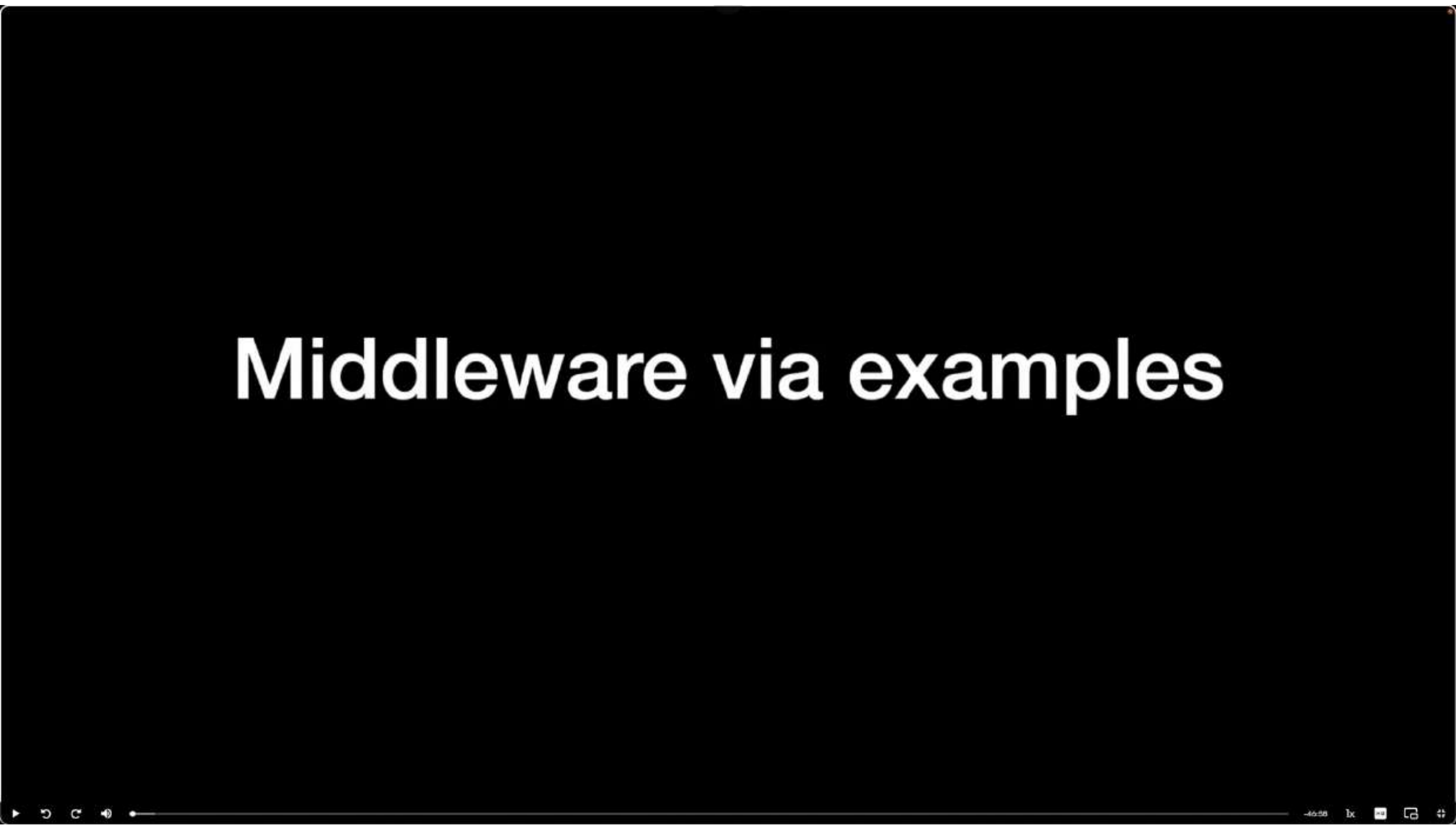
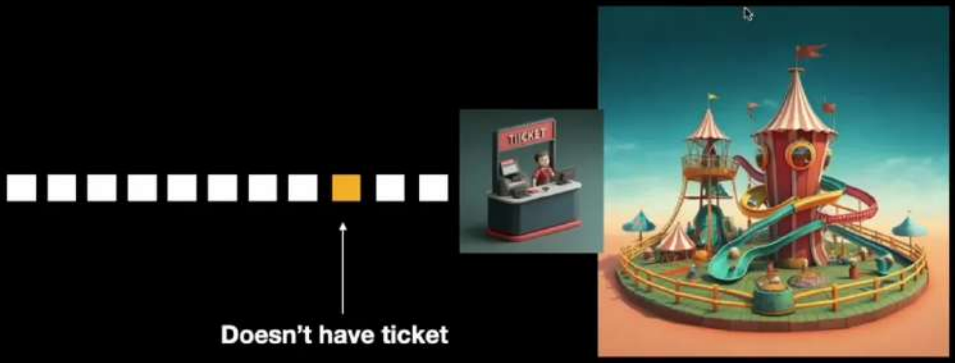
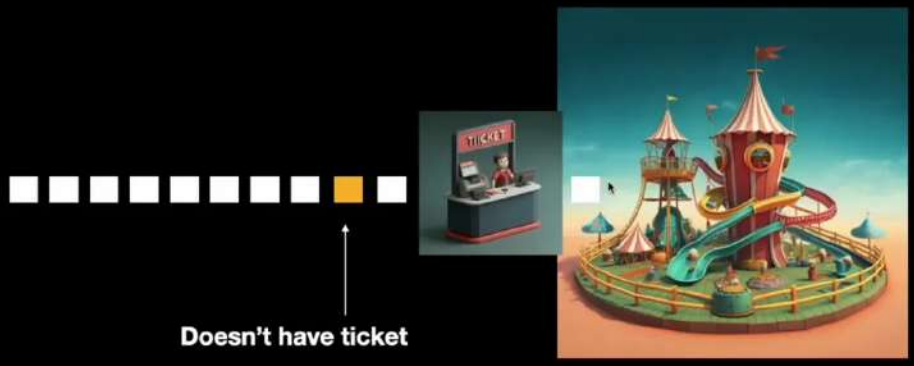


Middleware via examples

1. If you go to an amusement park



1. If you go to an amusement park

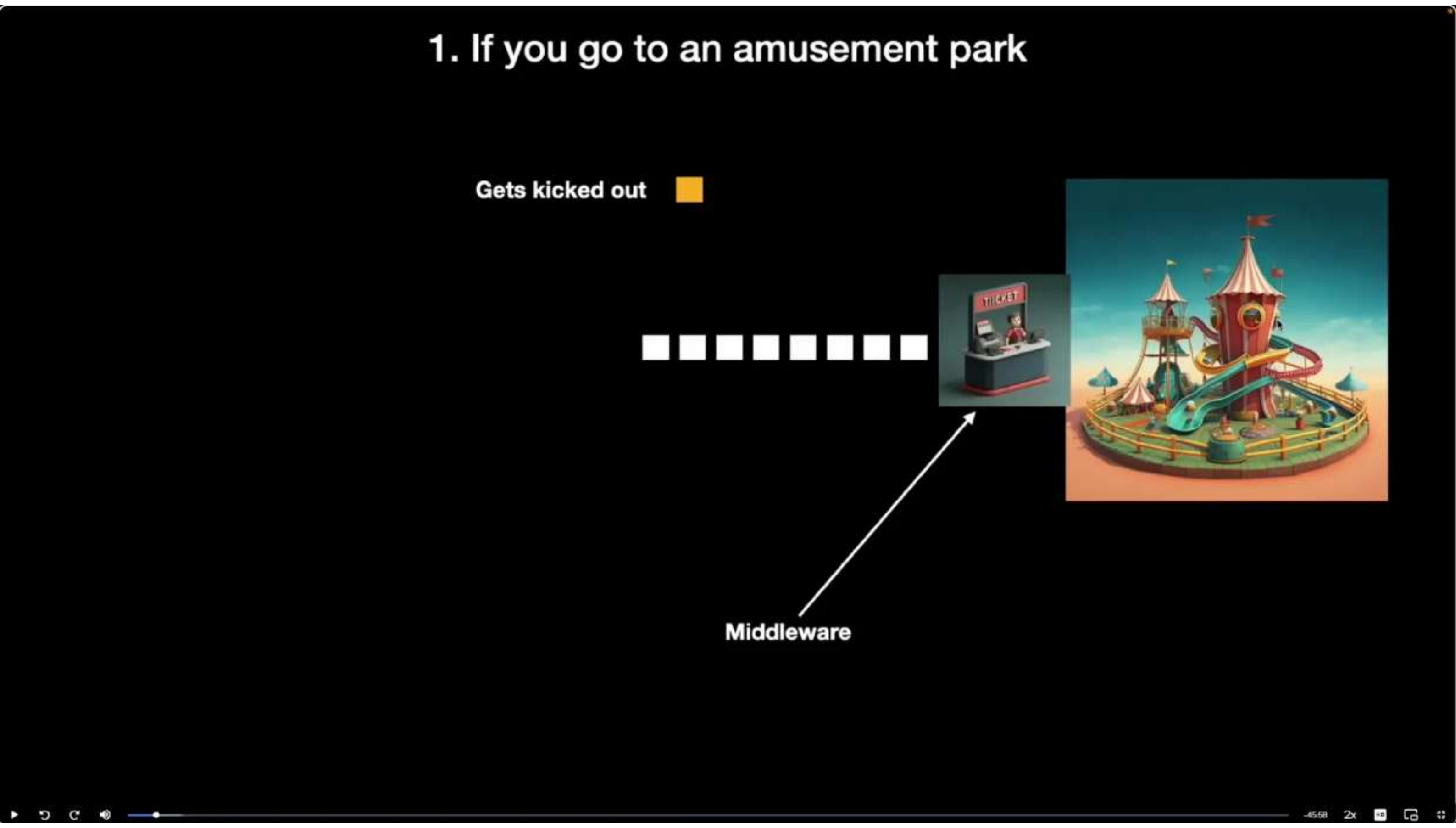


1. If you go to an amusement park

TICKET
Doesn't have ticket

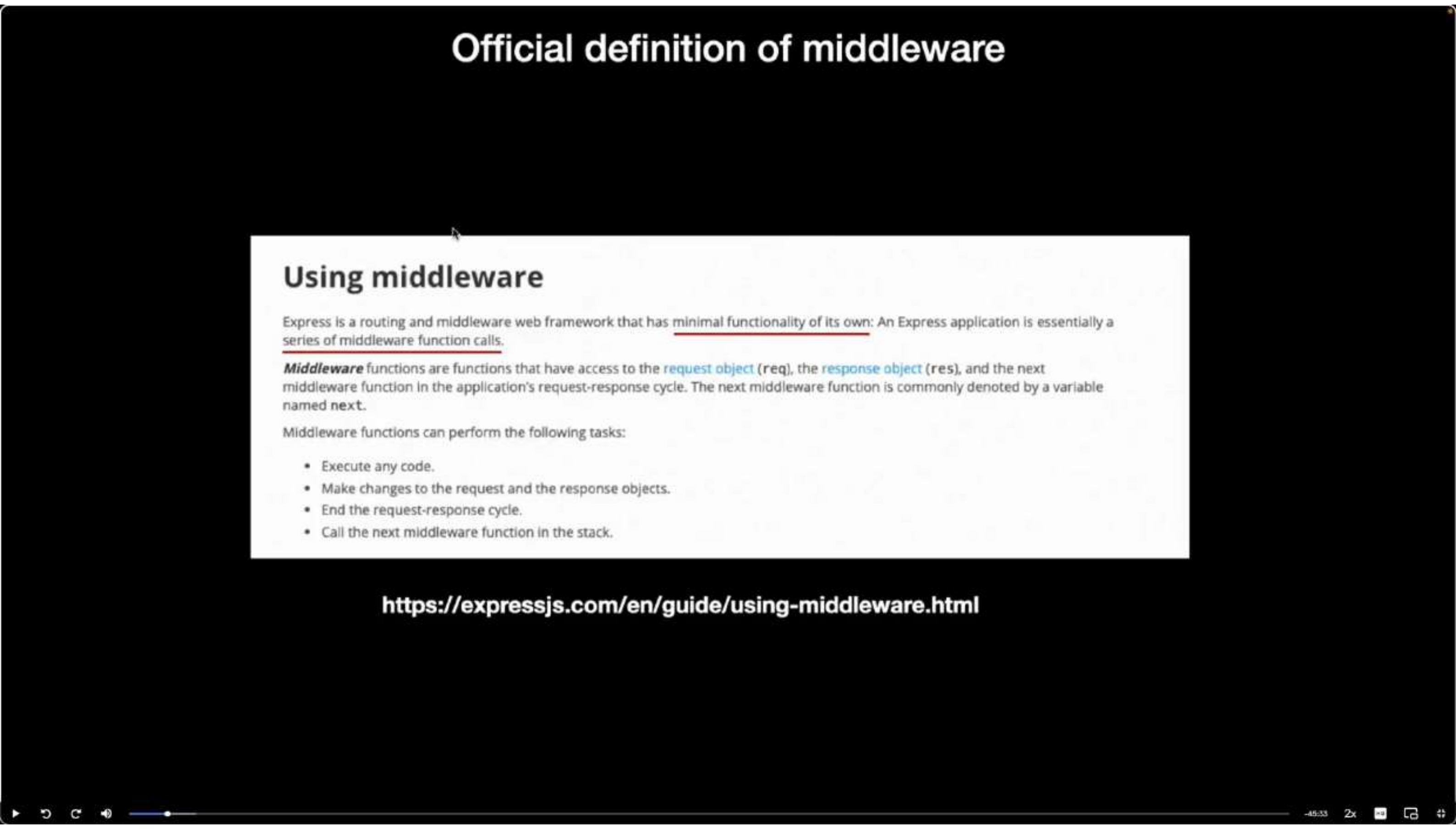
1. If you go to an amusement park

Gets kicked out 🟧



1. If you go to an amusement park

Gets kicked out 🟧



Official definition of middleware

Using middleware

Express is a routing and middleware web framework that has minimal functionality of its own: An Express application is essentially a series of middleware function calls.

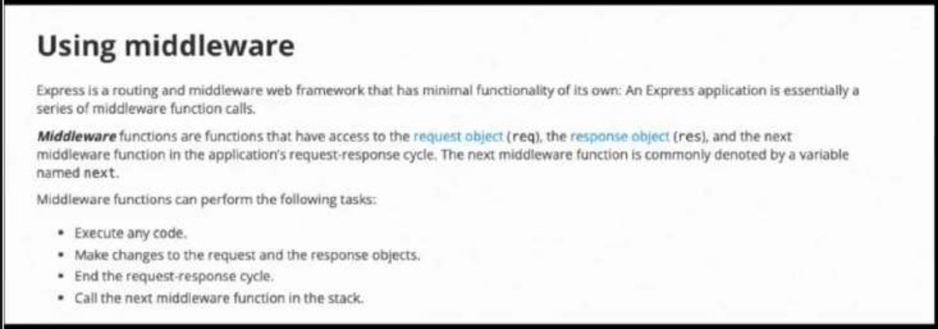
Middleware functions are functions that have access to the request object (`req`), the response object (`res`), and the next middleware function in the application's request-response cycle. The next middleware function is commonly denoted by a variable named `next`.

Middleware functions can perform the following tasks:

- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.

https://expressjs.com/en/guide/using-middleware.html

Official definition of middleware



Using middleware

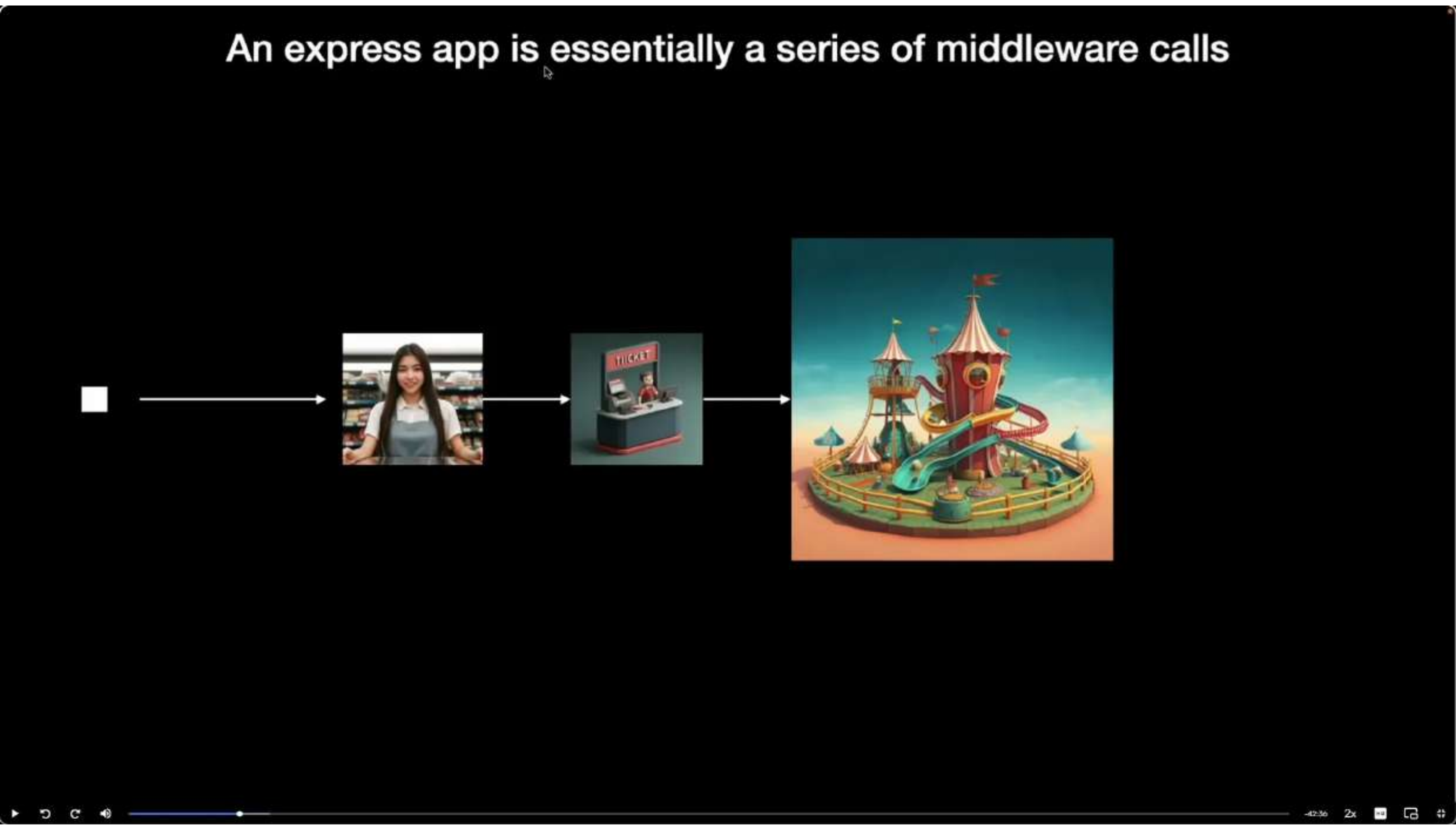
Express is a routing and middleware web framework that has minimal functionality of its own: An Express application is essentially a series of middleware function calls.

Middleware functions are functions that have access to the request object (`req`), the response object (`res`), and the next middleware function in the application's request-response cycle. The next middleware function is commonly denoted by a variable named `next`.

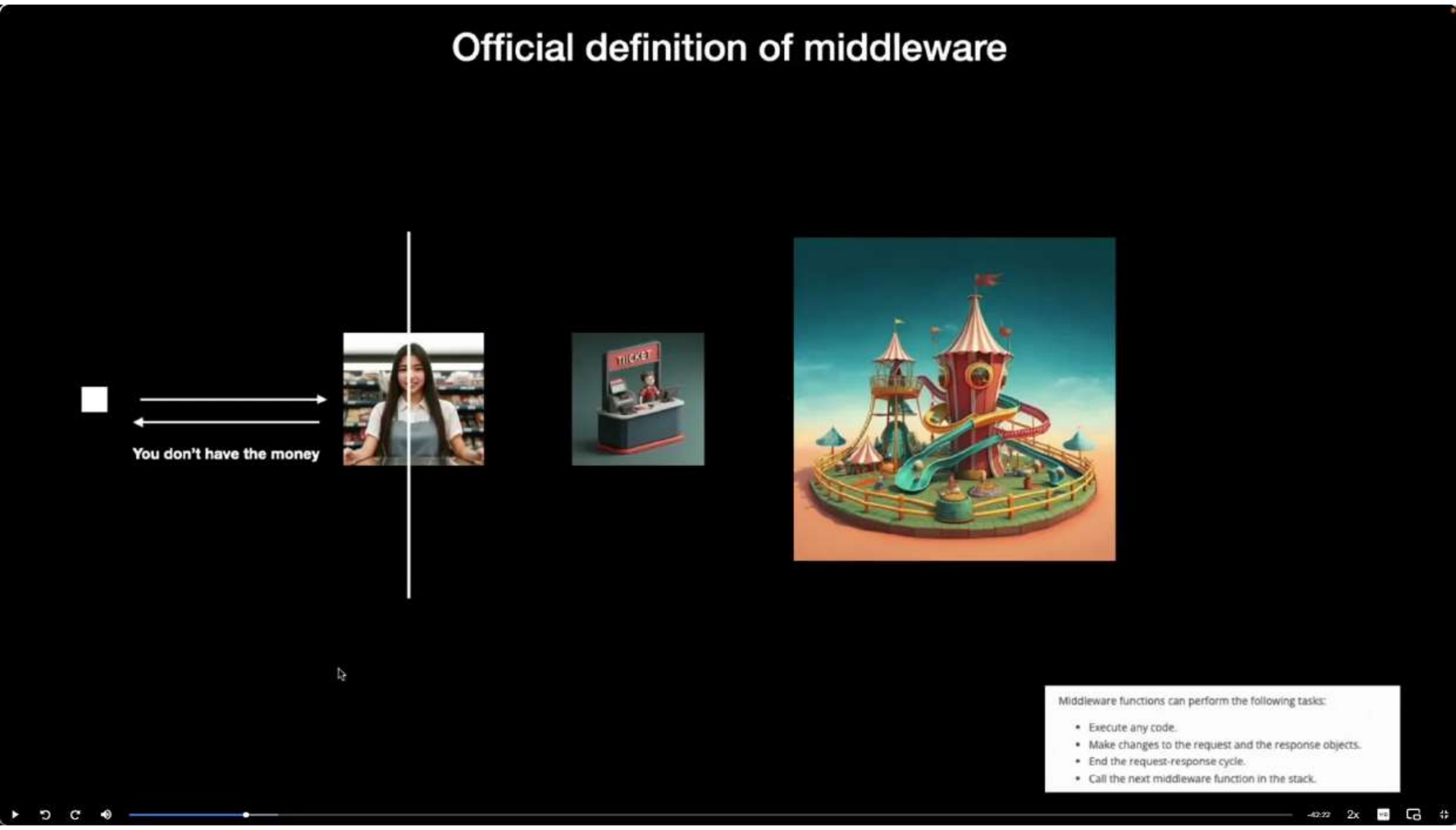
Middleware functions can perform the following tasks:

- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.

TICKET
Middleware



An express app is essentially a series of middleware calls

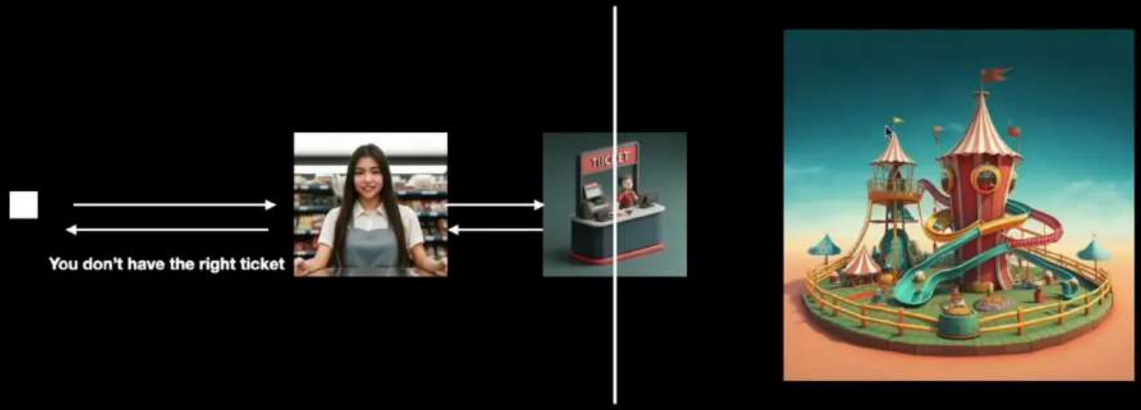


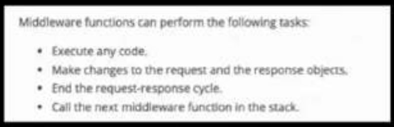
Official definition of middleware

Middleware functions can perform the following tasks:

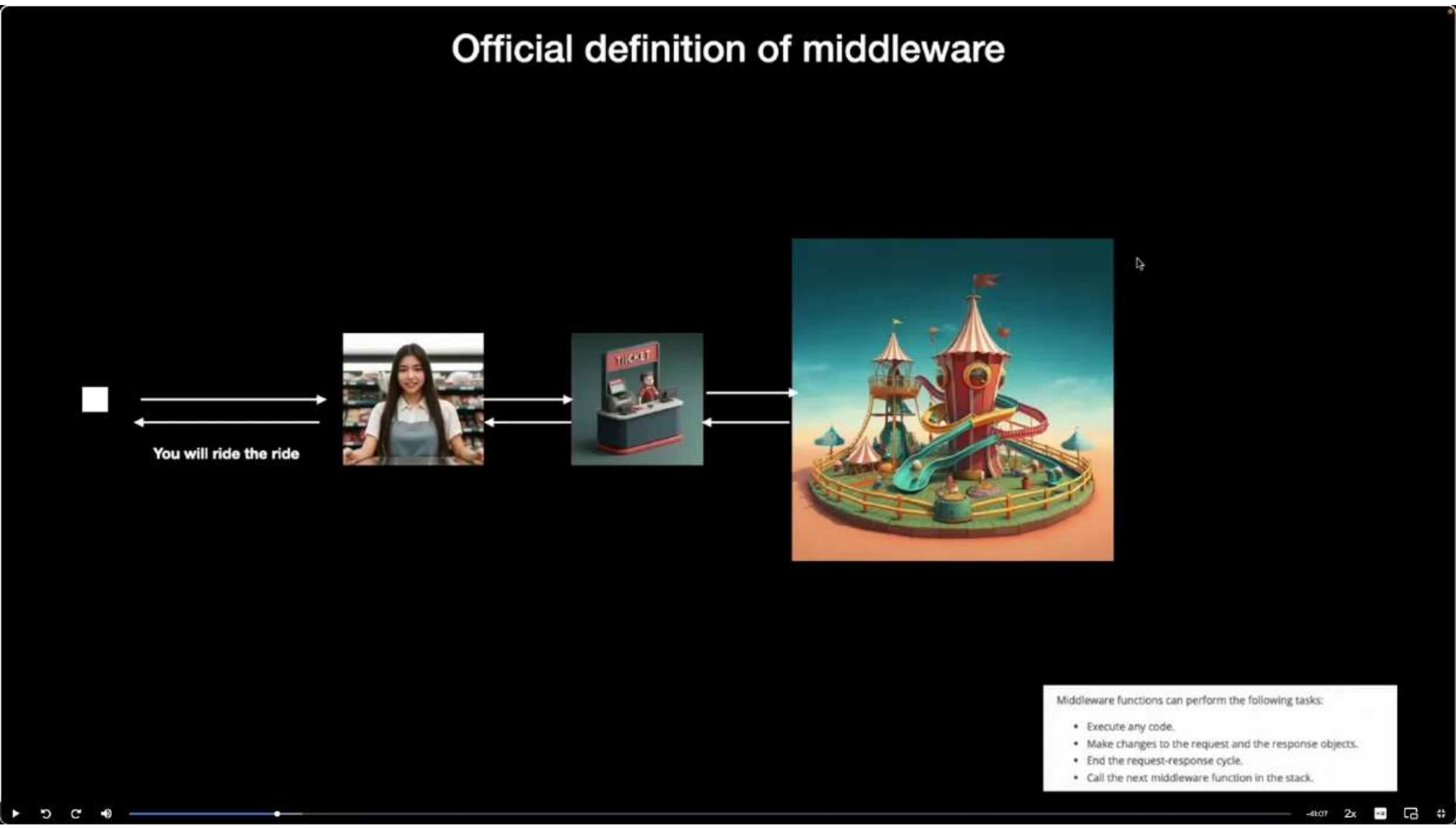
- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.

Official definition of middleware




Middleware functions can perform the following tasks:

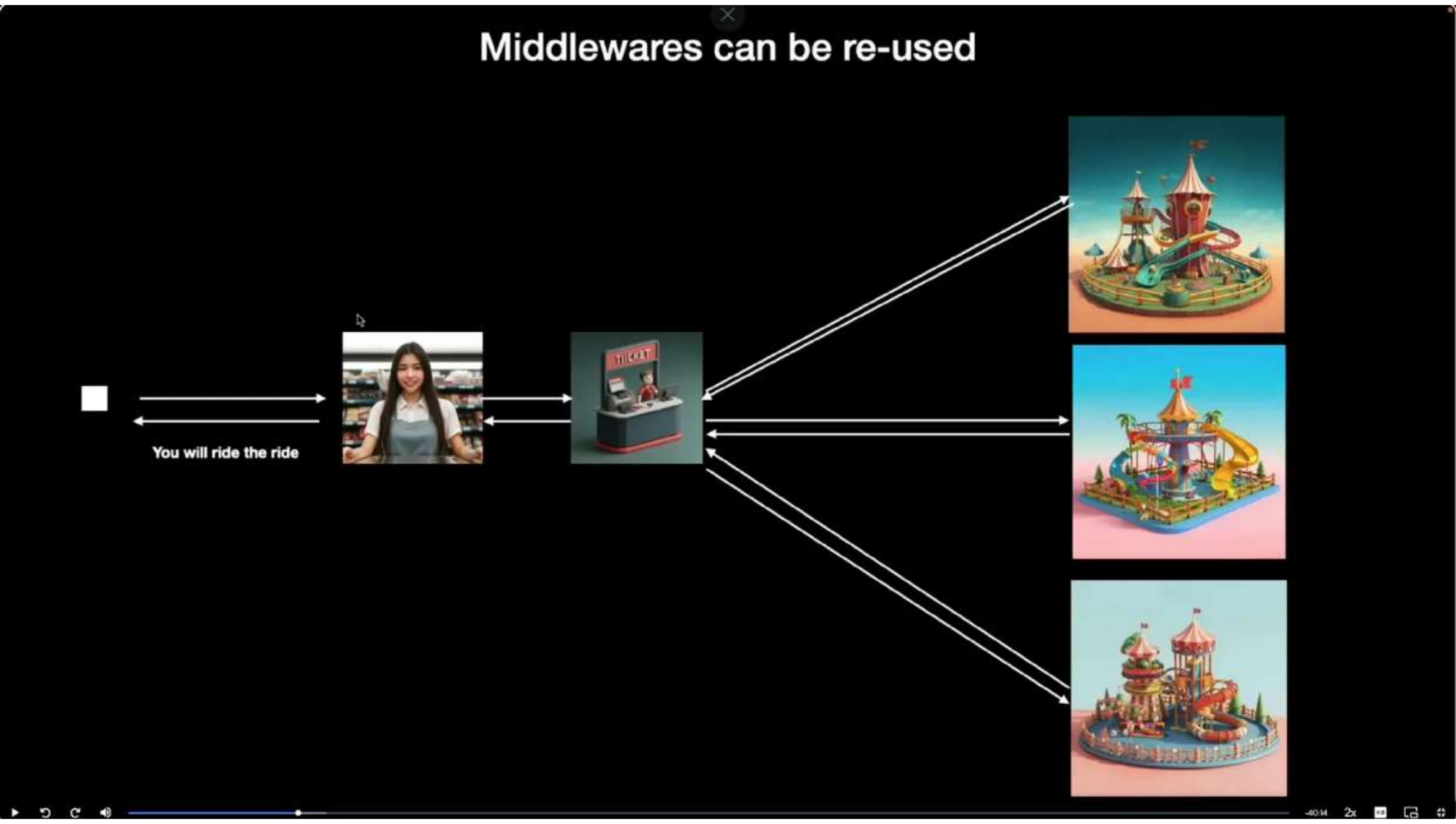
- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.



Official definition of middleware

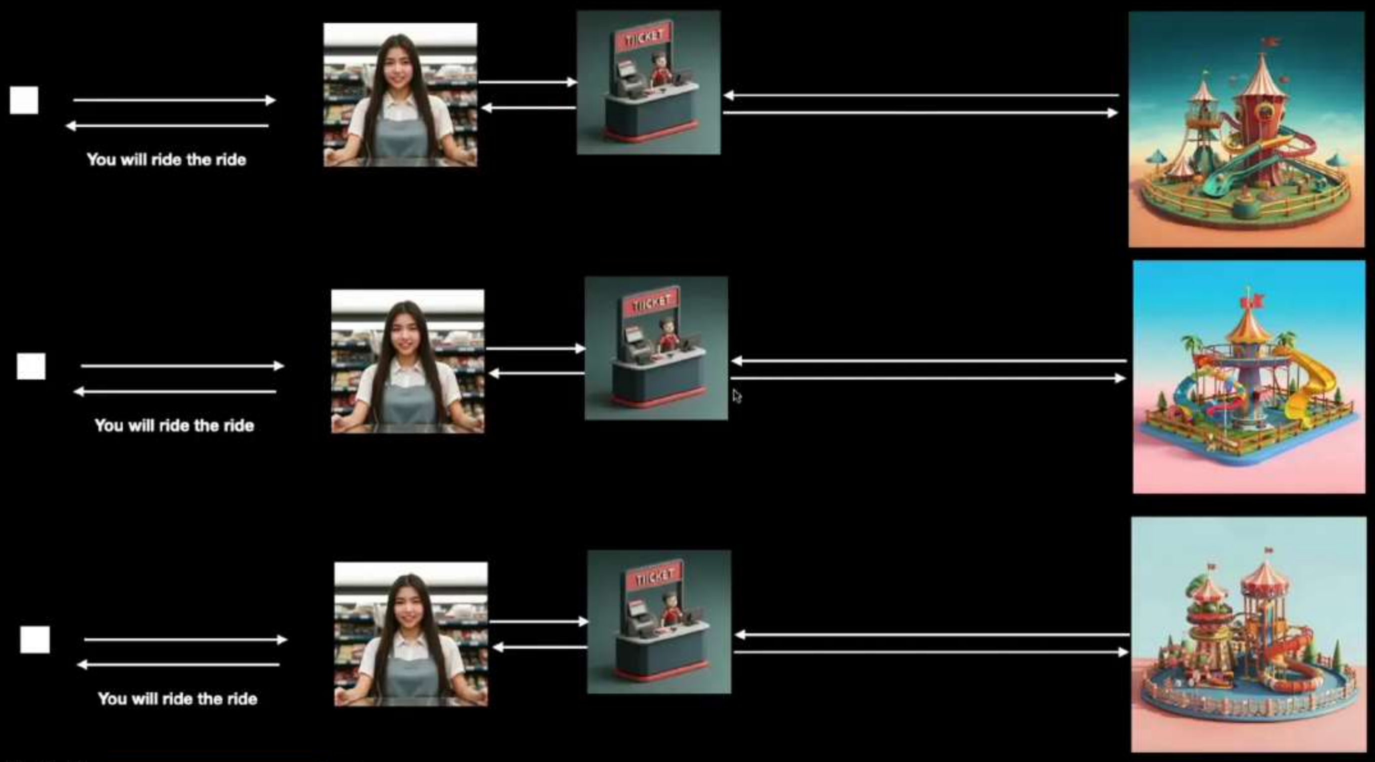
Middleware functions can perform the following tasks:

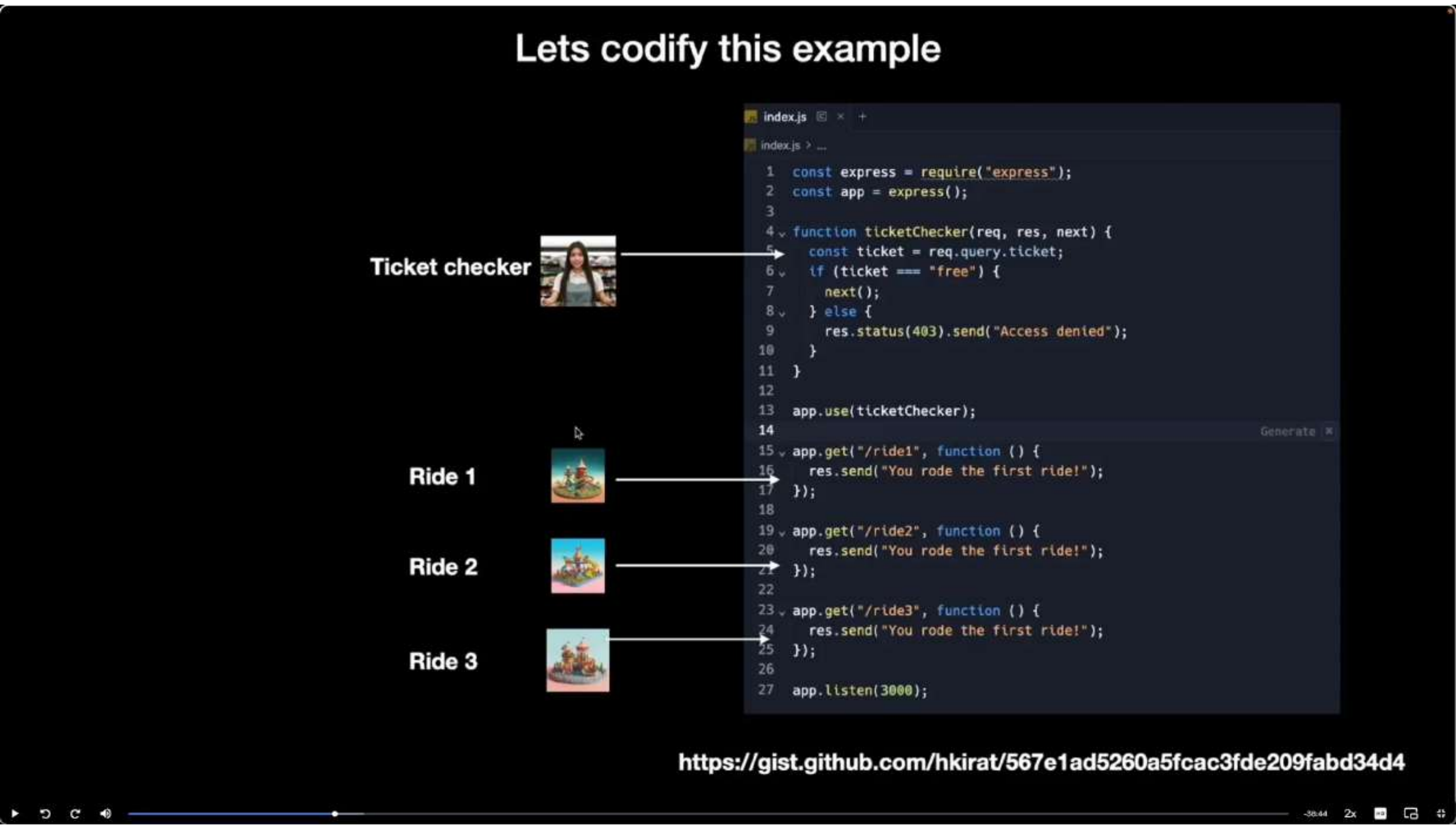
- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.



Middlewares can be re-used

This is bad (not re-using middleware)





Lets codify this example

```
const express = require("express");
const app = express();

function ticketChecker(req, res, next) {
  const ticket = req.query.ticket;
  if (ticket === "free") {
    next();
  } else {
    res.status(403).send("Access denied");
  }
}

app.use(ticketChecker);

app.get("/ride1", function () {
  res.send("You rode the first ride!");
});

app.get("/ride2", function () {
  res.send("You rode the first ride!");
});

app.get("/ride3", function () {
  res.send("You rode the first ride!");
});

app.listen(3000);
```

https://gist.github.com/hkirat/567e1ad5260a5fcac3fde209fabd34d4