

# Analiza porównawcza narzędzi typu Front-End Code Playground

Mateusz Magier\*, Beata Pańczyk

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

**Streszczenie.** Niniejszy artykuł zawiera analizę porównawczą wybranych narzędzi typu Front-End Code Playground. Zawarte tabele opracowane zostały na podstawie rezultatów testów rozważanych aplikacji i analizy ich dokumentacji. Porównanie zrealizowane zostało pod kątem kilku kryteriów (m.in.: cech edytorów, niestandardowych trybów tworzenia projektów, wsparcia dla preprocesorów i metod eksportu projektu). Wnioski wyciągnięte na podstawie porównania pozwoliły na podzielenie rozpatrywanych narzędzi na trzy grupy (aplikacje o: wysokim, umiarkowanym i niskim poziomie zaawansowania).

**Słowa kluczowe:** frontend; aplikacje internetowe; CSS; JavaScript

\* Autor do korespondencji.

Adres e-mail: mateusz.magier@pollub.edu.pl

## Comparative analysis of Front-End Code Playground tools

Mateusz Magier\*, Beata Pańczyk

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

**Abstract.** This article contains comparative analysis of selected Front-End Code Playground tools. The included tables were performed on the results of tests of the considered applications and the analysis of their documentation. The comparison was made for several criteria (including: features of code editors, custom modes of creating projects, support for preprocessors and project export methods). The conclusions drawn from the comparison allowed dividing the considered tools into three groups (applications with: high, moderate and low level of advancement).

**Keywords:** frontend; web applications; CSS; JavaScript

\*Corresponding author.

E-mail address: mateusz.magier@pollub.edu.pl

### 1. Wstęp

Według rezultatów corocznej ankiety – przeprowadzanej wśród członków społeczności StackOverflow - najpopularniejszymi technologiami programistycznymi w 2018 roku są odpowiednio: JavaScript (przez sześć lat z rzędu na pierwszym miejscu w zestawieniu), HTML oraz CSS [1]. Programiści korzystający z wyżej wymienionych języków nierzadko stoją przed dylematem związanym w wyborem narzędzia programistycznego. Rozwój aplikacji internetowych na przestrzeni lat przyczynił się do tego, że oprogramowanie desktopowe nie jest już jedyną opcją w kwestii środowisk służących do tworzenia kodu w HTML, CSS i JavaScript. Istnieją aplikacje spełniające to zadanie, które działają za pośrednictwem przeglądarki internetowej. Aplikacje te nazywane są „narzędziami typu Front-End Code Playground” (FECP).

Celem niniejszego artykułu jest przeprowadzenie analizy porównawczej wybranych aplikacji typu FECP. Opisane zostaną kryteria, względem których przeprowadzone zostanie porównanie. Rezultaty analizy zostaną zawarte w tabelach, które mogą okazać się pomocne dla osób stojących przed wyborem oprogramowania typu FECP spełniającego ich wymagania.

Ponieważ narzędzia typu Front-End Code Playground są właściwie nieobecne w literaturze fachowej, niniejszy artykuł stanowi również próbę uporządkowania informacji na ten temat. Przeprowadzone badania literaturowe wykazały, że tematyka ta jest poruszana jedynie w źródłach internetowych (na podstawie tych źródeł dokonano wyboru aplikacji będących przedmiotem porównania) [2-4]. Oprócz tego natrafiono na jedną próbę przeprowadzenia porównania narzędzi typu FECP [5].

Do przeprowadzanych testów wybrano dwanaście aplikacji typu FECP. Należą do nich: CodePen [6], CSSDeck [7], CSSDesk [8], Dabblet [9], JS Bin [10], JSFiddle [11], Kodnest [12], Liveweave [13], Pinocode [14], Plunker [15], SoloLearn [16] oraz Web Maker [17]. Analiza porównawcza została przeprowadzona za pomocą gruntownych testów wyżej wymienionych aplikacji (które zweryfikowały poszczególne kryteria porównania), a także przestudiowania ich dokumentacji.

### 2. Narzędzia typu Front-End Code Playground

Główna różnica między aplikacjami typu FECP a oprogramowaniem desktopowym polega na tym, że ten pierwszy rodzaj narzędzi działa za pośrednictwem przeglądarki internetowej. Aplikacja taka może działać w trybie online lub offline.

Większość tego typu narzędzi udostępnia użytkownikowi edytory HTML, CSS i JavaScript, dodatkowym elementem bywa również konsola JavaScript. Ważną cechą aplikacji typu FECP jest automatyczne odświeżanie widoku (tzw. *Live Preview*) [18]. Dostępne edytory często wyposażone są w szereg cech usprawniających proces tworzenia kodu (np. wsparcie dla wtyczki Emmet oraz sprawdzanie poprawności kodu). Niektóre narzędzia pozwalają na wykorzystanie preprocesorów, a także na dodawanie do projektu zasobów CSS i JavaScript. Niewątpliwie ciekawą funkcjonalnością jest możliwość rozpoczęcia tworzenia projektu od gotowego szablonu, bazującego na wybranej bibliotece CSS/JavaScript (tzw. *boilerplate*) [19].

Niektóre narzędzia udostępniają użytkownikom niestandardowe tryby tworzenia projektów, które pozwalają np. na równoczesną pracę wielu osób nad tym samym kodem [20]. Wiele aplikacji posiada opcję udostępniania i osadzania tworzonego kodu. Kolejną przydatną funkcjonalnością jest eksport projektu, dzięki czemu użytkownik może zapisać rezultaty swojej pracy np. w postaci pliku JSON lub ZIP.

Niektóre aplikacje dysponują bazą publicznych projektów użytkowników (członkowie społeczności mogą uzyskać dostęp do rezultatów pracy innych osób). Dość powszechną opcją jest również możliwość sklonowania kodu innego użytkownika i rozwijania go na swój sposób (tzw. *fork*) [21].

### 3. Analiza porównawcza narzędzi

#### 3.1. Tryb działania

Pierwszą cechą stanowiącą kryterium porównania aplikacji jest tryb działania – narzędzie może działać w trybach: online i/lub offline. W trybie online użytkownik musi posiadać dostęp do Internetu, aby korzystać z oprogramowania.

Tabela 2. Wsparcie dla wybranych funkcjonalności edytorów w narzędziach typu FECP

Narzędzie	Funkcjonalność edytora								
	Emmet	Kolorowanie składni	Podpowiedzi	Brace matching	Zawijanie kodu	Skróty klawiaturowe	Numerowanie linii	Sprawdzanie składni	Komunikaty o błędach
CodePen	tak	tak	tak	tak	tak	tak	tak	tak	tak
CSSDeck	tak	tak	nie	nie	nie	tak	tak	HTML, CSS	nie
CSSDesk	nie	tak	nie	CSS	nie	nie	tak	nie	nie
Dabblet	nie	tak	nie	nie	nie	tak	poz. kursora	nie	nie
JS Bin	tak	tak	JS	tak	tak	tak	tak	CSS, JS	JS
JSFiddle	tak	tak	tak	tak	tak	tak	tak	tak	JS
Kodnest	tak	tak	CSS	CSS, JS	tak	tak	tak	nie	nie
Liveweave	tak	tak	tak	JS	nie	tak	tak	tak	CSS, JS
Pinocode	tak	tak	tak	tak	tak	tak	tak	CSS, JS	JS
Plunker	nie	tak	tak	tak	tak	tak	tak	tak	tak
SoloLearn	nie	tak	tak	tak	tak	tak	tak	tak	tak
Web Maker	tak	tak	tak	tak	tak	tak	tak	CSS, JS	JS

#### 3.3. Tryb tworzenia i prezentowania projektów

Trzecim kryterium porównawczym jest dostępność niestandardowych trybów tworzenia i prezentowania projektów. Przeprowadzone testy pozwoliły na zidentyfikowanie trzech niestandardowych trybów. Pierwszym z nich jest tryb współpracy (tzw. *Collab Mode*), udostępniający możliwość równoczesnej pracy wielu osób nad

Natomiast w trybie offline nie jest wymagane połączenie z Internetem – narzędzie funkcjonuje jako rozszerzenie przeglądarki internetowej. Dostępność trybów dla poszczególnych narzędzi przedstawiono w tabeli 1.

Tabela 1. Zestawienie dostępnych trybów działania

Narzędzie	Tryb działania	
	online	offline
CodePen	tak	nie
CSSDeck	tak	nie
CSSDesk	tak	nie
Dabblet	tak	nie
JS Bin	tak	nie
JSFiddle	tak	nie
Kodnest	tak	nie
Liveweave	tak	nie
Pinocode	nie	tak
Plunker	tak	nie
SoloLearn	tak	nie
Web Maker	tak	tak

#### 3.2. Cechy edytorów kodu

Kolejne kryterium dotyczy cech edytorów kodu w rozważanych aplikacjach typu FECP. Rozpatrywane cechy dotyczą: wsparcia dla Emmet (będącego rozszerzeniem edytorów HTML i CSS, umożliwiającego stosowanie skróconej notacji), kolorowania składni (wyróżniania elementów składowych kodu za pomocą np. koloru), wyświetlania podpowiedzi (autouzupełniania słów kluczowych), podświetlania dopasowanych nawiasów, możliwości zawijania kodu (pozwalającego na uzyskanie lepszej czytelności zagnieżdżonego kodu), skrótów klawiaturowych, numerowania linii w edytorach, sprawdzania poprawności składni oraz wyświetlania komunikatów o błędach. Wsparcie dla wybranych funkcjonalności edytorów w poszczególnych narzędziach przedstawiono w tabeli 2.

tym samym kodem (aktualizacja zawartości edytorów dokonuje się automatycznie u wszystkich użytkowników w grupie, gdy stan projektu ulegnie zmianie).

Kolejny tryb – nauczania (w zależności od narzędzia określane jako *Professor Mode* [22] lub *Codecasting* [23]) – jest modyfikacją trybu współpracy. W grupie korzystającej wspólnie z narzędzia dokładnie jedna osoba może edytować kod – pozostali członkowie otrzymują projekt tylko do

odczytu. Ostatnim trybem jest tryb prezentacji, który dostosowuje wygląd i funkcjonalność narzędzia do sytuacji, w której kod jest np. wyświetlany za pomocą projektora dużej grupie odbiorców (jedną z jego cech jest ograniczenie widocznych komponentów interfejsu narzędzia do niezbędnego minimum). Wsparcie dla trybów niestandardowych w poszczególnych narzędziach przedstawiono w tabeli 3.

Tabela 3. Dostępność trybów niestandardowych w narzędziach typu FECP

Narzędzie	Tryb niestandardowy		
	współpracy	nauczania	prezentacji
CodePen	tak (premium)	tak (premium)	tak (premium)
CSSDeck	nie	nie	nie
CSSDesk	nie	nie	nie
Dabblet	nie	nie	nie
JS Bin	nie	tak	nie
JSFiddle	tak	nie	nie
Kodnest	nie	nie	nie
Liveweave	tak	nie	nie
Pinocode	nie	nie	nie
Plunker	nie	nie	nie
SoloLearn	nie	nie	nie
Web Maker	nie	nie	nie

### 3.4. Przeglądanie projektów innych użytkowników

Narzędzia typu FECP można również porównać pod względem możliwości przeglądania kodu utworzonego przez innych użytkowników aplikacji. Przeprowadzone testy pozwoliły wyróżnić dwa sposoby dostępu do publicznych projektów. Pierwszym z nich jest globalne przeglądanie projektów, realizowane zazwyczaj za pomocą podstrony istniejącej w obrębie serwisu, która zawiera bazę projektów innych użytkowników. Podstrona ta często zawiera interfejs pozwalający na filtrowanie projektów (najpopularniejsze, najwyżej oceniane, najnowsze). Drugim sposobem jest przeglądanie projektów z poziomu konta użytkownika (zawierającej kod utworzony przez danego użytkownika). Listy publicznych projektów często wyposażone są w informacje o opinii innych członków społeczności (wyrażanych za pomocą metryk takich jak: liczba wyświetleń, liczba polubień, oraz liczba komentarzy). Dostępność tej

funkcjonalności dla poszczególnych narzędzi przedstawiono w tabeli 4.

Tabela 4. Dostęp do projektów innych użytkowników

Narzędzie	Przeglądanie projektów	
	globalnie	z konta użytkownika
CodePen	tak	tak
CSSDeck	tak	tak
CSSDesk	nie	nie
Dabblet	nie	nie
JS Bin	nie	nie
JSFiddle	nie	nie
Kodnest	tak	tak
Liveweave	nie	nie
Pinocode	nie	nie
Plunker	tak	tak
SoloLearn	tak	tak
Web Maker	nie	nie

### 3.5. Dostępność narzędzi rozszerzających HTML, CSS i JavaScript

Niektóre narzędzia typu FECP udostępniają użytkownikom możliwość tworzenia kodu z wykorzystaniem preprocesorów i innych narzędzi (wielu programistów do tworzenia kodu frontend wykorzystuje właśnie preprocesory) [24]. Wsparcie dla preprocesorów i narzędzi HTML, CSS i JavaScript w poszczególnych narzędziach przedstawiono w tabelach 5-7.

Tabela 5. Dostępność preprocesorów HTML w poszczególnych aplikacjach

Narzędzie	Preprocesor HTML			
	Haml	Markdown	Slim	Pug
CodePen	tak	tak	tak	tak
CSSDeck	nie	nie	nie	nie
CSSDesk	nie	nie	nie	nie
Dabblet	nie	nie	nie	nie
JS Bin	nie	tak	nie	tak
JSFiddle	nie	nie	nie	nie
Kodnest	tak	nie	nie	tak
Liveweave	nie	nie	nie	nie
Pinocode	nie	tak	nie	tak
Plunker	nie	nie	nie	nie
SoloLearn	nie	nie	nie	nie
Web Maker	nie	tak	nie	tak

Tabela 6. Dostępność narzędzi rozszerzających CSS w poszczególnych aplikacjach typu FECP

Narzędzie	Preprocesor/narzędzie CSS						
	Less	Sass		Stylus	Myth	PostCSS	Atomizer
		SCSS	SASS				
CodePen	tak	tak	tak	tak	nie	tak	nie
CSSDeck	nie	nie	nie	nie	nie	nie	nie
CSSDesk	nie	nie	nie	nie	nie	nie	nie
Dabblet	nie	nie	nie	nie	nie	nie	nie
JS Bin	tak	tak	tak	tak	tak	nie	nie
JSFiddle	nie	tak	nie	nie	nie	nie	nie
Kodnest	tak	tak	tak	tak	nie	nie	nie
Liveweave	nie	nie	nie	nie	nie	nie	nie
Pinocode	tak	tak	tak	tak	nie	nie	tak
Plunker	nie	nie	nie	nie	nie	nie	nie
SoloLearn	nie	nie	nie	nie	nie	nie	nie
Web Maker	tak	tak	tak	tak	nie	nie	tak

Tabela 7. Dostępność narzędzi rozszerzających JavaScript w poszczególnych aplikacjach typu FECP

Narzędzie	Preprocesor/narzędzie JavaScript					
	CoffeeScript	LiveScript	TypeScript	Babel	Traceur	ClojureScript
CodePen	tak	tak	tak	tak	nie	nie
CSSDeck	nie	nie	nie	nie	nie	nie
CSSDesk	nie	nie	nie	nie	nie	nie
Dabblet	nie	nie	nie	nie	nie	nie
JS Bin	tak	tak	tak	tak	tak	tak
JSFiddle	tak	nie	tak	tak	nie	nie
Kodnest	nie	nie	tak	tak	nie	nie
Liveweave	nie	nie	nie	nie	nie	nie
Pinocode	tak	nie	tak	tak	nie	nie
Plunker	nie	nie	nie	nie	nie	nie
SoloLearn	nie	nie	nie	nie	nie	nie
Web Maker	tak	nie	tak	tak	nie	nie

### 3.6. Metody eksportowania projektów

Kolejnym kryterium porównawczym jest sposób, w jaki użytkownik może wyeksportować utworzony kod. Do zidentyfikowanych metod należą: plik ZIP, plik HTML, plik JSON (format pozwalający na zapisanie nie tylko zawartości edytorów, ale także ustawień projektu) oraz sklonowanie projektu do narzędzia CodePen. W tabeli 8. zaprezentowano dostępność rozpatrywanych metod eksportu w poszczególnych narzędziach typu FECP.

Tabela 8. Dostępność metod eksportu projektu w poszczególnych aplikacjach

Narzędzie	Metoda eksportu projektu			
	Plik ZIP	Plik HTML	Plik JSON	CodePen
CodePen	tak	nie	nie	nie
CSSDeck	nie	nie	nie	nie
CSSDesk	nie	tak	nie	nie
Dabblet	nie	nie	nie	nie
JS Bin	nie	tak	nie	nie
JSFiddle	nie	nie	nie	nie
Kodnest	tak	tak	nie	nie
Liveweave	tak	tak	nie	nie
Pinocode	nie	tak	tak	tak
Plunker	tak	nie	nie	nie
SoloLearn	nie	nie	nie	nie
Web Maker	nie	tak	tak	tak

### 3.7. Sposoby zapisywania projektów

Następną rozpatrywaną cechą jest metoda zapisywania kodu stworzonego przez użytkownika. Niektóre narzędzia wykorzystują mechanizm localStorage do tymczasowego zapisywania bieżącego stanu projektu. Powszechną metodą jest wykorzystanie adresu URL. W celu zapisywania kodu tworzonego przez użytkowników wykorzystywana jest także usługa Gist (służąca do szybkiego tworzenia małych repozytoriów). Kolejną zidentyfikowaną metodą jest zapisanie projektu po stronie serwera bez bezpośredniego odnośnika URL (ten sposób w zestawieniu reprezentuje kolumna „serwer”). Ostatnią metodą jest wykorzystanie systemu plików komputera użytkownika (sposób występujący w narzędziach działających w trybie offline). W tabeli 9. zaprezentowano wsparcie dla różnych mechanizmów zapisywania projektów w poszczególnych narzędziach typu FECP.

Tabela 9. Wsparcie dla metod zapisywania projektu w aplikacjach typu FECP

Narzędzie	Metoda zapisywania projektu				
	localStorage	URL	Gist	serwer	lokalnie
CodePen	nie	tak	tak	nie	nie
CSSDeck	tak	tak	nie	nie	nie
CSSDesk	tak	tak	nie	nie	nie
Dabblet	tak	tak	tak	nie	nie
JS Bin	tak	tak	tak	nie	nie
JSFiddle	nie	tak	nie	nie	nie
Kodnest	nie	tak	nie	nie	nie
Liveweave	tak	tak	nie	nie	nie
Pinocode	nie	nie	nie	nie	tak
Plunker	tak	tak	nie	tak	nie
SoloLearn	nie	tak	nie	nie	nie
Web Maker	tak	nie	nie	tak	tak

### 3.8. Dodawanie zasobów CSS i JavaScript

Wiele aplikacji typu FECP pozwala na dodawanie do projektu bibliotek oraz szkieletów programowych CSS i JavaScript. Wyróżniono trzy sposoby realizowania tej funkcjonalności. Pierwszą z nich jest tzw. metoda „quick-add” (użytkownik wybiera zasób z predefiniowanej listy). Drugi sposób polega na wykorzystaniu systemu dostarczania treści (np. CDNJS). Ostatnim sposobem jest podanie przez użytkownika bezpośredniego linku do zasobu. W tabeli 10. przedstawiono wsparcie dla różnych metod dodawania zasobów w rozważanych narzędziach typu FECP.

Tabela 10. Wsparcie dla różnych metod dodawania zasobów do projektu

Narzędzie	Metoda dodawania zasobów		
	lista	wyszukiwarka	link
CodePen	tak	tak	tak
CSSDeck	tak	nie	nie
CSSDesk	nie	nie	nie
Dabblet	nie	nie	nie
JS Bin	tak	nie	nie
JSFiddle	tylko JavaScript	tak	tak
Kodnest	tak	nie	tak
Liveweave	tak	nie	nie
Pinocode	tak	tak	tak
Plunker	tak	tak	tak
SoloLearn	nie	nie	nie
Web Maker	tak	tak	tak

#### 4. Wnioski

Powyższe zestawienia pozwoliły na wyciągnięcie kilku wniosków. Zdecydowana większość aplikacji typu FECF funkcjonuje w trybie online. Dokładnie jedno narzędzie – Web Maker – dostępne jest w dwóch trybach. Narzędzia typu FECF są zróżnicowane pod względem oferowanych funkcjonalności edytorów kodu. Do aplikacji dysponujących najlepiej wyposażonymi edytorami należą: CodePen, JS Bin, JSFiddle, Pinocode oraz Web Maker.

Stosunkowo niewielka liczba narzędzi wspiera niestandardowe tryby tworzenia i prezentowania projektów (wszystkie tryby wspierane są w aplikacji CodePen, jednak są one niedostępne dla użytkowników dysponujących bezpłatną wersją konta; oprócz tego trzy narzędzia wspierają tryb współpracy lub nauczania – JS Bin, JSFiddle i Liveweave).

Pięć z dwunastu rozważanych narzędzi udostępnia bazę publicznych projektów innych użytkowników (CodePen, CSSDeck, Kodnest, Plunker i SoloLearn). Pod względem dostępności preprocesorów i narzędzi HTML/CSS/JavaScript, najlepszymi aplikacjami okazały się: CodePen, JS Bin, Kodnest, Pinocode oraz Web Maker. Najczęściej wspieranymi metodami eksportu są: plik ZIP oraz plik HTML (cztery narzędzia nie udostępniają możliwości wyeksportowania utworzonego kodu). Najczęściej stosowanymi sposobami zapisywania projektów okazały się: magazyn localStorage oraz adres URL. Zdecydowana większość narzędzi typu FECF wspiera co najmniej jedną metodę dodawania zasobów CSS/JavaScript (najczęściej wspierają jest „quick-add”).

Analiza zestawień zawartych w niniejszym artykule pozwoliła na podzielenie rozpatrywanych narzędzi typu FECF na trzy grupy (według poziomu zaawansowania). Pierwszą z nich jest grupa o **wysokim poziomie zaawansowania**. Do jej cech charakterystycznych należą:

- zaawansowane, liczne funkcjonalności,
- wsparcie dla preprocesorów i bibliotek,
- dostępność co najmniej jednego niestandardowego trybu tworzenia projektów,
- mało znaczące wady lub ich brak,
- bardzo dobre wyposażenie edytorów.

Kolejną wyróżnioną grupą są narzędzia o **niskim poziomie zaawansowania**, które charakteryzują się:

- brakiem zaawansowanych funkcjonalności,
- brakiem wsparcia dla preprocesorów,
- niedziałającymi funkcjonalnościami dostępnymi w interfejsie aplikacji,
- słabym wyposażeniem edytorów,
- brakiem niestandardowych trybów tworzenia projektów.

Do ostatniej grupy – narzędzi o **umiarkowanym poziomie zaawansowania** – zaliczono aplikacje charakteryzujące się cechami wymienionymi zarówno przy opisie pierwszej, jak i drugiej grupy. Taki podział pozwolił na przyporządkowanie rozważanych aplikacji typu FECF do poszczególnych grup w następujący sposób:

- narzędzia o wysokim poziomie zaawansowania: CodePen, JS Bin, JSFiddle,

- narzędzia o umiarkowanym poziomie zaawansowania: Kodnest, Liveweave, Pinocode, Plunker, Web Maker,
- narzędzia o niskim poziomie zaawansowania: CSSDeck, CSSDesk, Dabblet, SoloLearn.

Powyższy podział – podobnie jak zestawienia porównawcze – może okazać się dobrym źródłem dla osób podejmujących decyzję dotyczącą wyboru narzędzia typu Front-End Code Playground. Narzędzia o wysokim poziomie zaawansowania wydają się być najlepszą opcją dla doświadczonych programistów frontend (np. korzystających z preprocesorów, lub tworzących kod w zespole). Aplikacje o niskim poziomie zaawansowania mogą z kolei stanowić wybór początkujących programistów o niewielkich wymaganiach względem narzędzi.

#### Literatura

- [1] Stack Overflow Developer Survey 2018, <https://insights.stackoverflow.com/survey/2018/#technology> [09.09.2018]
- [2] 7 of the Best Code Playgrounds — SitePoint, <https://www.sitepoint.com/7-code-playgrounds/> [09.09.2018]
- [3] 10 Best Code Playgrounds for Developers | Code Geekz, <https://codegeekz.com/best-code-playgrounds-for-developers/> [09.09.2018]
- [4] 8 Code Playgrounds to Test your JavaScript Applications and Skills | Jscrambler Blog, <https://blog.jscrambler.com/8-code-playgrounds-to-test-your-javascript-applications-and-skills/> [09.09.2018]
- [5] Comparison of online source code playgrounds - Wikipedia, [https://en.wikipedia.org/wiki/Comparison\\_of\\_online\\_source\\_code\\_playgrounds#Online\\_web\\_client-side\\_source\\_code\\_playgrounds](https://en.wikipedia.org/wiki/Comparison_of_online_source_code_playgrounds#Online_web_client-side_source_code_playgrounds) [09.09.2018]
- [6] CodePen, <https://codepen.io/> [09.09.2018]
- [7] HTML5, CSS3, JS Demos, Creations and Experiments | CSSDeck, <http://cssdeck.com/> [09.09.2018]
- [8] CSSDesk - Online CSS Sandbox, <http://www.cssdesk.com/> [09.09.2018]
- [9] dabblet.com, <http://dabblet.com/> [09.09.2018]
- [10] JS Bin - Collaborative JavaScript Debugging, <http://jsbin.com/> [09.09.2018]
- [11] Create a new fiddle – JSFiddle, <https://jsfiddle.net/> [09.09.2018]
- [12] Kodnest - Realtime online code editor & playground for Frontend developers, <http://kodhus.com/kodnest/> [09.09.2018]
- [13] HTML, CSS and JavaScript demo – Liveweave, <https://liveweave.com/> [09.09.2018]
- [14] Pinocode - Chrome Web Store, <https://chrome.google.com/webstore/detail/pinocode/kijgcmknihlfnlfpjgdmchlidplcmp> [09.09.2018]
- [15] Plunker, <http://plnkr.co/> [09.09.2018]
- [16] SoloLearn: Learn to Code for Free!, <https://www.sololearn.com/> [09.09.2018]
- [17] Web Maker - A blazing fast & offline web playground, <https://webmakerapp.com/> [09.09.2018]
- [18] Auto Updating Previews - CodePen Blog, <https://blog.codepen.io/documentation/editor/auto-updating-previews/> [09.09.2018]
- [19] What is boilerplate and why do we use it? Necessity of coding style guide, <https://medium.freecodecamp.org/whats-boilerplate-and-why-do-we-use-it-let-s-check-out-the-coding-style-guide-ac2b6c814ee7> [09.09.2018]

- [20] Collab Mode - CodePen Blog, <https://blog.codepen.io/documentation/pro-features/collab-mode/> [09.09.2018]
- [21] Forks - CodePen Blog, <https://blog.codepen.io/documentation/features/forks/> [09.09.2018]
- [22] Professor Mode - CodePen Blog, <https://blog.codepen.io/documentation/pro-features/professor-mode/> [09.09.2018]
- [23] What is codecasting?, <https://remysharp.com/2013/11/14/what-is-codecasting/> [09.09.2018]
- [24] R. Queirós, A Survey on CSS Preprocessors, 6th Symposium on Languages, Applications and Technologies (SLATE 2017), <http://drops.dagstuhl.de/opus/volltexte/2017/7943/pdf/OASlcs-SLATE-2017-8.pdf>.