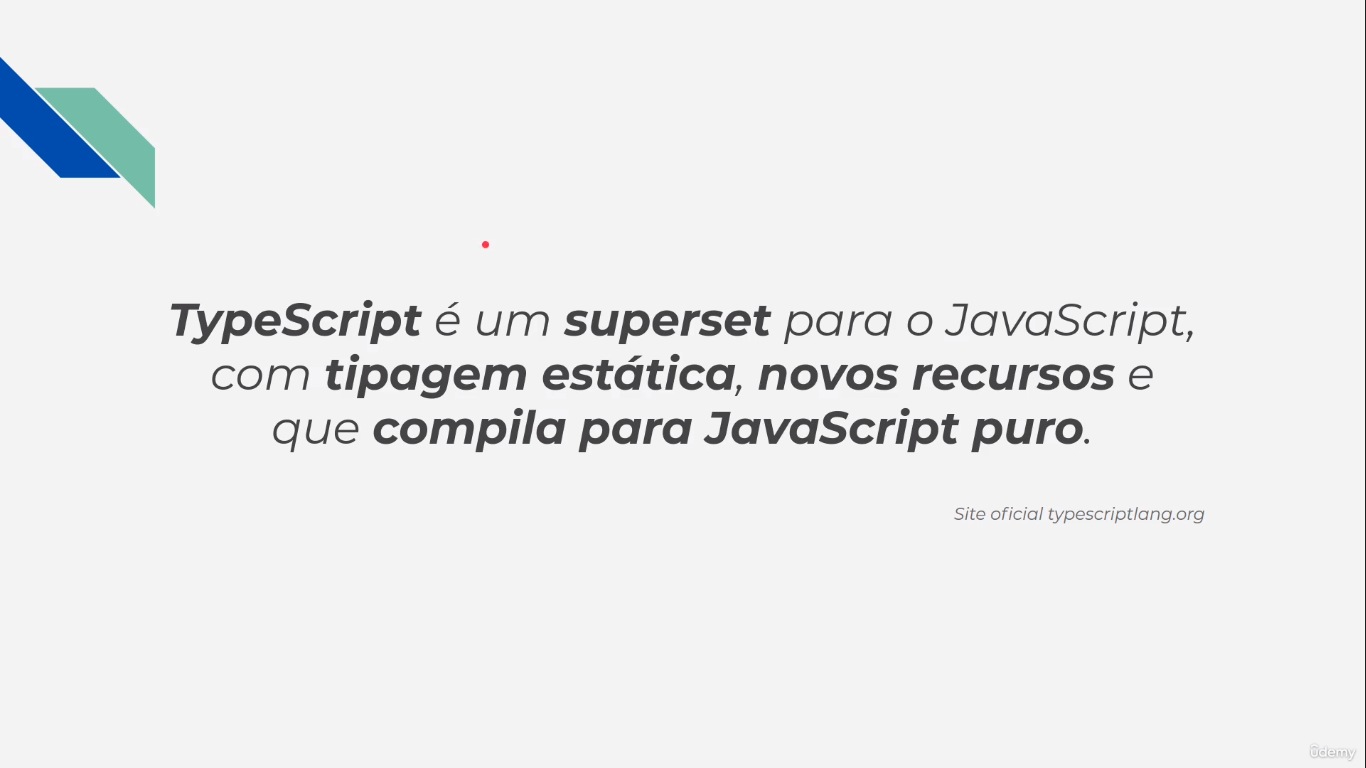
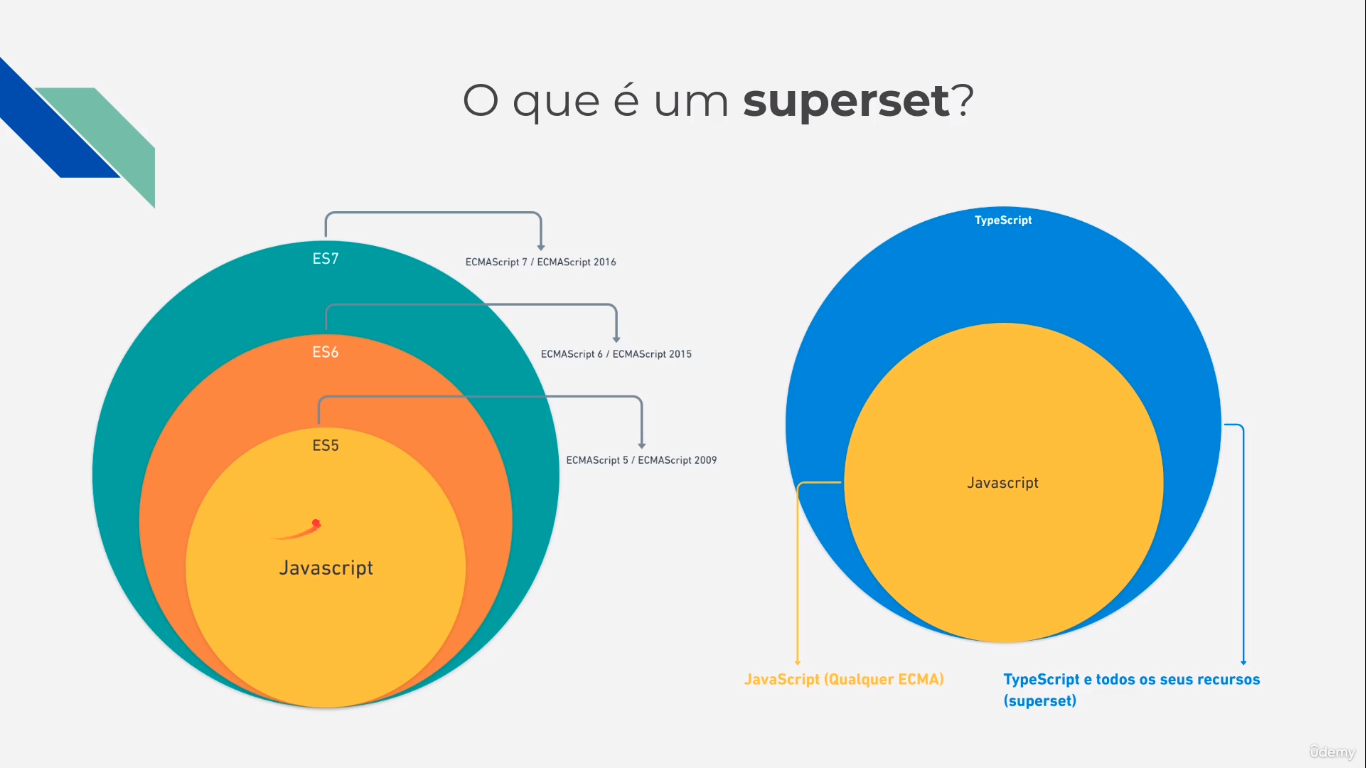
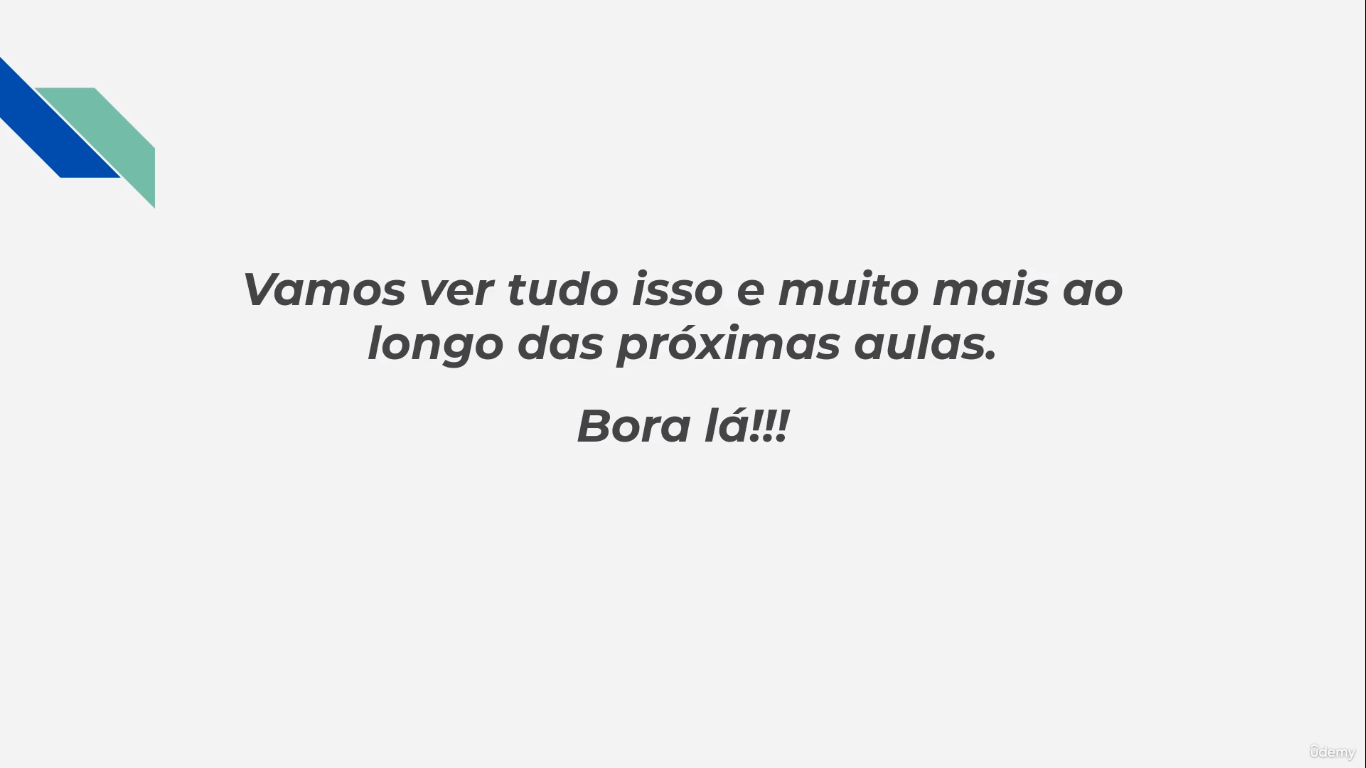
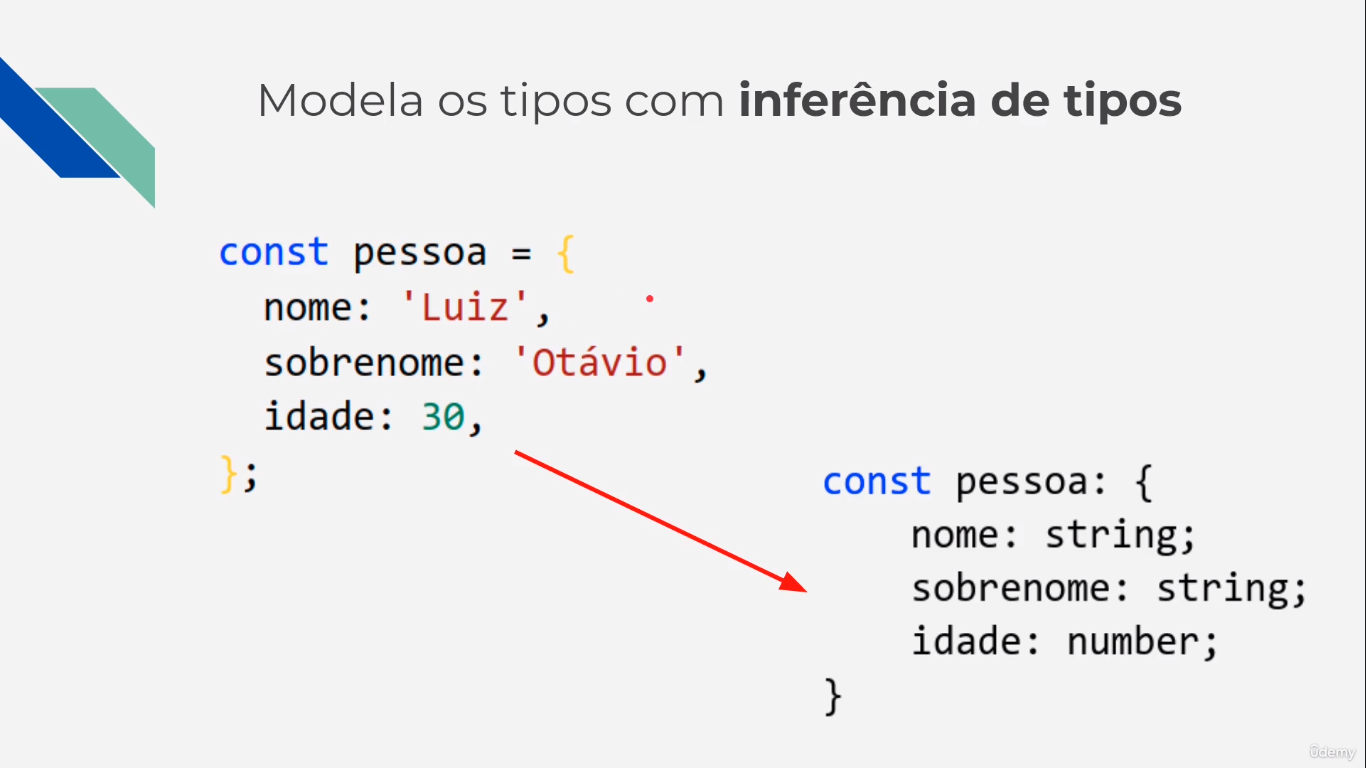
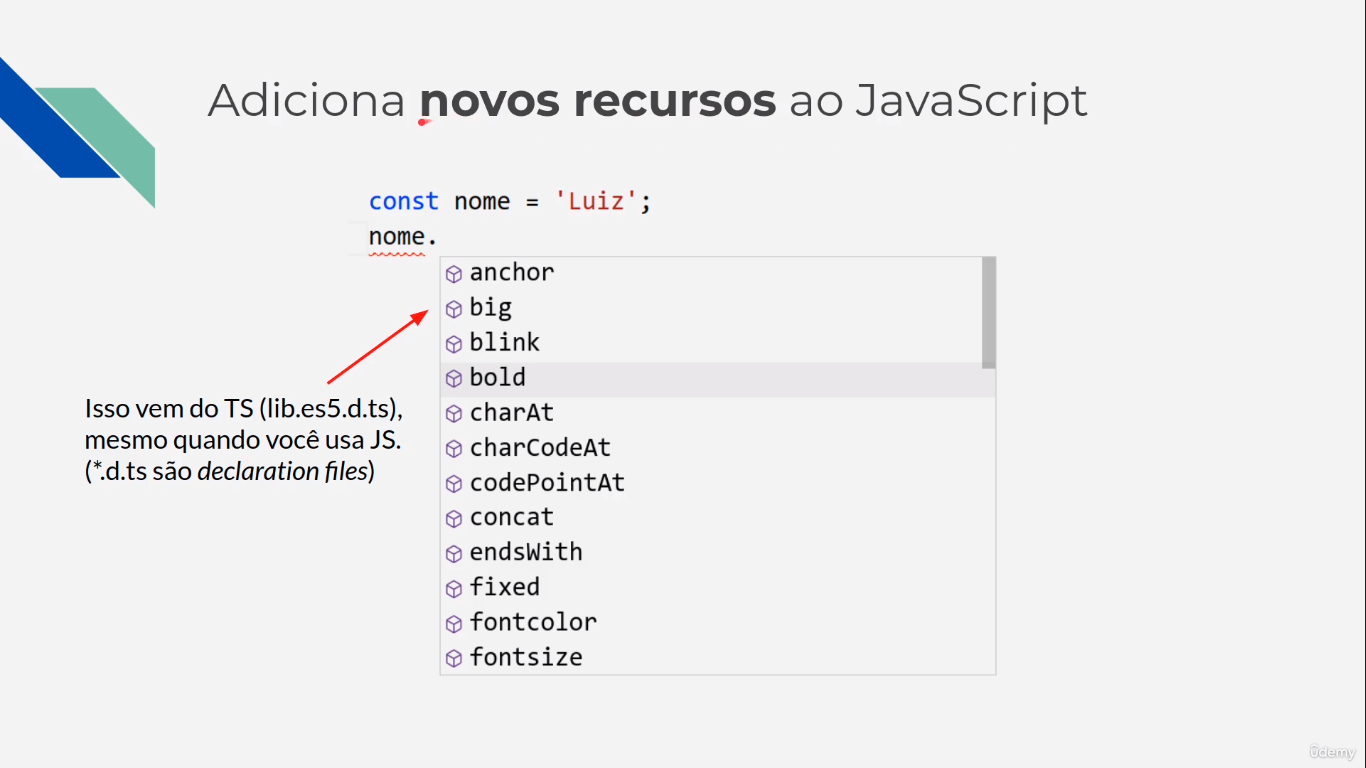
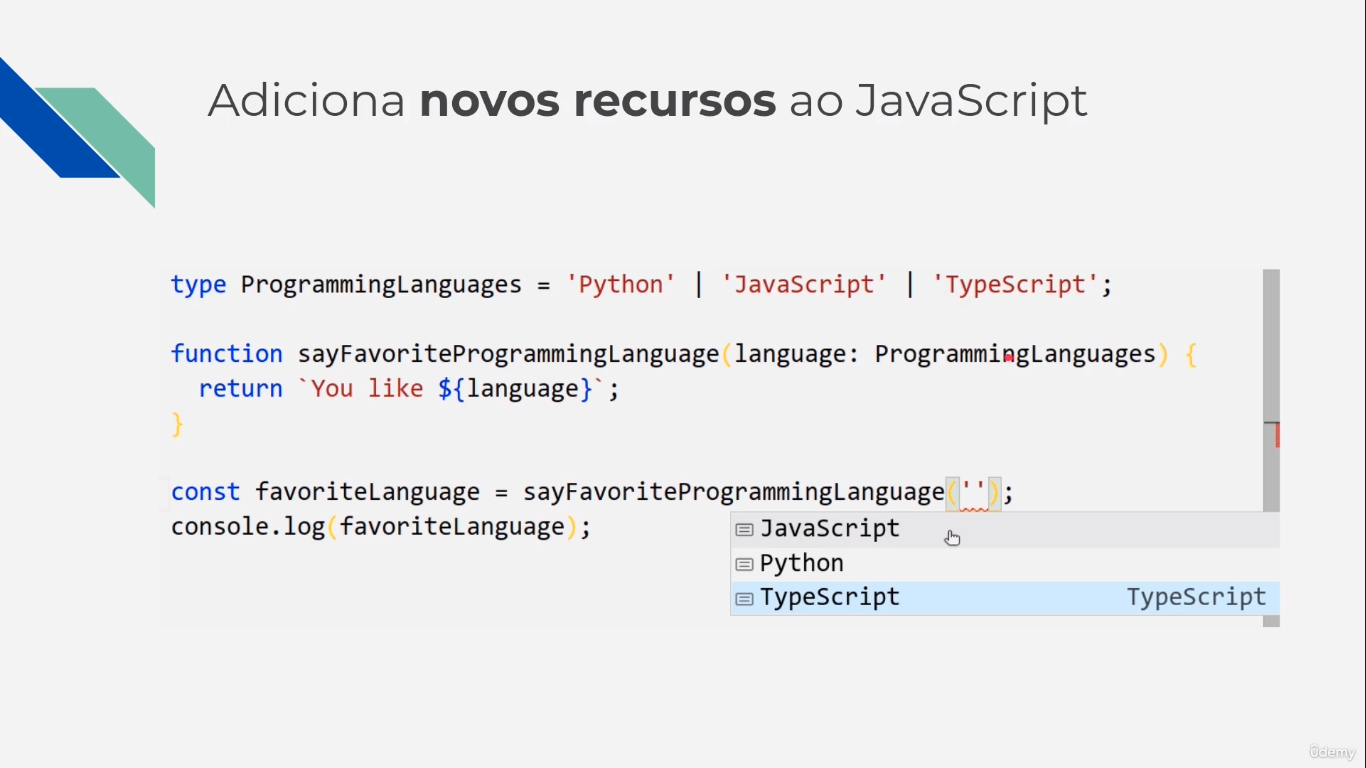
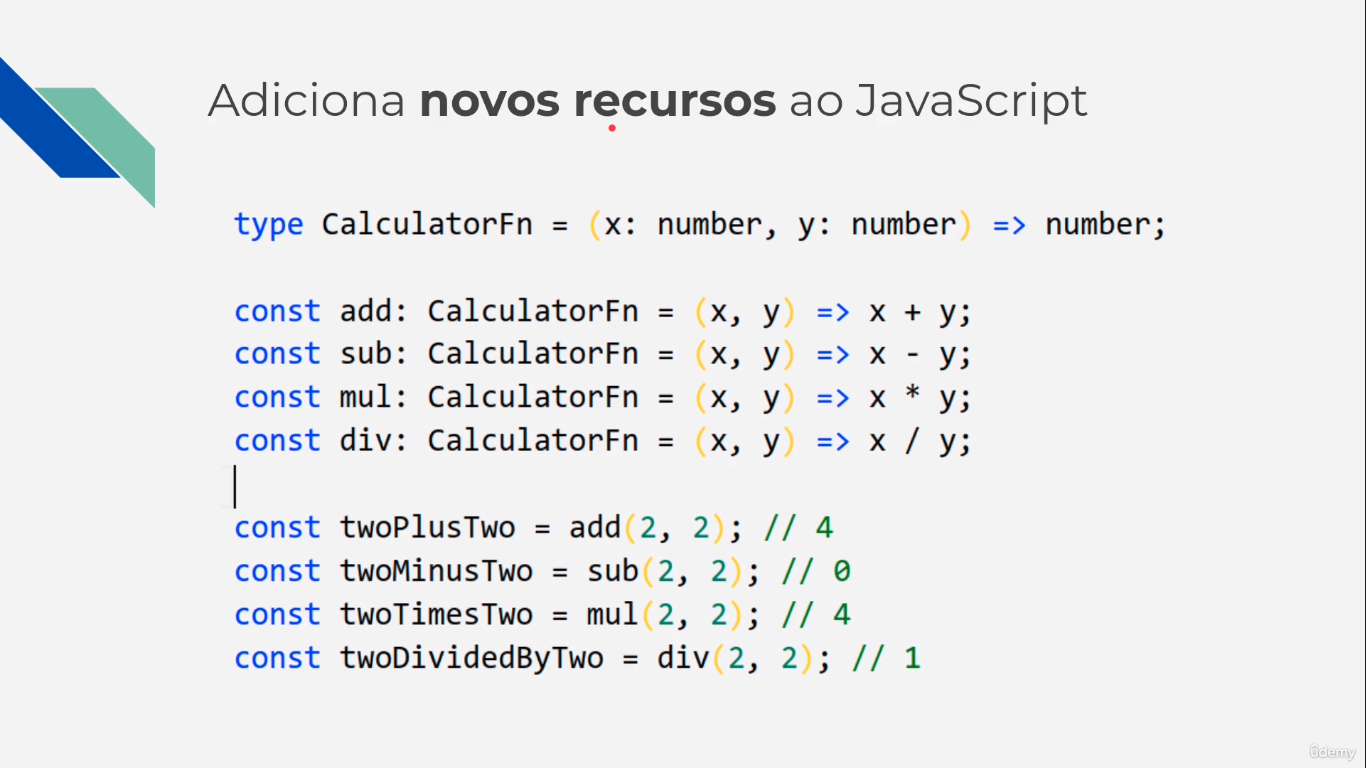
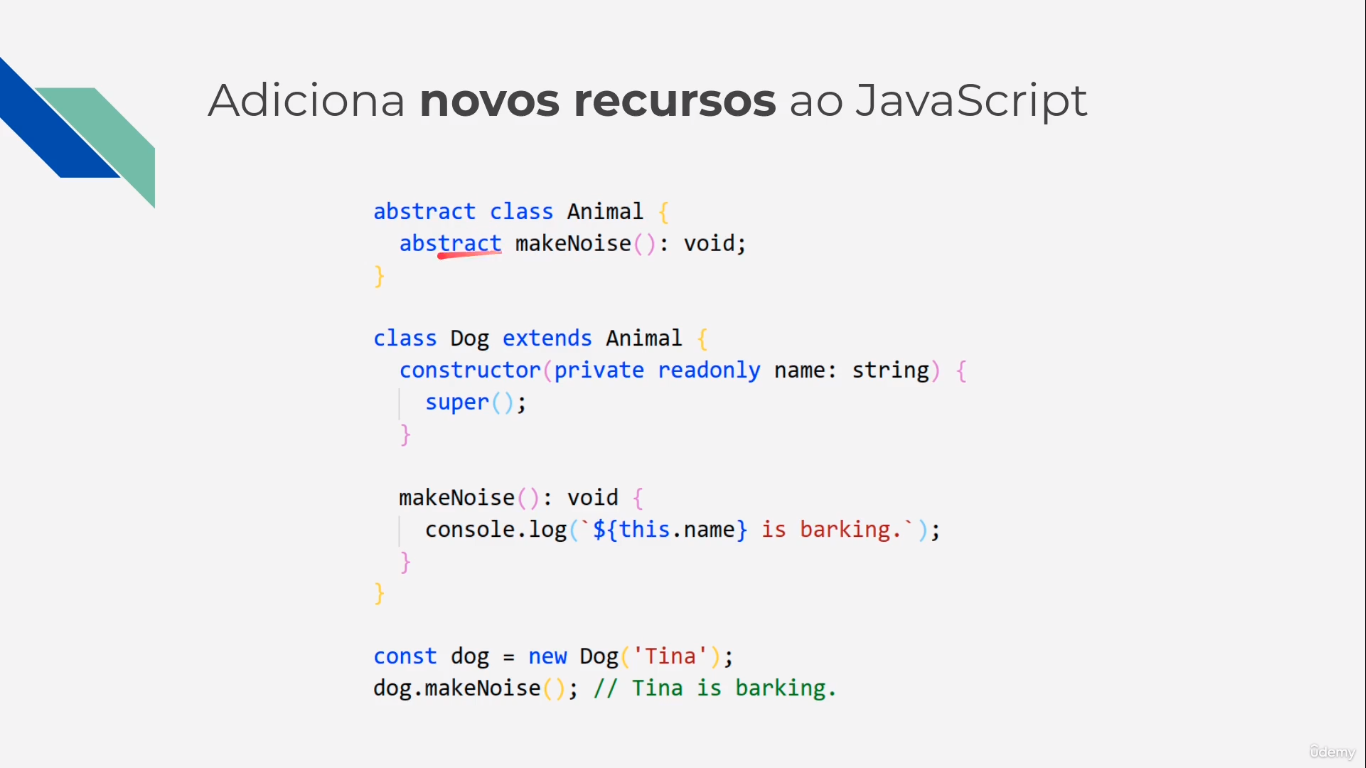
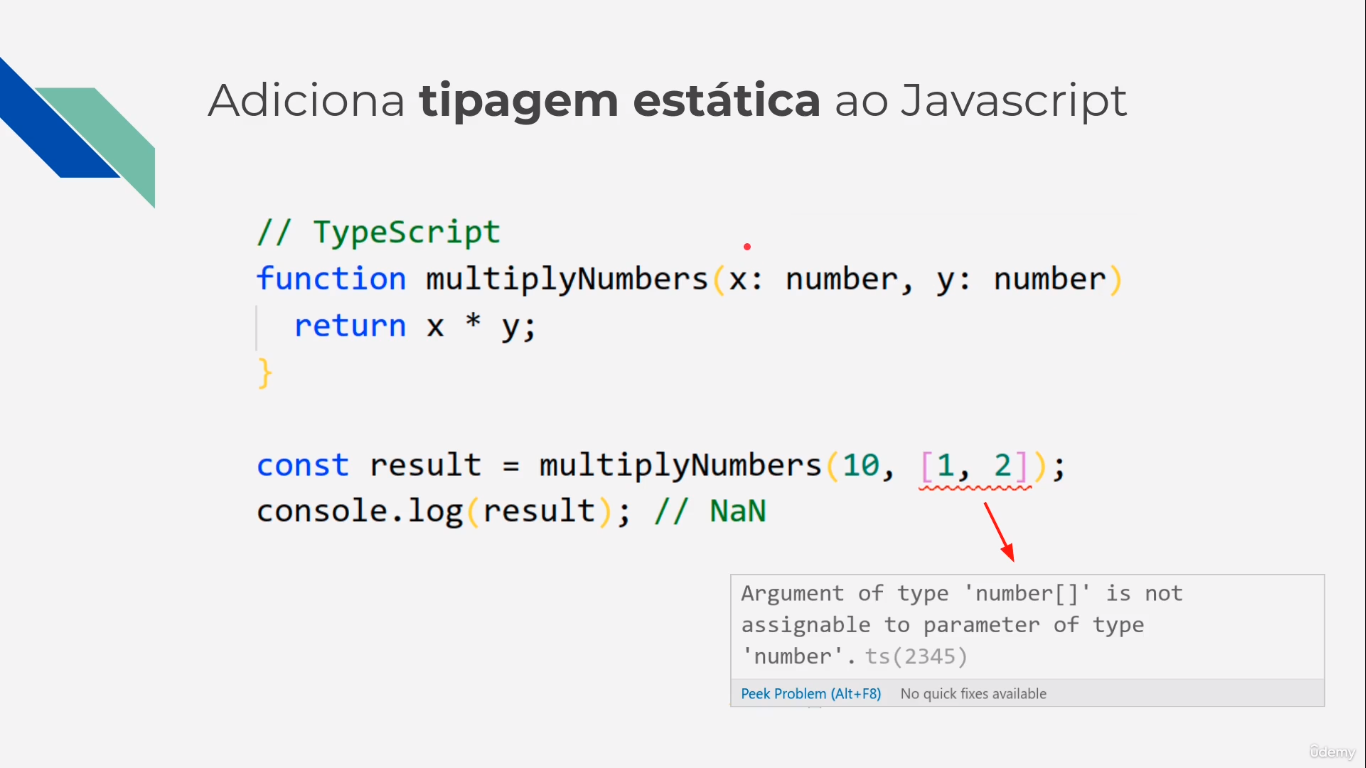
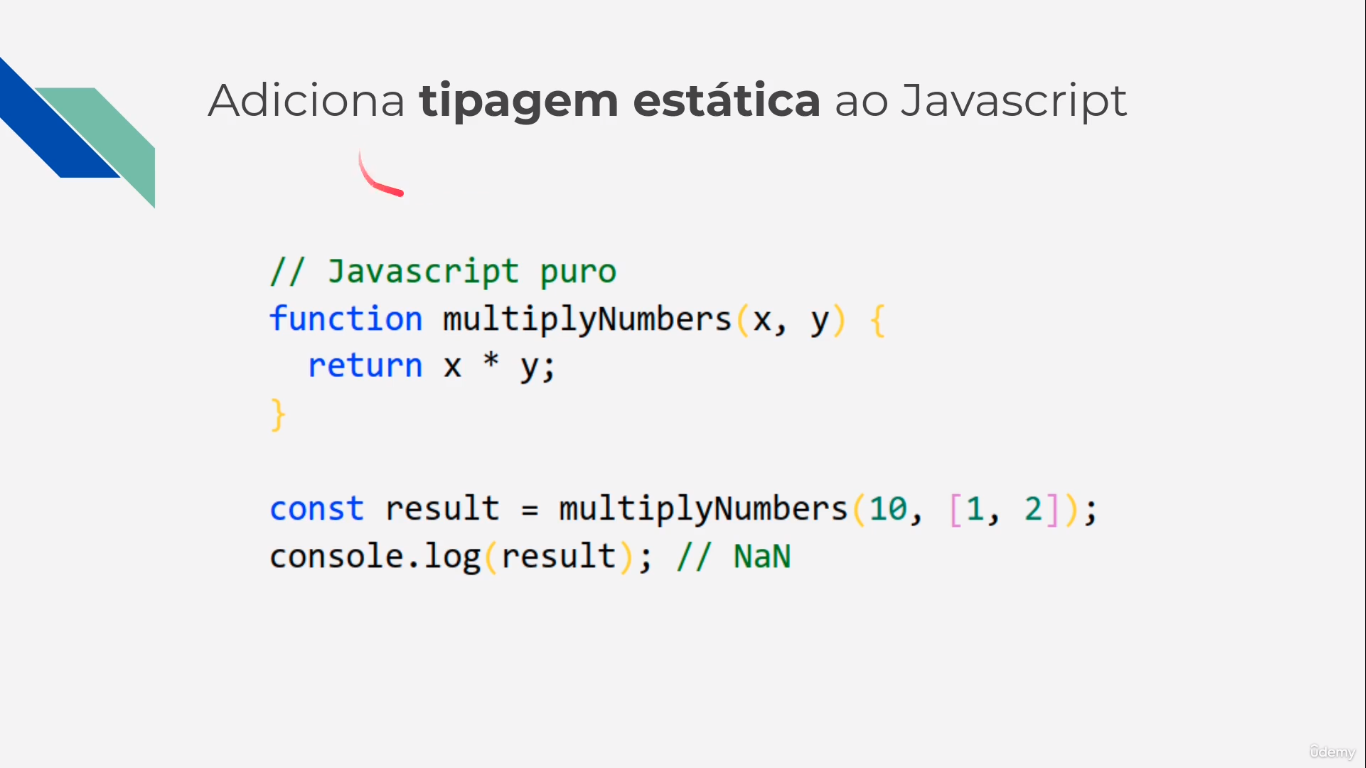
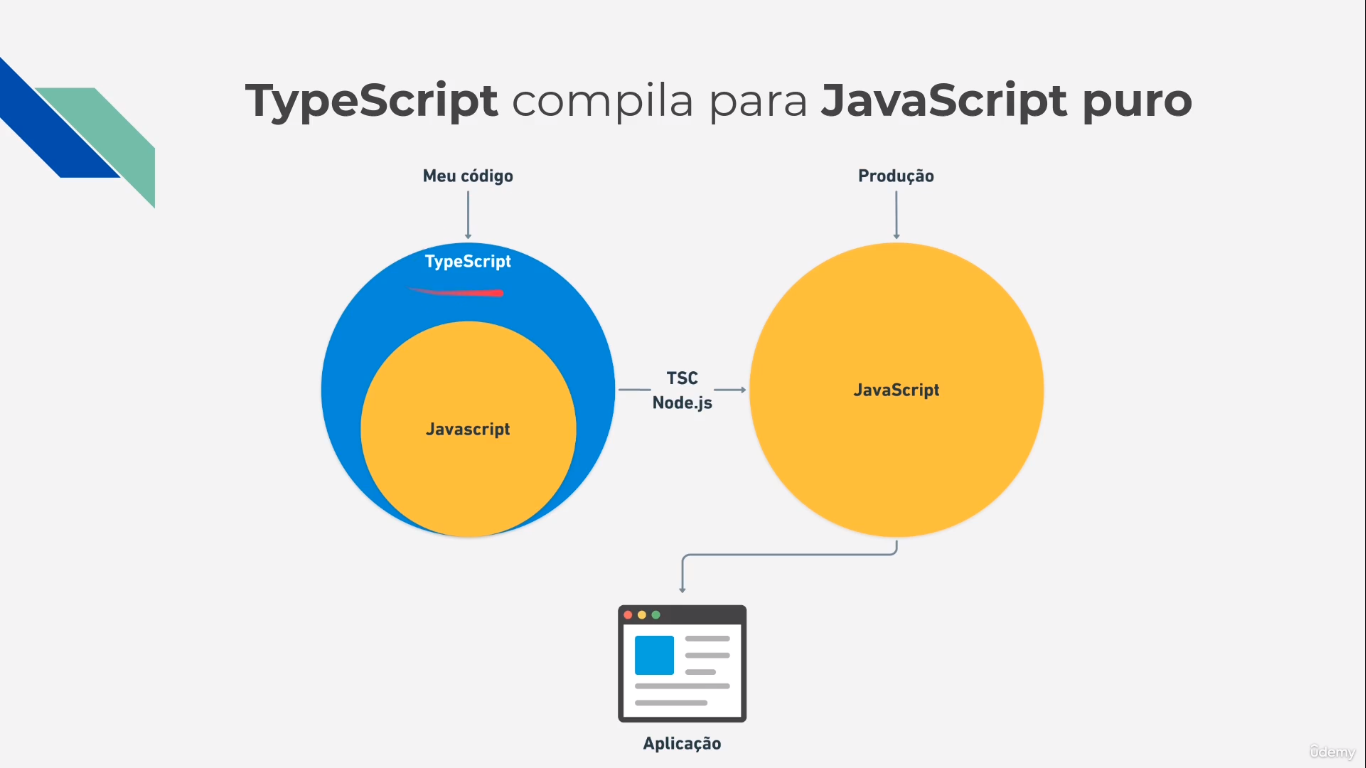
Secao 17 TypeScript - Instalação









Inicie seu projeto com:

npm init -y

* Instalando o Typescript

agora vamos instalar o typescript:

npm install typescript -D

seu typescript sera instalado em devDependencies

agora iremos criar um arquivo index.ts

function greet(name) {

    return `Olá ${name}`

}

console.log(greet('Rodrigo de Souza'));

por ser typescript o coderunner não irá ler, para resolver esse problemas iremos utilizar o:

npx tsc index.ts

que irá gerar um arquivo index.js, que ai sim poderar ser lido pelo coder runner

function greet(name) {

    return "Ol\u00E1 " + name;

}

console.log(greet('Rodrigo de Souza'));

index.ts = arquivo de desenvolvimento

index.js = arquivo de produção

O typescript tem dois modo de operação:

Model mode = modo de modulo

Script mode = mode operação de script

Script mode seria basicamente com um arquivo javascript enorme, como um arquivo único.

const nome = 'Rodrigo'

index.tj

const nome = 'Rodrigo'

index.js

no arquivo index.ts iria gerar um erro, por já ter uma variável já com esse nome, mesmo sendo em outro arquivo separado, ele consegue detectar que essa variável já foi definida no seu projeto.

Para sair desse modo so e preciso exporta alguma coisa ou importar alguma coisa no seu modulo, fazendo isso ele assume o model mode automaticamente.

const nome = 'Rodrigo'

export default nome;

* Instalando o ts-node e Code Runner

Para poder usar o coder runner diretamente nos códigos ts, você precisa instalar uma dependência de desenvolviemento chamada de ts-node:

npm i ts-node -D

sera instalado em devDependencies

em sequida cria uma pasta chamada .vscode, e crie um arquivo chamado settings.json

esse arquivo será as configuração do seu vscode desse projeto, não interferindo em outros projetos

insira em settings.json:

{

    "code-runner.executorMap": {

        "typescript": "npx ts-node --files --transpile-only"

    }

}

Agora quando voltarmos ao nosso script e mandar executar o coder runner ela já vai compilar e executar ao mesmo tempo

Agora temos um ambiente pronto para rodar os script pelo próprio terminal

* Instalando e configurando o ESint

Instale o eslint:

npm i eslint -D

npm i @typescript-eslint/eslint-plugin @typescript-eslint/parser -D

depois crie o arquivo .eslintrc.js na raiz do projeto

e cole o código:

module.exports = {

    env: {

      browser: true,

      es6: true,

      jest: true,

    },

    extends: [

      'react-app',

      'airbnb',

      'plugin:@typescript-eslint/recommended',

      'prettier/@typescript-eslint',

    ],

    globals: {

      Atomics: 'readonly',

      SharedArrayBuffer: 'readonly',

    },

    parserOptions: {

      ecmaFeatures: {

        jsx: true,

      },

      ecmaVersion: 2018,

      sourceType: 'module',

    },

    plugins: ['react', 'import', 'jsx-a11y'],

    rules: {

      'react/jsx-filename-extension': [

        'error',

        {

          extensions: ['.tsx'],

        },

      ],

      'import/prefer-default-export': 'off',

      '@typescript-eslint/explicit-function-return-type': 'off',

      '@typescript-eslint/explicit-member-accessibility': 'off'

    },

    settings: {

      'import/parsers': {

        '@typescript-eslint/parser': ['.ts', '.tsx'],

      },

      'import/resolver': {

        typescript: {},

      },

    },

  };

Aviso importantes:

Tenha instalado a extensão eslint no seu vscode

Opcional ter a extensão reload no seu vscode

* Tsconfig.json(inicial)

Digite em seu projeto:

npx tsc –init

no arquivo que foi criado o tsconfig.json, em lib mude para:

"lib": ["ESNext", "DOM"],