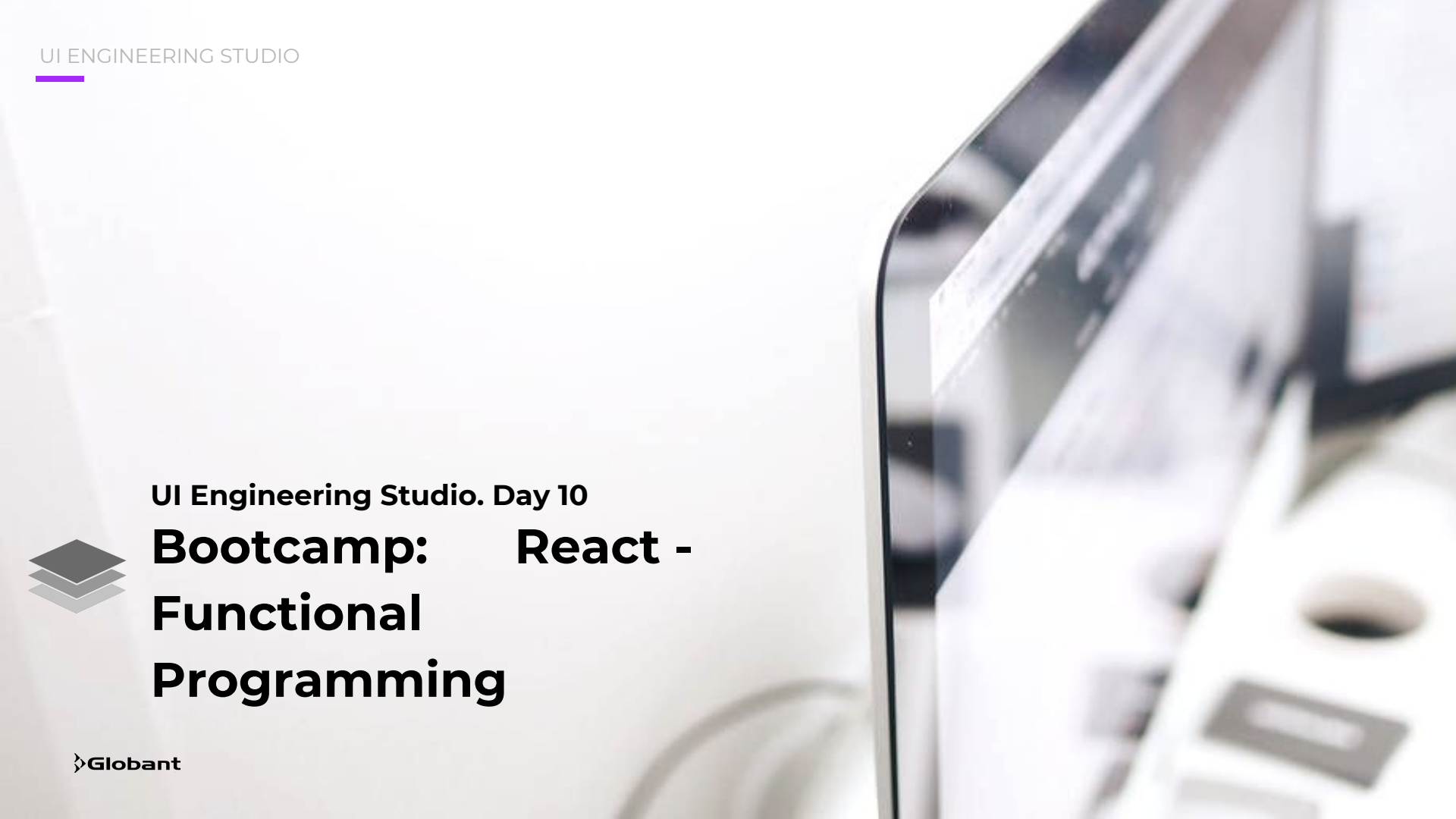

Globant
we are ready



UI Engineering Studio. Day 10

Bootcamp: React - Functional Programming

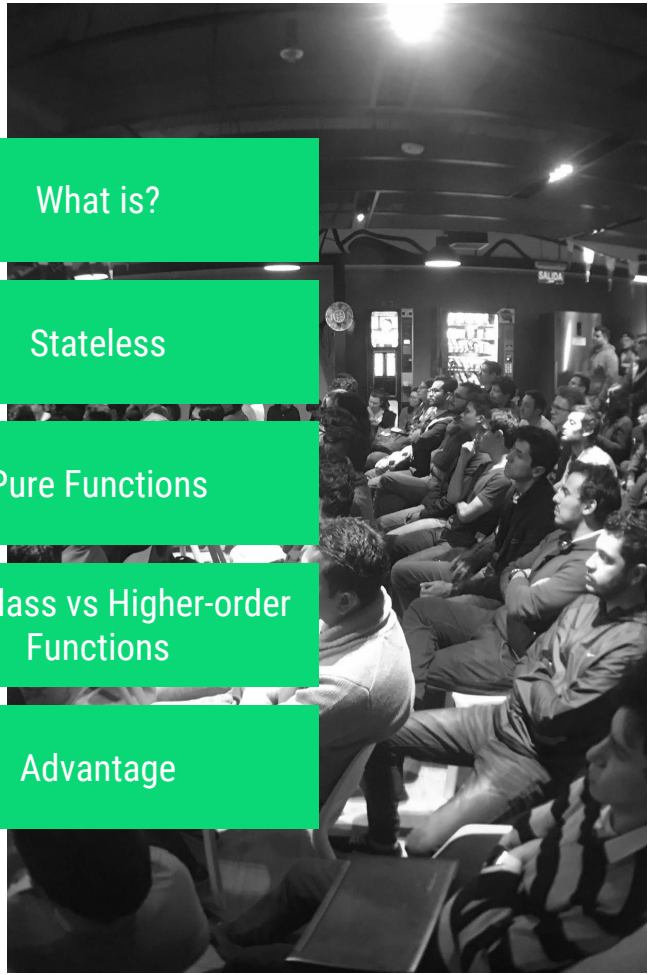
Globant

What is functional programming?

In computer science, functional programming is a programming paradigm—a style of building the structure and elements of computer **programs**—that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

-Wikipedia

- What is?
- Stateless
- Pure Functions
- First-class vs Higher-order Functions
- Advantage



UI Boot Camp: React

Stateless

Functional Programming is a programming paradigm in which the global state does not exist (*or is not considered*), it is not part of the basic logic of programming. In other words, it is stateless or stateless programming, in which the global state of an application is not accessed or attempted, but rather everything is immutable.

Source

UI Boot Camp: React

Pure Functions

A **pure function** is a function which:

- Given the same input, will always return the same output.
- Produces no side effects.

"A dead giveaway that a function is impure is if it makes sense to call it without using its return value. For pure functions, that's a noop."

- source

Return same output

```
function add(a, b) {
 let result = a + b
 return result;
}
add(1,2)//return 3
add(3,4)//return 7
//Always returns the add of two numbers
```

No side effects

```
let arr = [1, 2, 3, 4, 5, 6];
let even = arr.filter(function (element) {
    return element%2 === 0;
});
console.log(even);
```

UI Boot Camp: React

First-class vs Higher-order Functions

A programming language is said to have ***First-class*** functions when functions in that language are treated like any other variable.

A function is a **higher order** function if it takes a function as a parameter, or returns a function as its result. Both of these requirements rely on functions being first-class objects in a language.

source

First-class

```
const add = function (a, b) {
 let result = a + b
 return result;
}
```

Higher-order

```
const arr = [1, 2, 3, 4, 5, 6];
const even = arr.filter(function (element) {
    return element%2 === 0;
});
console.log(even);
```

UI Boot Camp: React

Advantage

- They are easier to test, since they will return the same result from the same parameters.
- They can be cached, since they always return the same result we can store these results in cache (memorize).
- Easy to understand.
- They can be executed in parallel, since they do not have any external dependence.
- It reduces the amount of bugs, since they do not generate alterations of the external state.
- If the result of a pure expression is not used, it can be eliminated without affecting other expressions.

UI Boot Camp: React

Homework

Make the next challenge. At this level it has to be easy...

Challenge


Globant
we are ready