

Combining Data Streams

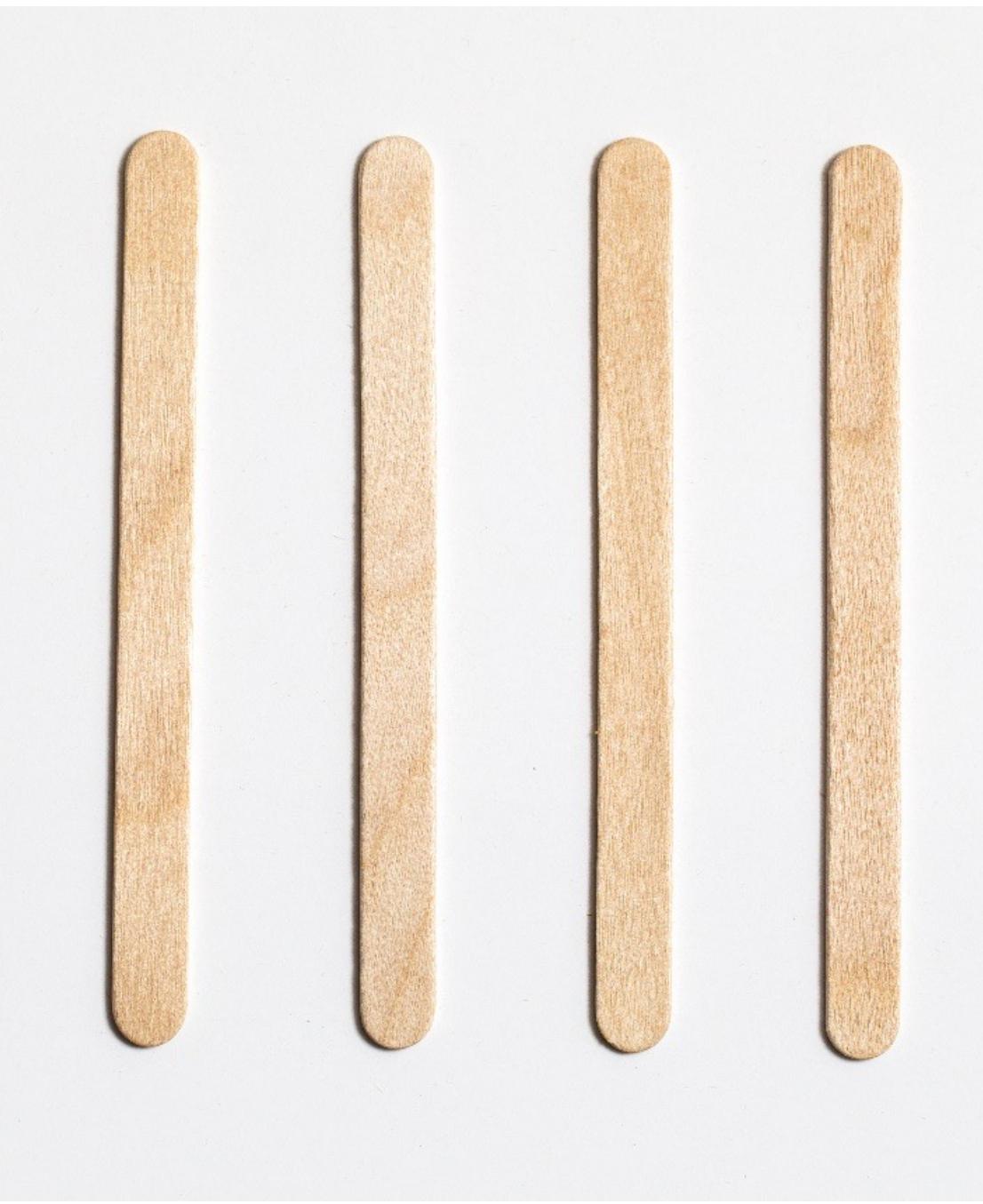


Deborah Kurata

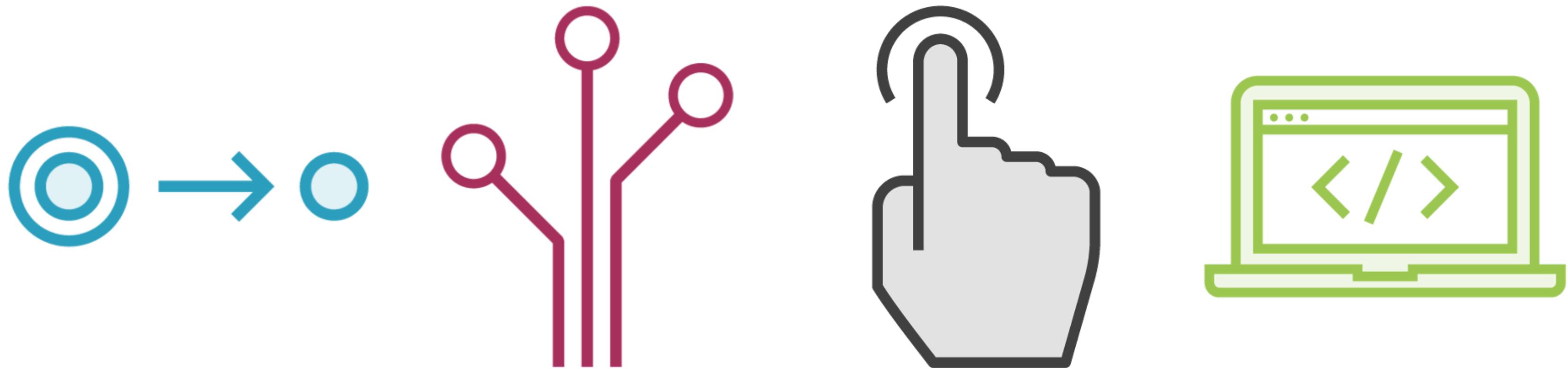
Consultant | Speaker | Author | MVP | GDE

@deborahkurata





Why Combine Data Streams?



**Map id to a
string**

**Work with
multiple data
sources**

React to actions

**Simplify
template code**



Module Overview

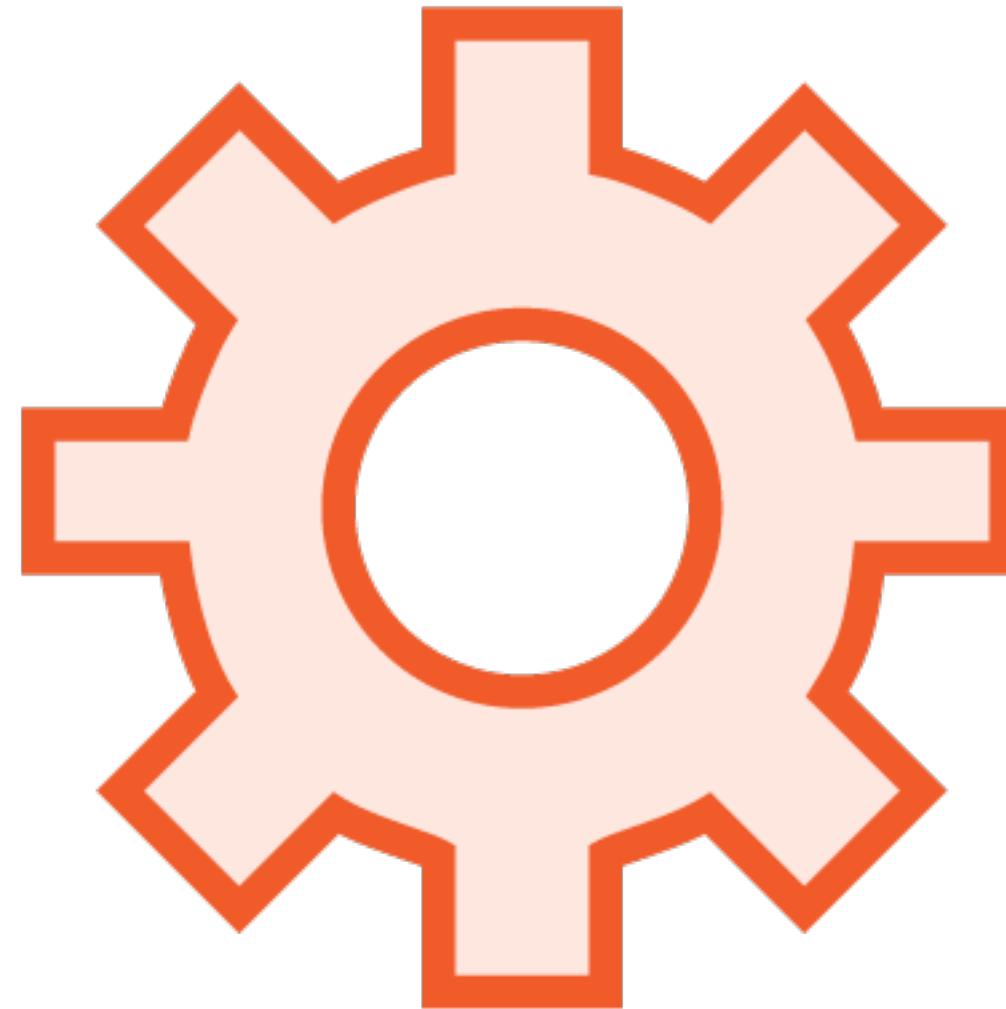


Combination operators and creation functions

Combining data streams to map an id to a string



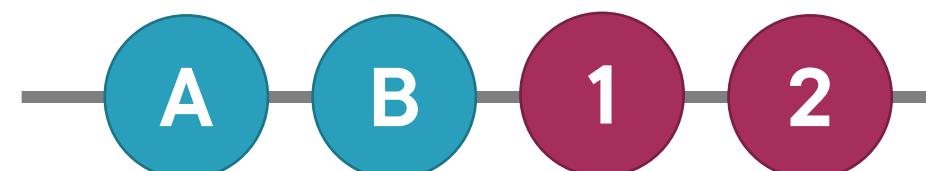
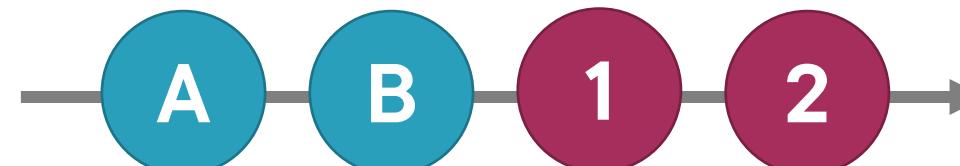
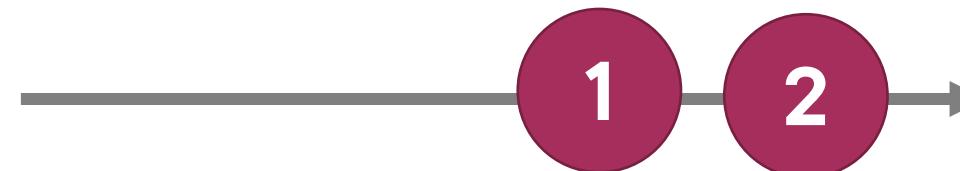
RxJS Features



combineLatest
forkJoin
withLatestFrom



Types of Combination Operators/Functions



Combine to a single Observable result
merge, concat

Flatten higher-order Observables
mergeAll

Emit a combined value
combineLatest,
forkJoin,
withLatestFrom



RxJS Creation Function: combineLatest



Creates an Observable whose values are defined:

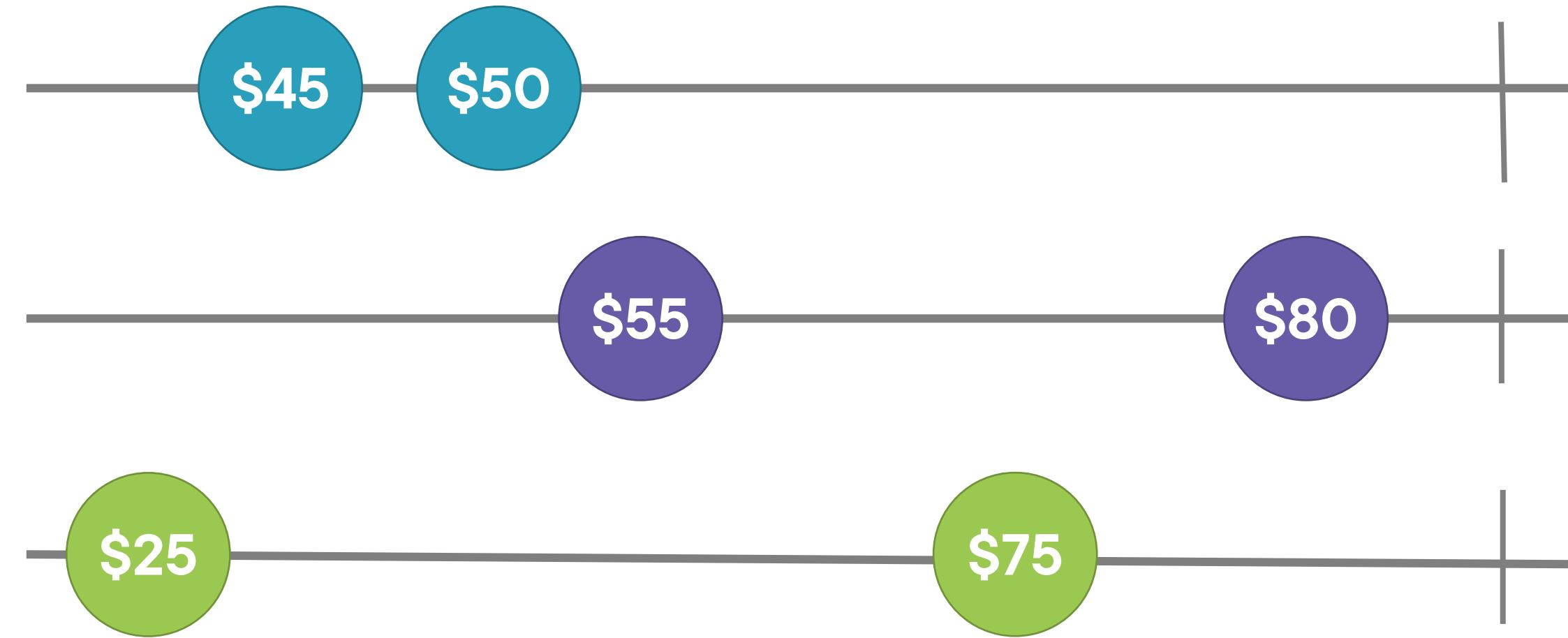
- Using the **latest** values from each input Observable

```
combineLatest([a$, b$, c$])
```

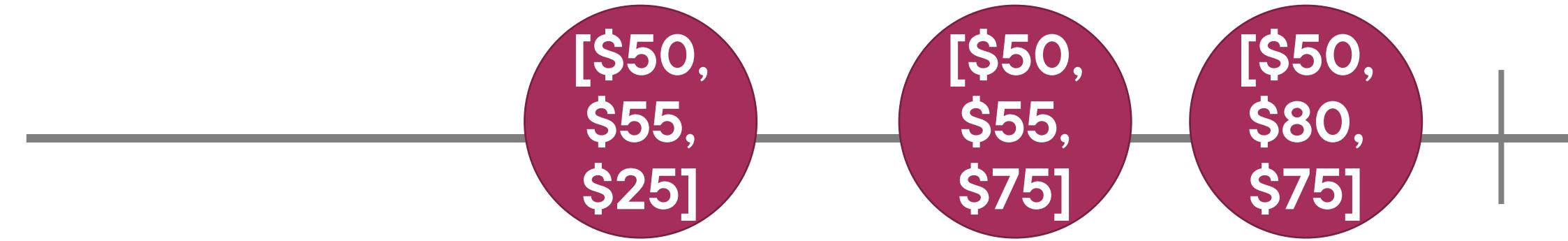
Static creation function, not a pipeable operator



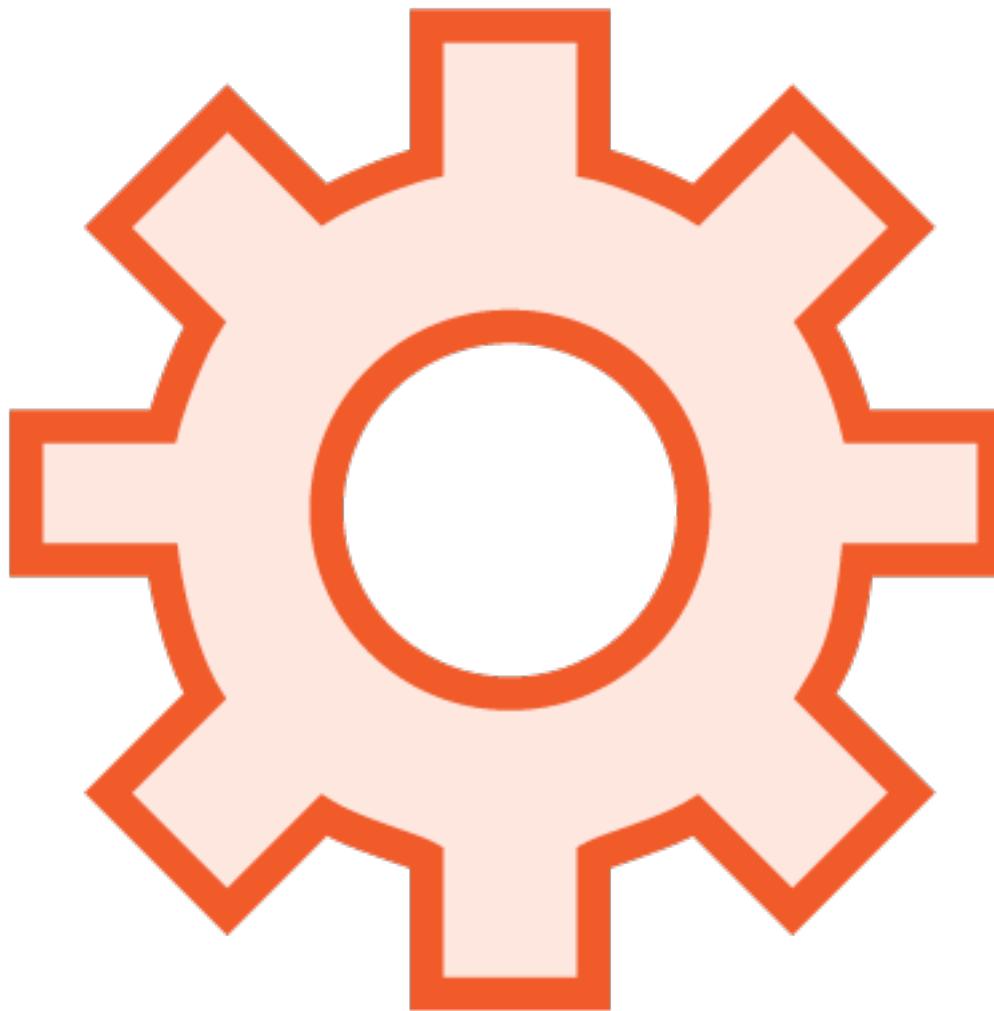
Marble Diagram: combineLatest



```
combineLatest([audience$, online$, phone$])
```



RxJS Creation Function: combineLatest



combineLatest is a combination function

- Takes in a set of Observables, subscribes
- Creates an output Observable

When an item is emitted from any Observable

- If all Observables have emitted at least once
- Emits a value to the output Observable

Emitted value combines the latest emitted value from each input Observable into an array

Completes when all input Observables complete

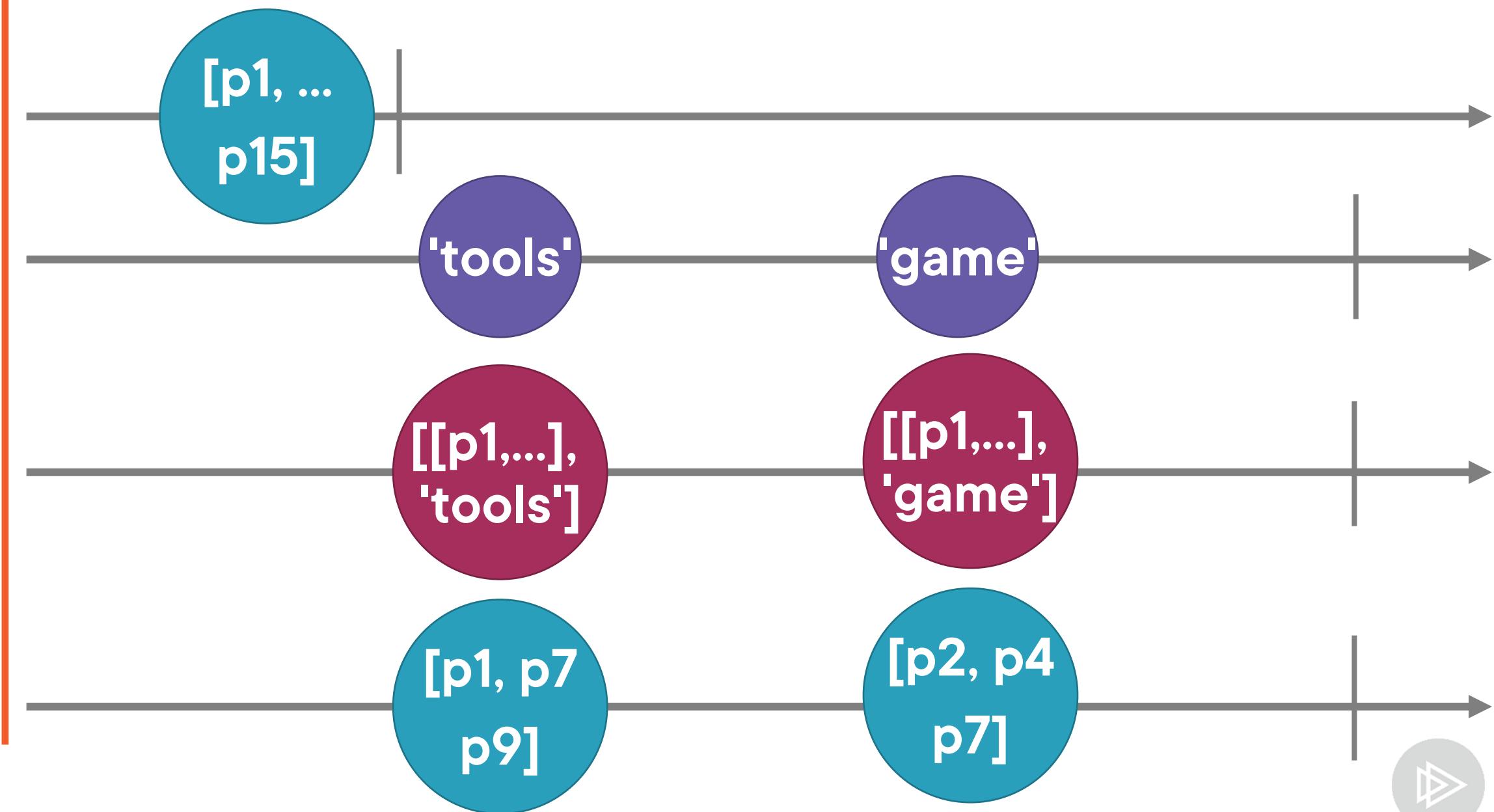


Use combineLatest



To work with multiple data sets

To reevaluate state when an action occurs



RxJS Creation Function: forkJoin



Creates an Observable whose value is defined:

- Using the **last** value from each input Observable

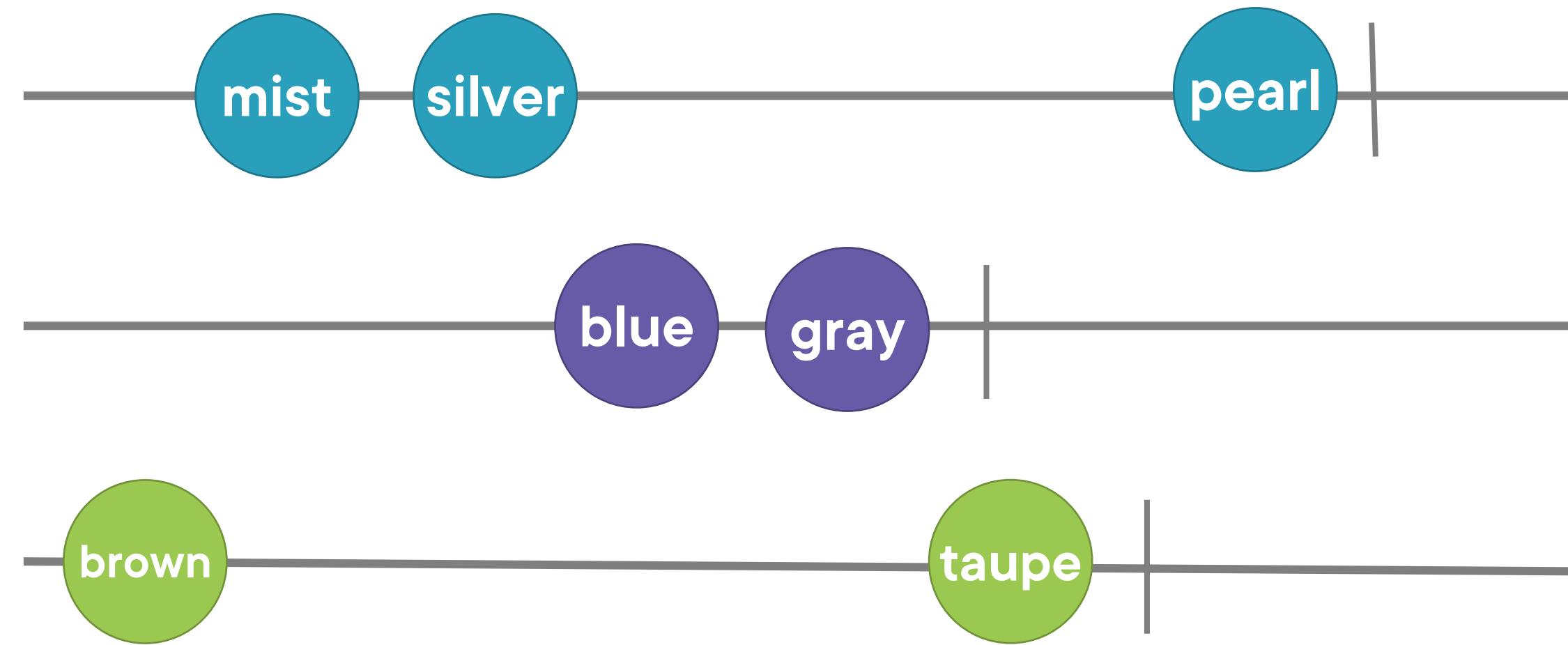
```
forkJoin([a$, b$, c$])
```

Only emits one time, when all input Observables complete

Static creation function, not a pipeable operator



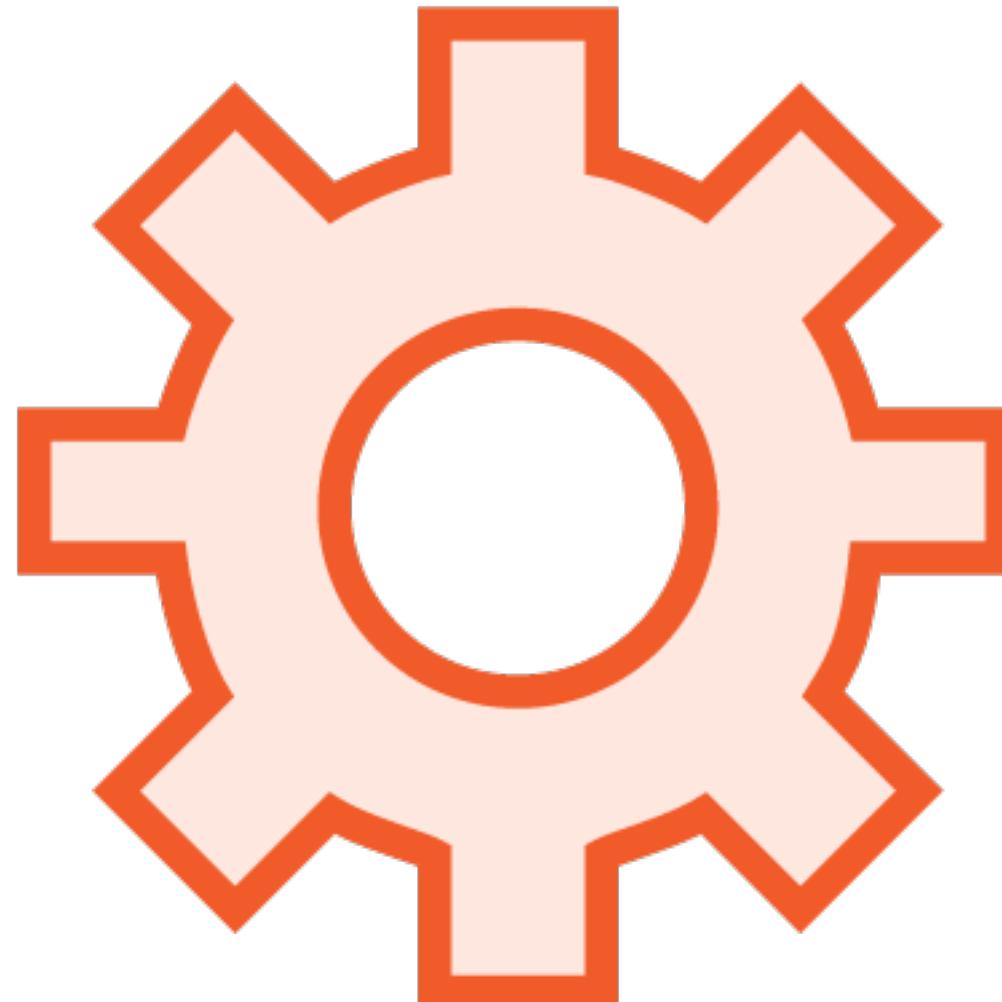
Marble Diagram: forkJoin



```
forkJoin([you$, yourPartner$, yourNeighbor$])
```



RxJS Creation Function: forkJoin



forkJoin is a combination function

- Takes in a set of Observables, subscribes
- Creates an output Observable

When all input Observables complete

- Emits a value to the output Observable
- And completes

Emitted value combines the last emitted value from each input Observable into an array



Use forkJoin



To wait to process any results until all Observables are complete

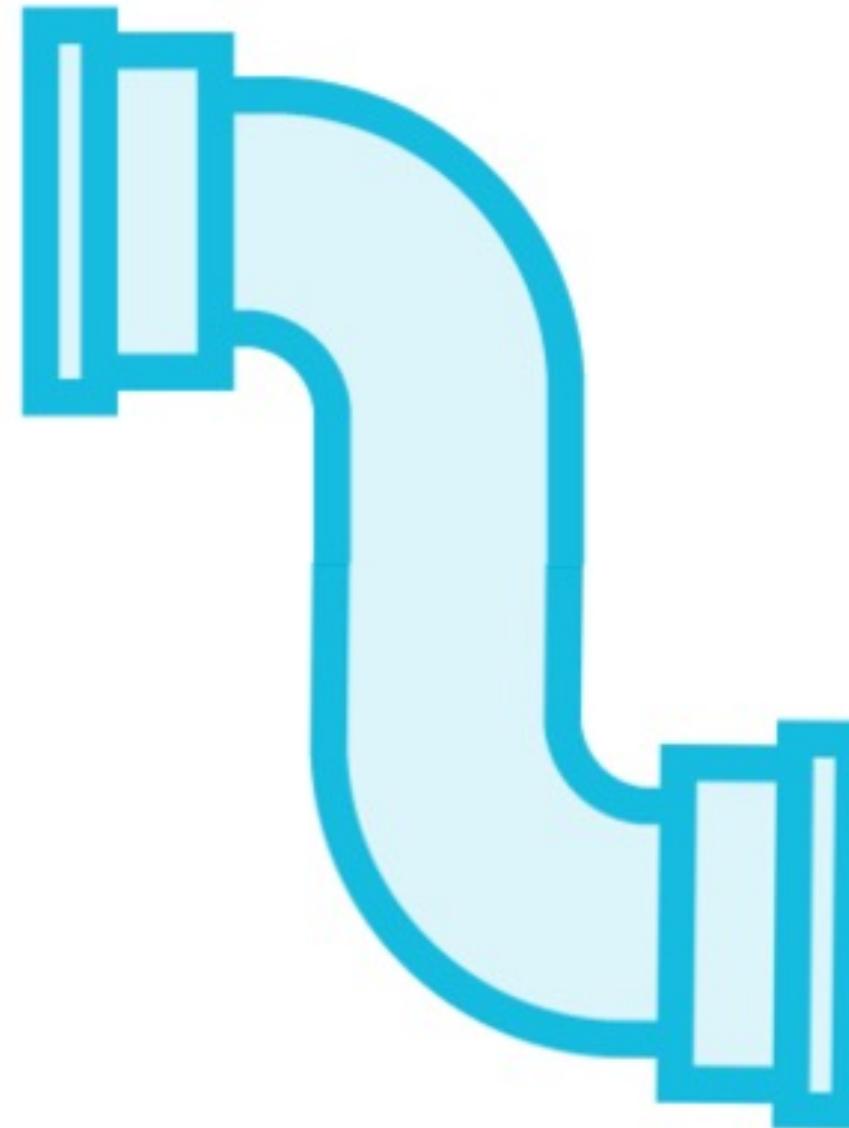
When emitting individual items you'd like emitted as a single array

Don't use when working with Observables that don't complete

- Such as action streams



RxJS Operator: withLatestFrom



Creates an Observable whose values are defined:

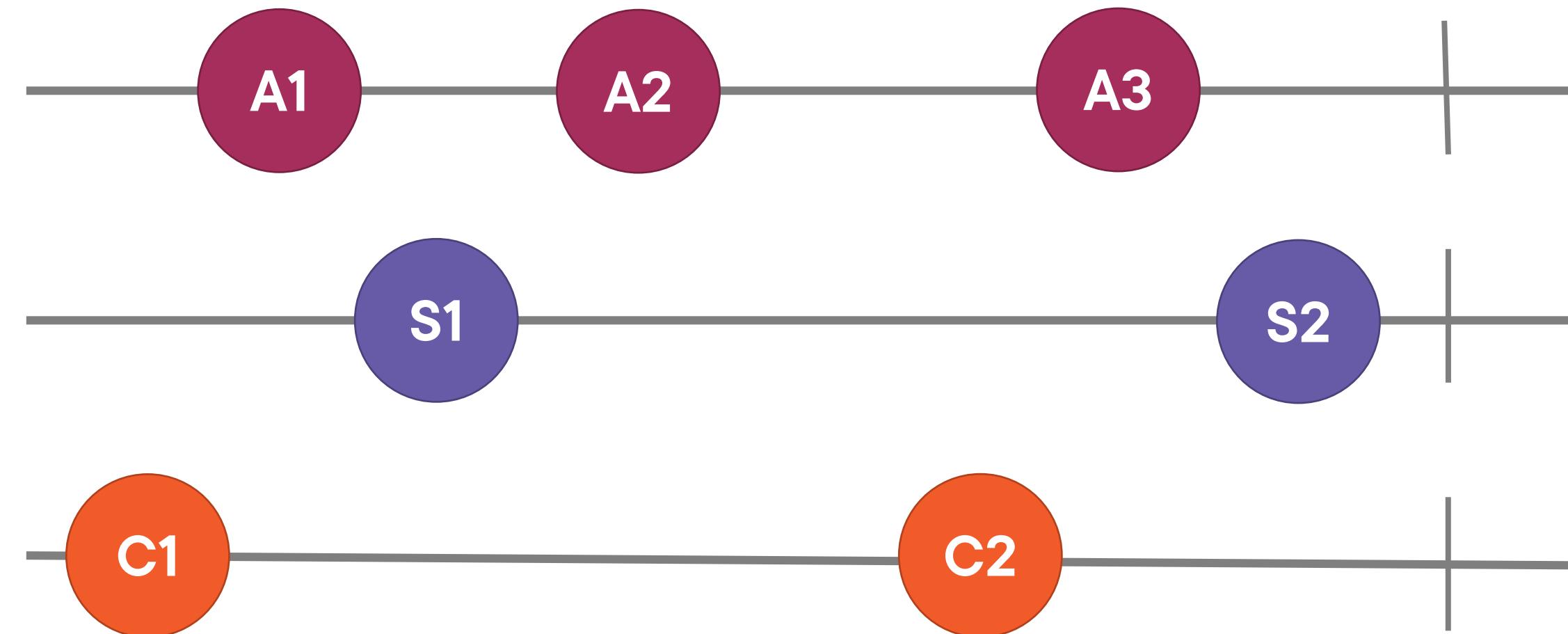
- Using the **latest** values from each input Observable
- But only when the **source** Observable emits

```
a$.pipe(withLatestFrom(b$, c$))
```

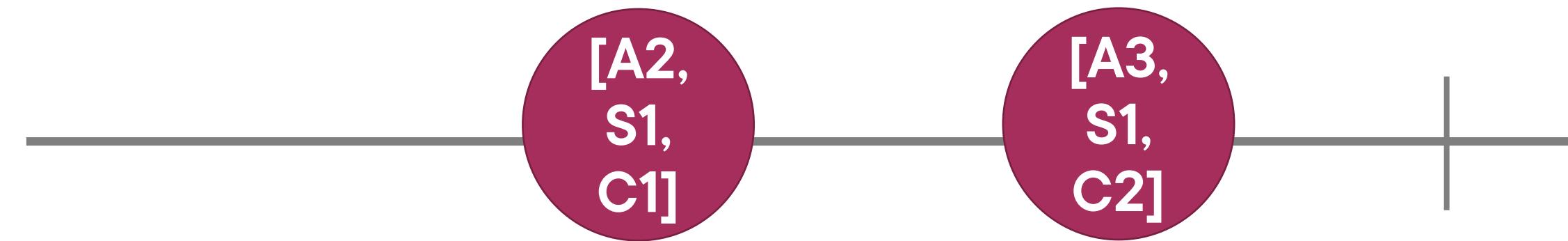
Pipeable operator



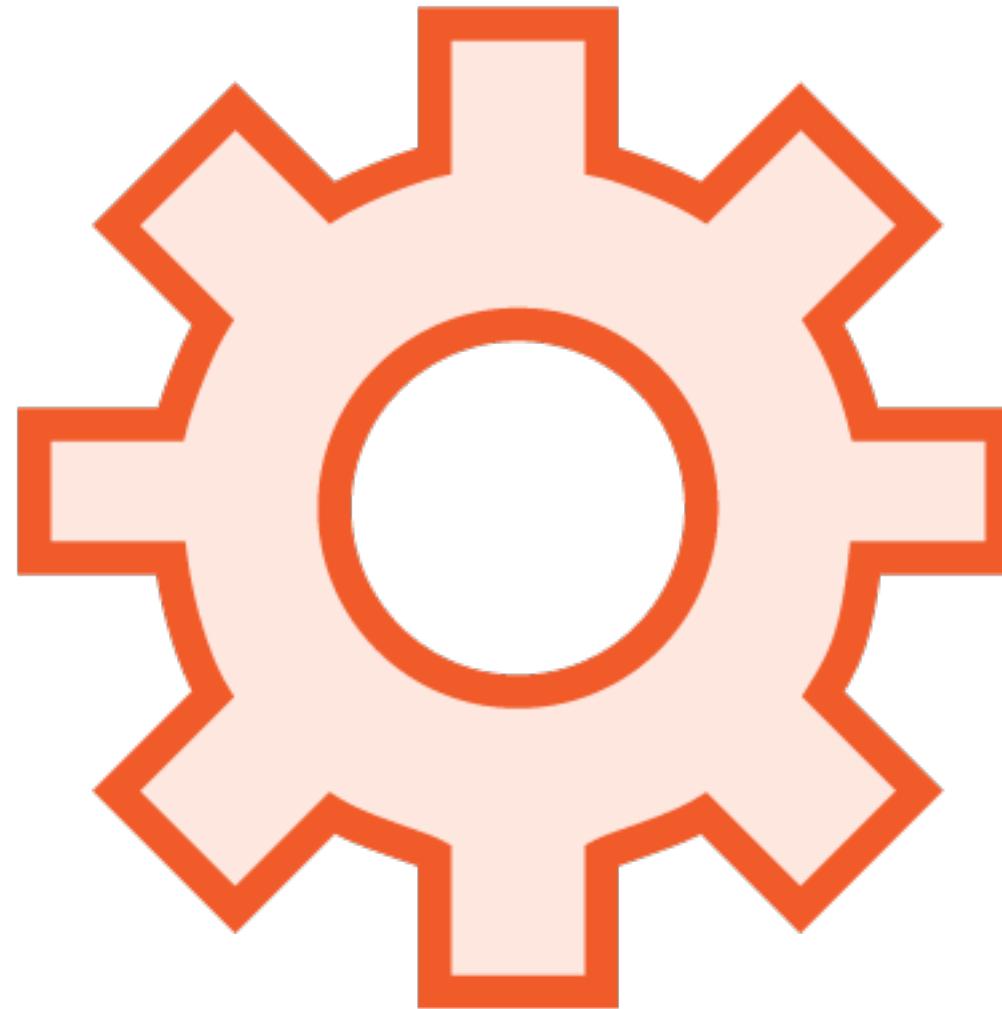
Marble Diagram: withLatestFrom



```
apples$.pipe(withLatestFrom(sticks$, caramel$))
```



RxJS Operator: withLatestFrom



withLatestFrom is a combination operator

- Takes in a set of Observables, subscribes
- Creates an output Observable

When an item is emitted from the source Observable

- If all Observables have emitted at least once
- Emits a value to the output Observable

Emitted value combines the latest emitted value from each input Observable into an array

Completes when the source Observable completes



Use withLatestFrom



To react to changes in only one Observable

To regulate the output of the other Observables



Product List

- Display All - ▾Add Product

Product	Code	Category	Price	In Stock
Leaf Rake	GDN-0011	1	\$29.92	15
Garden Cart	GDN-0023	1	\$49.49	2
Hammer	TBX-0048	3	\$13.35	8
Saw	TBX-0022	3	\$17.33	6
Video Game Controller	GMG-0042	5	\$53.93	12



Product List

- Display All - ▾Add Product

Product	Code	Category	Price	In Stock
Leaf Rake	GDN-0011	Garden	\$29.92	15
Garden Cart	GDN-0023	Garden	\$49.49	2
Hammer	TBX-0048	Toolbox	\$13.35	8
Saw	TBX-0022	Toolbox	\$17.33	6
Video Game Controller	GMG-0042	Gaming	\$53.93	12



Mapping an Id to a String

```
export interface Product
{
    id: number;
    productName: string;
    productCode?: string;
    description?: string;
    price?: number;
    categoryId?: number;
}
```

```
export interface Product
{
    id: number;
    productName: string;
    productCode?: string;
    description?: string;
    price?: number;
    categoryId?: number;
    category?: string;
}
```



Mapping an Id to a String

```
products$ = this.http.get<Product[]>(this.url)
  .pipe(
    map(products =>
      products.map(product => ({
        ...product,
        price: product.price * 1.5,
        category: ???  
      } as Product)))
  );
```

```
category: this.getCategory(product.categoryId)
```

```
categories$ = this.http.get<ProductCategory[]>(this.categoriesUrl);
```



Combining the Streams

```
productsWithCategory$ = combineLatest([  
    this.products$,  
    this.productCategories$  
]);
```



[product[], category[]]

```
productsWithCategory$ = forkJoin([  
    this.products$,  
    this.productCategories$  
]);
```



[product[], category[]]

```
productsWithCategory$ = this.products$  
.pipe(  
    withLatestFrom(this.productCategories$)  
);
```



[product[], category[]]



Mapping an Id to a String

```
products$ = this.http.get<Product[]>(this.url);
```

```
productCategories$ = this.http.get<ProductCategory[]>(this.categoriesUrl);
```

```
productsWithCategory$ = combineLatest([
  this.products$,
  this.productCategories$
]).pipe(
  map(([products, categories]) =>
    products.map(product => ({
      ...product,
      price: product.price * 1.5,
      category: categories.find(
        c => product.categoryId === c.id
      ).name
    } as Product)))
);
```



Mapping an Id to a String

```
productsWithCategory$ = combineLatest([
  this.products$,
  this.productCategories$
]).pipe(
  map(([products, categories]) =>
    products.map(product => ({
      ...product,
      price: product.price * 1.5,
      category: categories.find(
        c => product.categoryId === c.id
      ).name
    } as Product)))
);
```



[product[], category[]]



Demo



**Combining data streams
to map an id to a string**



RxJS Checklist: Combination features



combineLatest: Emits any time a new value is emitted from any of the input Observables

```
combineLatest([a$, b$, c$])
```

forkJoin: Emits only the last emitted values from each input Observable after they complete

```
forkJoin([a$, b$, c$])
```

withLatestFrom: Emits any time a new value is emitted from the source Observable

```
a$.pipe(withLatestFrom(b$, c$))
```



RxJS Checklist: Emitted Item



`combineLatest([a$, b$, c$])`

`forkJoin([a$, b$, c$])`

`a$.pipe(withLatestFrom(b$, c$))`

`[a1, b1, c1]`



RxJS Checklist: Mapping a data stream to a lookup stream value



Combine the data streams

Map the items

```
productsWithCategory$ = combineLatest([
  this.products$,
  this.productCategories$
]).pipe(
  map(([products, categories]) =>
    products.map(product => ({
      ...product,
      price: product.price * 1.5,
      category: categories.find(
        c => product.categoryId === c.id).name
    } as Product)))
);
```





Coming up next...

Reacting to Actions

