TypeScript للمبتدئين - شرح مبسط

What is TypeScript?

TypeScript = JavaScript + types

It helps you write safer, more maintainable code by catching errors before running.

Why use TypeScript?

- Prevents runtime errors
- Makes code easier to maintain
- Provides intelligent code suggestions and autocomplete

Basic Data Types

```
let name: string = "Nour";
let age: number = 25;
let isLoggedIn: boolean = true;
```

Arrays

```
let numbers: number[] = [1, 2, 3];
let names: string[] = ["Ali", "Sara"];
```

Objects

```
let user: { name: string; age: number } = {
  name: "Nour",
  age: 25
};
```

Interfaces

```
interface User {
  name: string;
```

```
  age: number;
  isAdmin?: boolean;
}
```

Functions

```
function greet(name: string): string {
  return `Hello, ${name}`;
}
```

Union Types

```
let id: number | string;
id = 5;
id = "abc";
```

Type Aliases

```
type Status = "loading" | "success" | "error";
let currentStatus: Status = "loading";
```

TypeScript with React

```
type Props = {
  name: string;
  age: number;
};

function Profile({ name, age }: Props) {
  return <div>{name} - {age}</div>;
}
```

Using useState

```
const [count, setCount] = useState<number>(0);
```

TypeScript with Events

- onClick: React.MouseEvent<HTMLButtonElement>
- onChange: React.ChangeEvent<HTMLInputElement>
- onSubmit: React.FormEvent<HTMLFormElement>
- onKeyDown: React.KeyboardEvent<HTMLInputElement>

Example: onChange Event

```
const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  console.log(e.target.value);
};
```

Important Notes

- `e` is the event object
- Each HTML element has a specific type
- TypeScript prevents common event handling mistakes