infoShare ACADEMY

Javascript

Narzędzia, środowiska uruchomieniowe, podstawy składni

infoShare Academy

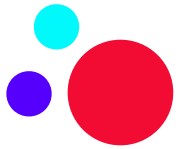
infoShareAcademy.com

HELLO

Jakub Wojtach

Senior **full stack** developer

infoShare ACADEMY



Rozgrzewka i agenda

Zacznijmy ten dzień z przytupem!

Rozgrzewka

Chłopaki nie płaczą

Poranek kojota

Rozgrzewka

Robert Lewandowski

Adam Małysz

Rozgrzewka

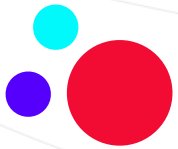
Jogurt naturalny

Kefir

Rozgrzewka

NFT

Obraz Da Vinci

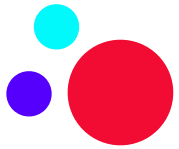


Agenda

- Podstawy teorii
- Zakładki Console i Sources
- Środowisko Node.js
- Debugowanie

Współpraca

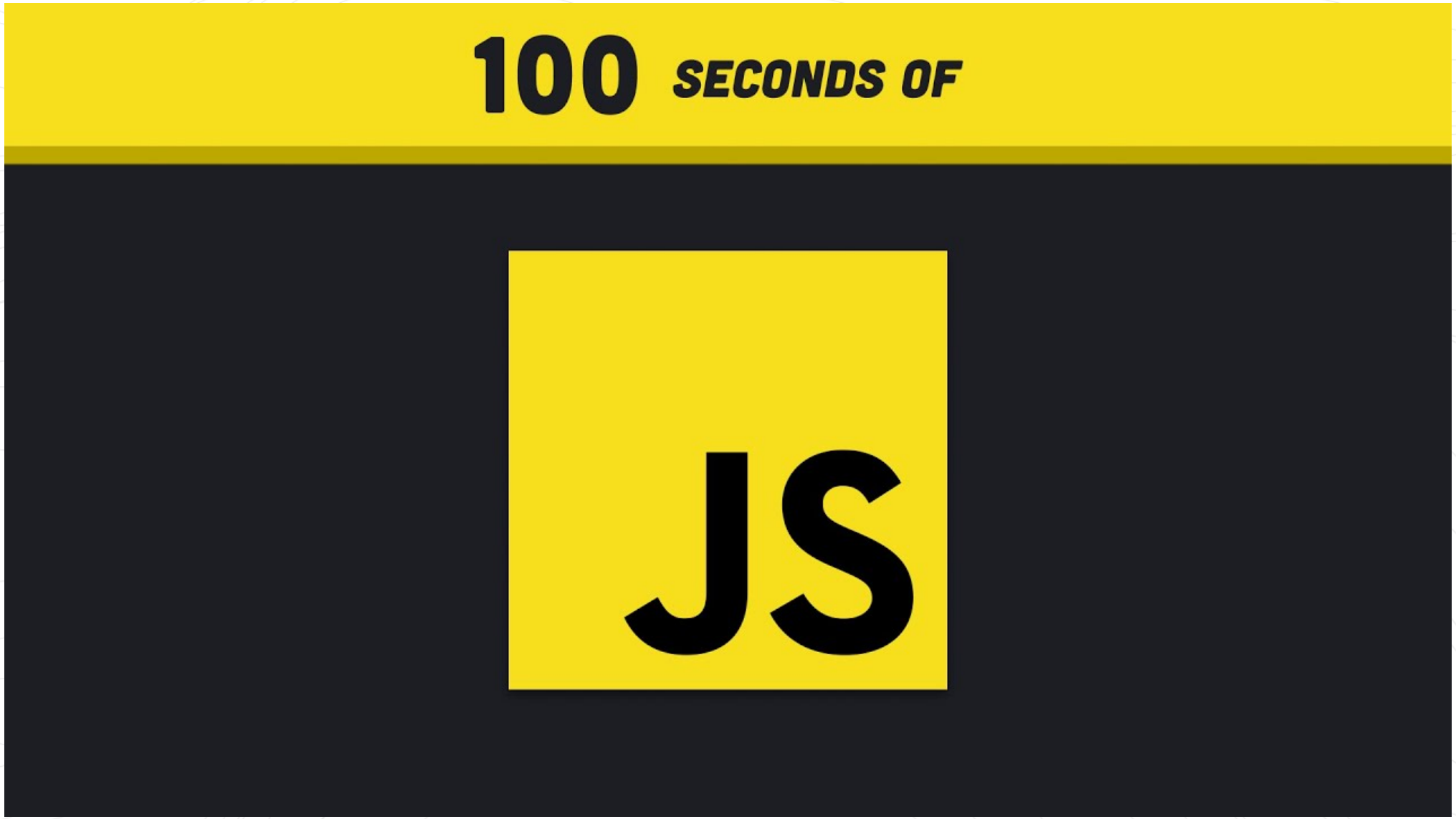
- Zadajemy pytania w dowolnym momencie - kanał **merytoryka**
- Krótkie przerwy (**5 min**) co godzinę
- Długa przerwa (**20 min**) po ostatnim bloku

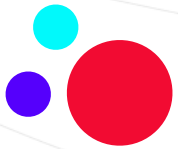


Podstawy teorii

infoShareAcademy.com

Po krótce

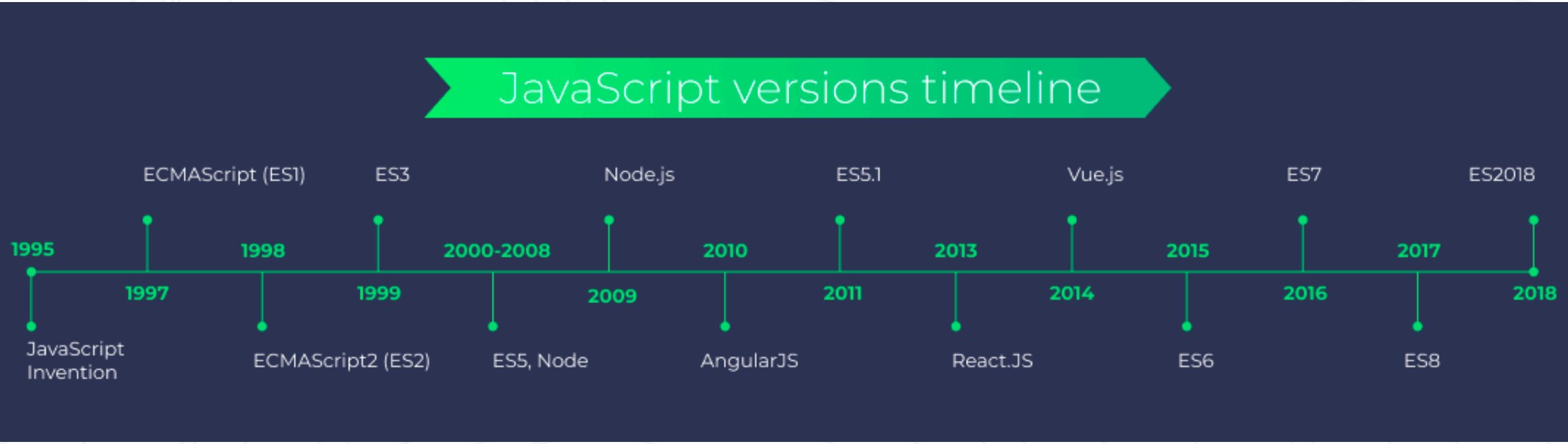


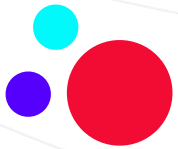


Przeznaczenie języka

- Tworzenie i kontrolowanie dynamicznej zawartości strony internetowej
- Pozwala na wykonywanie akcji bez konieczności przeładowania witryny
- Możemy obsłużyć nim zdarzenia, budować i walidować formularze, tworzyć skomplikowane systemy, gry i... tak naprawdę ograniczeń brak.

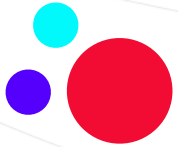
Linia czasu





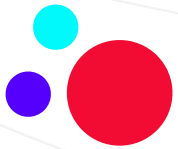
TC39 I ECMAScript

- **TC39** - komisja przygotowująca tekst standardu ECMAScript, pracuje nad przeglądaniem i oceną propozycji zmian i przygotowuje testy używane do weryfikacji standardu ECMA. Powstała w 1996 roku.
- **ECMAScript** - ustandaryzowaną specyfikacja obiektowego języka programowania (JavaScript, Script, ActionScript). Język opisany nią będący początkowo skryptowym, w tym momencie jest pełnoprawnym językiem programowania ogólnego zastosowania. Pojawiła się w 1997 roku.



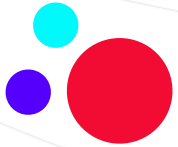
Historia rozwoju języka

- 1993 rok - **Brendan Eich** tworzy język w 10 dni. Złośliwi dodają, że na kacu.
- Javascript trafia do przeglądarki Netscape Navigator 2.0
- Microsoft kopiuje język w swojej przeglądarce.
- Jest rok 1997. Javascript zaczyna podbijać rynek..
- Pierwsza wersja standardu ECMA. 300 000+ stron używających **JS**
- 1999 rok - poważna transformacja wraz z wejściem ECMA Script 3 - pętle, wyjątki, try/catch. Ciągi, tablice, poprawiona obsługa błędów. **ASYNCHRONICZNE żądania do serwera!!**
- **2005** - pojawienie się AJAX, Prototype, jQuery, ułatwienie pracy z DOM.



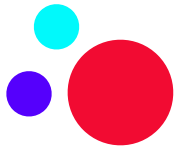
Historia rozwoju języka

- Niewielkie aktualizacje przez lata wraz z kolejnymi standardami aż do lat **2015** i **2016.** Wbudowane podklasy, standard unicode, rest operator.
- **2017** Object.values, Object.entries, Object Spread/Rest, asynchroniczne iteratory.



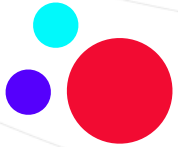
Wykorzystanie JS

- Przeglądarki internetowe.
- Serwer (Node.js)
- IoT (np. JerryScript)
- Mobile (React Native, Flutter)
- Desktop (Electron)



Zakładki DevTools

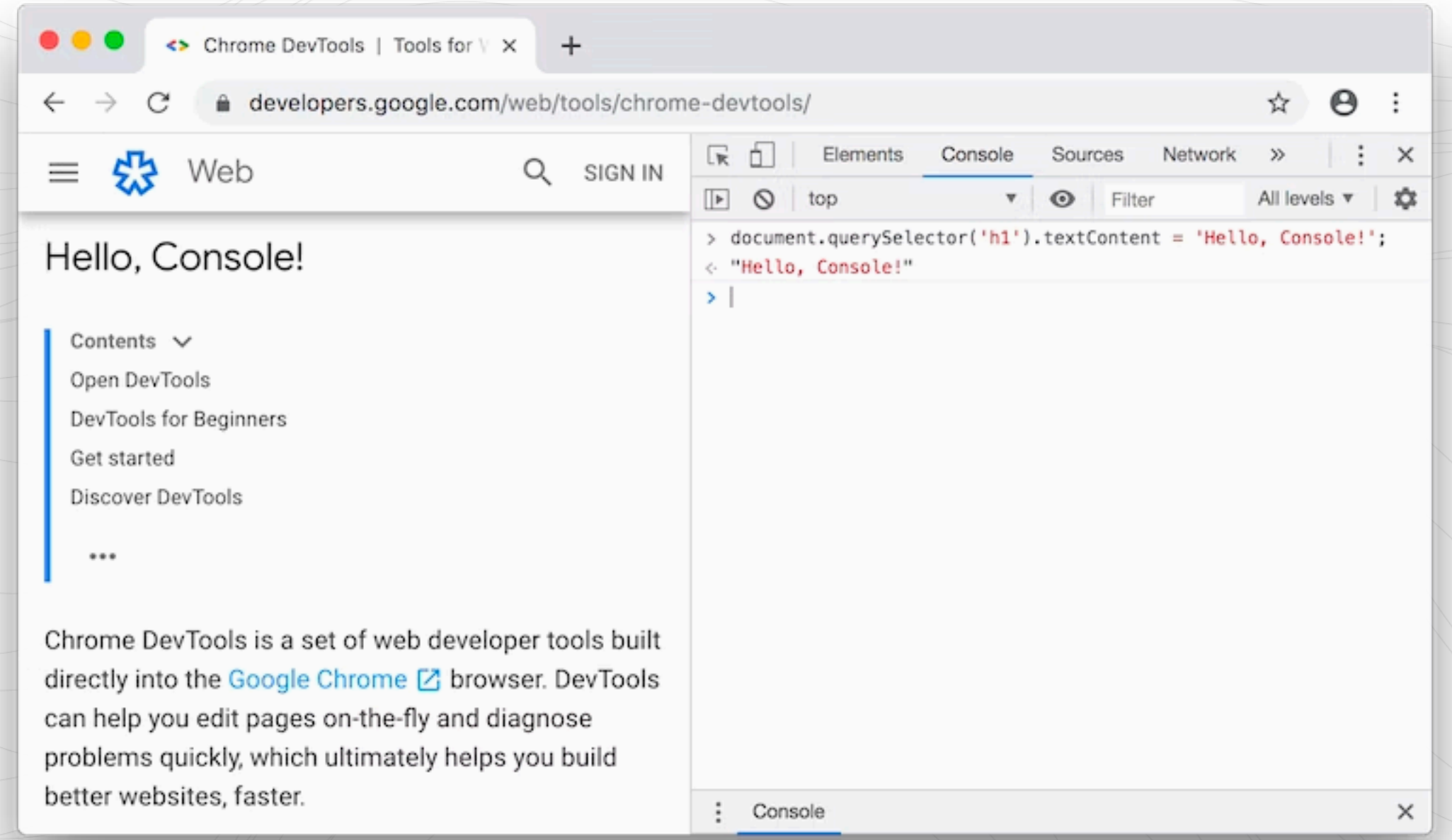
Console, Sources



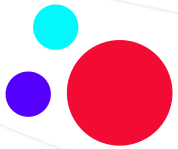
Console

- assert() - wyświetla error msg w momencie gdy założenie **nie** jest spełnione
- clear() - czyści konsolę gdy jest to możliwe.
- dir() - lista interaktywna propertiesów elementu. (Przykład **document.location**)
- error() - wyświetla error message do konsoli
- group() - tworzy nową grupę w konsoli, powodując że elementy do niej należące będą wcięte na tej samej wysokości
- groupCollapsed() - tak jak group, tyle, że lista jest domyślnie zwinięta.
- groupEnd() - zakańcza daną grupę rozpoczętą poleceniem group()
- info() - wyświetla info message w konsoli, w firefoxie z dodatkową ikoną “i”.
- log() - wyświetla wiadomość do konsoli. Najczęściej stosowana.

REPL przeglądarki



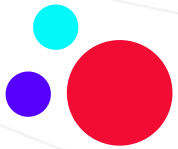
infoShare ACADEMY



Obiekty wbudowane

- Window
- Location
- Document
- String
- Date
- Math

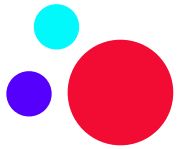
Konsola - skróty klawiszowe



Sources

- Tworzymy własny kod w zakładce snippets.
- Piszemy kod
- Uruchamiamy przyciskiem play na dole

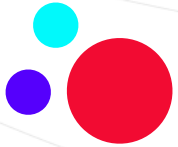
Zadanie



Środowisko

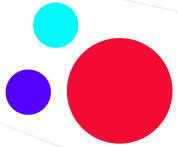
Node.js

infoShare ACADEMY



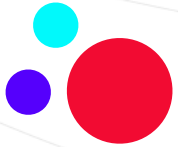
Wprowadzenie

- Wieloplatformowe środowisko uruchomieniowe JavaScript - można uruchomić za jego pomocą kod Javascript bez wykorzystania przeglądarki.
- Oparty o silnik V8, używany chociażby w Google Chrome. Szybki i niezawodny.
- Node.js sam w sobie nie jest serwerem, ale pozwala na stworzenie własnego serwera HTTP lub innych usług sieciowych.
- Daje dostęp do systemu plików, metod pracy z plikami oraz wykonanie wielu operacji w systemie operacyjnym.



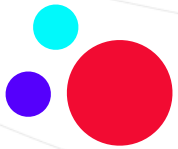
Node.js a przeglądarka

- W Node.js brakuje wielu interfejsów API przeglądarki, takich jak wszystko związane z DOM i CSS, wydajnością, dokumentem, interfejsami API związanymi z oknem. Tak więc ze względu na logikę obiekt globalny został przemianowany na **global**, ponieważ nie odnosi się do window i nie ma właściwości podobnych do niego



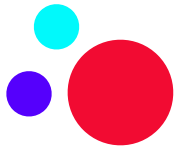
Node.js zastosowanie

- Node.js jest przede wszystkim używany do tworzenia serwerów internetowych i narzędzi CLI, jednak node.js może zrobić o wiele więcej.
- Może zrobić całkiem dużo wszystkiego, co można zrobić z natywną aplikacją. Gry. Aplikacje na PC. Slack i doświadczenie Geforce jest zrobione z node.js poprzez electron. Nawet całe IDE można zrobić, vscode został wykonany w node.js przez electron i działa świetnie.
- Kompilator typescript działa w node.js.
- Npm działa w node.js.
- Możesz nawet rozszerzyć to, co node.js może zrobić za pomocą C ++, jeśli biblioteka dla tego, co chcesz, jeszcze nie istnieje.
- Z node.js prawie wszystko, czego możesz chcieć, jest w zasięgu kilku instalacji npm.



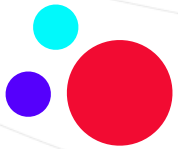
Node.js REPL i parametry wejściowe

- https://nodejs.dev/en/learn/how-to-use-the-nodejs-repl/
- https://nodejs.org/en/knowledge/command-line/how-to-parse-command-line-arguments/



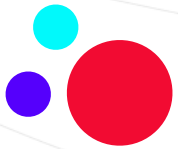
Debugowanie

Czyli analiza błędów



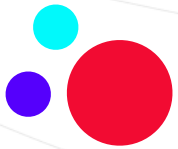
Sources

- https://developer.chrome.com/docs/devtools/javascript/



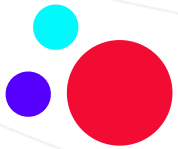
VS:Code Debugger

- https://youtu.be/_QtUGdaCb1c?t=550 i robimy to potem samemu
- Debugujemy kod z pliku script



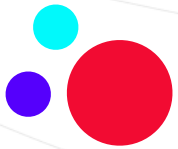
Praktyczny debug

- https://stackify.com/a-practical-guide-to-javascript-debugging/



Praktyczny debug - REST API

- Przejdziemy do mojego API, przejdziemy na branch na którym jest bug. Znajdziemy problem(y) i naprawimy go(je).
- Potem możemy popatrzeć na kod aplikacji i trochę go poanalizować - popatrzeć jak wygląda gotowe API działające w środowisku produkcyjnym

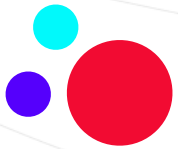


BONUS

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects

Pytania

infoShareAcademy.com



Linki dla chętnych

- https://tc39.es/
- https://www.ecma-international.org/publications-and-standards/standards/ecma-262/
- https://es6.io/ - płatny ale ŚWIETNY kurs z ES6
- JavaScript30.com - darmowy i ŚWIETNY kurs JS
- https://beginnerjavascript.com/ - kurs JS dla początkujących

Dziękuję za uwagę

Jakub Wojtach