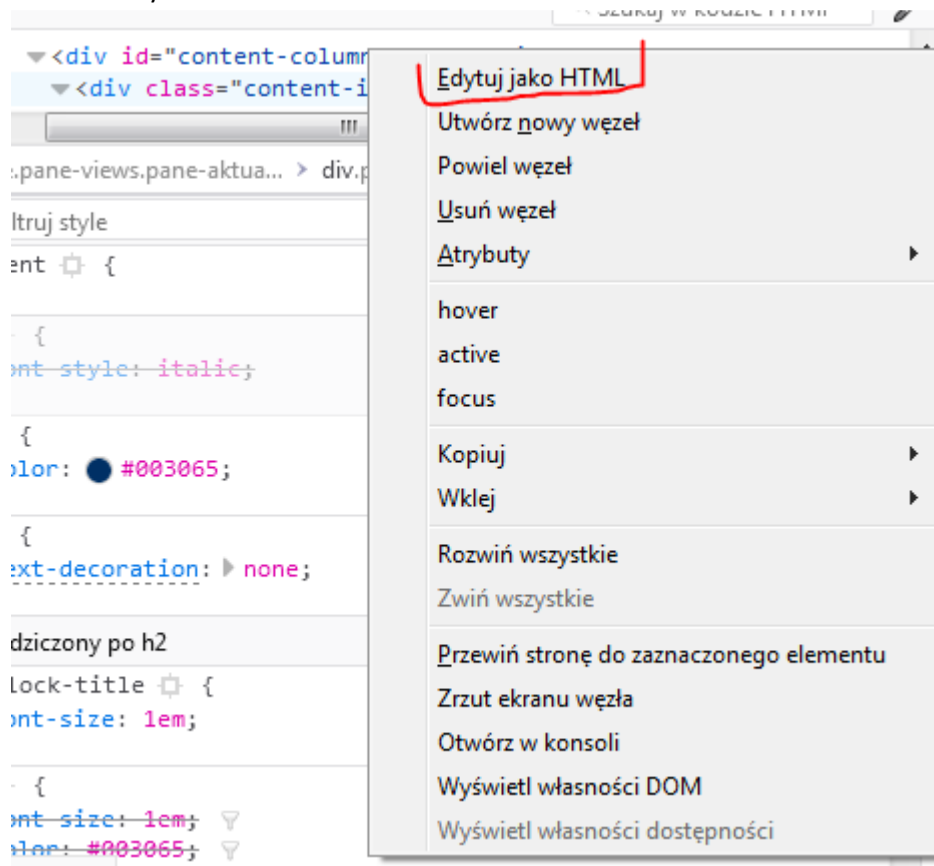


Aktualności

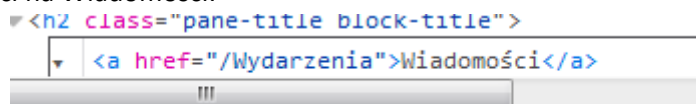


WYNIKI KONKURS
FOTOGRAFICZNEG

Możemy edytować elementy HTML.



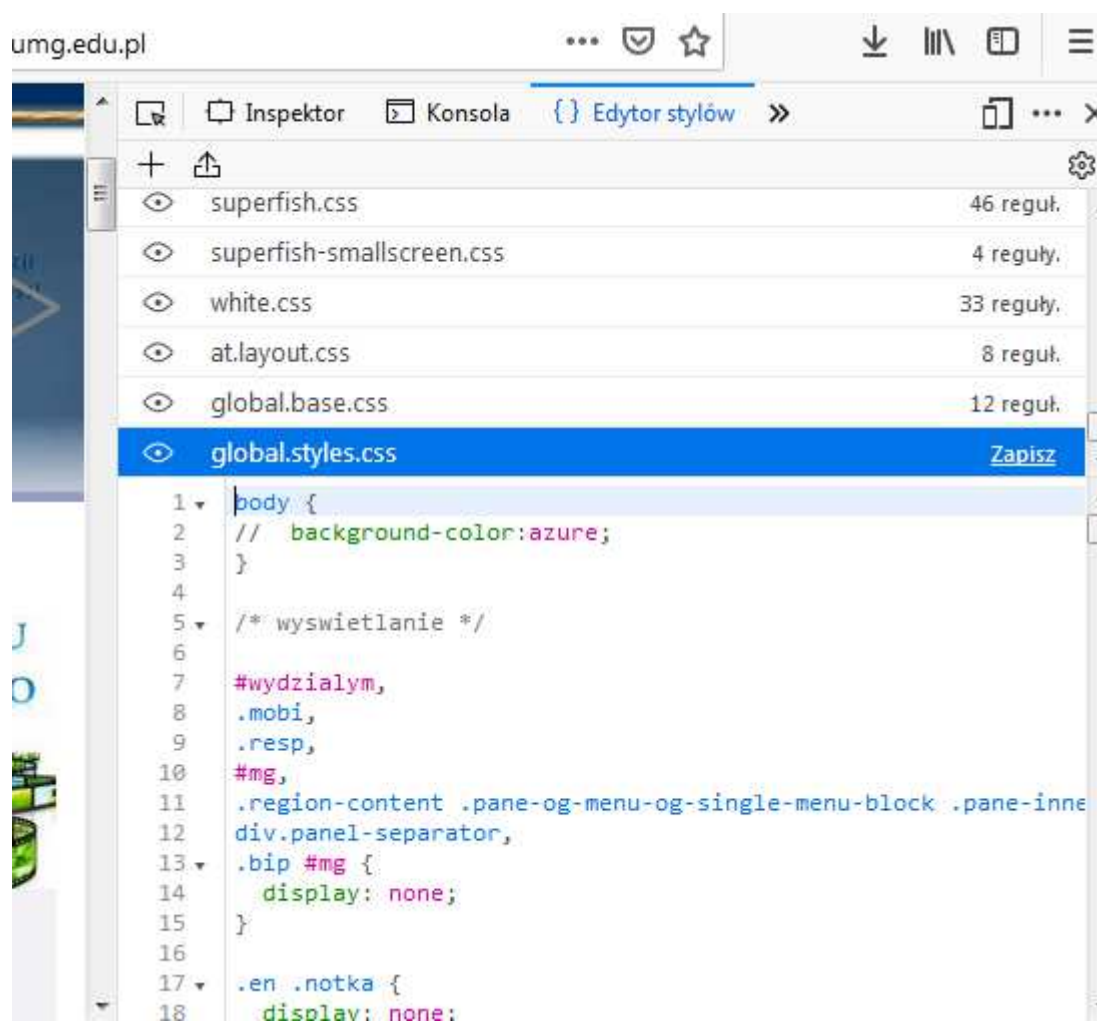
Zmieńmy napis Aktualności na Wiadomości.



W rezultacie na stronie zamiast napisu Aktualności pojawi się napis Wiadomości.

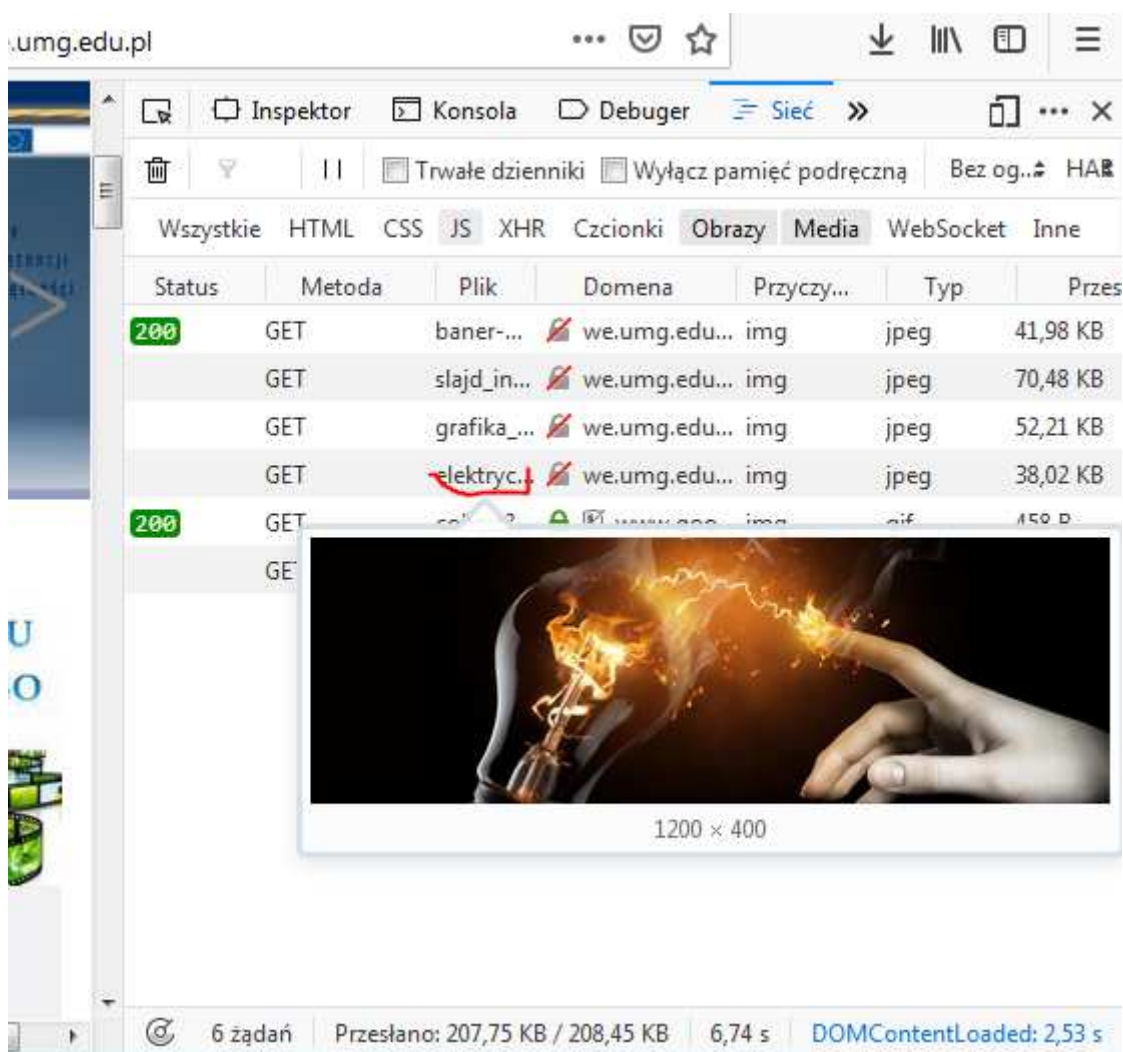


Kolejnym przydatnym narzędziem jest *Edytor stylów*. Pozwala nam wyświetlić wszystkie aktywne na stronie Kaskadowe Arkusze Stylów, edytować je i zapisywać lokalnie na komputerze.



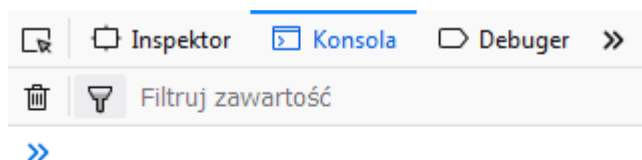
Jest to bardzo przydatna opcja, jeśli widzimy na jakiejś stronie ciekawie sformatowany obiekt i chcielibyśmy zapożyczyć odpowiadające za nie formatowanie do swojego projektu.

Bardzo przydatną zakładką jest *Sieć*. Pozwala nam wyświetlić wszystkie pobrane pliki, czyli pliki HTML, CSS, JS, ale także wszystkie obrazy i media. Najeżdżając na źródło pliku w kolumnie *Plik* wyświetli nam się jego podgląd.



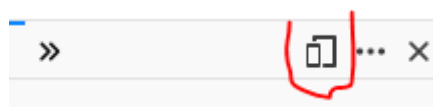
Opcja bardzo przydatna do pobierania zasobów strony internetowej, często używana do pobierania plików wideo, obrazków i muzyki zamieszczonej na stronie. Warto także zauważyć, że w kolumnie *Domena* wyświetlone jest poprzez jaki protokół poprawny został element HTTP czy HTTPS (symbol kłódki).

Kolejnym narzędziem jest *Konsola*, pozwala on wyświetlić komunikaty związane z np. niedziałającymi skryptami JS. Wyświetlmy zawartość dziennika *Konsoli*.

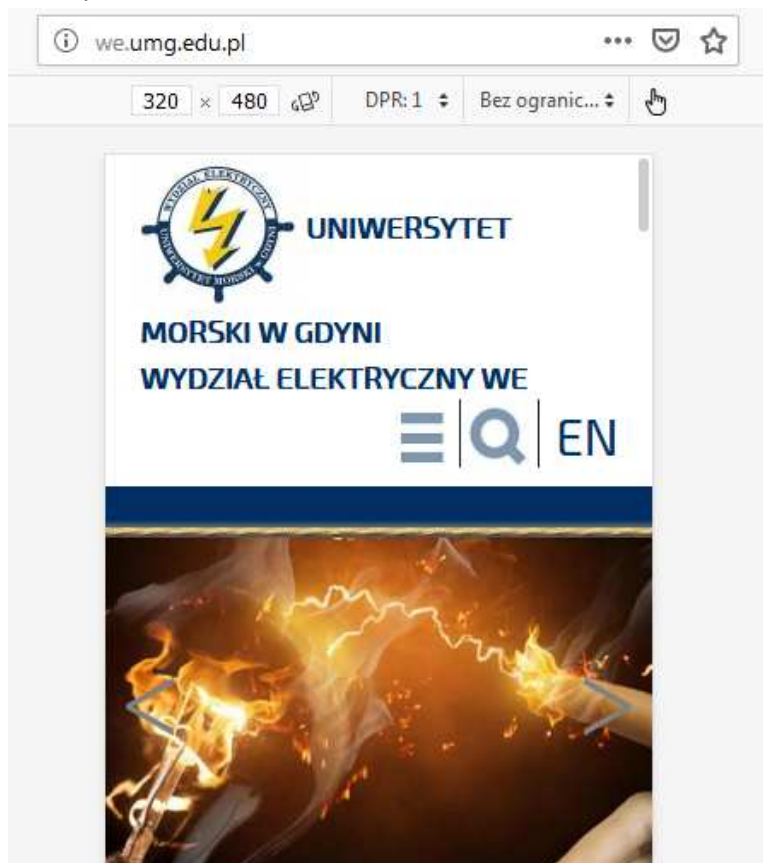


Wygłąda na to, że dziennik konsoli jest pusty, oznacza to, że na stronie nie występują aktualnie żadne błędy w kodzie ani ostrzeżenia.

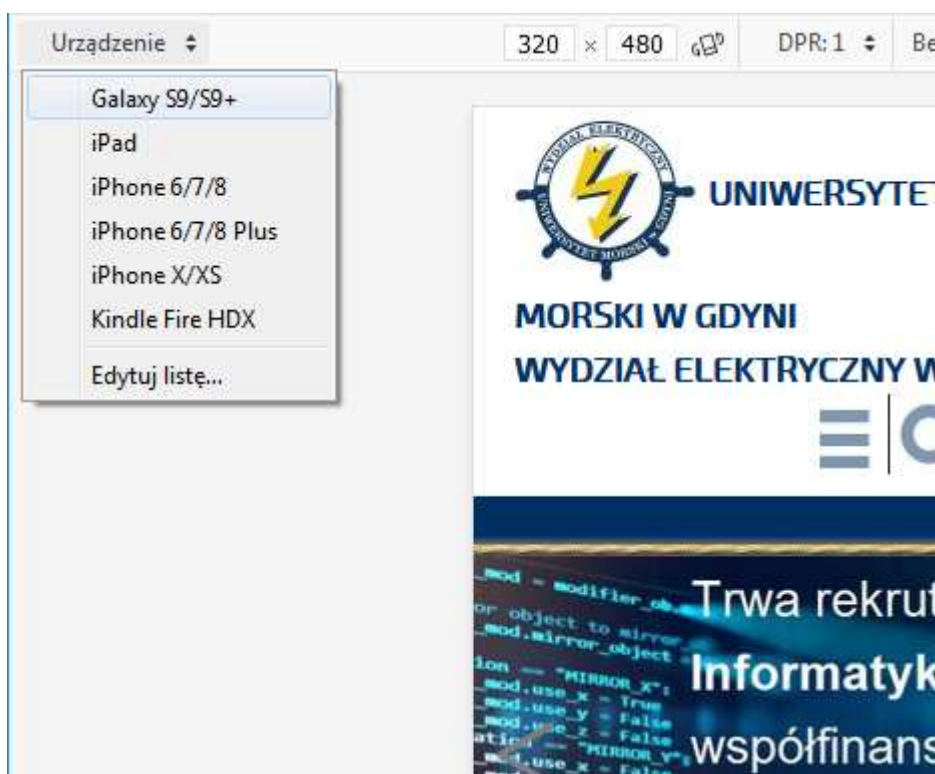
Kolejną opcją wartą uwagi, jest tryb responsywny. Pozwala zasymulować przeglądarce środowisko urządzenia mobilnego.



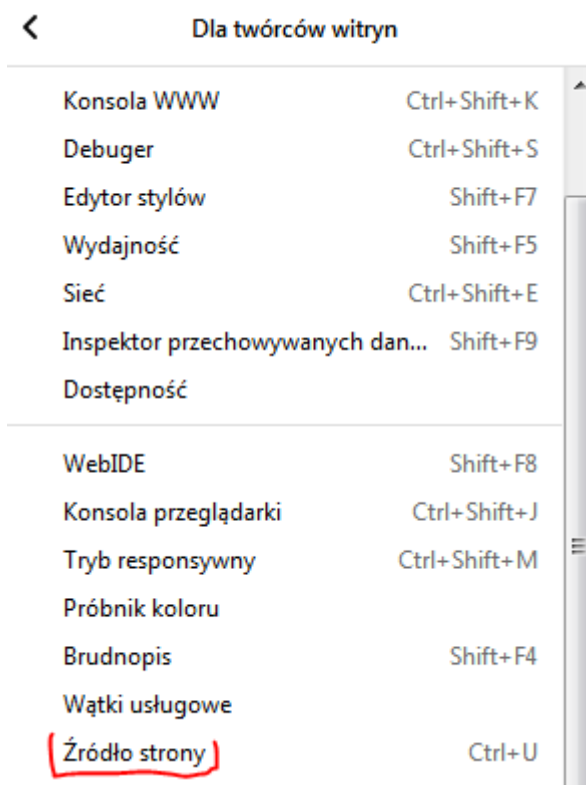
Naciśnijmy przycisk i zobaczymy efekt.



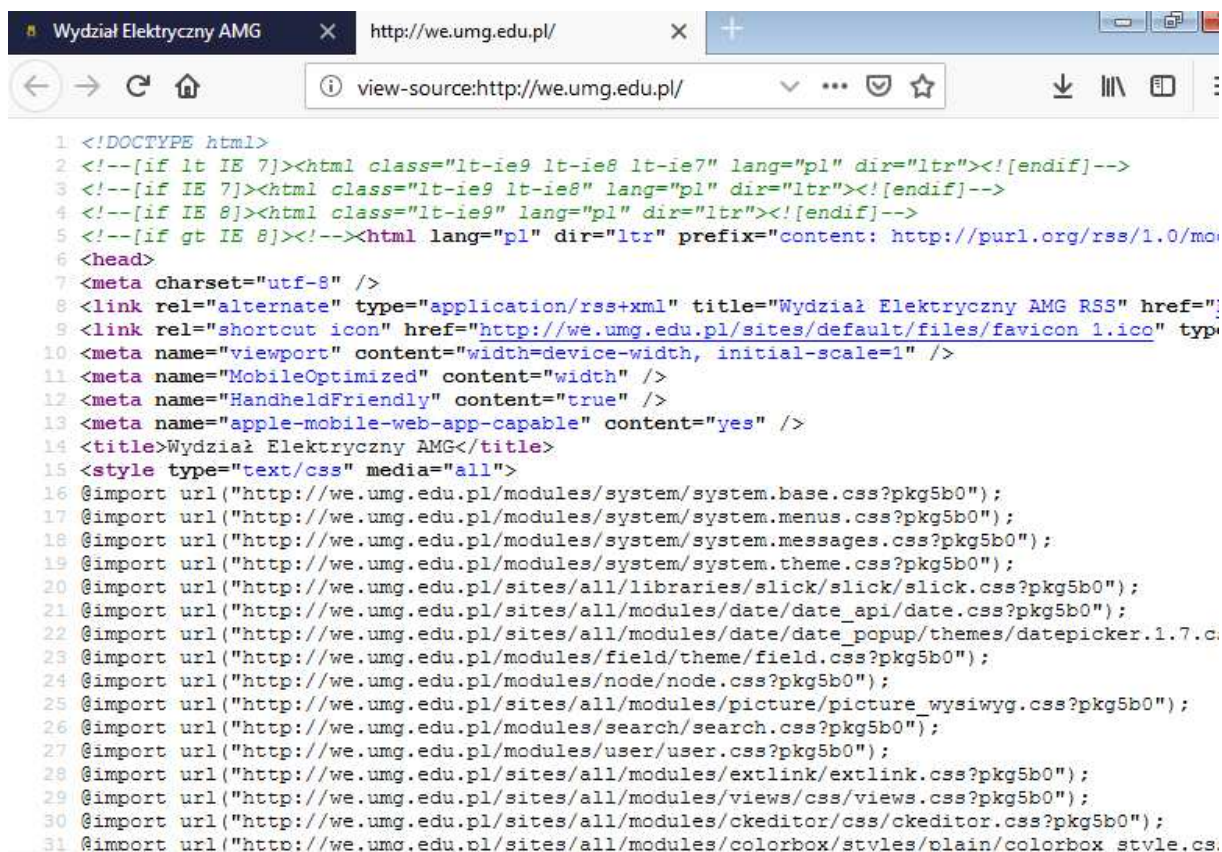
Tak wygląda strona internetowa na urządzeniu o rozdzielczości ekranu 320x480. Tryb responsywny pozwala nam na wybieranie gotowych szablonów najpopularniejszych urządzeń.



W ten sposób możemy wyświetlić jak wyglądała by strona na tych urządzeniach.
Kolejną przydatną opcją jest wyświetlenie źródła strony.



Ukazany nam zostaje kod pliku HTML.



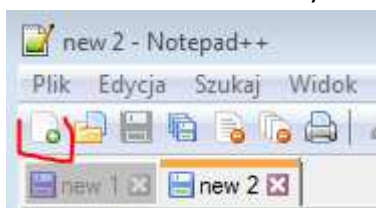
```

1 <!DOCTYPE html>
2 <!--[if lt IE 7]><html class="lt-ie9 lt-ie8 lt-ie7" lang="pl" dir="ltr"><![endif]-->
3 <!--[if IE 7]><html class="lt-ie9 lt-ie8" lang="pl" dir="ltr"><![endif]-->
4 <!--[if IE 8]><html class="lt-ie9" lang="pl" dir="ltr"><![endif]-->
5 <!--[if gt IE 8]><!--><html lang="pl" dir="ltr" prefix="content: http://purl.org/rss/1.0/mo
6 <head>
7 <meta charset="utf-8" />
8 <link rel="alternate" type="application/rss+xml" title="Wydział Elektryczny AMG RSS" href="
9 <link rel="shortcut icon" href="http://we.umg.edu.pl/sites/default/files/favicon.1.ico" typ
10 <meta name="viewport" content="width=device-width, initial-scale=1" />
11 <meta name="MobileOptimized" content="width" />
12 <meta name="HandheldFriendly" content="true" />
13 <meta name="apple-mobile-web-app-capable" content="yes" />
14 <title>Wydział Elektryczny AMG</title>
15 <style type="text/css" media="all">
16 @import url("http://we.umg.edu.pl/modules/system/system.base.css?pkg5b0");
17 @import url("http://we.umg.edu.pl/modules/system/system.menus.css?pkg5b0");
18 @import url("http://we.umg.edu.pl/modules/system/system.messages.css?pkg5b0");
19 @import url("http://we.umg.edu.pl/modules/system/system.theme.css?pkg5b0");
20 @import url("http://we.umg.edu.pl/sites/all/libraries/slick/slick/slick.css?pkg5b0");
21 @import url("http://we.umg.edu.pl/sites/all/modules/date/date_api/date.css?pkg5b0");
22 @import url("http://we.umg.edu.pl/sites/all/modules/date/date_popup/themes/datepicker.1.7.d
23 @import url("http://we.umg.edu.pl/modules/field/theme/field.css?pkg5b0");
24 @import url("http://we.umg.edu.pl/modules/node/node.css?pkg5b0");
25 @import url("http://we.umg.edu.pl/sites/all/modules/picture/picture_wysiwyg.css?pkg5b0");
26 @import url("http://we.umg.edu.pl/modules/search/search.css?pkg5b0");
27 @import url("http://we.umg.edu.pl/modules/user/user.css?pkg5b0");
28 @import url("http://we.umg.edu.pl/sites/all/modules/extlink/extlink.css?pkg5b0");
29 @import url("http://we.umg.edu.pl/sites/all/modules/views/css/views.css?pkg5b0");
30 @import url("http://we.umg.edu.pl/sites/all/modules/ckeditor/css/ckeditor.css?pkg5b0");
31 @import url("http://we.umg.edu.pl/sites/all/modules/colorbox/stvles/plain/colorbox stvle.cs

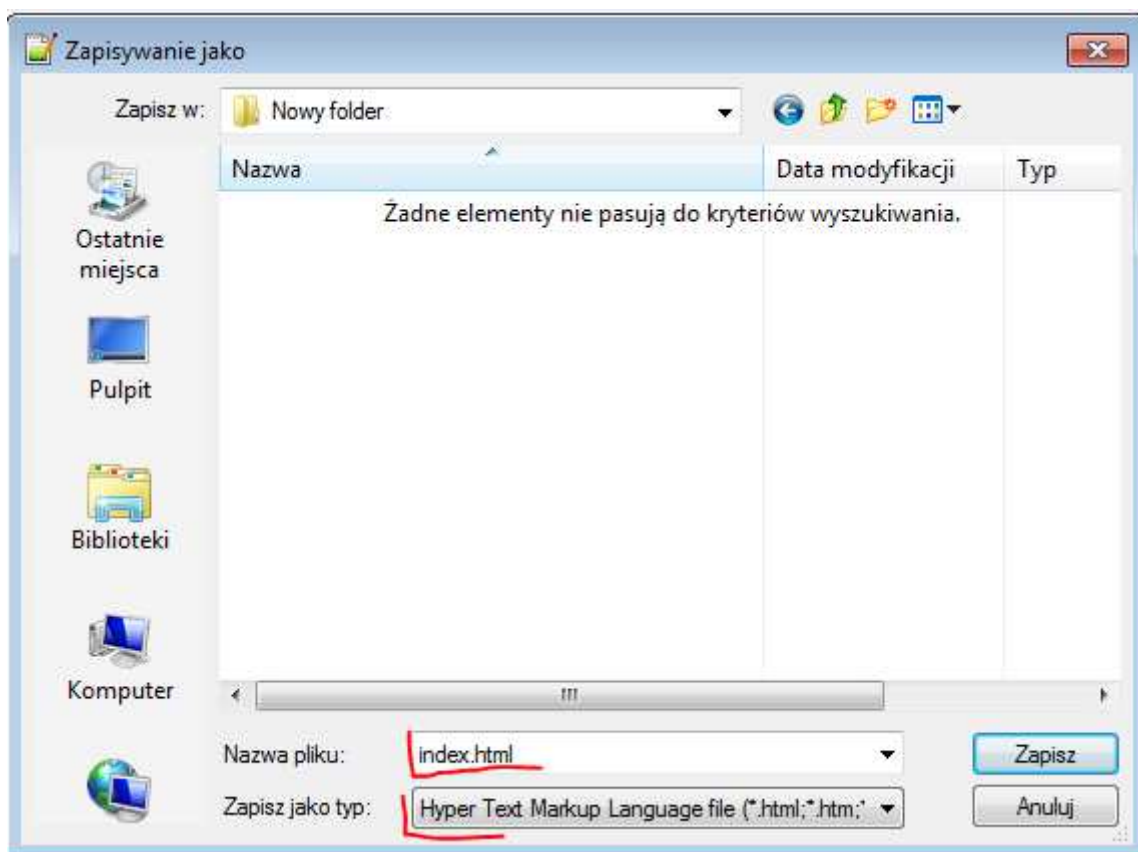
```

W takiej formie stronę otrzymuje przeglądarka internetowa, jej zadaniem jest przetransformować otrzymany kod źródłowy na bardziej przyjazną użytkownikowi formę wizualną.

W jaki sposób tworzy się jednak stronę internetową? Pobierzmy jeden z popularnych edytorów ATOM (<https://atom.io/>) lub Notepad++ (<https://notepad-plus-plus.org>). ATOM jest narzędziem dojrzalszym, oferującym więcej możliwości, lecz ma dużo większą wagę. W tym przypadku posłużymy się Notepadem++, ponieważ wystarczy nam zwykłe kolorowanie składni. Uruchommy Notepad++ oraz utwórzmy nowy plik.



Zapiszmy plik jako stronę HTML.



Stworzymy sobie prosty szablon, na którym będziemy pracować.

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Przykładowa strona</title>
    <link rel="stylesheet" href="styl.css" />
  </head>
  <body>
    <header>
      
    </header>
    <div class="zawartosc">
      <p>Witaj na stronie</p>
    </div>
    <footer>
      <a href="http://we.umg.edu.pl">Uniwersytet Morski w Gdyni</a>
    </footer>
  </body>
</html>
```

Stworzymy teraz plik css i dokonajmy dowolnej edycji wybranych elementów.



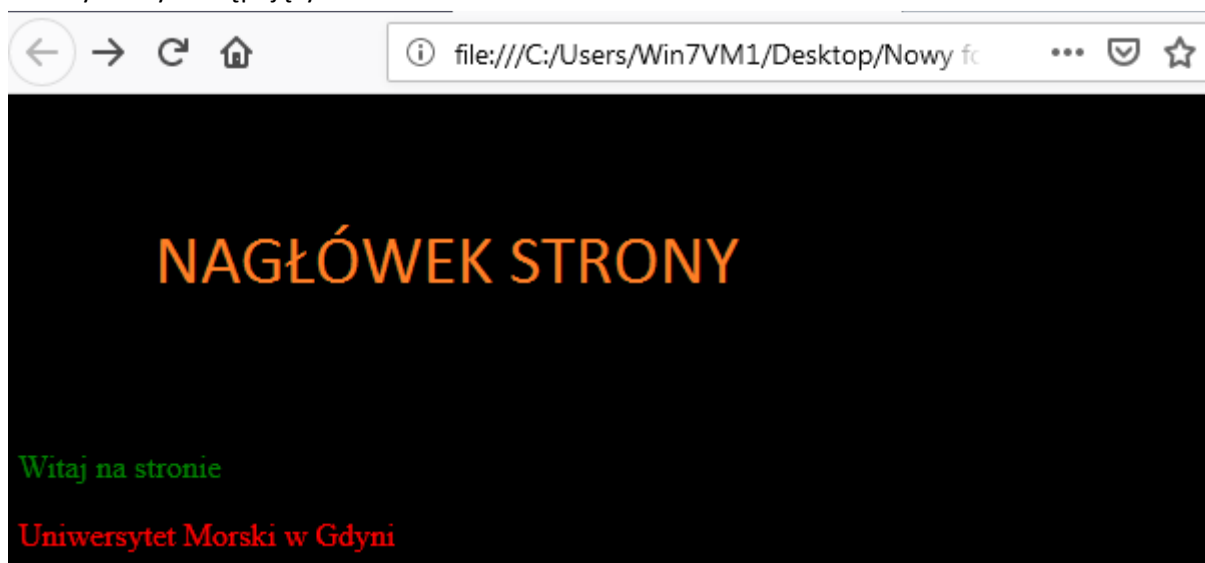
```
body {
  background-color: black;
  color: white;
}

.logo {
  text-align: center;
}

p {
  font-family: Helvetica;
  color: green;
}

a {
  text-decoration: none;
  color: red;
}
```

Przykładowy plik css zmienia kolor tła na czarny, a tekstu domyślnie na biały. Środkowany jest tekst w klasie logo. Tekst akapitów (znacznik p) korzysta z czcionki Helvetica i ma kolor zielony. Hiperłącza (znacznik a) posiadają usunięte podkreślenie i kolor zmieniony na czerwony. W wyniku zastosowania Kaskadowego Arkusza Stylów otrzymamy następujący rezultat.



Stwórzmy prostą animację. Sprawmy, aby „Witaj na stronie” zmieniało kolory.

Aby tego dokonać musimy zdefiniować animację w css'ie, a następnie przypisać ją do nowo stworzonej klasy kolorowane.

Jako pierwszą operację przypiszmy akapitowi klasę *kolorowane*.

```
<div class="zawartosc">
  <p class="kolorowane">Witaj na stronie</p>
</div>
```

Następnie zdefiniujmy animację oraz klasę *kolorowane*.



```
@keyframes zmiana_koloru {
  0% {color: green;}
  25% {color: yellow;}
  50% {color: red;}
  75% {color: pink;}
  100% {color: aqua;}
}

.kolorowane {
  animation-name: zmiana_koloru;
  animation-duration: 5s;
  animation-iteration-count: infinite;
}
```

W wyniku tekst „Witaj na stronie” ulegnie animacji.

Witaj na stronie

Witaj na stronie

Dodajmy przycisk na stronie, który umożliwi wyłączenie animacji. W celu implementacji tej funkcjonalności posłużymy się *JavaScript'em*.

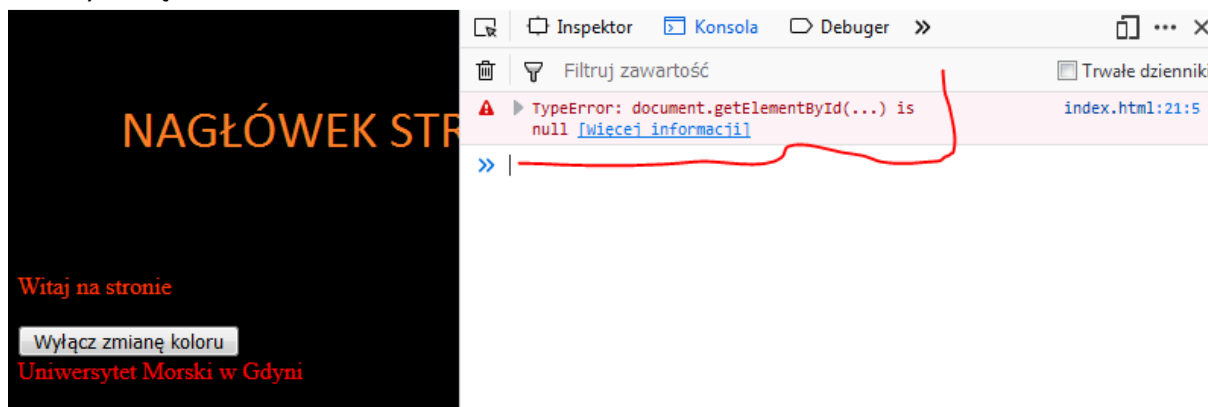
```
<div class="zawartosc">
  <p class="kolorowane" id="do_wylaczenia">Witaj na stronie</p>
  <button type="button" onclick="wylacz();">Wyłącz zmianę koloru</button>
</div>
<footer>
  <a href="http://we.umg.edu.pl">Uniwersytet Morski w Gdyni</a>
</footer>
<script>
  function wylacz() {
    document.getElementById('dowylaczenia').style.animation='none';
  }
</script>
```

Zdefiniowana została funkcja, która zmienia wartość atrybutu *animation* na *none*, co powoduje wyłączenie animacji. Funkcja działa jedynie na elementy strony o identyfikatorze w niej podanym. Należy zauważyć, że w aktualnym momencie identyfikatory się nie zgadzają, co wywoła błąd. Zaprezentujemy jak wykryć ten błąd za pomocą przeglądarki.



Po naciśnięciu przycisku nic się nie dzieje. Developer nie dostaje żadnej informacji o błędzie i nie wie co się dzieje. W takich sytuacjach przydaje się narzędzie *Konsola* znajdujące się w przeglądarce internetowej.

Uruchommy narzędzie konsola.



Konsola informuje nas, że funkcja nie posiada jako parametr żadnej wartości. Nic dziwnego, skoro nazwy się nie zgadzają to próbuje przyjmować nieistniejący element. Poprawmy argument funkcji.

```
<script>
function wylacz() {
    document.getElementById('do_wylaczenia').style.animation='none';
}
</script>
```

Następnie sprawdzimy czy przycisk działa jak należy.



Po dokonaniu korekty przycisk zyskuje odpowiednią funkcjonalność.

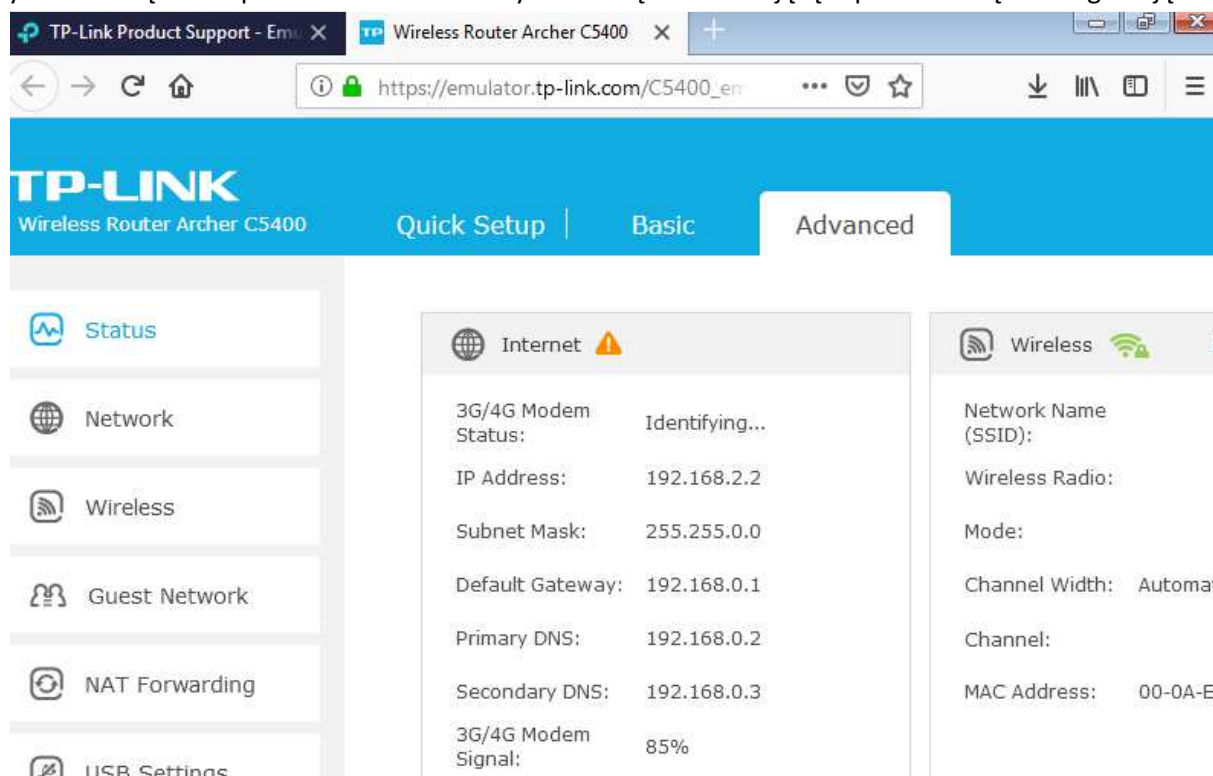
Nie jest to jednak jedyna funkcjonalność przeglądarek. Mogą one służyć także do obsługi WebGUI. Jednym z przykładowych WebGUI są konfiguratory routerów i switchy. Producenci często udostępniają emulatory owych konfiguratorów. Przyjrzyjmy się jednemu z nich. Przejdźmy na stronę <https://www.tp-link.com/us/support/emulators>.

Następnie wybierzmy jeden z routerów.

Home > Routers > Wi-Fi Routers

Archer AX6000	Archer C5400	Archer C4000	Archer C3200	Archer C5400X
TL-WR842ND	Archer C3150 V2	Archer C3150	Archer C2600	AD7200
Archer C2300	Archer C9	Archer A7	TL-WR700N	Archer C1900
Archer C8	Archer C7	Archer C5	Archer C1200	Archer C20
Archer C59	Touch P5	Archer C50	Archer C900	Archer C2
TL-WR902AC	TL-WR743ND	TL-WR710N	TL-WR740N	TL-WR1043N
TL-WR1043ND	TL-WR940N	TL-WR720N	TL-WR802N	TL-WDR3500
TL-WR702N	TL-WDR4300	TL-WR810N	TL-WDR3600	TL-WR941ND
TL-WR841ND	TL-WR841N	TL-WR741ND	Archer C5200	Archer C3000
Archer A2300				

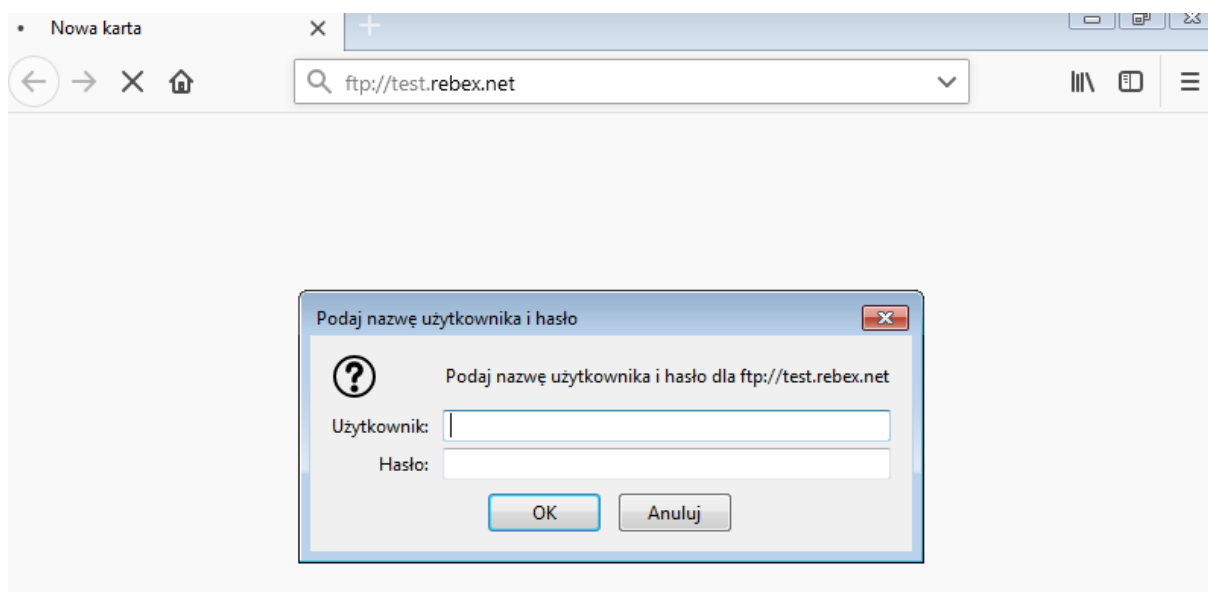
Po wybraniu urządzenia przeniesieni zostaniemy na stronę umożliwiającą zapoznanie się z konfiguracją.



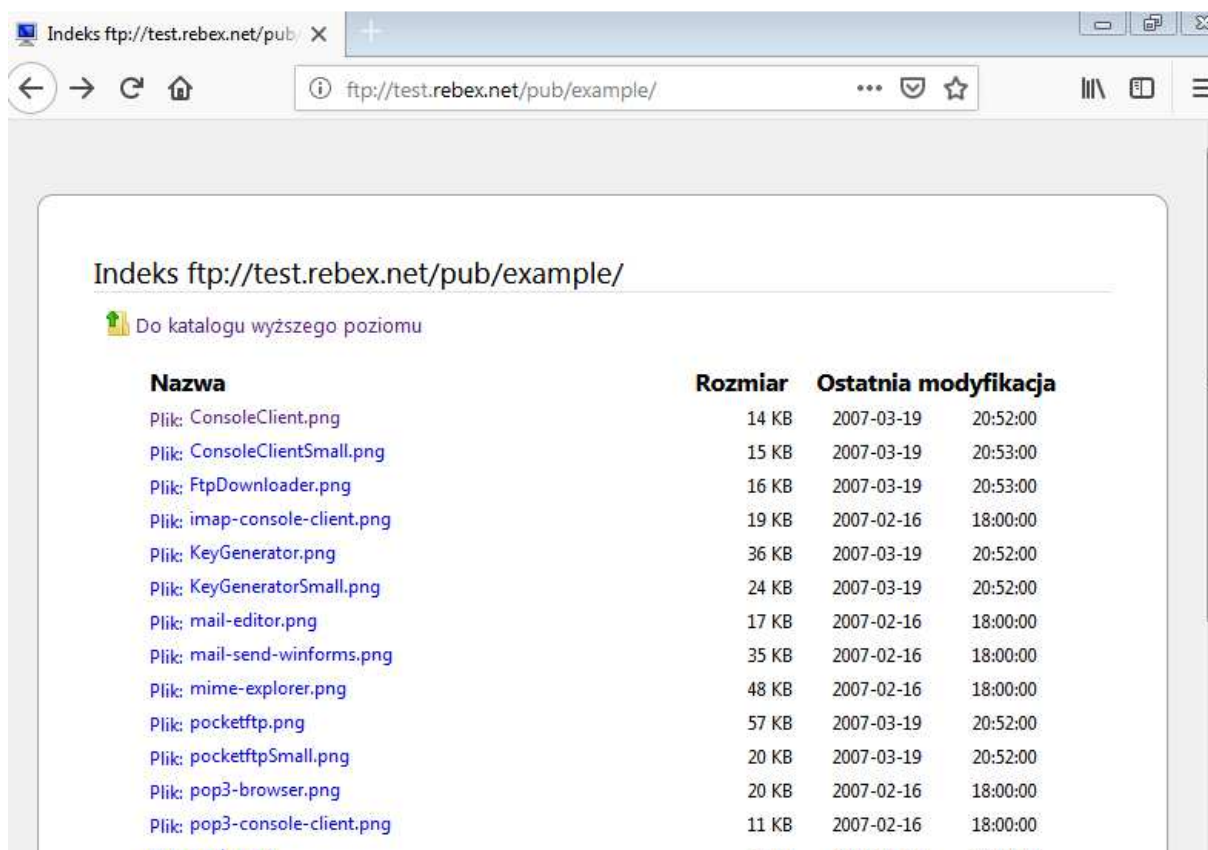
Możemy tutaj sprawdzić dostępne funkcjonalności routera np. przed zakupem oraz sprawdzić jak się go konfiguruje.

Kolejną opcją przeglądarki jest uzyskiwanie dostępu do zasobów FTP. Przeglądarka potrafi wykonywać tylko najprostsze operacje na serwerach FTP m.in. pobieranie umieszczonych tam plików.

Połączmy się z testowym serwerem ftp pod adresem: <ftp://test.rebex.net>



Należy podać nazwę użytkownika i hasło. W tym przypadku jest to *demo* i *password*
Wyświetlona zostanie nam zawartość serwera



Możemy te pliki pobrać, lecz aby dokonać poważniejszych operacji należy skorzystać z klienta FTP. Przeglądarka jednak świetnie nadaje się, by sprawdzić czy dostęp do serwera jest prawidłowy.

4. Zadania

Na każdym etapie zadania twórz screenshot'y (w nazwach powinna znajdować się numeracja wskazująca wykonywaną kolejność kroków), a następnie spakuj do archiwum zip o nazwie WdZOW_LAB_NrAlbumu oraz wyślij na adres e-mail podany przez prowadzącego zajęcia laboratoryjne.

4.1 Skonfiguruj maszynę wirtualną z dostępem do Internetu

- Zainstaluj dwie przeglądarki internetowe: Firefox i Chrome

4.2 Przetestuj funkcjonalność przeglądarki Firefox

- Na podstawie strony internetowej podanej przez prowadzącego sprawdź funkcjonalność narzędzi wbudowanych w przeglądarkę
- Sporządź ...

4.3 Przetestuj funkcjonalności przeglądarki Chrome

- Zidentyfikuj narzędzia przeglądarki Chrome analogiczne do narzędzi Firefoxa
- Przedstaw ...
- Sporządź ...

4.4 Stwórz prostą stronę HTML

- Stwórz prostą stronę w HTML5, skorzystaj z ...
- Stwórz plik CSS zmieniający ...
- Stwórz dowolną prostą animację ...
- Wykonaj skrypt w języku JavaScript ...
- Sporządź ...

4.5 Dokonaj za pomocą przeglądarki przez WebGUI konfiguracji urządzenia sieciowego

- Dokonaj konfiguracji WebGUI routera (lub switcha) obsługując ...
- Przedstaw istotę ...
- Sporządź ...

4.6 Dokonaj połączenia z serwerem FTP przez przeglądarkę internetową

- Połącz się z dowolnym serwerem FTP ...
- Pobierz plik ...
- Sporządź ...

Literatura

1. A. Kisielewicz, Wprowadzenie do informatyki, Helion, Gliwice 2002
2. Scott H. A. Clark, W sercu PC – wg Petera Nortona, Helion, Gliwice 2002
3. J. Shim, J. Siegel, R. Chi, Technologia Informacyjna, Dom Wydawniczy ABC, Warszawa, 1999
4. A. Silberschatz, P.B. Galvin, G. Gagne, Podstawy systemów operacyjnych, WNT, Warszawa 2006
5. A. S. Twnenbaum, Systemy operacyjne, Helion, Gliwice 2010
6. P. Beynon-Davies, Systemy baz danych, WNT, Warszawa 2000
7. W. Stallings, Systemy operacyjne, Struktura i zasady budowy, PWN, Warszawa 2006
8. A. Jakubowski, Podstawy SQL. Ćwiczenia praktyczne, Helion, Gliwice 2004