

Konfiguracja edytoru Atom dla programistów Django

Piotr Kulas

March 4, 2016

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Wstęp

Czym jest Atom?
Historia
Zalety
Alternatywy

Wbudowane pakiety

Wstęp
Tree View
Snippets

Dodatkowe pakiety

Wstęp
Autocomplete
Lint
Flake8
Formatowanie składni
Terminal
Emmet
Historia
Git
Snippets

Co dalej?

- czym jest Atom?
- historia
- zalety

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Desktopowy edytor programistyczny dystrybuowany na otwartej licencji MIT. Jest on zbudowany na frameworku Electron korzystającym z Node.js oraz przeglądarki Chromium.

Skąd pobrać

Aktualne stabilne wydanie Atom to wersja 1.5.4
<https://github.com/atom/atom/releases/>.

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Krótką historią przebiegu prac nad edytorem Atom.

- 11-2011 - projekt stworzenia edytora Atom oficjalnie
- 02-2014 - pierwsza publiczna beta
- 05-2014 - Atom został wypuszczony na licencji MIT
- 25-06-2015 - pierwsze stabilne wydanie oznaczone wersją 1.0
- 02-03-2016 - obecna wersja 1.5.4

Statystyki z okresu maj 2014 - maj 2015

750k pobrań, 102 wydane wersje, 300k aktywnych użytkowników, 1861 paczek, 596 motywów

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Co oferuje nam Atom?

- wieloplatformowość
- wbudowany manager pakietów
- autouzupełnianie
- podział interfejsu na podzielne panele
- wiele projektów w 1 oknie
- znajdź i zamień
- setki dostępnych motywów
- open source

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Darmowe alternatywy:

- Eclipse + PyDev
- PyCharm Community Edition
- Vim + masa wtyczek

Płatne alternatywy:

- PyCharm
- Komodo Edit
- Sublime Edit

Co oferuje nam Atom po instalacji?

Wstęp

Czym jest Atom?
Historia
Zalety
Alternatywy

Wbudowane pakiety

Wstęp
Tree View
Snippets

Dodatkowe pakiety

Wstęp
Autocomplete
Lint
Flake8
Formatowanie składni
Terminal
Emmet
Historia
Git
Snippets

Co dalej?

- podstawowe autouzupełnianie + css, html
- kolorowanie składni dla większości języków programowania
- początek i koniec nawiasów (ctrl+m)
- zawijanie klas (ctrl+alt+[])
- znajdź i zamień (ctrl+alt+F)
- git diff
- Tree View
- snippets

Tree view pozwala nam widzieć strukturę naszego projektu. W większości edytorów, IDE nie jest to żadna nowość. Lecz i tutaj mamy kilka dodatkowych funkcjonalności.

- ukrywanie wybranych plików (.gitignore bądź własny plik)
- Reveal in Tree View
- wsparcie dla kontroli wersji git (zmienione pliki)
- Tree View po lewej bądź prawej stronie edytora

Chowanie Tree View

Jeżeli chcesz chować/wyświetlać Tree View użyj skrótu klawiszowego `Ctrl + \`

Snippets

Snippet – pojęcie programowania oznaczające zwykle mały wycinek kodu źródłowego, kodu maszynowego lub tekstu do wielokrotnego użycia. Snippetów używa się zazwyczaj w celu zminimalizowania powtarzalności tego samego kodu lub tekstu. Atom pozwala nam w łatwy sposób tworzyć nowe snippety, publikować je jako pakiety.

Przykładowy snippet?

```
'source.js':  
'if, else if, else':  
  'prefix': 'ieie'  
  'body': """  
    if (${1:true}) {  
      $2  
    } else if (${3:false}) {  
      $4  
    } else {  
      $5  
    }  
  """
```

Konfiguracja edytora Atom dla programistów Django

Piotr Kulas

Wstęp

- Czym jest Atom?
- Historia
- Zalety
- Alternatywy

Wbudowane pakiety

- Wstęp
- Tree View
- Snippets

Dodatkowe pakiety

- Wstęp**
- Autocomplete
- Lint
- Flake8
- Formatowanie składni
- Terminal
- Emmet
- Historia
- Git
- Snippets

Co dalej?

Czy "czysty" Atom nam wystarczy?

autocomplete-python

Obecnie najlepsza dostępny pakiet do podpowiadania składni dla Pythona

- inteligente autouzupełnianie
- wsparcie dla virtualenv
- go to definition
- override methods

Instalacja pakietów

Aby zainstalować dodatkowe pakiety trzeba użyć menedżera pakietów dostępnego w Atomie bądź używając terminala oraz komendy
`apm install <packagename >`

Nie zapomnij !

Po instalacji trzeba ustawić ścieżki dla interpretera Python oraz pakietów dystrybucyjnych.

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

W Atomie mamy dostępnych dużo dostępnych paczek automatycznego sprawdzania błędów podczas pisania kodu.

Najbardziej przydatne dla programistów Django będą:

- linter-flake8
- linter-js-standard
- linter-bootlint
- linter-htmlhint

Nie zapomnij !

Każda z powyższych paczek wymaga także by w Atomie była zainstalowana jego bazowa wersja - linter.

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Co zawiera w sobie flake8?

- sprawdzanie czy kod jest zgodny ze standardem pep8
- pyflakes
- Ned Batchelder's McCabe script

Nie zapomnij !

Wtyczka wymaga zainstalowanego flake8 w systemie
`sudo pip install flake8`

Pakiet atom-beautify pozwala na automatyczne formatowanie kodu w pliku dla większości języków programowania.

Dla każdego wspieranego języka można dodatkowo tworzyć własną konfigurację.

Skrót klawiszowy

Ctrl+Alt+B będzie używany do formatowania kodu po instalacji pakietu.

Nie zapomnij !

Wtyczka wymaga zainstalowanego autopep8 w systemie
`sudo pip install autopep8`

Terminal Plus

Przy pisaniu aplikacji webowych w Django konsolę musimy mieć zawsze pod ręką.

Po co zaśmiewać sobie listę otwartych aplikacji, skoro możemy mieć to od razu w Atomie używając terminal-plus?

Zalety

- możliwość otwierania więcej niż 1 terminala
- nie zabiera miejsca, wysuwamy tylko gdy potrzebujemy
- możliwość konfiguracji terminala (czcionka, kolory)
- możliwość nazewnictwa oraz kolorowania otwartych tabów
- każdy terminal może mieć inną wielkość
- domyślnie otwiera się w katalogu projektu
- można mapować terminale do konkretnych plików

Skrót klawiszowy

Ctrl+ pozwala nam chować/wysuwać panel terminali.
ctrl+enter pozwala wykonać zaznaczony w edytorze tekst

Emmet jest narzędziem do szybkiego tworzenia kodu, przede wszystkim używa się go do HTML oraz CSS. Pozwala on nam wygenerować w prosty sposób bloki html z uzupełnionymi klasami, polami.

Wspiera mnożenie, numerację, wspinanie, atrybuty oraz grupowanie.

```
nav>ul>li
wygeneruje nam
<nav>
  <ul>
    <li></li>
  </ul>
</nav>
```

Cheat Sheet

<http://docs.emmet.io/cheat-sheet/>

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Tak jak w popularnych IDE Atom także zawiera wtyczki, które pozwalają nam na lokalną historię pliku oraz historię pliku dla GIT. Jeśli chodzi o historię lokalną, przy każdym zapisie pliku wtyczka ta generuje kopię pliku, więc w każdym momencie możemy wrócić do kodu, który był pewien czas temu.

Instalacja

```
apm install local-history
```

```
apm install git-history
```

Obsługa git z poziomu Atomu

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Większość osób woli obsługiwać git graficznie. Z pomocą przychodzi tu wiele wtyczek, manager pakietów zawiera kilkanaście wtyczek do współpracy edytora z git. Oprócz tego, co mamy wbudowane w Atomie warto zainstalować:

- git-plus, który zawiera większość poleceń do obsługi git.
- tree-view-git-status pokazujący na którym branchu jesteśmy, oraz ile commitów z przodu/do tyłu
- git-history

Paczki ze snippetami

Wstęp

Czym jest Atom?

Historia

Zalety

Alternatywy

Wbudowane pakiety

Wstęp

Tree View

Snippets

Dodatkowe pakiety

Wstęp

Autocomplete

Lint

Flake8

Formatowanie składni

Terminal

Emmet

Historia

Git

Snippets

Co dalej?

Spółeczność stworzyła kilkadziesiąt paczek ze snippetami, te które mogą nam się przydać to:

- css-snippets
- atom-bootstrap3
- javascript-snippets

Django

Nie ma aktualnie godnych polecenia wtyczek zawierających snippety dla Django.

Co dalej?

Wstęp

Czym jest Atom?
Historia
Zalety
Alternatywy

Wbudowane pakiety

Wstęp
Tree View
Snippets

Dodatkowe pakiety

Wstęp
Autocomplete
Lint
Flake8
Formatowanie składni
Terminal
Emmet
Historia
Git
Snippets

Co dalej?

Atom ciągle się rozwija w zawrotnym tempie, lecz to rozwój wsparcia dla konkretnych języków i frameworków zależy od społeczności!
Więc dlaczego by nie zrobić czegoś dla Atoma?

- snippety dla Django
- auto import

Pomysły?

Dziękuję za uwagę !

Literatura

<https://pl.wikipedia.org/>

<https://atom.io/>

<https://flake8.readthedocs.org/>

Kontakt

Piotr Kulas

B207