

Instrukcja wygenerowania dokumentacji w KSS.

1. Instalujemy node.js wraz z npm, zawsze wersję LTS (latest stable):

<https://nodejs.org/en/learn/getting-started/how-to-install-nodejs>

<https://nodejs.org/en/download>

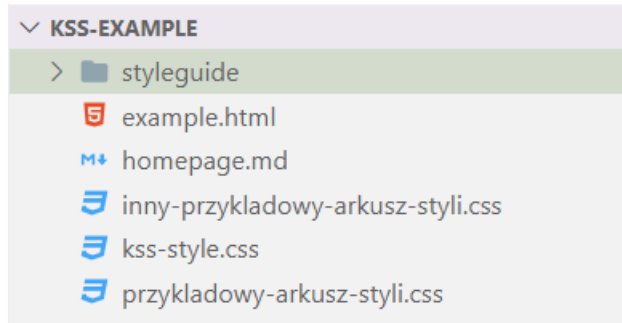
2. Jeżeli instalacja node pójdzie pomyślnie w konsoli będzie dostępne polecenie npm:

```
PS C:\Users\pracownik> npm -version  
8.1.2  
PS C:\Users\pracownik>
```

3. Wpisujemy polecenie (globalna instalacja pakietu kss):

```
npm install -g kss
```

4. Następnie posiadając następującą strukturę projektu:

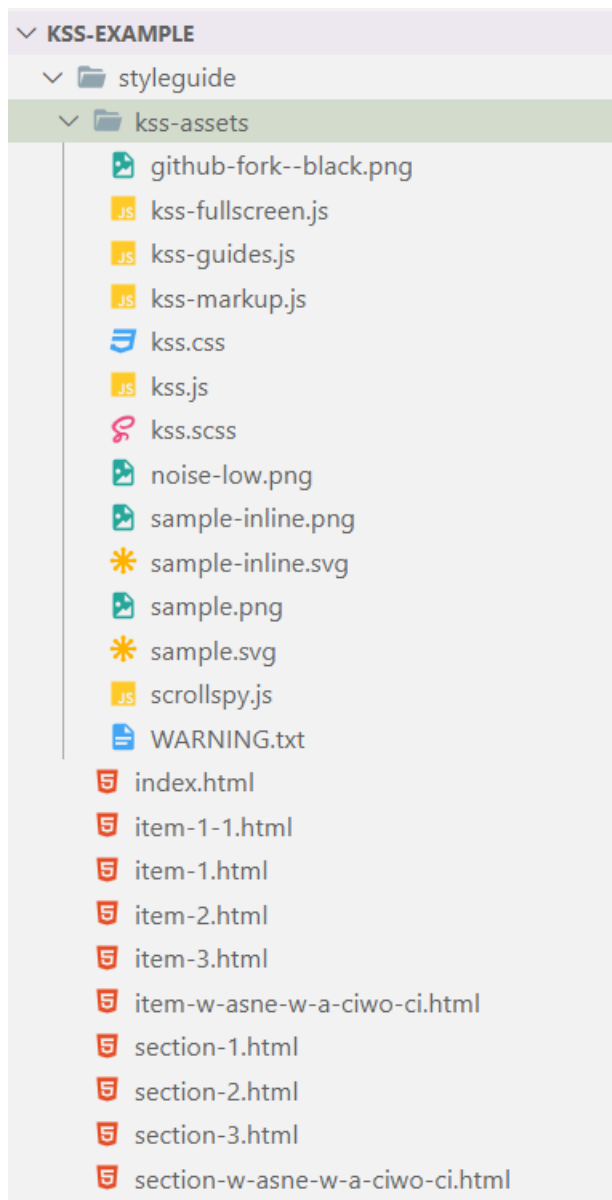


5. Możemy wpisać polecenie (z poziomu folderu kss-example w konsoli):

```
kss --css ../kss-style.css --source ./
```

Ustawiając source na bieżący folder spowodujemy że wszystkie pliki w folderze kss-example będą brane pod uwagę przy tworzeniu dokumentacji (więc jeżeli posiadamy więcej niż jeden arkusz CSS, to każdy z nich zostanie przeanalizowany czy nie zawiera dokumentacji; dotyczy to też przykładów użycia w HTML). Arkusz **kss-style.css** będzie załączony do wygenerowanej dokumentacji i musi zawierać wszystkie dokumentowane klasy/selektory/itd., jednak bez dokumentacji (gdyby ją też zawierał, to wtedy dokumentacja zostanie wygenerowana podwójnie). Żeby polecenie zadziałało musi istnieć **plik homepage.md** oraz **pusty folder styleguide**.

6. Jeżeli wszystko będzie poprawnie uzyskamy następujący efekt:



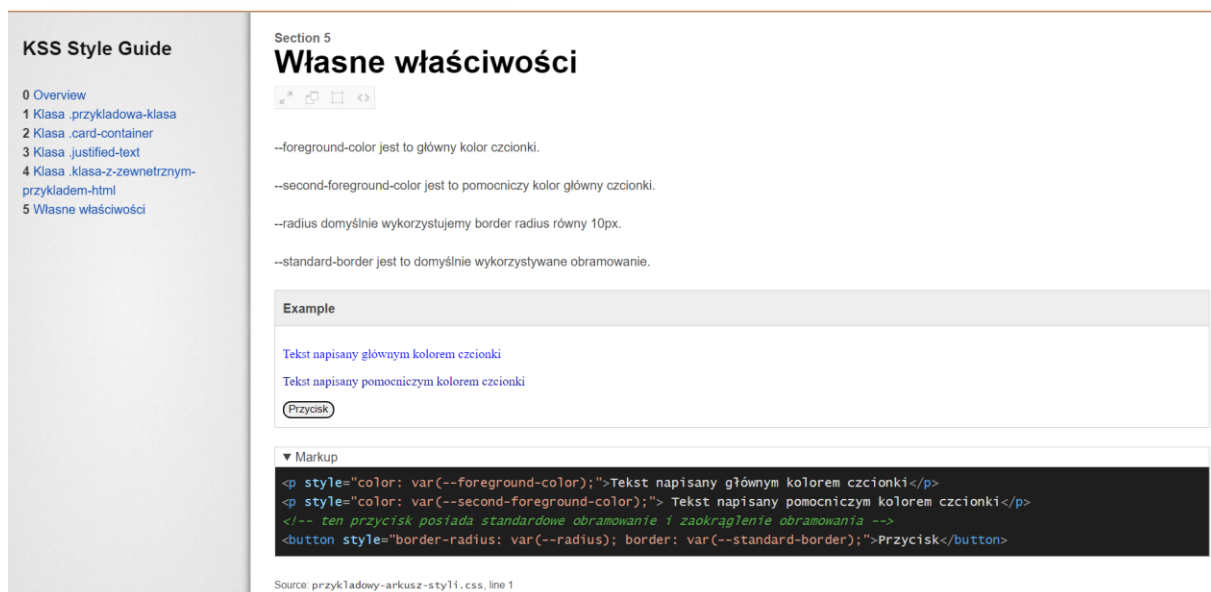
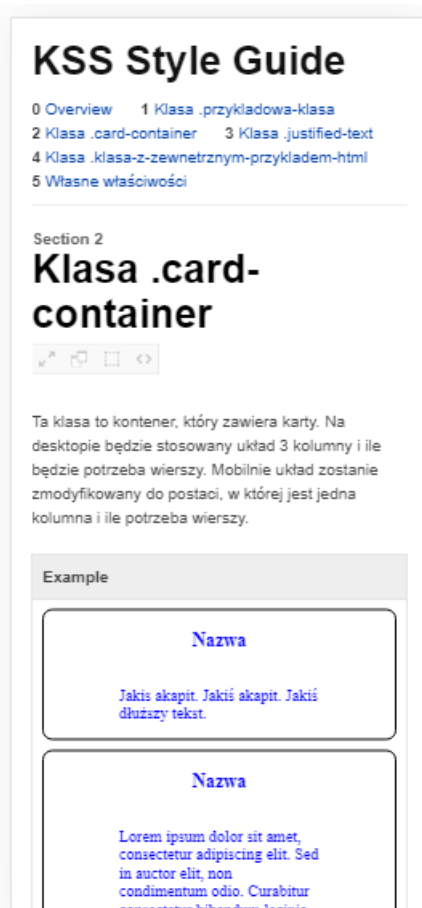
7. Otwierając plik index.html (albo lokalnie, albo za pomocą live server):



- 2 Klasa .card-container
- 3 Klasa .justified-text
- 4 Klasa .klasa-z-zewnetrznym-
przykladem-html
- 5 Własne właściwości

Tekst w karcie posiada ograniczoną maksymalną szerokość ustawioną na 200px.

```
<div class="card-container">  
<div class="przykladowa-klasa">
```



8. Więcej przykładów jak można tworzyć dokumentację w KSS można znaleźć tutaj: <https://github.com/kss-node/kss/blob/spec/SPEC.md>
9. UWAGI odnośnie składni KSS
 - Style guide przyjmuje albo wartości liczbowe, albo nazwy własne.
 - Markup poza inline może być zewnętrznym plikiem HTML (jak w odpowiednim przykładzie)
 - Jeżeli napiszemy media queries, to będą one zachowane w przykładach podczas zmiany rozmiaru okna

- *Możliwe jest również lekko inne podejście do generowania dokumentacji; bardziej oparte o tworzenie własnych komponentów; jednak trzeba tutaj wykorzystywać plik .hbs (<https://handlebarsjs.com/>). **Jest to rozwiązanie nieobowiązkowe, wyłącznie dla osób chętnych (jeżeli wolą pracować i dokumentować w tym duchu):**

```
/*
Button

Your standard button suitable for clicking.

.success - Everything fine, click to accept.
.warning - Warn the user, a click might be dangerous.
.error - Indicate that an action failed.

Markup: ./button.hbs

Style guide: components.button
*/
.button {
  background: transparent;
}

.button.success {
  background: green;
}

.button.warning {
  background: orange;
}

.button.error {
  background: red;
}
```

Section 2

components

Section 2.1

Button



Your standard button suitable for clicking.

Examples

Default styling

`test`

.success Everything fine, click to accept.

`test`

.warning Warn the user, a click might be dangerous.

`test`

.error Indicate that an action failed.

`test`

▼ Markup: button.hbs

```
<button class="button {{modifier_class}}">test</button>
```

Source: kss-style.css, line 1