

Learning Functional Programming with JS

@AnjanaVakil

The Recurse Center



<https://www.recurse.com/>

What is functional programming?

What is functional programming?

a programming paradigm

What is functional programming?

a coding style

What is functional programming?

a mindset

What is functional programming?

a sexy, buzz-wordy trend

Why functional JavaScript?

Why functional JavaScript?

object-oriented JS gets tricky
(prototypes? this?!?)

Why functional JavaScript?

safer, easier to debug/maintain

Why functional JavaScript?

established community

OK, let's do it!

OK, let's do it!

...how?

Do everything with functions

input -> output

Not functional:

```
var name = "Anjana";
```

```
var greeting = "Hi, I'm ";
```

```
console.log(greeting + name);
```

```
=> "Hi, I'm Anjana"
```

Functional:

```
function greet(name) {  
    return "Hi, I'm " + name;  
}
```

```
greet("Anjana");
```

```
=> "Hi, I'm Anjana"
```


Avoid side effects

use “pure” functions

Not pure:

```
var name = "Anjana";  
  
function greet() {  
    console.log("Hi, I'm " + name);  
}
```

Pure:

```
function greet(name) {  
    return "Hi, I'm " + name;  
}
```

Use higher-order functions

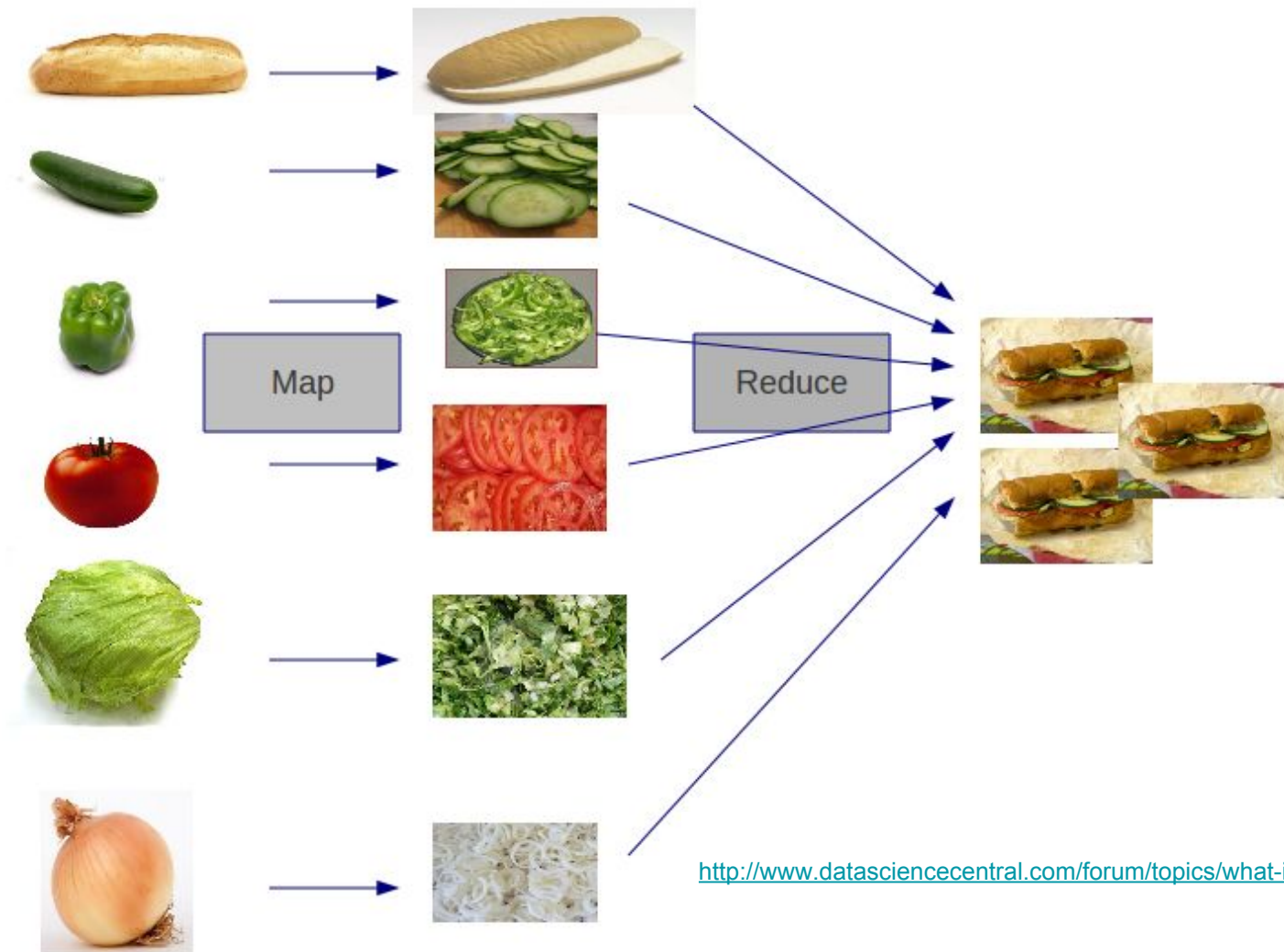
functions can be inputs/outputs

```
function makeAdjectifier(adjective) {  
    return function (string) {  
        return adjective + " " + string;  
    };  
}
```

```
var coolifier = makeAdjectifier("cool");  
coolifier("conference");  
=> "cool conference"
```

Don't iterate

use map, reduce, filter



<http://www.datasciencecentral.com/forum/topics/what-is-map-reduce>

Avoid mutability

use immutable data

Mutation (bad!):

```
var rooms = ["H1", "H2", "H3"];
```

```
rooms[2] = "H4";
```

```
rooms;
```

```
=> ["H1", "H2", "H4"]
```

No mutation (good!):

```
var rooms = ["H1", "H2", "H3"];
```

```
Var newRooms = rooms.map(function (rm) {  
    if (rm == "H3") { return "H4"; }  
    else { return rm; }  
});
```

```
newRooms; => ["H1", "H2", "H4"]
```

```
rooms; => ["H1", "H2", "H3"]
```

Persistent data structures for efficient immutability

Mori, Immutable.js

Ready to try it out?

FP libraries for JS

- Mori (<http://swannodette.github.io/mori/>)
- Immutable.js (<https://facebook.github.io/immutable-js/>)
- Underscore (<http://underscorejs.org/>)
- Lodash (<https://lodash.com/>)
- Ramda (<http://ramdajs.com/>)
- ...and more!

Want to learn more?

“An introduction to functional programming”

by Mary Rose Cook

<https://codewords.recurse.com/issues/one/an-introduction-to-functional-programming>

Thanks for listening!

I'm @AnjanaVakil

Huge thanks to:

JSUnconf organizers

Diversity sponsors

The Recurse Center & alums