# Code review

# Wyłączenie addEventListener

*const* buttonRock = document.getElementById('button\_rock'),

    buttonPaper = document.getElementById('button\_paper'),

    buttonScissors = document.getElementById('button\_scissors');

buttonRock.addEventListener ('click', *function*() {

    playGame (1)

    console.log(roundNumber, ' Numer rundy');

});

buttonPaper.addEventListener ('click', *function*() {

    playGame (2)

});

buttonScissors.addEventListener ('click', *function*() {

    playGame (3)

});

# Początkowa wartość zmiennych globalnych

'use strict';

{

*let* playerPoints = 0,

        computerPoints = 0,

        roundNumber = 0,

        score;

# Dlaczego <span> został tutaj użyty?

<ul class="list titles">

<li><a href="#article-1" class="active"><span>Article 1</span></a></li>

<li><a href="#article-2"><span>Article 2</span></a></li>

<li><a href="#article-3"><span>Article 3</span></a></li>

<li><a href="#article-4"><span>Article 4</span></a></li>

<li><a href="#article-5"><span>Article 5</span></a></li>

<li><a href="#article-6"><span>Article 6</span></a></li>

<li><a href="#article-7"><span>Article 7</span></a></li>

<li><a href="#article-8"><span>Article 8</span></a></li>

<li><a href="#article-9"><span>Article 9</span></a></li>

<li><a href="#article-10"><span>Article 10</span></a></li>

</ul>

# Miejsce const a działanie metody classList.remove & classList.add

*const* activeArticles = document.querySelectorAll('article.active');

    console.log(activeArticles, ' -> active articles');

*const* activeLinks = document.querySelectorAll('.titles a.active');

    console.log(activeLinks, ' -> all the articles active title links from a sidebar');

* Wyciągnięcie powyższych zmiennych poza funkcję titleClickHandler powodowało niewłaściwe działanie metod (możliwe było nadanie klasy .active kilku elementom naraz i jednocześnie nie dało się nadać .active ponownie na Article 1, który domyślnie taką klasę posiadał

// Function titleClickHandler - delete an active class from sidebar's links && articles

*const* titleClickHandler = *function* (*event*) {

    console.dir(titleClickHandler);

    // this

*const* clickedElement = this;

    console.log(clickedElement, ' -> titleClickHandler clickedElement');

    // Event element & currentTarget

    console.log(*event*, ' -> titleClickHandler event');

    console.log(*event*.currentTarget, ' -> titleClickHandler currentTarget');

    // Articles' active title links from a sidebar

*const* activeLinks = document.querySelectorAll('.titles a.active');

    console.log(activeLinks, ' -> all the articles active title links from a sidebar');

    // Loop - a single active link from a sidebar - removing class 'active'

    for (*let* activeLink of activeLinks) {

        activeLink.classList.remove('active');

    }

    // Add an active class to the clicked element

    clickedElement.classList.add('active');

    // All active articles

*const* activeArticles = document.querySelectorAll('article.active');

    console.log(activeArticles, ' -> active articles');

    // Loop - a single active article - removing class 'active'

    for (*let* activeArticle of activeArticles) {

        activeArticle.classList.remove('active');

    }

}

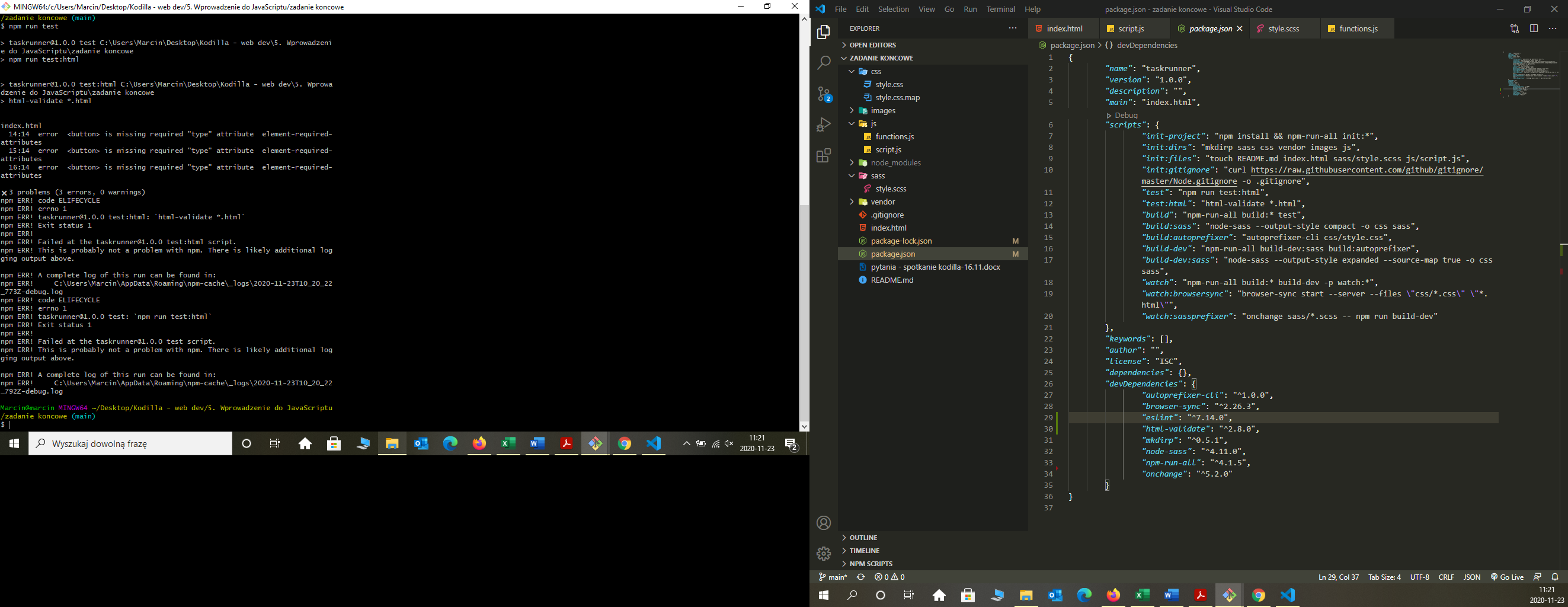
# Zmienna odnosząca się do tej samej wartości – czy lepiej tworzyć osobno (powielać) w każdej funkcji czy globalne

# Pobieranie atrybutu href a miejsce odwołania do funkcji titleClickHandler

# Npm run test

*"test"*: "npm run test:html",

*"test:html"*: "html-validate \*.html",



# Eslint

