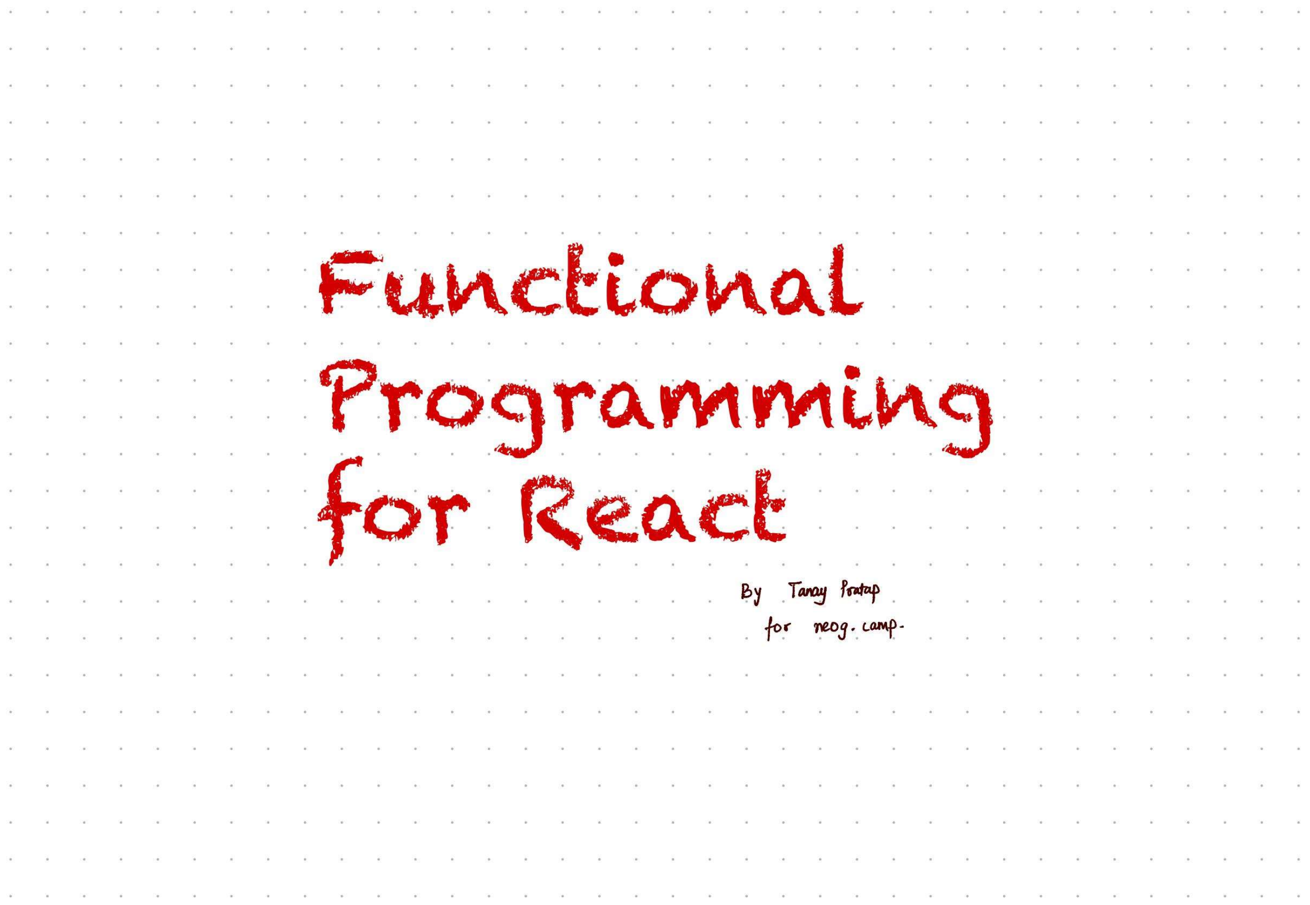
Functional Programming Exercises

Functional Programming



Functional Programming for React

By Tanay Pratap for neog.camp.

- ★ can be assigned to variables

- ★ sent to other functions as arguments

FUNCTIONS AS FIRST CLASS CITIZENS

- ★ can be added to objects.

- ★ can be returned from other functions

- ★ can be added to arrays as well.

- FP is part of larger programming paradigm.

DECLARATIVE PROGRAMMING

✓ What should happen

✗ How it should happen.

- We have seen how React abstracts away DOM creation.

Why FP?

- Less Bugs (no global changes)
- More code reuse
- Easier to understand & read.
- Super easy to test

? How to do Functional Programming

☐ Do not change any data.
Always return a new copy.

★ In Javascript, function arguments are reference to the actual data.

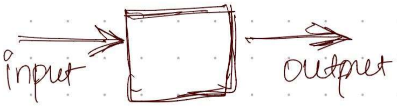
IMMUTABILITY

📌 Take an object with your mother's name and your age.

Now, create an obj for your sibling by age difference.

📌 Take an array with 5 colors.
Create another array by adding two more colors to it.

PURE FUNCTIONS



for same input the output will always be same.

3 rules

- at least one argument
- return a value or other function
- should not mutate any of its arguments.

Q. Write a function birthday() to take a person's name and age in an object and increase age.

HIGHER ORDER FUNCTIONS

Q. write a function which can tell whether a number is less than 10 or not. Supply this function to Array.filter() to get an array with no 10s in it.

Q. Given an array of numbers return an object for each item.

Q. Given an array of numbers calculate the sum.

* Do both questions using two different functions. and then supplying it to map and reduce.

- □ Can take functions as arguments.
- □ Return functions.
- □ Or both.

To return a new array with modifications on items.

[ ].map(fn)

fn (item, index)

where item ⇒ One item of array

index ⇒ index of current item

returns ⇒ new item for the new array

To return one final value after iterating on every item in an array.

[ ]. reduce(fn)

fn (accumulater, currentValue)

accumulator ⇒ coming from previous run

currentValue ⇒ value of array one at a time.

returns ⇒ updated value of accumulator.

H/W

Q. Can you write your own reduce using for loop?

Homework Questions

Q. Given an array of integers

- (a) Find the sum of all odd numbers.
- (b) Find the sum of all numbers at odd indices.
- (c) Find the biggest number in the array.
- (d) Find the numbers divisible by 10.
- (e) Return an array of numbers where odd numbers are incremented by one and even numbers are decremented by one.
- (f) return an object with sum of all odd numbers and even numbers separately.

Q. Given an array of strings

- (a) Find the number of strings with similar number of characters.

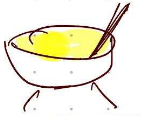
```
const input = ["apple", "orange",
               "mango", "papaya"]
// output : { 5:3 , 6:1 }
```

- (b) return an array with strings which have vowels.
- (c) Return an array of objects with key as item and value as number of characters in the string.

#Rules

1. Use pure functions.
2. Create functions separately for reuse.

CURRYING



```
const add = (num1,
             num2)
=>   num1 + num2
```

```
add(2,4)

add(2)(4)

const addTwo = add(2)
      addTwo(4); //6
```

```
const add = (num1)
   => (num2) =>
   num1 + num2
```

Q. Create a function which takes your name and returns a function which would add your name to anything that function says.

```
const giveYourname()...
const tanaySays
        = giveyourname
            ("Tanay")

tanaySays("hello")
   // "Tanay says, "Hello" "
```

Q. Write a function which can log any text with your username.
Another function which can write any text with your userID.
Now, compose both functions to give one function which can log any text with both username + userID.

Composition

Two functions $f(x)$, $g(x)$

Output of $f(x)$ should go to $g(x)$.

$$\boxed{g\,(f(x))}$$

Q. Write a function compose() which can take any number of functions and return a function which will run the given functions in order when called with an argument.

Hint This exercise would need

ⓐ Currying

ⓑ Reduce

ⓒ rest operator cuz you don't know how many values can be passed to compose.

THE ONE HOMEWORK

ex

```
const increment = num => num+1;

const square = num => num*num;

const incrementThenSquare = compose(
                              increment,
                              square);
incrementThenSquare(2); //9
```

Write this function compose as a util so that you can use it for all your functional programming needs.

THANK YOU

Property of neog. camp.
Don't share outside.

Previous

‹ Chrome Dev Tools + How Web Works Exercises

Next

How Web Works Post-Reads ›