Filter falsy values explanation

const arr = ["hello", "", null, "react", undefined, "js"];

const cleaned = arr.filter(x => x);

explain this code what is happening and why and how it is removing "", null, undefined

Here is a **very simple, beginner-friendly explanation** of what is happening.

✅ Code

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

✅ Step-by-step Explanation

1. `filter()` keeps only values where the callback returns `true`.

`arr.filter(x => x)`
means:

- For every element `x` in the array
- Return `x` itself
- If `x` is **truthy**, it will be kept
- If `x` is **falsy**, it will be removed

✅ 2. What is truthy and falsy?

JavaScript automatically converts values to `true` or `false` when used in conditions.

Falsy values (removed by filter here):

- `""` (empty string)
- `null`
- `undefined`
- `0`

- `false`
- `NaN`

Truthy values (kept):

- Non-empty strings → `"hello"`, `"react"`, `"js"`
- Any object
- Any array
- Any number except 0

✅ 3. Why does `filter(x => x)` remove `""`, `null`, `undefined`?

Because:

Value	Boolean(x)	Kept?	Reason
`"hello"`	true	Yes	Non-empty string is truthy
`""`	false	No	Empty string is falsy
`null`	false	No	Falsy
`"react"`	true	Yes	Truthy
`undefined`	false	No	Falsy
`"js"`	true	Yes	Truthy

✅ Final Output

```js
["hello", "react", "js"]
```

⭐ Why do people use `filter(Boolean)`?

Because:

```js
filter(x => x)
```

and

```js
filter(Boolean)
```

are the same.

`Boolean` is a function that converts a value to true/false.
So it is a cleaner shortcut.

If you want, I can also explain this using diagrams or animations.