Модули и сборка приложения

Модули будут иметь возможности импорта и экспорта, различные функции и др. Использование утилит для взаимообмена файлов js.

PS C:\plusi\00-js\project-start> npm --version

* 9.8.1

Npm – node package manager.

npm install chalk– установка библиотеки chalk.

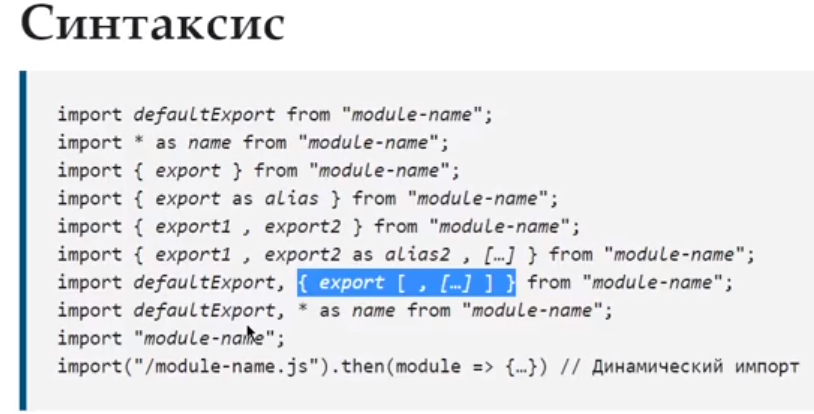
Деструктуризация массива

Обратите внимание, что в приведенном выше фрагменте Обратите внимание, что в приведенном выше фрагменте много повторяющегося кода, что не удовлетворяет принципу DRY.

А теперь посмотрим, как деструктуризация массива позволит нам сделать код более аккуратным и лаконичным.

* **const** profile = ["Oluwatobi", "Sofela", "codesweetly.com"];  
    
  **const** [firstName, lastName, website] = profile;  
    
  console.log(firstName); // "Oluwatobi"  
  console.log(lastName); // "Sofela"  
  console.log(website); // "codesweetly.com"

**const** profile = ["Oluwatobi", "Sofela", "codesweetly.com"];  
  
**const** [firstName, lastName, website] = profile;  
  
console.log(firstName); // "Oluwatobi"  
console.log(lastName); // "Sofela"  
console.log(website); // "codesweetly.com"



index.js

import {sum, addOne as add1} from './first';

import { firstName } from './second';

console.log('Hello World', firstName);

console.log(sum(2,5));

console.log(add1(10));

first.js

**const** x = 1

**function** addOne(num){

    return x + num;

}

**function** sum(a,b){

    return a+b

}

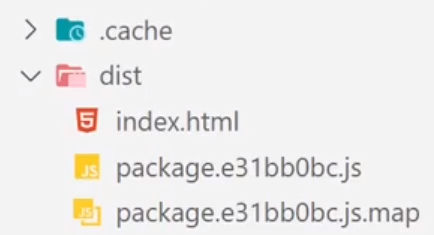
export {

    sum,

    addOne

}

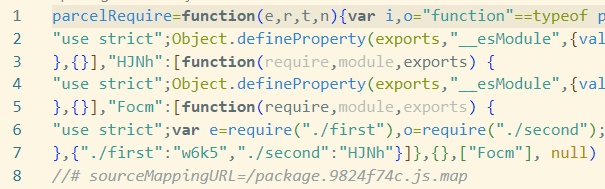
Parsing, rollup, esbuild etc.



Создание папок cahce, dist.

Во второй папке у нас создаются три файла. В js файле

У нас хранится исходный файл, только переформированный.

Js.map – делает соответствие с нашими файлами. Для удобного поиска ошибок.

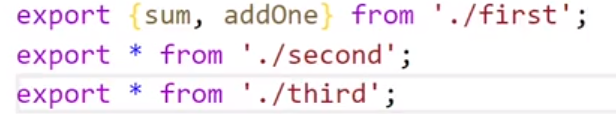
На выходе мы получаем бандл – зачастую это javascript файл.

Экспорт по-умолчанию экспортирует только что-то одно.

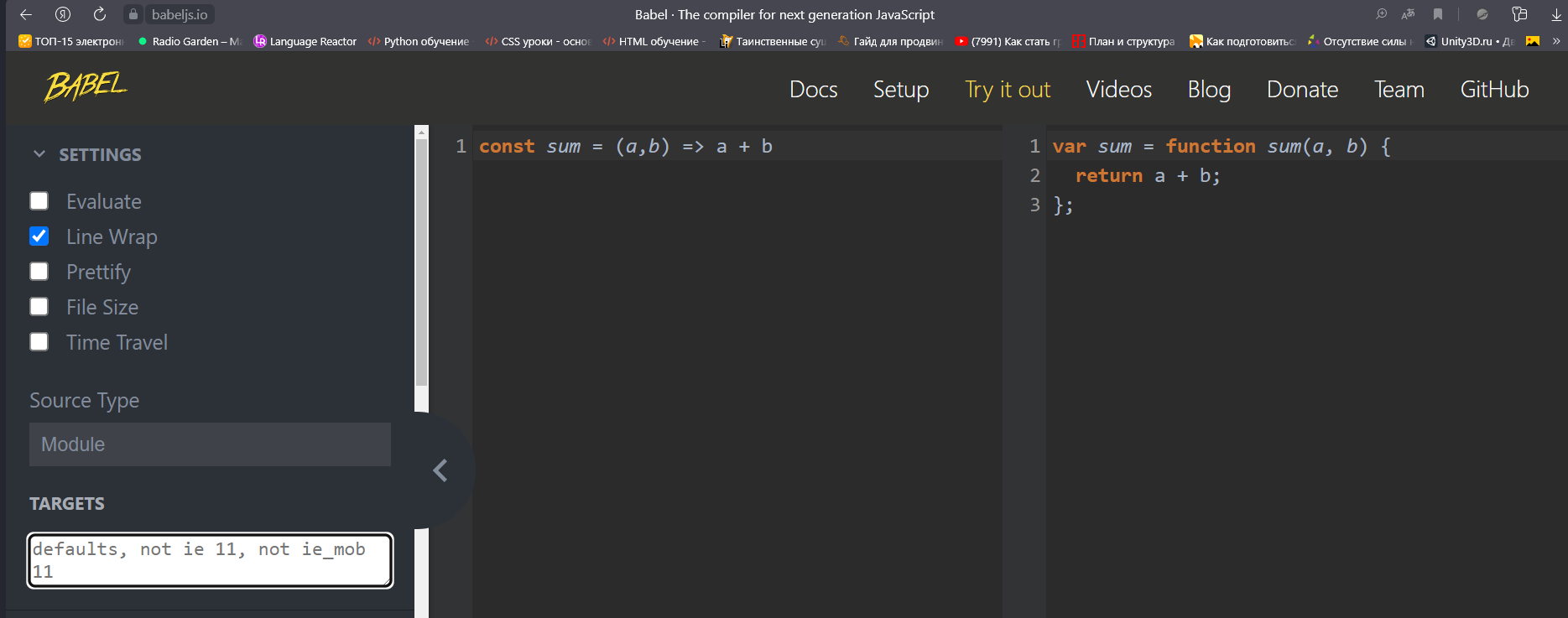
npm install chalk --save-dev– установка не в зависимостью

Реэкспорт – импорт + экспорт.

Создаём два файла с одинаковым названием – index.js. Один файл помещаем в новую папку heplers (где лежат остальные три js).

Затем во внутреннем файле производим экспорт first, second and third.js.

Во втором фале (изначальном index) мы по итогу импортируем информацию (данные).



Кроссбраузерность приложений

Работа с модулями. Mjs. Экспорт и импорт. Command js – для локальной

ES 6 Modules – для фронтенд-приложений. Их js можем в mjs, а наоборот – нет

Прилагаемые файлы

Package.json

{

  "name": "package",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "start": "parcel index.html",

    "build": "parcel build index.html",

    "print": "node index.js"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "lodash": "^4.17.21"

  },

  "devDependencies": {

    "chalk": "^5.3.0",

    "parcel-bundler": "^1.12.5"

  }

}

