

# PHASE 2

## WEEK 3

# DAY 4



# React TS: Intro Day

# План

---

1. Create React App, CRA (клиентский React)
2. TypeScript
3. CRA + TypeScript

# Create React App, CRA

# Create React App: что это?

---

Инструмент для создания React-приложений на JavaScript.  
Устанавливает пакеты для удобной разработки.

```
// создать папку my-app с dev окружением  
npx create-react-app my-app
```

```
// наполнить текущую папку dev окружением  
npx create-react-app .
```

# ES modules

---

Ранее мы использовали CommonJS модули для разбивки приложения на файлы — с синтаксисом `require/exports`.

Для проектов на CRA будем использовать ES модули — с синтаксисом `import/export`.

```
// App.jsx  
export default App;
```

```
// index.jsx  
import App from './App';
```

# TypeScript

# TypeScript: что это?

---

TypeScript — это язык программирования, в котором исправлены многие недостатки JavaScript.

TS компилируется в JS и подходит для разработки любых проектов под любые браузеры — тем более что можно выбрать версию JS, в которую будет компилироваться код.



# TypeScript: плюсы

---

- TS это open source проект Microsoft с большим сообществом
- ошибки допущенные в процессе модификации кода, видны сразу, а не во время выполнения кода
- TS индустриальный стандарт типизации в веб-разработке
- TS это актуальное требование рынка труда
- ООП с ключевыми словами: `private`, `protected`, `public`, `abstract`, `extends`, `implements`

# TypeScript: минусы

---

- порог вхождения в TS выше, чем в JS
- несмотря на статическую типизацию, можно допустить прохождение любого типа данных, прописав слово `any`
- если в работе вы используете пакет который не портирован на TS, то вам придётся описывать типы всех функций и методов самостоятельно

# TypeScript: простейшая типизация

---

// простейшая типизация

let a = 10; // тип number для переменной 'a'

a = '20'; // TS ошибка, т.к. происходит смена типа, что запрещено на уровне ЯП

const b = a \* 2; // тип number, т.к. переменная 'a' число

# TypeScript: типизация функций

---

// типизация параметров функций и возвращаемого значения

```
function sum(a: number, b: number): number {  
  return a + b;  
}
```

```
const s = sum(1, 2); // 3
```

// ещё вариант типизации функции

```
function makePair(a: number, b: string): [number, string] {  
  return [a, b];  
}
```

```
sum(2022, 'year'); // [2022, 'year']
```

# CRA + TS

# CRA + TS: установка

---

Инструмент для создания React-приложений на TypeScript.

Устанавливает пакеты для удобной разработки.

```
// создать папку my-app с dev окружением  
npx create-react-app my-app --template typescript
```

```
// наполнить текущую папку dev окружением  
npx create-react-app . --template typescript
```

# ES modules: JSX > TSX

---

Ранее мы использовали CommonJS модули для разбивки приложения на файлы — с синтаксисом `require/exports`.

Для проектов на CRA с TypeScript будем использовать ES модули — с синтаксисом `import/export`.

JSX формат превращается в формат TSX.

```
// App.tsx
```

```
export default App;
```

```
// index.tsx
```

```
import App from './App';
```