thinkdev #4

Making decisions

Explore featured courses

The University of Glasgow

The Museum as a Site and Source for Learning

4.6 (75 reviews)

Find out more

Sentinel|9 & FutureLearn

NEW

Fundamentals of Business Strategy

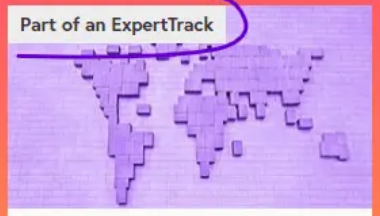
Find out more

University of Groningen, University of Cambridge & University Medical Center Groningen (UMCG)

Young People and Mental Health

4.7 (649 reviews)

Find out more



Coventry University

International Logistics: A Beginner's Guide to Logistics Management

Find out more

First, let's learn how to do some basic comparison

The relational operators:

```
3 > 2    // true
8 < 5    // false
9 >= 13  // false
6 <= 6   // true
```

Use the *strict equality operator* `===` to determine if two values are equal:

```
2 === 5 - 3           // true
89.0 === 89           // true
10 === '10'           // false
'hello' === 'Hello'   // false
true === false        // false
```

There’s a strict inequality counterpart:

```
2 !== 5 - 3           // false
89.0 !== 89           // false
10 !== '10'           // true
'hello' !== 'Hello'   // true
true !== false        // true
```

No two objects have the same value, even if they look alike:

```
const obj1 = { prop: "value" }
const obj2 = { prop: "value" }
console.log(obj1 === obj2) // false 😕
```

But, same object, same value:

```
const obj1 = { prop: "value" }
const obj2 = obj1
console.log(obj1 === obj2) // true
```

Let’s get to making decisions now.

The `if` statement

```
if (expression) {
  statement1
  statement2
  ...
}
```

```
if (expression) {
  statement1
  statement2
  ...
}
```

If `expression` is `true`, execute the statements in the curly brackets. Otherwise, ignore the statements.

Let's consider the FutureLearn example.

We represented a course like so in the previous lesson:

```
const course = {
  title: 'The Museum as a Site and ...',
  rating: 4.6,
  reviewsCount: 75,
  isNew: false,
  isPartOfAnExpertTrack: false,
}
```

Let’s determine if the course has a rating:

```
const course = {
  rating: 4.6,
  // ...
}

course.rating !== 0.0 // true
```

Now we can act accordingly:

```
const course = {
  rating: 4.6,
  // ...
}

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

Let's add some logs around the `if` statement for clarity:

```
const course = {
  rating: 4.6,
  // ...
}

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

Here’s the output:

```
Before decision
Rating: 4.6
After decision
```

What if the rating is 0?

```
const course = {
  rating: 0,
  // ...
}

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

Then the output is just this:

```
Before decision
After decision
```

How do we print a different message?

else

```
const course = {
  rating: 0,
  // ...
}

console.log('Before decision')
if (course.rating !== 0) {
  console.log(`Rating: ${course.rating}`)
} else {
  console.log('No rating')
}
```

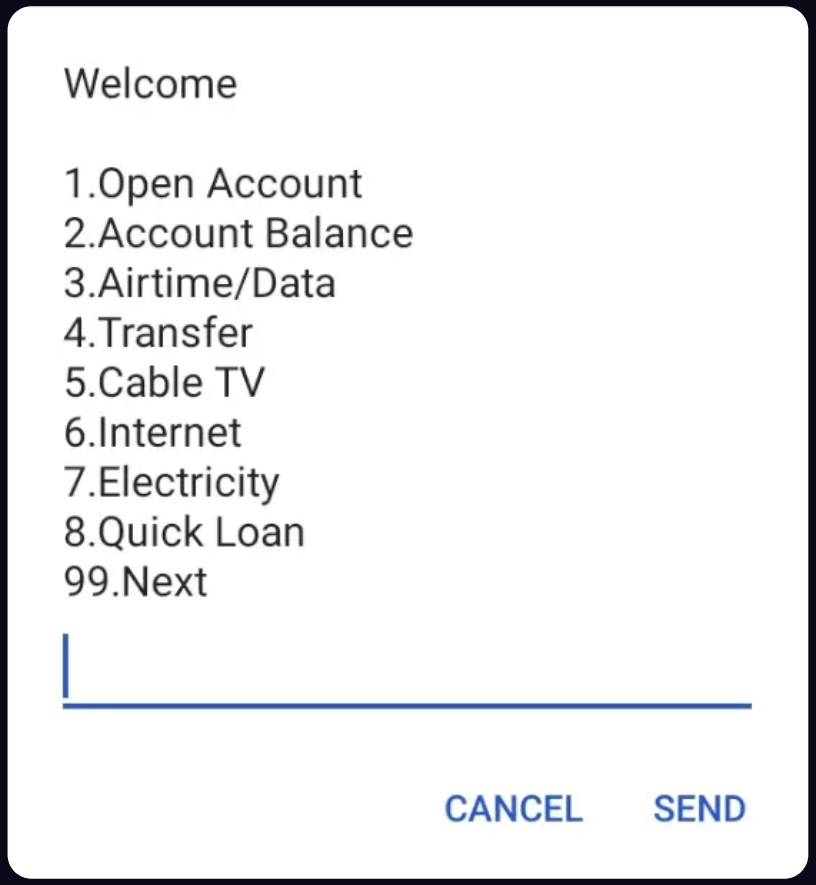
```
  rating: 0,
  // ...
}

console.log('Before decision')
if (course.rating !== 0) {
  console.log(`Rating: ${course.rating}`)
} else {
  console.log('No rating')
}
console.log('After decision')
```

The result:

```
Before decision
No rating
After decision
```

Let's consider another example.



A USSD menu with different options to choose from.

How do we express the several alternatives?

else if

```
const choice = 1 // Could be any other number

if (choice === 1) {
  console.log('Open Account')
} else if (choice === 2) {
  console.log('Account Balance')
} // ...
```

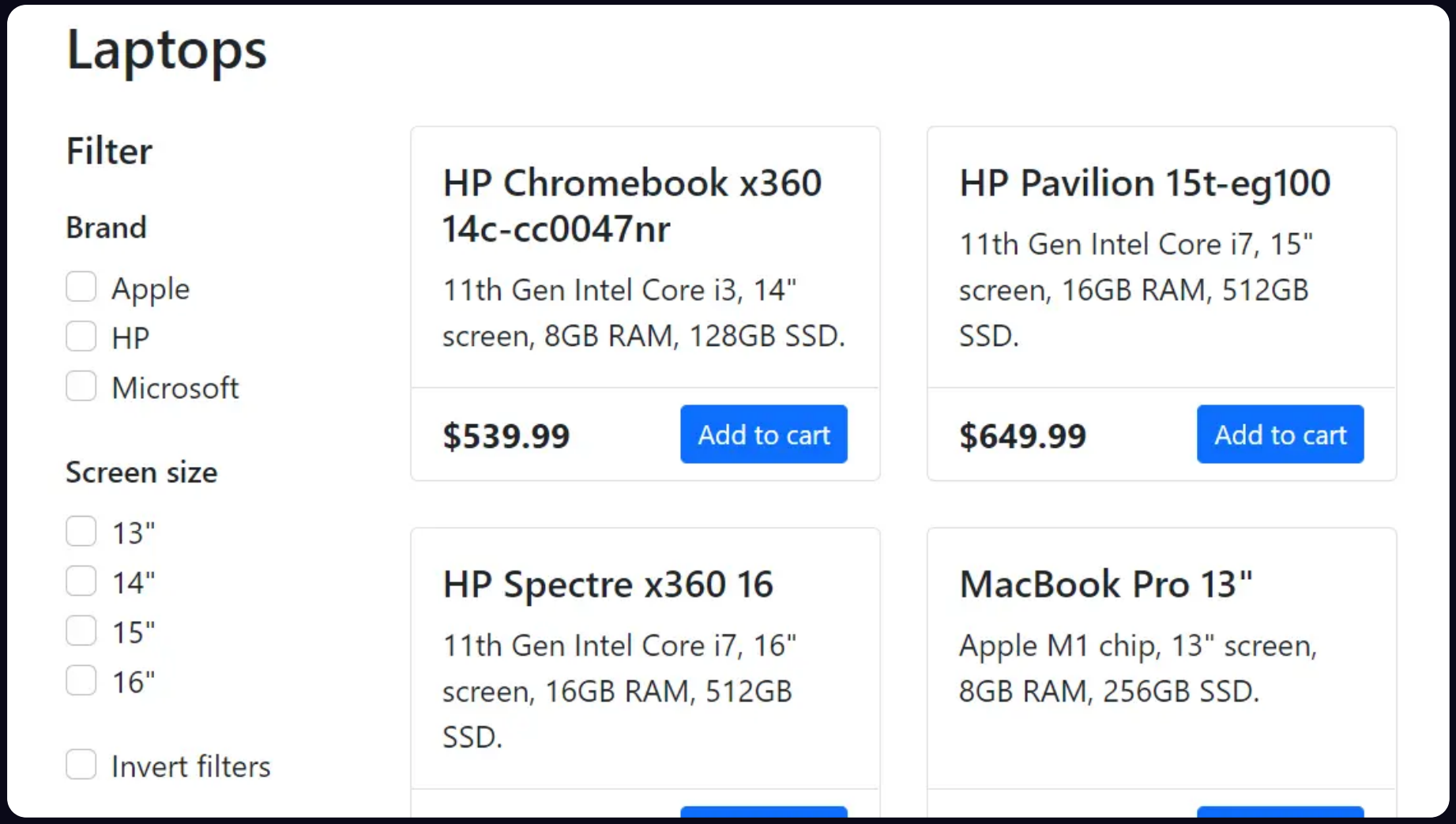
You can use a final `else` to handle any other choice.

```
const choice = 1 // Could be any other number

if (choice === 1) {
  console.log('Open Account')
} else if (choice === 2) {
  console.log('Account Balance')
} // ...
else {
  console.log('Invalid choice')
}
```

We can now make decisions based on single conditions.

What if we have many 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: '',
  brand: '',
  screenSize: 0,
  // ...
}
```

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.")
}
```

NOT

Laptops

Filter

Brand

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

Screen size

- [x] 13"
- [ ] 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 14"

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

NOT syntax:

```
!expression
```

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

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

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

One final thing ...

The expressions used in making decisions don’t have to be boolean; JavaScript automatically converts them.

We did this before:

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

We could write it this way too:

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

Same applies when using logical operators:

```
if (course.rating && course.reviewsCount) {
  console.log('Course has both rating and reviews.')
}
```

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
```