

Filter questions solutions



Provide me 20 questions and their solutions of filter most frequently asked questions 5 easy , 5 medium , 10 hard questions

Below are **20 most frequently asked JavaScript filter()** interview questions with **clear beginner-friendly explanations + final solutions**.

✓ 5 Easy Questions

1. Remove negative numbers from an array

Question:

Input: [-2, 5, -1, 7, -9, 3]

Output: [5, 7, 3]

Solution:

```
js
const arr = [-2, 5, -1, 7, -9, 3];
const result = arr.filter(num => num >= 0);
```

2. Filter even numbers

Input: [1,2,3,4,5,6]

Output: [2,4,6]

Solution:

```
js
const arr = [1,2,3,4,5,6];
const result = arr.filter(num => num % 2 === 0);
```

3. Filter strings with length > 3

Input: ["a","hello","ok","react"]

Output: ["hello","react"]

Solution:

```
js
```

```
const words = ["a", "hello", "ok", "react"];
const result = words.filter(w => w.length > 3);
```

4. Filter truthy values

Input: [0, 1, false, 3, "", "hello"]

Output: [1, 3, "hello"]

Solution:

```
js
```

```
const arr = [0, 1, false, 3, "", "hello"];
const result = arr.filter(Boolean);
```

5. Filter people age ≥ 18

```
js
```

```
const people = [
  {name: "A", age: 17},
  {name: "B", age: 22},
  {name: "C", age: 15},
  {name: "D", age: 30}
];

const adults = people.filter(p => p.age >= 18);
```

5 Medium-Level Questions

6. Filter unique values

Input: [1,2,2,3,4,4,5]

Output: [1,2,3,4,5]

Solution:

```
js
```

```
const arr = [1,2,2,3,4,4,5];
const unique = arr.filter((item, index) => arr.indexOf(item) === index);
```

7. Filter users whose status is “active”

```
js

const users = [
  {name: "A", status: "inactive"},
  {name: "B", status: "active"},
  {name: "C", status: "active"}
];

const activeUsers = users.filter(u => u.status === "active");
```

8. Filter words that contain letter "a"

```
js

const words = ["react", "html", "css", "node"];
const result = words.filter(w => w.includes("a"));
```

9. Filter objects where salary > 50,000

```
js

const employees = [
  {name: "A", salary: 30000},
  {name: "B", salary: 70000},
  {name: "C", salary: 90000}
];

const rich = employees.filter(e => e.salary > 50000);
```

10. Remove null, undefined, and empty strings

```
js

const arr = ["hello", "", null, "react", undefined, "js"];
const cleaned = arr.filter(x => x);
```

10 Hard-Level Questions

11. Filter deeply nested objects by a condition

Input:

```
js

const data = [
  {id: 1, info: {active: true}},
  {id: 2, info: {active: false}},
```

```
{id: 3, info: {active: true}}
];
```

Output → items where `info.active === true`

Solution:

```
js

const result = data.filter(item => item.info.active);
```

12. Filter based on multiple conditions

Find students with score > 70 and age < 20

```
js

const students = [
  {name: "A", age: 19, score: 80},
  {name: "B", age: 21, score: 90},
  {name: "C", age: 17, score: 60}
];

const filtered = students.filter(s => s.score > 70 && s.age < 20);
```

13. Filter based on index (keep only odd index items)

```
js

const arr = [10,20,30,40,50];

const result = arr.filter((_, index) => index % 2 === 1);
```

14. Filter objects by key existence

Return objects that contain the key "role"

```
js

const users = [
  {name: "A", role: "dev"},
  {name: "B"},
  {name: "C", role: "tester"}
];

const hasRole = users.filter(u => u.role !== undefined);
```

15. Filter dates between a range

```
js

const dates = [
  "2022-01-10", "2022-02-15", "2022-03-20"
];

const start = new Date("2022-02-01");
const end = new Date("2022-03-01");

const result = dates.filter(d => {
  const date = new Date(d);
  return date >= start && date <= end;
});
```

16. Remove duplicates from array of objects

Based on `id` field:

```
js

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)
);
```

17. Filter items that match ANY value from another list

Example: show products whose category is in `allowedCategories` array

```
js

const products = [
  {id:1, cat:"food"},
  {id:2, cat:"tech"},
  {id:3, cat:"fashion"}
];

const allowed = ["tech", "fashion"];

const result = products.filter(p => allowed.includes(p.cat));
```

18. Logical NOT filter

Remove all falsy values except 0

Input: `[0,1,"",null,2,false,3]`

Output: `[0,1,2,3]`

```
js

const arr = [0,1,"",null,2,false,3];

const result = arr.filter(v => v || v === 0);
```

19. Filter array based on search text (case insensitive)

```
js

const items = ["React", "HTML", "css", "node", "JavaScript"];

const search = "a";

const result = items.filter(i =>
  i.toLowerCase().includes(search.toLowerCase())
);
```

20. Custom filter to simulate SQL "WHERE IN (...) AND ..."

Find users whose role is "admin" AND id is in allowed list.

```
js

const users = [
  {id:1, role:"admin"},
  {id:2, role:"user"},
  {id:4, role:"admin"},
  {id:5, role:"admin"}
];

const allowedIds = [1,4];

const result = users.filter(
  u => u.role === "admin" && allowedIds.includes(u.id)
);
```

 **Want this as a PDF, notes sheet, or flashcards for memorizing?**