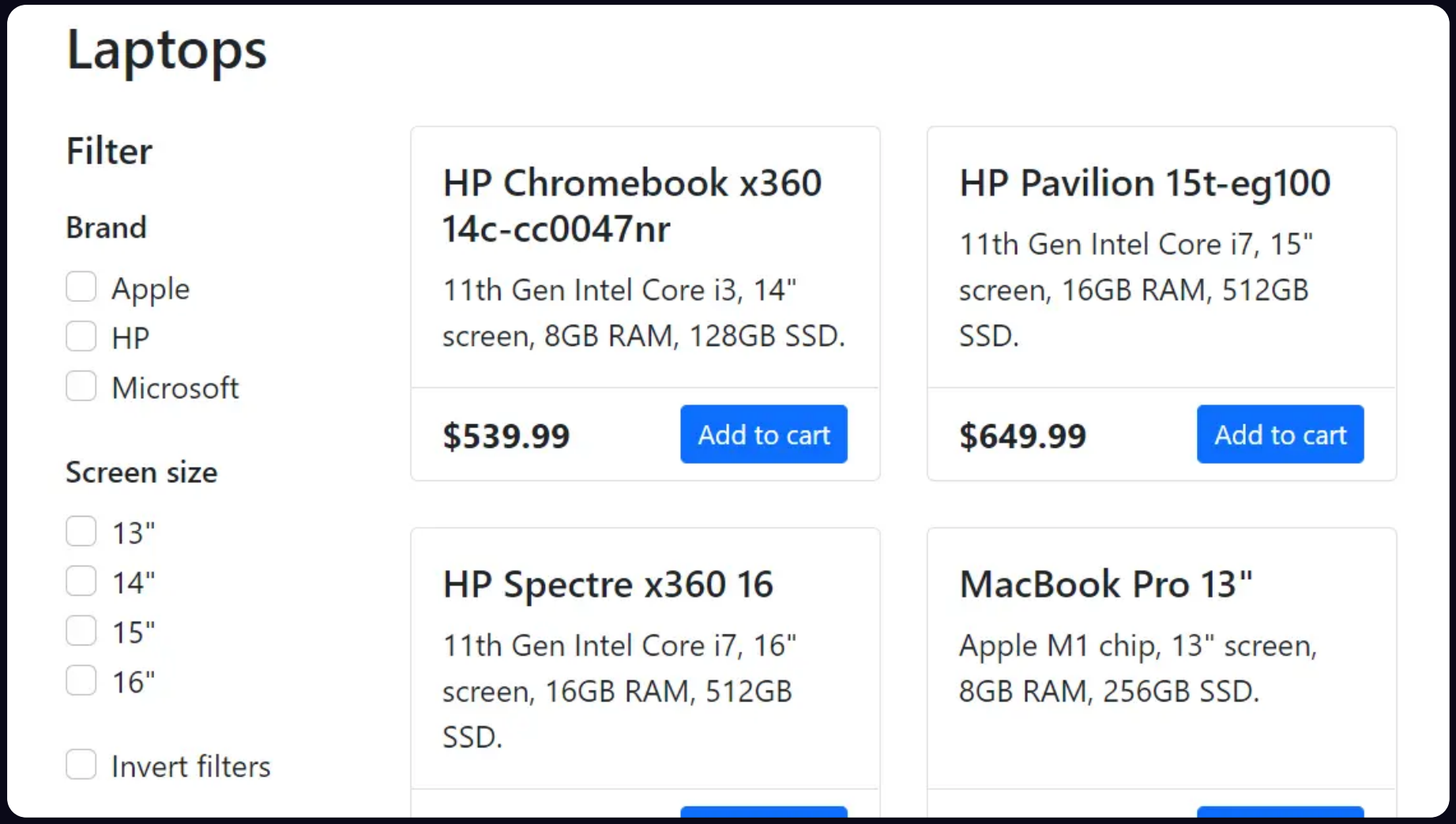
thinkdev #5

Logical operators

We learnt how to make decisions based on simple conditions.

But what if we have complex conditions?



A hypothetical e-commerce app with a list of laptops for sale and several options to filter the laptops.

Logical operators

Logical operators

OR

Logical operators

OR, AND

Logical operators

OR, AND, and NOT.

OR

Laptops

Filter

Brand

- [x] Apple
- [ ] HP
- [x] Microsoft

Screen size

- [ ] 13"
- [ ] 14"
- [ ] 15"
- [ ] 16"

- [ ] Invert filters

MacBook Pro 13"

Apple M1 chip, 13" screen, 8GB RAM, 256GB SSD.

$1299 Add to cart

MacBook Pro 14"

Apple M1 Pro chip, 14" screen, 16GB RAM, 512GB SSD.

$1999 Add to cart

MacBook Pro 16"

Apple M1 Pro chip, 16" screen, 16GB RAM, 512GB SSD.

Surface Laptop 4

11th Gen Intel Core i5, 13" screen, 8GB RAM, 256GB SSD.

Let’s assume each laptop looks like this:

```
const laptop = {
  title: 'MacBook Pro 13"',
  brand: 'Apple',
  screenSize: 13,
  // ...
}
```

OR syntax:

```
expression1 || expression2
```

One expression must be `true` for the result to be `true`.

```
const { brand } = laptop
if (brand === 'Apple' || brand === 'Microsoft') {
  console.log("It's an Apple or Microsoft laptop.")
}
```

AND

Laptops

Filter

Brand

- [x] Apple
- [ ] HP
- [ ] Microsoft

Screen size

- [ ] 13"
- [x] 14"
- [ ] 15"
- [ ] 16"

- [ ] Invert filters

MacBook Pro 14"

Apple M1 Pro chip, 14" screen, 16GB RAM, 512GB SSD.

$1999 Add to cart

AND syntax:

```
expression1 && expression2
```

Both expressions must be `true` for the result to be `true`.

```
const { brand, screenSize } = laptop
if (brand === 'Apple' && screenSize === 14) {
  console.log("It's a 14-inch Apple laptop.")
}
```

Let’s take a more complex example

Laptops

Filter

Brand

- [x] Apple
- [ ] HP
- [x] Microsoft

Screen size

- [x] 13"
- [x] 14"
- [ ] 15"
- [ ] 16"

- [ ] Invert filters

MacBook Pro 13"

Apple M1 chip, 13" screen, 8GB RAM, 256GB SSD.

$1299 Add to cart

MacBook Pro 14"

Apple M1 Pro chip, 14" screen, 16GB RAM, 512GB SSD.

$1999 Add to cart

Surface Laptop 4

11th Gen Intel Core i5, 13" screen, 8GB RAM, 256GB SSD.

$799.99 Add to cart

```
const { brand, screenSize } = laptop
if (
  brand === 'Apple' || brand === 'Microsoft'

) {
  console.log(
    "It's a 13 or 14-inch Apple or Microsoft laptop."
  )
}
```

```
const { brand, screenSize } = laptop
if (
  brand === 'Apple' || brand === 'Microsoft'
  screenSize === 13 || screenSize === 14
) {
  console.log(
    "It's a 13 or 14-inch Apple or Microsoft laptop."
  )
}
```

AND has higher precedence than OR:

```
const { brand, screenSize } = laptop
if (
  (brand === 'Apple' || brand === 'Microsoft') &&
  (screenSize === 13 || screenSize === 14)
) {
  console.log(
    "It's a 13 or 14-inch Apple or Microsoft laptop."
  )
}
```

AND has higher precedence than OR:

```
const { brand, screenSize } = laptop
if (
  (brand === 'Apple' || brand === 'Microsoft') &&
  (screenSize === 13 || screenSize === 14)
) {
  console.log(
    "It's a 13 or 14-inch Apple or Microsoft laptop."
  )
}
```

NOT

Laptops

Filter

Brand

- [x] Apple
- [ ] HP
- [ ] Microsoft

Screen size

- [ ] 13"
- [x] 14"
- [ ] 15"
- [ ] 16"

- [x] Invert filters

HP Chromebook x360 14c-cc0047nr

11th Gen Intel Core i3, 14" screen, 8GB RAM, 128GB SSD.

$539.99 Add to cart

HP Pavilion 15t-eg100

11th Gen Intel Core i7, 15" screen, 16GB RAM, 512GB SSD.

$649.99 Add to cart

HP Spectre x360 16

11th Gen Intel Core i7, 16" screen, 16GB RAM, 512GB SSD.

MacBook Pro 13"

Apple M1 chip, 13" screen, 8GB RAM, 256GB SSD.

NOT syntax:

```
!expression
```

- If `expression` is `true`, the result is `false`.
- If `expression` is `false`, the result is `true`.

We wrote this previously:

```
const { brand, screenSize } = laptop
if (brand === 'Apple' && screenSize === 14) {
  console.log("It's a 14-inch Apple laptop.")
}
```

Let’s invert it (NOT has higher precedence than AND):

```
const { brand, screenSize } = laptop
if (!(brand === 'Apple' && screenSize === 14)) {
  console.log("It's not a 14-inch Apple laptop.")
}
```

Let’s invert it (NOT has higher precedence than AND):

```
const { brand, screenSize } = laptop
if (!(brand === 'Apple' && screenSize === 14)) {
  console.log("It's not a 14-inch Apple laptop.")
}
```

“Truthiness”

Conditions don’t have to be boolean; JavaScript automatically converts them.

We did this in the previous lesson:

```
if (course.rating !== 0) {
  console.log(`Rating: ${course.rating}`)
}
```

We could write it this way too:

```
if (course.rating) {
  console.log(`Rating: ${course.rating}`)
}
```

The following values convert to `false`; we call them *falsy* values:

```
Boolean(0)          // false
Boolean("")         // false
Boolean(null)       // false
Boolean(undefined)  // false
Boolean(false)      // false
```

All other values convert to `true`, so they are *truthy*:

```
Boolean(3.4)                  // true
Boolean("thinkdev")           // true
Boolean({ key: 'value' }) // true
Boolean({})                   // true
Boolean([1, 2])               // true
Boolean([])                   // true
Boolean(true)                 // true
```

Redefining the logical operations

The NOT operation returns `true` if the *operand* is falsy, and `false` if the operand is truthy.

```
!true   // false
!0      // true
!"hi"   // false
```

The OR operation returns the first truthy operand.
If there's none, it returns the last operand.

```
true || false;              // true
0 || null;                  // null
"hi" || "hey" || "hello"; // "hi"
```

The AND operation returns the first falsy operand.
If there's none, it returns the last operand.

```
true && false;              // false
0 && null;                  // 0
"hi" && "hey" && "hello"; // "hello"
```

Short-circuit evaluation

Open your REPL and type:

```
abc && false;
```

There’s no reference (i.e. variable) called abc:

```
abc && false;
// Uncaught ReferenceError: abc is not defined
```

Rearrange and abc will be ignored:

```
false && abc;
// false
```

It applies to the OR operation too:

```
abc || true;
// Uncaught ReferenceError: abc is not defined

true || abc;
// true
```