

Hello, World!



KAJ BIAŁAS

JavaScript Developer

| Node JS

CZYM JEST **Node JS**

Środowisko pozwalające uruchamiać `JavaScript` poza przeglądarką.

Wykorzystuje silnik `JavaScript V8` - używany w przeglądarce: `Google Chrome` .

CZYM JEST **Node JS**

w **DUŻYM** uproszczeniu:

Technologia **back-end**'owa, pozwalająca uruchamiać język `JavaScript` na serwerze.

Back-end

VS

Front-end

Czym to się **różni**?

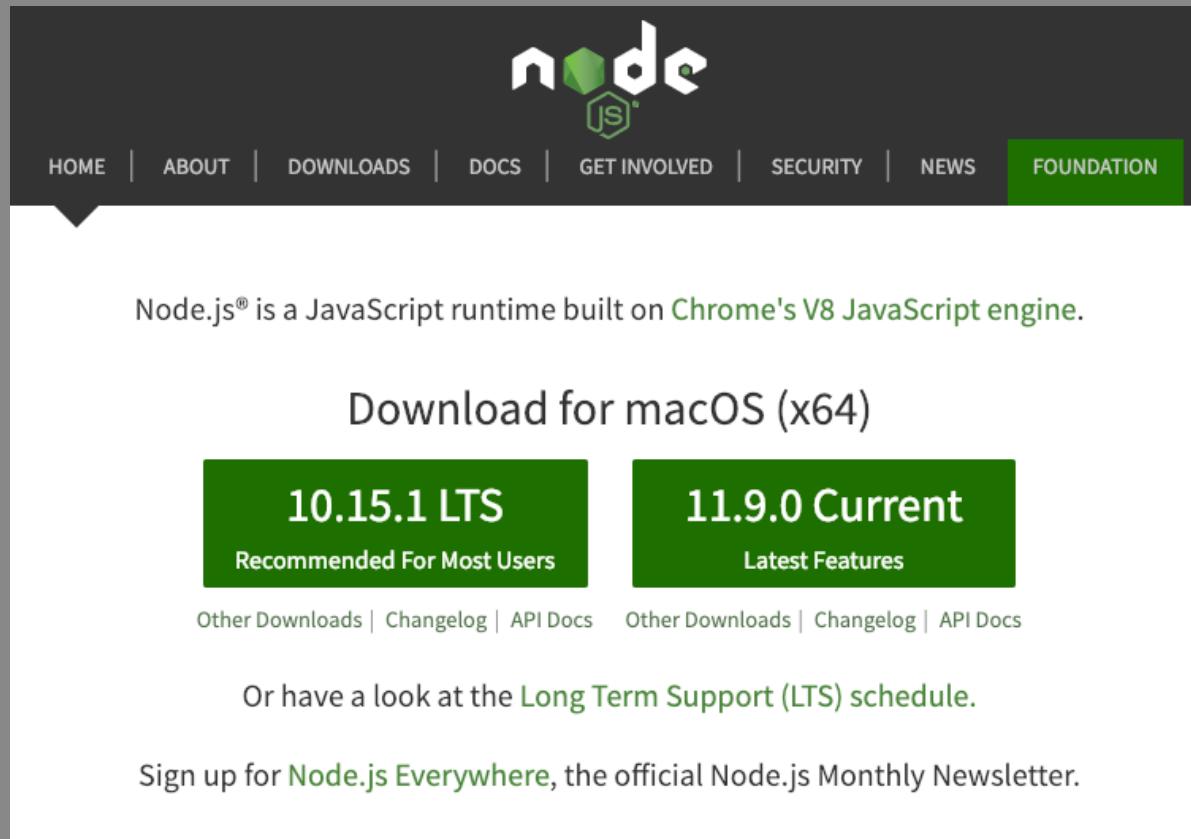
WERSJE BIBLIOTEKI Node JS

Long Time Support

VS

Current

WERSJE BIBLIOTEKI Node JS



The screenshot shows the Node.js website with a dark header containing the Node.js logo and a navigation menu. The main content area is white and features a description of Node.js, download links for macOS (x64), and information about LTS and Current versions.

node JS

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

Download for macOS (x64)

10.15.1 LTS
Recommended For Most Users

11.9.0 Current
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

WERSJE BIBLIOTEKI Node JS

Long Time Support

VS

Current

Node JS

Środowisko lokalne

Uaktualnienie `apt-get` .

```
$ sudo apt-get update && sudo apt-get -y upgrade
```

Instalacja `node JS` oraz `NPM` .

```
$ curl -sL https://deb.nodesource.com/setup_9.x | sudo -E bash -
```

```
$ sudo apt-get install -y nodejs
```

Sprawdzenie aktualnie zainstalowanej wersji `node JS` oraz `NPM` .

```
$ node -v
```

```
$ npm -v
```

Issue #0

Sprawdź posiadaną przez siebie wersję `node JS` oraz `NPM` .

Jeżeli nie posiadasz zainstalowanych tych pakietów, wykonaj ich instalację.

PIERWSZE KROKI Z Node JS

Pierwsze programy można pisać bezpośrednio w terminalu systemu operacyjnego, wystarczy wpisać polecenie:

```
$ node
```

Następnie możemy wykonać dowolny program, którego działanie będzie wyświetlone

```
> 2 + 3
```

Możemy też napisać program wyświetlający napis na konsoli

```
> console.log('Hello Node')
```

PIERWSZE KROKI Z Node JS

Działanie konsoli z uruchomionym wykonywaniem programów `node JS` można w dowolnej chwili przerwać, poprzez kombinację klawiszy:

CTRL + Z

Issue #1

Napisz program `node JS`, uruchamiany w terminalu, w którym utworzysz dwie zmienne: `x` oraz `y` i przypiszesz do nich wartości liczbowe.

Wyświetl w konsoli napis: "Sumą liczb {x} oraz {y} jest {z}."

PIERWSZE KROKI Z **Node JS**

Uruchamianie plików za pomocą `node JS` .

```
node example.js
```

Issue #2

Utwórz w nowym katalogu plik: `example-node.js`

Zaimplementuj w nim program, napisany na potrzeby: `#Issue 1`.

SERWER DANYCH W Node JS

W node JS, można obsłużyć protokół: `HTTP` , dzięki czemu wynik programu będzie wyświetlany, w postaci strony internetowej.

```
var http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World!');
}).listen(8080);
```


Issue #3

Zmodyfikuj program znajdujący się obecnie w pliku: `example-node.js`, tak aby wynik jego działania został wyświetlony, jako strona internetowa.

Strona internetowa powinna być dostępna na porcie: `3000`

Postaraj się, aby strona była dostępna lokalnie, pod adresem: `http://my-webpage.com:3000`

SERWER HTTP W Node JS

W node JS, możemy mieć dostęp do plików.

```
const http = require('http');
const fs = require('fs')
const index = fs.readFileSync('./index.html')

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer(function(req, res) {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/html');
  res.end(index);
});

server.listen(port, hostname, function() {
  console.log('Server running at http://' + hostname + ':' + port + '/');
});
```

Issue #4

Utwórz plik: `.html` oraz stwórz w nim prostą stronę internetową.

Zmodyfikuj program znajdujący się obecnie w pliku: `example-node.js`, tak żeby wyświetlił utworzony wcześniej plik: `.html`

| NPM

CZYM JEST **NPM**

Główne idee **NPM**, czyli **Node Package Manager**:

CZYM JEST **NPM**

Główne idee **NPM**, czyli **Node Package Manager**:

- Łatwość publikacji i instalacji bibliotek.

CZYM JEST **NPM**

Główne idee **NPM**, czyli **Node Package Manager**:

- Łatwość publikacji i instalacji bibliotek.

- Proste zarządzanie zależnościami.

CZYM JEST **NPM**

Główne idee **NPM**, czyli **Node Package Manager**:

- Łatwość publikacji i instalacji bibliotek.

- Proste zarządzanie zależnościami.

- Łatwe wersjonowanie aplikacji.

CZYM JEST **NPM**

Główne idee **NPM**, czyli **Node Package Manager**:

- Łatwość publikacji i instalacji bibliotek.

- Proste zarządzanie zależnościami.

- Łatwe wersjonowanie aplikacji.

- Ustrukturyzowane zarządzanie projektem.

PIERWSZE KROKI Z **NPM**

Tworzenie nowego projektu

```
$ npm init
```

Issue #5

Zainicjuj nowy projekt `NPM` .

Przyjrzyj się strukturze pliku: `package.json`

PIERWSZE KROKI Z NPM

Dodawanie nowych skryptów.

```
"scripts": {  
  "start": "node index.js",  
  "test": "node test.js"  
},
```

Uruchamianie skryptów

```
$ npm run start
```

```
$ npm run test
```

PIERWSZE KROKI Z NPM

Uruchamianie skryptów

```
$ npm start
```

```
$ npm test
```

Skróty, dotyczą **tylko** skryptów: `start` oraz `test` .

Issue #6

Wykorzystaj istniejący projekt `NPM` i rozwiązanie `#Issue 4` .

Utwórz skrypt uruchamiający stronę komendą `npm run serve` .

Wykorzystaj mechanizm zmiennych środowiskowych żeby podawać numer portu w skrypcie: `NPM` .

PIERWSZE KROKI Z **NPM**

Instalowanie zależności lokalne

```
$ npm install jquery
```

Instalowanie zależności globalne

```
$ npm install jquery -g
```

PIERWSZE KROKI Z **NPM**

Instalowanie zależności lokalne z dodaniem do package.json

```
$ npm install jquery --save
```

```
$ npm install jquery --save-dev
```

Instalowanie zależności zawartych w package.json

```
$ npm install
```


Issue #7

Wykorzystaj istniejący projekt z `#Issue 4` i dodaj do niego bibliotekę `jQuery` .

Biblioteka `jQuery` , powinna zostać w przyszłości automatycznie zainstalowana po wybraniu polecenia: `$ npm install` .

| Webpack

CZYM JEST **WEBPACK**

Jest to najpopularniejsze narzędzie - `bundler` , używany w projektach java-script.

Potrafi on spakować wiele różnych typów zasobów do jednego wynikowego zasobu.