

**Stanisław Topolewicz**  
**stopolewicz@gmail.com**  
**Małgorzata Klisowska**  
**goskli@op.pl**  
**Katedra Fizyki Doświadczalnej**  
**Wydział Matematyczno-Przyrodniczy**  
**Uniwersytet Rzeszowski**  
**Rzeszów**

## **Prezentacje multimedialne – otwartoźródłowe alternatywy dla *PowerPointa***

### **Wstęp – dlaczego NIE *PowerPoint***

Dzięki polityce firmy Microsoft, preferencji wśród środowisk korporacyjnych oraz początkowemu brakowi konkurencyjnych produktów, program *PowerPoint* stał się pewnym standardem oprogramowania przeznaczonego do tworzenia prezentacji multimedialnych. Niestety, program ten nie jest wolny od wad. Wadą zasadniczą jest brak całkowitej separacji treści od wyglądu prezentacji – nie pozwala to skoncentrować się na treści prezentacji i toku myślenia. Autor prezentacji musi zatroszczyć się o całość formatowania lub wykorzystać wbudowane style – ich edycja nie jest trudna, ale problematyczne może stać się utrzymanie jednorodności wyglądu prezentacji i „powstrzymanie się” przed stosowaniem zbyt wielu efektów graficznych, animacji i przejść slajdów. Brak umiaru w stosowaniu obrazów lub efektów specjalnych sprawia, że stają się one *distraktorami* (czynnikami rozpraszającymi uwagę). Brak jednolitej koncepcji wyglądu prezentacji prowadzi do nadmiaru bodźców, sprawia wrażenie chaosu i przeładowania, utrudniając skupienie się na kluczowych treściach prezentacji.

Można znaleźć mnóstwo podręczników i e-booków uczących, jak stworzyć prezentację w programie *PowerPoint* lub jego otwartoźródłowym odpowiedniku. Zasadniczo to instrukcje obsługi programu. Większość z nich tylko bardzo ogólnie zajmuje się strukturą prezentacji. Najczęściej dominantą treści takiego przewodnika jest efektowny wygląd prezentacji, a nie jej semantyczna, logiczna bądź chronologiczna struktura czy też implementacje aplikacyjne, właściwe w kontekście prezentowanych treści merytorycznych.

### **Rodzaje alternatywnych systemów prezentacji**

Mimo faktu, że *PowerPoint* jest najczęściej wykorzystywanym i najpopularniejszym oprogramowaniem do tworzenia prezentacji multimedialnych, to w środowisku naukowym dość szybko powstała alternatywa, wykorzystująca system składu tekstu TeX, a właściwie zestaw makr oraz poleceń opartych na nim, upraszczających posługiwanie się tym systemem – LaTeX. Alternatywą tą jest *Beamer*, który jest *klasą* dokumentów opisanych językiem LaTeX, stanowiącą pomoc w tworzeniu prezentacji multimedialnych [zob. *LaTeX...*, 2014].

Korzystanie z *Beamer'a* wymaga znajomości LaTeX'a oraz dodatkowych poleceń specyficznych dla tworzenia poszczególnych slajdów prezentacji. W przeciwieństwie do programu *PowerPoint* (i jego odpowiedników do tworzenia grafiki prezentacyjnej, jak np. *LibreOffice Impress*), *Beamer* nie posiada graficznego interfejsu, pozwalającego na edycję poszczególnych slajdów. Układ slajdów generowany jest automatycznie według wybranego szablonu. Użytkownik tworzy dokument źródłowy w LaTeX i przetwarza go (najczęściej do pliku PDF), stosując ten system składu tekstu oraz *Beamer*. Klasa *Beamer* posiada kilkanaście różnych (ok. dwudziestu), spójnych stylów graficznych prezentacji – ich zaletą jest elegancki minimalizm.

W „beamerowskiej” prezentacji to LaTeX dba o poprawne rozłożenie tekstu na slajdzie, pozwala na pokazywanie listy wypunktowanej, wiersz po wierszu i przede wszystkim – co ważne dla fizyków, inżynierów, matematyków, ale i ekonomistów, socjologów, nauczycieli – umożliwia tworzenie slajdów, zawierających równania matematyczne (zdefiniowane w identyczny sposób, jak w systemie LaTeX) oraz prezentacje w nich rysunków i wykresów – bezpośrednio z programów matematycznych. W slajdach *Beamer'a* możemy również załączać linki, które po kliknięciu przypisanego elementu (tekstu, obrazu, lub „przycisku”) uruchomią wybraną zewnętrzną aplikację: film, animację. Pomocą są liczne przykłady dostępne w źródłach internetowych [zob. *TeXample.net...*, 2014].

Dla użytkownika, zaznajomionego z systemem składu tekstu LaTeX, klasa *Beamer* będzie przydatnym narzędziem, np. pisząc naukowy artykuł oraz przygotowując prezentację na konferencję można skorzystać z tego samego kodu źródłowego (np. tych samych równań matematycznych, wykresów, zestawień). W pracy dydaktycznej może okazać się nowym, ciekawym narzędziem, wykorzystywanym do projektów indywidualnych, jak i grupowych [*WriteLaTeX...*, 2014].

Kolejną alternatywą do tworzenia prezentacji mogą być strony WWW. Powstała duża liczba otwartoźródłowych projektów, pozwalających na takie, stosunkowo łatwe, generowanie grafiki prezentacyjnej. Najprostszym podejściem do utworzenia prezentacji w takiej formie jest pewna liczba plików HTML (np. po jednym dla każdego slajdu), które wyposażone zostaną w odpowiedni system nawigacji (mogą to być np. przyciski lub skrypt JavaScript odczytujący stan klawiatury). Podejście proste, ale poważnie utrudnia edycję. Lepszym pomysłem jest wykorzystanie jednego z projektów, który używa języka znaczników HTML5 oraz języka skryptowego JavaScript w celu umieszczenia całości prezentacji w pojedynczym pliku (np. poszczególne slajdy znajdują się w znacznikach lub odpowiednio zdefiniowanej klasie).

Głównymi zaletami wykorzystania plików HTML są:

- przenośność prezentacji – w przeciwieństwie do pliku PowerPointa takie slajdy można wyświetlić na dowolnym urządzeniu – komputerze stacjonarnym, laptopie, projektorze multimedialnym, telewizorze, a nawet tablecie czy też smartfonie. Nowe systemy tworzenia prezentacji dają użytkownikowi do dyspozycji widok responsywny, dzięki czemu prezentacja automatycznie dopasowuje się do rozdzielczości ekranu danego urządzenia;

- prezentacja może zostać właściwie w dowolny sposób sformatowana (poprzez edycję kaskadowych arkuszy stylów CSS – w podobny sposób, jak zwyczajna strona WWW);
- może wykorzystywać dowolną technologię używaną na stronach WWW (np. framework CSS – *Bootstrap*, różnego rodzaju biblioteki pozwalające na dodanie do prezentacji np. efektów graficznych – *jQuery*);
- może zostać opublikowana w Internecie – jako strona internetowa;
- do prezentacji możemy dołączyć każdy rodzaj mediów, z którymi spotykamy się w Internecie, poczynając od grafik, infografik, filmów wideo, animacji aż po dedykowane symulacje, bezpośrednią prezentację danych (numerycznych, matematycznych lub innych), a skończywszy na dołączeniu zawartości całych stron internetowych lub dokumentów HTML;
- prezentacje w ten sposób wykonane mogą być nie tylko multimedialne (tekst, grafika, dźwięk, wideo), ale również interaktywne: elementy prezentacji mogą reagować na działania użytkownika (np. może to być wykres obrazujący dane, którego wygląd, formatowanie oraz zakres danych jest modyfikowany podczas przedstawiania prezentacji). Ważnym aspektem interaktywności może być zastosowanie skryptów w języku JavaScript w celu symulacji i wizualizacji zjawisk fizycznych.

### Przykłady narzędzi o otwartym kodzie źródłowym

W przypadku alternatywnych systemów prezentacji brak jest właściwie dedykowanych rozwiązań (programów) służących przygotowaniu danej prezentacji. Wykorzystywanie ich wymaga od użytkownika otwartości na nowe – chęci do nauki nowych rzeczy, a w efekcie – większej wiedzy. Pozytywnym aspektem jest fakt, że często, ucząc się zadaniowo obsługi oprogramowania (poprzez tworzenie prezentacji) użytkownik uczy się też podstaw HTML, JavaScript, innych języków znacznikowych czy dodatkowego oprogramowania.

Przykładem takiego zestawu narzędzi, może być użycie edytora tekstowego do napisania tekstu w jednym z kilku prostych, a co ważne czytelnych dla człowieka, nawet w postaci nieprzetworzonej, języków znacznikowych (w przypadku systemu Windows może to być np. otwartoźródłowy program *Notepad++*). Jednym z najczęściej stosowanych (np. przez portal [github.com](https://github.com)) jest język Markdown, ale istnieją również inne, np. *reStructuredText* (stosowany w dokumentowaniu kodu źródłowego w języku Python) lub *Textile*. Języki znacznikowe są, ogólnie rzecz biorąc, podobne do języka HTML lub LaTeX, lecz definiują tylko strukturę dokumentu, a nie jego wygląd (jak może to mieć miejsce w przypadku języka HTML). Ponadto wszystkie znaczniki są proste, nie wymagają elementów zamykających, dzięki czemu nie przeszkadzają w odczytywaniu tekstu źródłowego przez człowieka.

Po utworzeniu takiego tekstu, możemy wykorzystać wszechstronny konwerter plików tekstowych *Pandoc*, do przetworzenia go na plik np. w języku HTML. Program ten pozwala na przekształcanie plików między dokumentami zapisanymi w językach znacznikowych, HTML'u, formatach edytorów, takich jak Word czy OpenOffice Writer, języku LaTeX, plikach PDF (lub dowolnie innych [zob. *Pandoc...*, 2014]). Jest to oprogramowanie otwartoźródłowe rozprowadzane na licencji GPL (ang. *General Public License*) – co pozwala na dowolne

modyfikacje kodu programu, a także na tworzenie dodatkowych skryptów konwertujących (pisanych w języku skryptowym LUA, który jest bardzo podobny do języka Pascal). Umożliwia utworzenie prezentacji jako pliku HTML w jednym z systemów prezentacyjnych: *HTML Slidy*, *Reveal.js*, *Slideous*, *S<sup>5</sup>* [zob. Aplikacje].

Sekwencja działań wg schematu: {plik tekstowy → *Pandoc* → plik HTML} ma tę zaletę, że można stosunkowo łatwo i szybko przygotować wersję prezentacji przeznaczoną do druku lub materiały dla uczestników konferencji – wystarczy dokonać konwersji z pliku tekstowego do dokumentu Microsoft Office lub do pliku PDF (gdybyśmy chcieli otrzymać prezentację w jeszcze innym systemie, jest to możliwe – wystarczy napisać skrypt konwertujący).

## Porównanie systemów prezentacji w języku HTML

Wraz ze wzrostem znaczenia Internetu, jako jednego z istotnych kanałów przekazywania informacji, wzrosła liczba projektów mających na celu ułatwienie zwykłym użytkownikom tworzenie prezentacji multimedialnych bazujących na języku HTML.

Jednym z pierwszych był *S<sup>5</sup>* (ang. *Simple Standards-Based Slide Show System*), składający się z zestawu plików, w skład których wchodzi kod JavaScript, ostylowanie (domyślny szablon wyglądu wyraźnie przypomina slajdy generowane przez *Beamer'a*) oraz plik HTML, który możemy edytować, dodawać slajdy i ich elementy. *S<sup>5</sup>* jest prosty, łatwy do opanowania i edycji [zob. *S5: A Simple...*, 2014]. Konwertowanie dokumentów do tego systemu prezentacji multimedialnych wspiera program *Pandoc*. Wygląd poszczególnych slajdów i całej prezentacji zależy od zastosowanych stylów CSS, więc jeżeli nie podoba się nam podstawowy szablon, wystarczy zmodyfikować odpowiedni plik (np. kolory, tło itp.).

Kolejnym uproszczonym systemem prezentacji multimedialnych, udostępnionym przez W3C, jest *HTML Slidy*, składający się z trzech zasadniczych plików: pliku *index.html* – zawierającego prezentację, pliku stylów CCS – definiującego wygląd oraz pliku *slidy.js* – odpowiedzialnego za nawigację. Każdy slajd jest umieszczany w znaczniku o klasie „slide”. *HTML Slidy* posiada wbudowane funkcje, takie jak: wyświetlanie czasu do końca prezentacji, wyświetlanie wiersz po wierszu listy wypunktowanej, okienka, które mogą być rozwinięte (otwarte) przez prezentującego. Co ważne – został w pełni wykonany w języku polskim.

Innym prostym systemem prezentacji, którego zawartość również można wygenerować przez program *Pandoc* jest *Slideous* – skrypt do sterowania pokazu slajdów inspirowany na *S<sup>5</sup>* i *HTML Slidy*, ale zawierający dodatkowo funkcje, uwzględniające inne potrzeby (jak np. ręczne dostosowanie wielkości czcionek, automatyczne tworzenie spisu treści wszystkich slajdów czy też przyciski nawigacyjne w belce statusu). Wygląd prezentacji w tym systemie jest „spartański” – brak tutaj efektów graficznych, przejść między slajdami i jakichkolwiek animacji, jednak należy pamiętać, że wygląd graficzny zależy tak naprawdę od arkusza stylów i można go modyfikować.

Bardziej zaawansowanym, integrującym w sobie większą liczbę otwartoźródłowych narzędzi i bibliotek jest *Reveal.js*. System ten został napisany w języku HTML5 oraz korzysta z możliwości kaskadowych arkusza stylów CSS3. Posiada kilka domyślnych szablonów, jednak istnieje możliwość spersonalizowania prezentacji: umożliwia to ok.

40 kombinacji zestawów kolorów slajdów i rodzajów przejść między slajdami; ponadto każdy slajd może otrzymać swoje własne tło. Nietypową dla dotychczasowych systemów prezentacyjnych, przydatną funkcją jest możliwość *zagnieżdżenia* jednego slajdu w drugim: prezentacja staje się „dwuwymiarowa” – opcją „*strzałka w dół*” na klawiaturze przechodzimy do dodatkowych slajdów. Innym funkcjonalnym pomysłem jest powiększanie fragment slajdu poprzez wybór *Alt+kliknięcie* (to szczególnie przydatne podczas zwracania uwagi słuchaczy na niewielkie, a ważne dla meritum prezentacji szczegóły). Jest też możliwość zatrzymania w dowolnym momencie prezentacji i wygaszenie slajdów, aby ich wyświetlanie nie rozpraszało i nie przeszkadzało w toku zajęć czy innej fazie prelekcji. Prezentacje utworzone w *Reveal.js* posiadają również wersję stylu przeznaczoną do druku, np. do pliku PDF.

W przeciwieństwie do pozostałych systemów *Reveal.js* wspiera obsługę gestów na tabletach oraz składnię Markdown bezpośrednio w slajdach (bez pomocy programu *Pandoc*). Skrypty są udostępniane w zasobach otwartych [zob. *Reveal.js#markdown*]. Przydatną (i istotną też z punktu widzenia edukacji informatycznej) funkcją w *Reveal.js* jest wyświetlanie kodu źródłowego programów w czytelny sposób, z podświetleniem składni danego języka programowania [<http://lab.hakim.se/reveal.js/#/18>, 2014].

*Reveal.js* jest projektem dojrzałym, funkcjonalnym, uwzględniającym nową generację mediów. Jest też produktem implementującym aplikacje i modele, nie tylko dla nowej generacji urządzeń, ale i sposobów ich wykorzystania – przede wszystkim w otwartych społecznościach, w których swobodna wymiana informacji stała się kluczem do stworzenia warunków, w których ludzie mogą się wzajemnie uczyć oraz wykorzystywać istniejące zasoby w kierunku tworzenia nowych pomysłów. Warto wykorzystać otwartość zasobów i atrybuty integrujące, by podobną funkcję mógł spełniać również i w systemie m-edukacji [zob. Ibáñez, 2013]. Wszak tworzenie slajdów poprzez edycję HTML, może być przyjaznym sposobem na wprowadzenie podstaw tego języka na zajęciach edukacyjnych.

## Programy pomocnicze

Przygotowując prezentację, wykorzystującą jako sposób zapisu kod HTML, możemy posłużyć się wieloma przydatnymi narzędziami i bibliotekami. Jednocześnie musimy pamiętać, by stosować różne nazwy klas i identyfikatorów dla różnych bibliotek i elementów z nimi związanych (celem uniknięcia konfliktów nazw i nieprzewidzianych efektów).

Jedną z przydatnych bibliotek jest *TimelineJS*, pozwalająca na przygotowanie pliku HTML zawierającego „*linię czasu*” – jest to pewnego rodzaju prezentacja, której elementy są uporządkowane chronologicznie. Biblioteka ta jest rozprowadzana na licencji *Mozilla Public Licence*. Każde zdarzenie (slajd) może zawierać tytuł, tekst, datę oraz różnego rodzaju media:

- automatyczne zrzuty stron internetowych,
- fragmenty wpisów w Wikipedii (wystarczy podać link),
- cytaty (wyróżnione większą czcionką),
- wpisy z serwisu Twitter,
- dźwięk (z portalu SoundCloud),

- pliki wideo (np. z YouTube, Vimeo lub dostępne lokalnie),
- fotografie (np. z serwisu Flickr lub dostępne lokalnie),
- mapy Google itp.

*TimelineJS* jest biblioteką przydatną wszędzie tam, gdzie chcemy zaprezentować informacje, mające uporządkowanie chronologiczne. To narzędzie dla reprezentantów wszystkich dziedzin: umożliwia umiejscowienie na osi czasu zagadnień związanych z rozwojem danej dyscypliny bądź też chronologie tematyczne w danej dyscyplinie (np. ery historyczne, biologiczne, geologiczne; epoki literackie; w fizyce zaś, np. ewolucja gwiazdy, ewolucja i wiek Wszechświata, oś czasu dalekiej przyszłości itp.).

Pewną niedogodnością jest fakt, że dane wykorzystywane podczas generowania „linii czasu” muszą zostać zapisane w formacie JSON (ang. *JavaScript Object Notation*), który – choć znacznie prostszy od formatu XML – wymaga precyzji i dokładności, jeżeli chodzi o umieszczanie nawiasów, przecinków lub cudzysłowów: jeden niewłaściwie postawiony znak sprawi, że plik nie zostanie załadowany. By ułatwić korzystanie z *TimelineJS* opublikowano generator, pozwalający na utworzenie „linii czasu” z wcześniej przygotowanego arkusza kalkulacyjnego Google Spreadsheet. W tym przypadku pobieramy odpowiedni plik ze strony projektu *TimelineJS*, wypełniamy go danymi (uważając, aby nie zmienić nagłówek tabeli), a następnie korzystamy z generatora dostępnego na stronie projektu (lub z *TimelineJS Embed Generator*). Otrzymujemy kod do dodania w prezentacji bądź na stronie internetowej. *TimelineJS* możemy użyć jako jeden ze slajdów w każdym innym systemie – wystarczy tylko zastosować znacznik, by dołączyć utworzony kod do danego slajdu. Warunkiem takiego rozwiązania jest konieczność zezwolenia na publiczny dostęp do naszego arkusza – innymi słowy dowolny użytkownik, znający jego adres, będzie miał dostęp do zgromadzonych danych.

Inną przydatną biblioteką wykorzystującą język JavaScript jest *D3.js* (*D3* – akronim *Data-Driven Documents*). Pozwala ona na dodanie do prezentacji w języku HTML (lub strony WWW) wykresów, grafów i animacji przedstawiających różnego rodzaju dane (nie tylko numeryczne, ale również np. grafy pokazujące częstotliwość występowania słów w danym tekście). Umożliwia napisanie prostych skryptów JavaScript pobierających dane, wizualizujących je, natomiast wygląd danej grafiki (kolory, animacje, cieniowanie itp.) są realizowane tylko i wyłącznie przy pomocy kaskadowych arkuszy stylów. Biblioteka *D3* pozwala na manipulowanie elementami na stronie HTML w kontekście zbioru danych. Elementy te mogą być tagami HTML i grafikami SVG lub Canvas. Wizualizacje generowane przez *D3* wykonywane są podczas wczytywania strony internetowej i składają się wyłącznie ze skryptu JavaScript, zatem można dołączyć je do każdego z przedstawianych wcześniej systemów prezentacyjnych i indywidualnie skonfigurować ich wygląd.

Biblioteką, przydatną z punktu widzenia użytkowników (matematyków, fizyków, informatyków, ekonomistów itp.), mającą do czynienia z matematyką, jest skrypt *MathJax*, który umożliwia wygenerowanie równań, symboli i wyrażeń matematycznych na stronach internetowych. W przeciwieństwie do wielu innych rozwiązań nie generuje on obrazków zawierających równania, ani nie wykorzystuje Flash'a – używane są style CSS oraz czcionki web w celu generowania wysokiej jakości wyrażeń. *MathJax* interpretuje wyrażenia matematyczne zapisane w dwóch formatach – jako matematyczne wyrażenia

w języku LaTeX (co z pewnością ucieszy wszystkich naukowców) oraz MathML (ang. *Mathematical Markup Language*), przy czym ten drugi język wywodzi się z języka XML i – jako taki – jest mało czytelny dla ludzi (natomiast znakomicie oddaje strukturę wyrażeń matematycznych i jest przydatny dla komputerów). *MathJax* można w prosty sposób zintegrować z dowolną stroną internetową, a więc i z przedstawionymi tutaj systemami prezentacji multimedialnych. Niektórzy autorzy dostrzegli przydatność wsparcia dla symboli i wyrażeń matematycznych, dołączając *MathJax* do swoich systemów, np. opisany wcześniej – *Reveal.js*.

Informatycy doceniają możliwość zaprezentowania kodu źródłowego programu lub skryptu z wyróżnioną, pokolorowaną składnią. Dlatego kolejną biblioteką, która również została dodana m.in. do *Reveal.js* jest *highlight.js*. To kolejne przydatne narzędzie (i składnik kodu naszej prezentacji) pozwala na wyświetlenie kodu źródłowego w 71 językach programowania, które mogą zostać pokolorowane na ok. 44 różne sposoby (z których część jest identyczna z tymi stosowanymi w zwykłych edytorach lub zintegrowanych środowiskach programistycznych – IDE). Biblioteka ta pozwala na automatyczne wykrycie języka programowania, a nawet na użycie kilku języków w tym samym kodzie.

Jedną z najczęściej używanych bibliotek napisanych w języku JavaScript jest *jQuery* – pozwala ona na znaczne uproszczenie kodu manipulującego wyglądem i strukturą danego dokumentu HTML. W przypadku systemów prezentacji multimedialnych wykorzystywana jest do manipulacji dokumentem, np. w celu pokazywania poszczególnych slajdów, przejść między nimi, czy też elementów list wypunktowanych lub numerowanych. Aktualnie, większość efektów graficznych, animacji, przejść i efektów multimedialnych na stronach WWW powstaje z wykorzystaniem tej biblioteki.

## Podsumowanie

Możliwość tworzenia prezentacji multimedialnych z wykorzystaniem formatu HTML jest propozycją dla tych, których interesują: przedstawiana treść, elastyczność rozwiązań i nowe możliwości. Mimo, że krzywa uczenia dla takiego zestawu narzędzi jest dość stroma i często wymaga od użytkownika poznania kilku programów/bibliotek, to jakość otrzymywanych prezentacji, możliwość dodawania elementów multimedialnych, fragmentów innych stron WWW, działających skryptów (np. symulacji fizycznych) i elementów interaktywnych – nadają nowy wymiar w przestrzeni podejmowanych przez użytkownika czynności. Takie podejście do prezentacji zachęca do skupienia się na treści, zawartości merytorycznej i strukturze przedstawianej informacji, natomiast warstwa graficzna jest zazwyczaj generowana z wykorzystaniem gotowych szablonów lub modyfikowana później przy pomocy odpowiednich plików CSS (wyjątkiem jest tu biblioteka D3, która wymaga sporo wysiłku podczas przygotowania grafik, również multimedialnych i interaktywnych).

Alternatywne systemy tworzenia prezentacji z wykorzystaniem oprogramowania *Open Source* w pewnym stopniu wymuszają samodzielną naukę ich stosowania. Praktycznie brak jest klasycznych podręczników. Mamy jednak do czynienia z projektami posiadającymi bogate e-zasoby: obszerną dokumentację, dużą liczbę przykładów, wsparcie w formie samouczków lub na forach danego projektu. Samodzielne zdobywanie wiedzy

i umiejętności aktywizuje, wykraczając poza sformalizowane, tradycyjne ramy i formy (samo)kształcenia, choćby w zakresie tworzenia prezentacji multimedialnych i pozwala poznawać zarówno nowe narzędzia, jak i nowe implementacje aplikacyjne w kontekście prezentowanych treści merytorycznych.

## **Aplikacje [dostęp 15.06.2014]**

*Beamer*. <https://bitbucket.org/rivanvx/beamer/>

*D3*. <http://d3js.org/>

*highlight.js*. <http://highlightjs.org/>

*HTML Slidy*. <http://www.w3.org/Talks/Tools/Slidy/help.html>

*jQuery*. <http://jquery.com/>

*LibreOffice Impress*. <http://pl.libreoffice.org/>

*MathJax*. <http://www.mathjax.org/>

*Notepad++*. <http://notepad-plus-plus.org/>

*Pandoc*. <http://johnmacfarlane.net/pandoc/>

*Reveal.js*. <http://lab.hakim.se/reveal-js/#/>

*S5*. <http://meyerweb.com/eric/tools/s5/>

*Slideous*. <http://goessner.net/articles/slideous/slideous.html>

*Slides*. <http://slides.com/>

*TimelineJS*. <http://timeline.knightlab.com/>

*TimelineJS Embed Generator*. <http://embed.verite.co/timeline/generator.html>

*WriteLaTeX*. <https://www.writelatex.com/>

## **Bibliografia**

Ibáñez L.: *Teaching with open source presentation service Reveal.js*. 2013.  
<http://opensource.com/education/13/10/teaching-with-revealjs/> [dostęp 15.06.2014]

*S5: A Simple Standards-Based Slide Show System*.  
<http://meyerweb.com/eric/tools/s5/primer.html/> [dostęp 15.06.2014]

*LaTeX Templates. Beamer Presentation*. <http://www.latextemplates.com/template/beamer-presentation/> [dostęp 15.06.2014]

*Reveal.js#markdown*. <https://github.com/hakimel/Reveal.js#markdown> [dostęp 15.06.2014]

*Pandoc. A universal document converter. User Guide*.  
<http://johnmacfarlane.net/pandoc/README.html/> [dostęp 15.06.2014]

Popov D.: *Workspace: HTML-owe prezentacje. Przygotowujemy prezentacje w przeglądarce*. „Linux Magazine” 2014, nr 5(123), s. 76-79

*TeXample.net*. <http://www.texample.net/tikz/examples/> [dostęp 15.06.2014]

*WriteLaTeX for Education*. <https://www.writelatex.com/edu/> [dostęp 15.06.2014]



*What is open source?* <http://opensource.com/resources/what-open-source/> [dostęp 15.06.2014]