DAY 2: INPUT AND OUTPUT

SOCKETS AND SERVERS



- Built on top of TCP
- Used by browsers to serve web pages



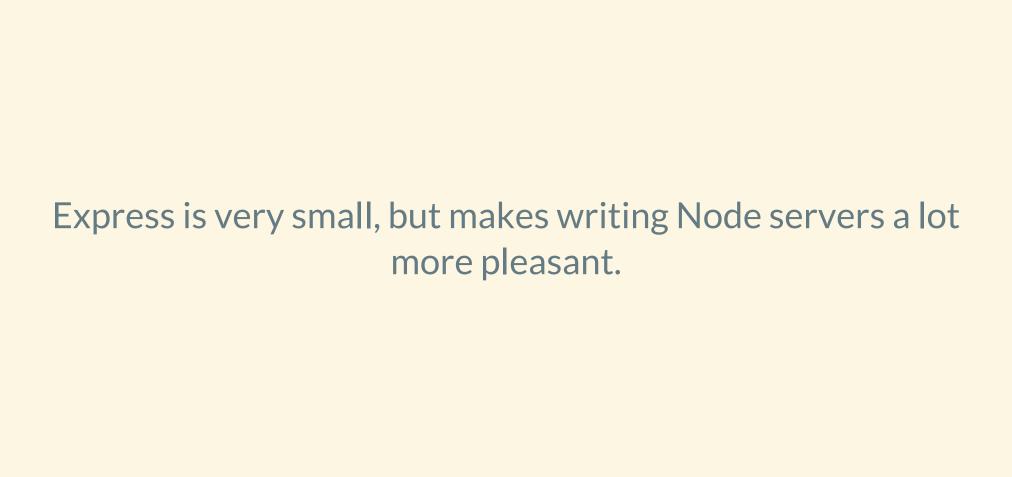
REQUEST/RESPONSE







Fast, unopinionated, minimalist web framework for Node.js



EXAMPLE: A SIMPLE EXPRESS SERVER

examples/day-2/01-express-hello-world

```
// Serves the current script's directory as a static server
app.use(express.static('public'));
```

EXAMPLE: USING STATIC-SERVE

examples/day-2/03-express-static-server

SOCKET.10

Realtime messaging between browsers

EXAMPLE: SETTING UP SOCKET.10

examples/day-2/04-socketio-setup

EXAMPLE: BROWSER TO SERVER

examples/day-2/05-socketio-send

EXAMPLE: BROADCASTING BROWSER TO BROWSER

examples/day-2/05-socketio-recieve

ASSIGNMENT #3

GIPHY CHAT

Using the examples provided, create a web-based chat server/client using Express and socket.io.

However, instead of sending text between clients, use the giphy translate API we experimented with on Tuesday to turn everybody's messages into GIFs.