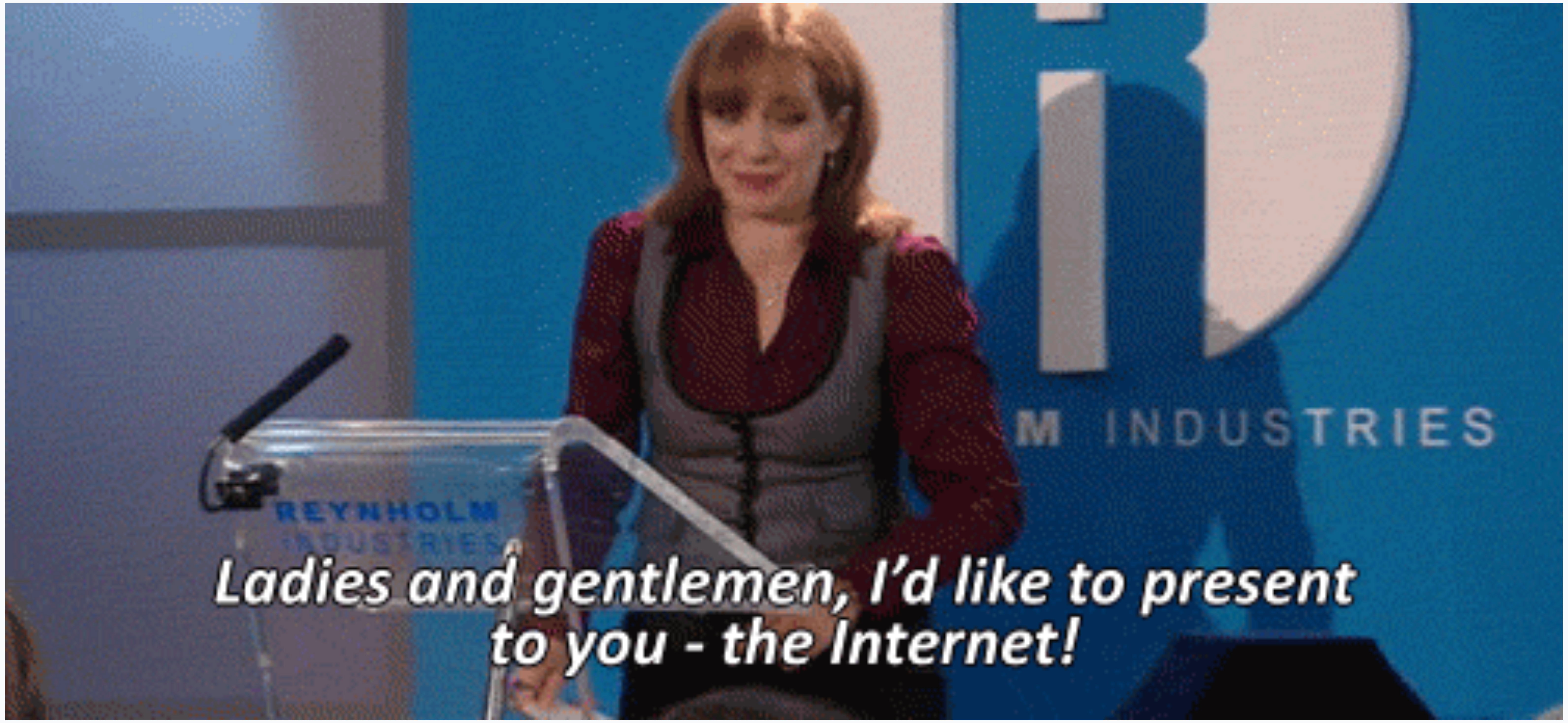
/express.js

“You think, you can”

GRACE HOPPER ACADEMY


REYNHOLM INDUSTRIES
M INDUSTRIES
Ladies and gentlemen, I'd like to present to you - the Internet!

internet communication

BitTorrent

AFP

TCP

HTTPS

POP

9P

IMAP

BGP

HTTP

UDP

SCP

SSL

ICMP

FTP

SSH

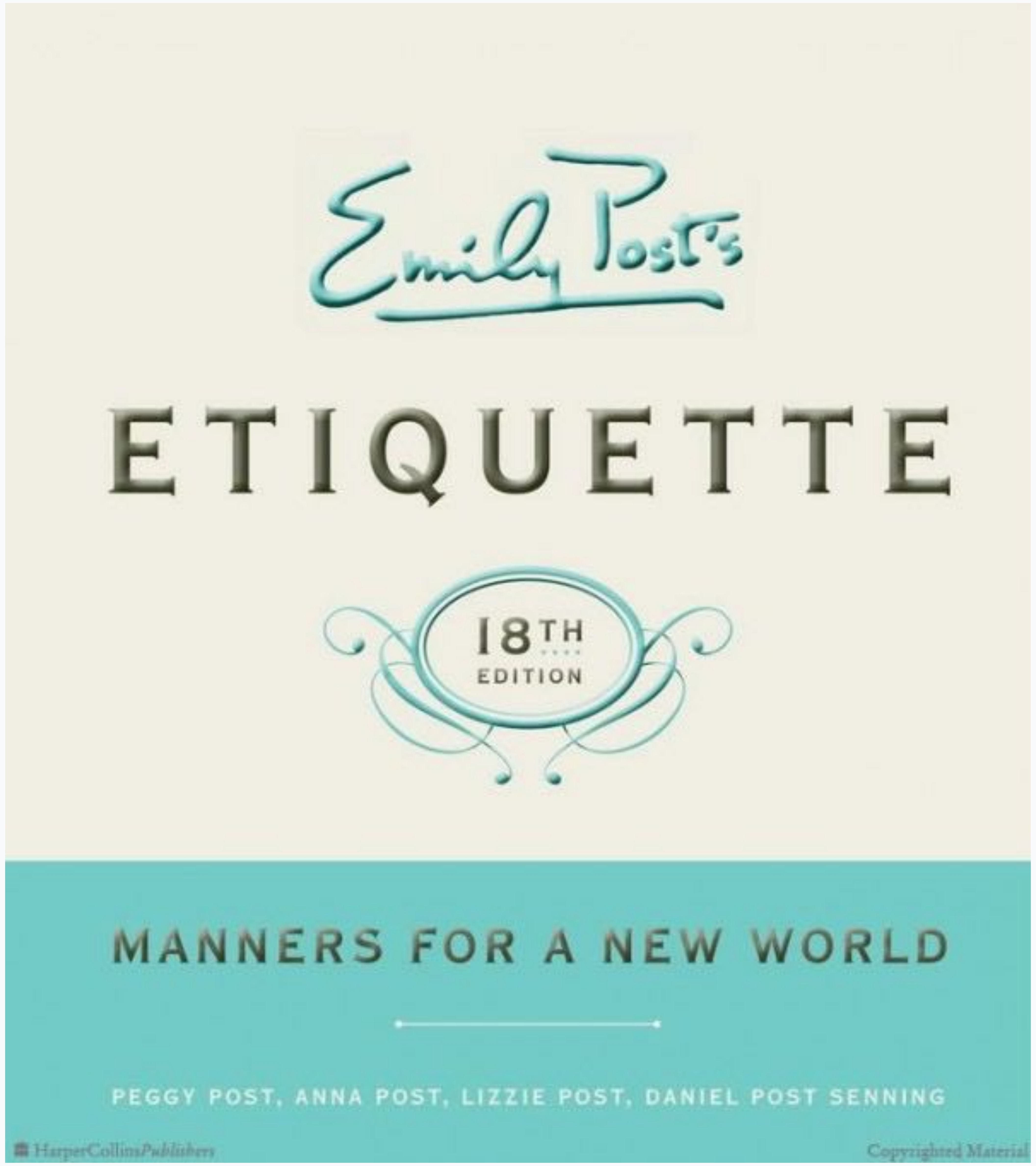
IP

PPP

NFS

SMTP

2006.04.12 13:43


Emily Post's

ETIQUETTE

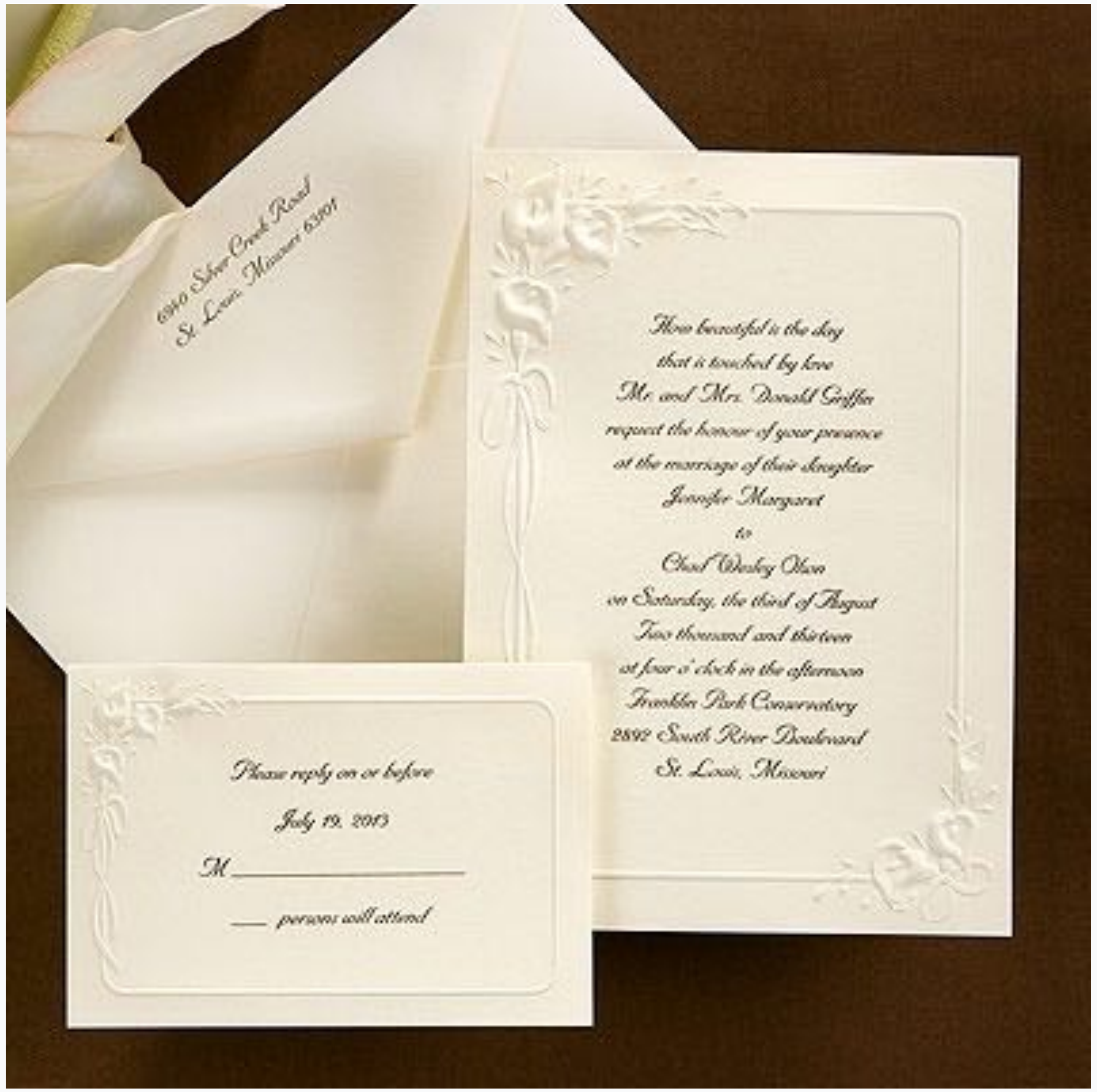
18TH EDITION

MANNERS FOR A NEW WORLD

PEGGY POST, ANNA POST, LIZZIE POST, DANIEL POST SENNING

HarperCollinsPublishers

Copyrighted Material


6940 Silver Creek Road
St. Louis, Missouri 63101

How beautiful is the day
that is touched by love
Mr. and Mrs. Donald Griffin
request the honour of your presence
at the marriage of their daughter
Jennifer Margaret
to
Chad Wesley Olson
on Saturday, the third of August
Two thousand and thirteen
at four o'clock in the afternoon
Franklin Park Conservatory
2892 South River Boulevard
St. Louis, Missouri

Please reply on or before
July 19, 2013
M__________________
___ persons will attend

GRACE HOPPER ACADEMY

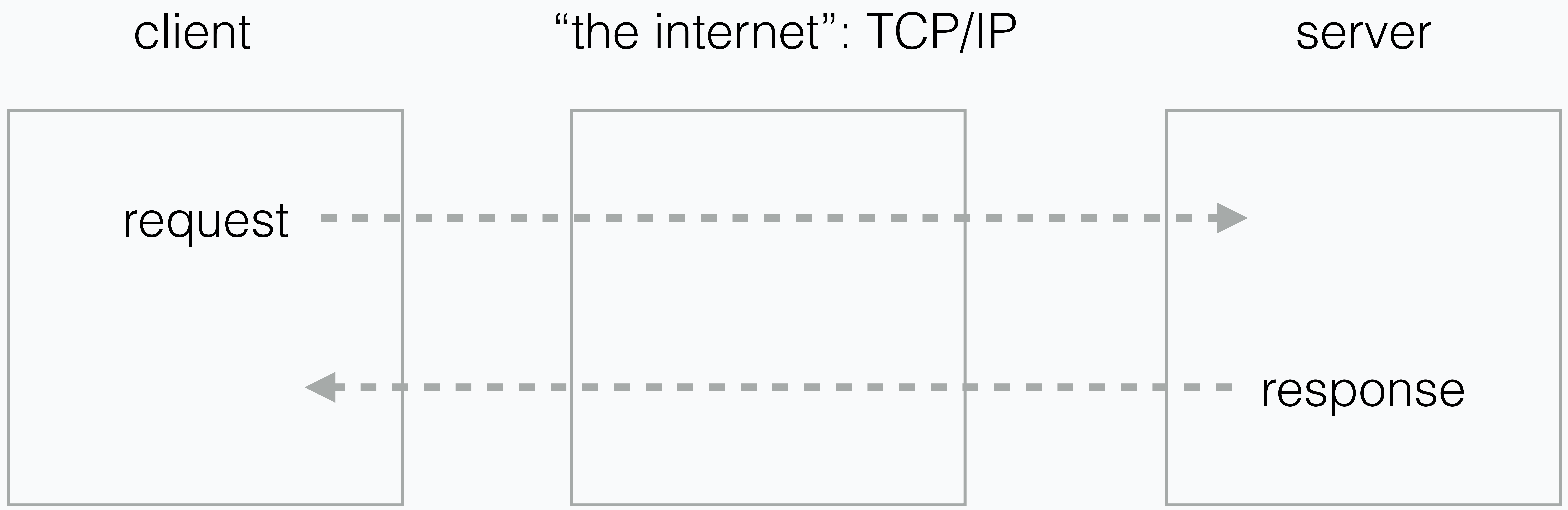
http

client

“the internet”: TCP/IP

server

http



http

Every request gets exactly one response

protocol

- Rules
- Specification NOT implementation
- Often used for communication

node's `http` library is an implementation of HTTP

http request

just a message with a certain format

verb URI

```
        POST /docs/1/related HTTP/1.1
            Host: www.test101.com
      Accept: image/gif, image/jpeg, */*
          Accept-Language: en-us
        Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

           bookId=12345&author=Nimit
```

headers

body

(from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)

Common verbs

GET	“read”
POST	“create”
PUT	“update”
DELETE	“delete”

http response

status

headers

```
HTTP/1.1 200 OK
Date: Sun, 18 Oct 2009 08:56:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Sat, 20 Nov 2004 07:16:26 GMT
ETag: "10000000565a5-2c-3e94b66c2e680"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html
X-Pad: avoid browser bug

<html><body><h1>It works!</h1></body></html>
```

payload/body

(from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)

Common statuses

200	“OK”
201	“created”
304	“cached”
400	“bad request”
401	“unauthorized”
404	“not found”
500	“server error”

express

A node library for request handling

<code>

</code>

express

- Treats requests as objects, created by event
- Matches on verb AND route
- Allows chaining of many handlers
- Enables modular layering with “routers”

pop quiz

client

Something that makes (HTTP) requests

server

Something that responds to (HTTP) requests

request

A formatted message sent over the network by a client. Contains VERB, URI (route), headers, and body.

response

A server’s reply to a request (formatted message).
Contains headers, payload, and status.

request-response cycle

The client **always** initiates by sending a request, and the server completes it by sending **exactly one** response

express middleware

A function that handles responding to requests.
Of the form: **`function(req, res, next){…}`**

request query string

A way to pass data from client to server.

verb route

headers

```
POST /docs/index?x=123&foo=that HTTP/1.1
Host: www.test101.com
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

bookId=12345&author=Nimit
```

body

In express…

`request.query = {x:123, foo: ‘that’}`

(from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)
(from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/)

request body

A way to pass data from client to server.

verb route

headers

```
POST /docs/index?x=123&foo=that HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
```

In express…

```
request.body = {bookId:12345, author: ‘Nimit’}
```

body

```
bookId=12345&author=Nimit
```

(from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)

request params

A variable portion of the URI.

router

A “layer” of route handlers (middlewares).

gotchas

- `app.get` v `app.get`
- routes are not file paths
- order matters
- `req.params` v `req.query` v `req.body`
- `app.use` v `app.all`

Workshop