Learning
Functional Programming

What is functional programming?

What is functional programming?

A programming paradigm

What is functional programming?

A Coding style or mindset

Why another programming paradigm?

Languages come and go. When programming paradigms change, something serious is out of balance.

What makes a software project hard?

What makes a software project hard?

Inherent Complexity

What makes a software project hard?

The difficulty of the actual problem that the software is trying to make simpler and better

What makes a software project hard?

Incidental Complexity

What makes a software project hard?

Anything about software that is hard but doesn't really have to be.

“I had a problem
So, i wrote a code to solve it
Now, I have
10 compiler errors
3 runtime errors
So, 13 new problems”

We want our software to be safe and easy to maintain and debug

Software is not meeting modern demands

- The CPU is not getting faster
- We need to take advantage of concurrency

Where is it Used?

- Facebook, Whatsapp (Erlang)
- Facebook (Haskell)
- DiscordApp, Pinterest, Zuppler, Spreedly (Elixir)
- Linkedin, Twitter , SoundCloud (Scala)
- Thoughtworks, Akamai (Clojure)
- And many more ....

How to get started?

#1 Treat Functions as your King

- Functions are like Variables
- We can pass functions as arguments to other functions, return functions as well store functions in variables

Non-Functional:

```
json_response = {
          student_id: 1,
          name: “Chatur”
     }

Name = Map.get(json_response, “name”)

console.log(“Hi, my name is ” + Name)
```

Functional:

```
function extract_name(json_response) {
     Map.get(json_response, “name”)
     console.log(“Hi, my name is” + name)
}

extract_name(json_response)
```

Functional:

```
function extract_name(json_response) {
     Map.get(json_response, “name”)
}

function parse(name){
     console.log(“Hi, my name is” + name)
}

Name = extract_name(json_response)

parse(Name)
```

#2 Stop Iteration

Use map, reduce, filter instead of loops

Imperative:

```
test_array = [1, 2, 3, 4, 5, 6 ,7 ,8 ,9]
function getOdds(arr){
  let odds = [ ];
  for(i = 0; i < test_array.length + 1; i++){
         if ( i % 2 !== 0 ){
            odds.push( i )
         };
       };
   return odds
 };

console.log(getOdds(test_array))


=> [1, 3, 5, 7, 9]
```

Declarative:

```
test_array = [1, 2, 3, 4, 5, 6 ,7 ,8 ,9]


function getOdds(arr){
   return arr.filter(function (num){ num % 2 !== 0 })
}

console.log(getOdds(test_array))


=> [1, 3, 5, 7, 9]
```

Functional:

```
test_array = [1, 2, 3, 4, 5, 6 ,7 ,8 ,9]

getOdds = arr => arr.filter(num => num % 2 !== 0)

console.log(getOdds(test_array))


=> [1, 3, 5, 7, 9]
```

#3 Avoid Side Effects

Use Pure functions to avoid side effects.

Impure Function:

```
test_array = [1, 2, 3, 4, 5, 6 ,7 ,8 ,9]

val = 2
function getOdds(arr){
   return arr.filter(num => num % val !== 0)
}

console.log(getOdds(test_array))


=> [1, 3, 5, 7, 9]
```

Pure Function:

```
test_array = [1, 2, 3, 4, 5, 6 ,7 ,8 ,9]


function getOdds(arr){
   return arr.filter(num => num % 2!== 0)
}

console.log(getOdds(test_array))


=> [1, 3, 5, 7, 9]
```

#4 Avoid Mutability

```
list = [1, 2, 3, 4]

list.pop
# => 4

list.push(1)
# => [1, 2, 3, 1]
```

Avoid Mutability

- Big problem for concurrency
- Race Condition

How to prevent problems?

Using Thread and lock mechanisms

How to prevent problems?

Using Immutable Data

Immutability

```
list = [1, 2, 3, 4]

list.pop()
# => 4

list.push(1)
# => [1,2,3,4,1]

list
# => [1, 2, 3, 4]
```

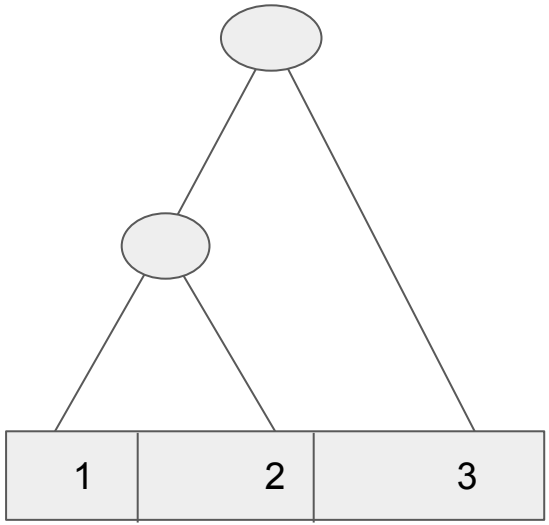
Does Immutability Slow down system?

```
list = [1, 2, 3, 4]

list.pop()
# Memory => [1,2,3,4] , [1,2,3] ??

list.push(1)
# Memory => [1,2,3,4] , [1,2,3,4,1] ??

list
# => [1, 2, 3, 4]
```

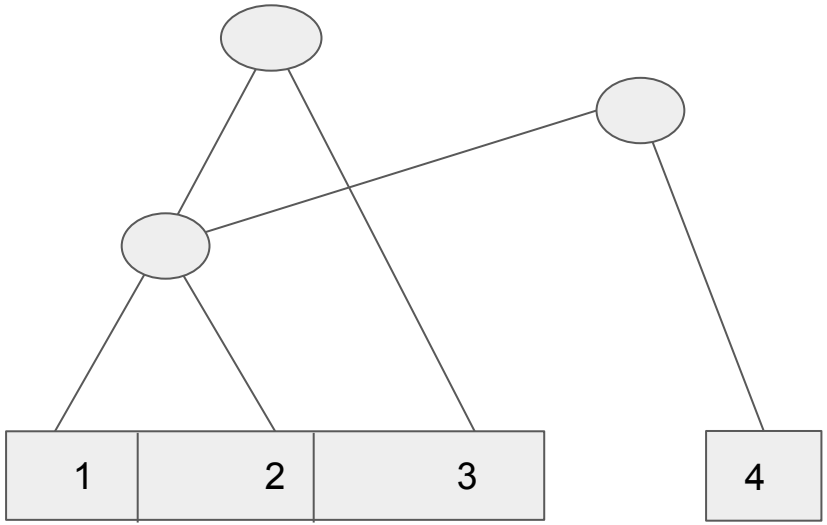
Persistent Data Structures

1. Preserves the previous version of itself even when it is modified
2. New version meets the Complexity Guarantee
3. Old version also meets the Complexity Guarantee

Persistent Data Structures



Persistent Data Structures



Thanks!

Himanshu Gupta

@himanshu0128

decentname (Himanshu Gupta) · GitHub

himanshu0128gupta@gmail.com

Delhi |> Elixir