

Filter unique objects by id



```
const arr = [
  {id:1, name:"A"},
  {id:2, name:"B"},
  {id:1, name:"A2"}
];

const unique = arr.filter((obj, index, self) =>
  index === self.findIndex(o => o.id === obj.id)
);

explain it line by line unable to understand it
```

Here is the **line-by-line, beginner-friendly, simple explanation** of your code.

✓ Your Array

```
js

const arr = [
  {id:1, name:"A"},
  {id:2, name:"B"},
  {id:1, name:"A2"}
];
```

This array contains **three objects**, but **id = 1** appears **two times**.

Goal → **keep only the first object of each id** (remove duplicates by id).

✓ Filter Code

```
js

const unique = arr.filter((obj, index, self) =>
  index === self.findIndex(o => o.id === obj.id)
);
```

Let's break this into **small parts**.

🔥 filter() runs for every element

Filter gives you access to:

- **obj** → current object

- `index` → current element index
- `self` → the whole array (`arr`)

Inside filter, this line runs for each object

```
js
index === self.findIndex(o => o.id === obj.id)
```

This condition decides **whether the object should be kept**.

Let's understand `self.findIndex()` first.

What does `findIndex` do here?

```
js
self.findIndex(o => o.id === obj.id)
```

`findIndex` finds the *first position* in the array where:

```
python
o.id === obj.id    (same id as current object)
```

So for each element:

✓ **When `obj = {id:1, name:"A"}, index = 0`**

`findIndex(o => o.id === 1) → 0`
(first time id 1 appears is at index 0)

✓ **When `obj = {id:2, name:"B"}, index = 1`**

`findIndex(o => o.id === 2) → 1`

✗ **When `obj = {id:1, name:"A2"}, index = 2`**

`findIndex(o => o.id === 1) → 0`
(because id 1 appeared earlier)

Now compare

```
js
index === firstIndexOfSameId
```

For {id:1, name:"A"}

index = 0

findIndex = 0

0 === 0 → **true** → kept ✓

For {id:2, name:"B"}

index = 1

findIndex = 1

1 === 1 → **true** → kept ✓

For {id:1, name:"A2"}

index = 2

findIndex = 0

2 === 0 → **false** → removed ✗

Final Output

```
js
[
  { id: 1, name: "A" },
  { id: 2, name: "B" }
]
```

Duplicates removed based on **id**.

★ One-line summary

Keep the object only if its index is the **first place** where that id appears in the array.

If you want, I can also explain this with a **diagram**, **dry run table**, or give **alternative simpler methods** (like using a Set or reduce).