



# CSc 337

## Node and Express

Benjamin Dicken

# Serve static files



- Can serve static files with express
  - HTML, CSS, Images, etc
- If you have a site that only needs static resources from the server, can just set up a simple static server!
  - Guitar World, Welcome to the Jumble, etc

# Static Server



```
const express = require('express');
const app = express();
const port = 80;

app.use(express.static('public_html'));

app.listen(port, () =>
  console.log(`App listening at http://localhost:${port}`));
```

# Run a local server

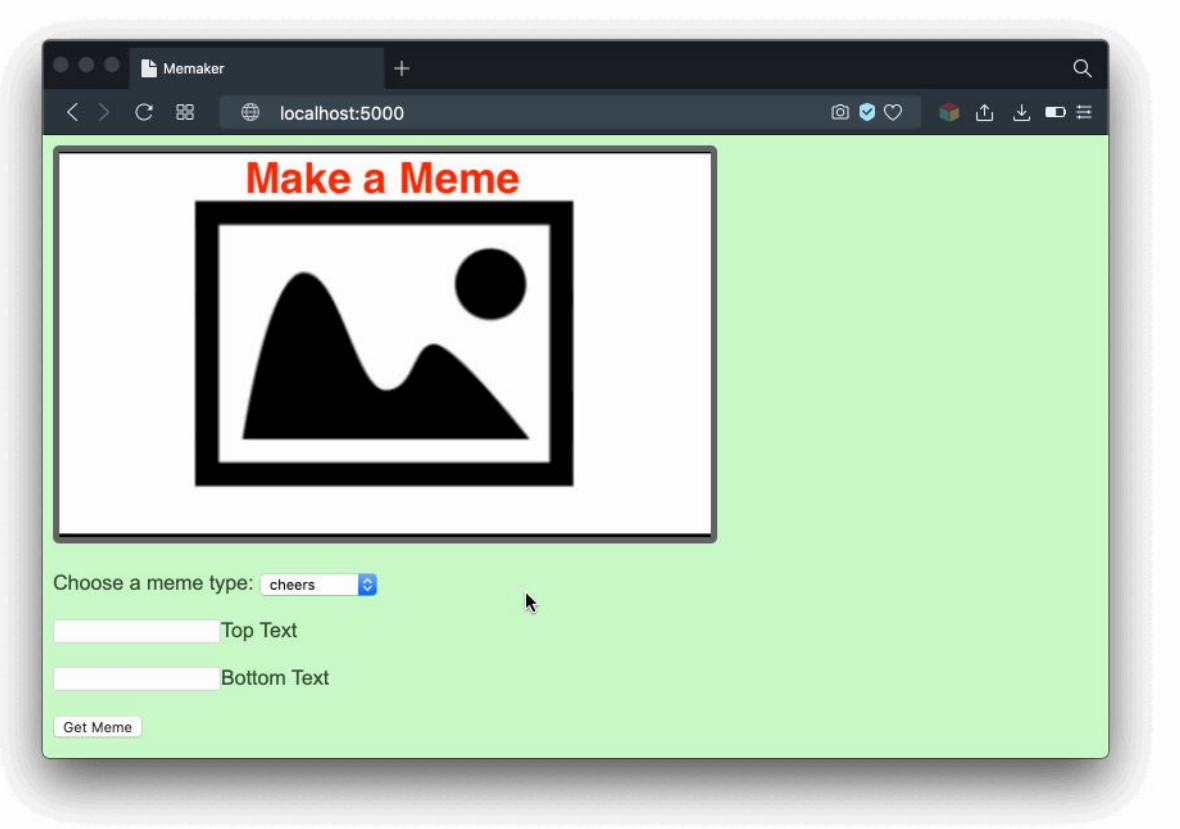
- Choose one of your past PAs (3, 4, or 5)
- Create a simple express static server on your computer to serve the PA files
- Run the server and connect

```
const express = require('express')
const app = express()
const port = 80
app.use(express.static('public_html'))
app.listen(port, () =>
  console.log(`App listening at http://localhost:${port}`))
```

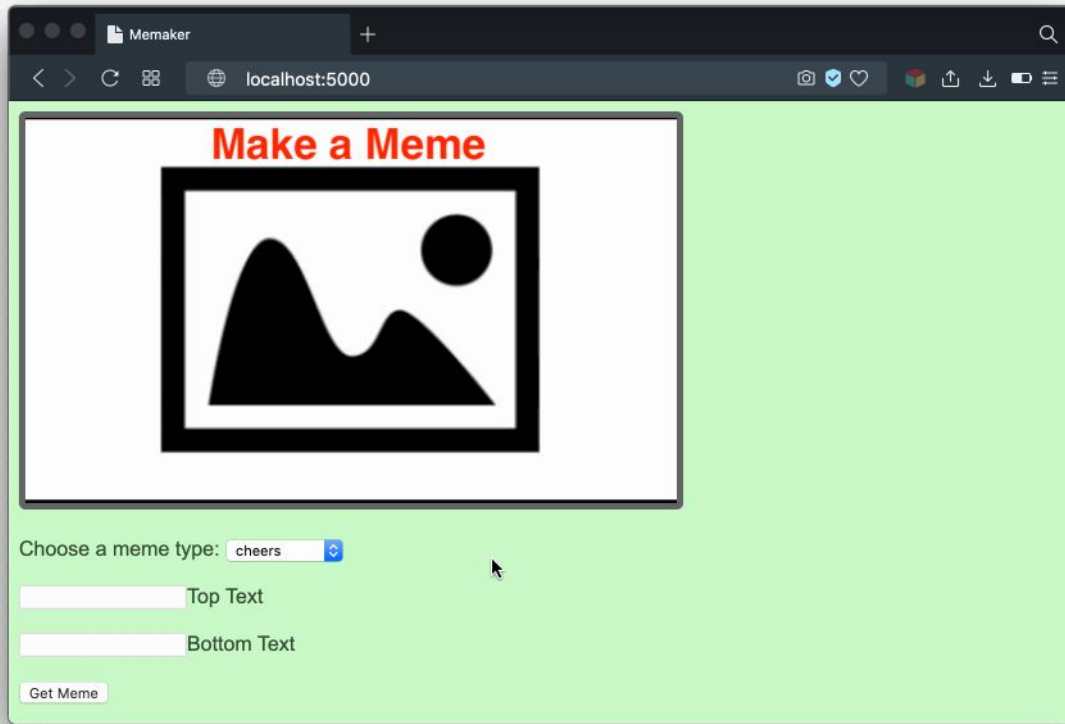
# Static + Dynamic

- Can combine static and dynamic capabilities in a single server
- For instance
  - ***Statically*** serve image resources, CSS, or client-side javascript
  - ***Dynamically*** generate html with custom content based on the URL

# Build the site

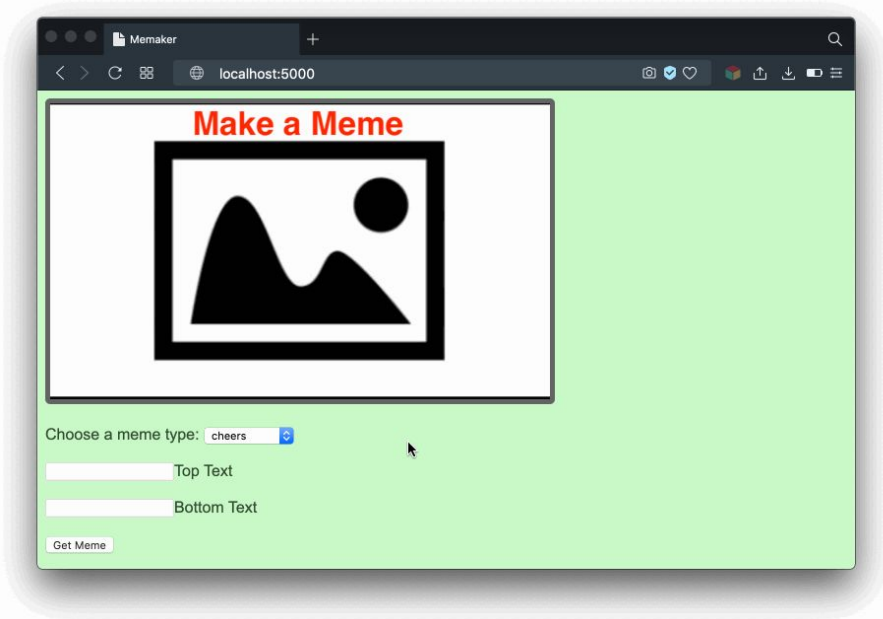


# Build the site



What will we need to create this?

# Build the site



- A combo of serving static files and dynamic response
- A module that we can use for the image processing
  - <https://www.npmjs.com/package/jimp>
- Express
- Client-side Js
- Client-side CSS
- Promises (optional)
  - <https://www.digitalocean.com/community/tutorials/understanding-javascript-promises>



```
jimp.read(inputFileName,
  function(err, loadedImage){
    if (err) { return console.log(err); }
    jimp.loadFont(jimp.FONT_SANS_64_WHITE,
      function(err, font){
        if (err) { return console.log(err); }
        loadedImage
          .print(font, 10, 10, topText)
          .print(font, 10, 326, bottomText)
          .write('./public_html/' + outputFileName ,
            function (err, data) {
              if (err) { return console.log(err); }
              fs.readFile('./public_html/index.html', 'utf8',
                function (err, data) {
                  if (err) { return console.log(err); }
                  var result = data.replace(/.\default.png/g, '/' + outputFileName);
                  res.send(result);
                });
            });
          });
    });
  });
});
```