

hello!





what value am I
adding?

Difficulties
moving from
just programming
to building
REAL WORLD
apps?

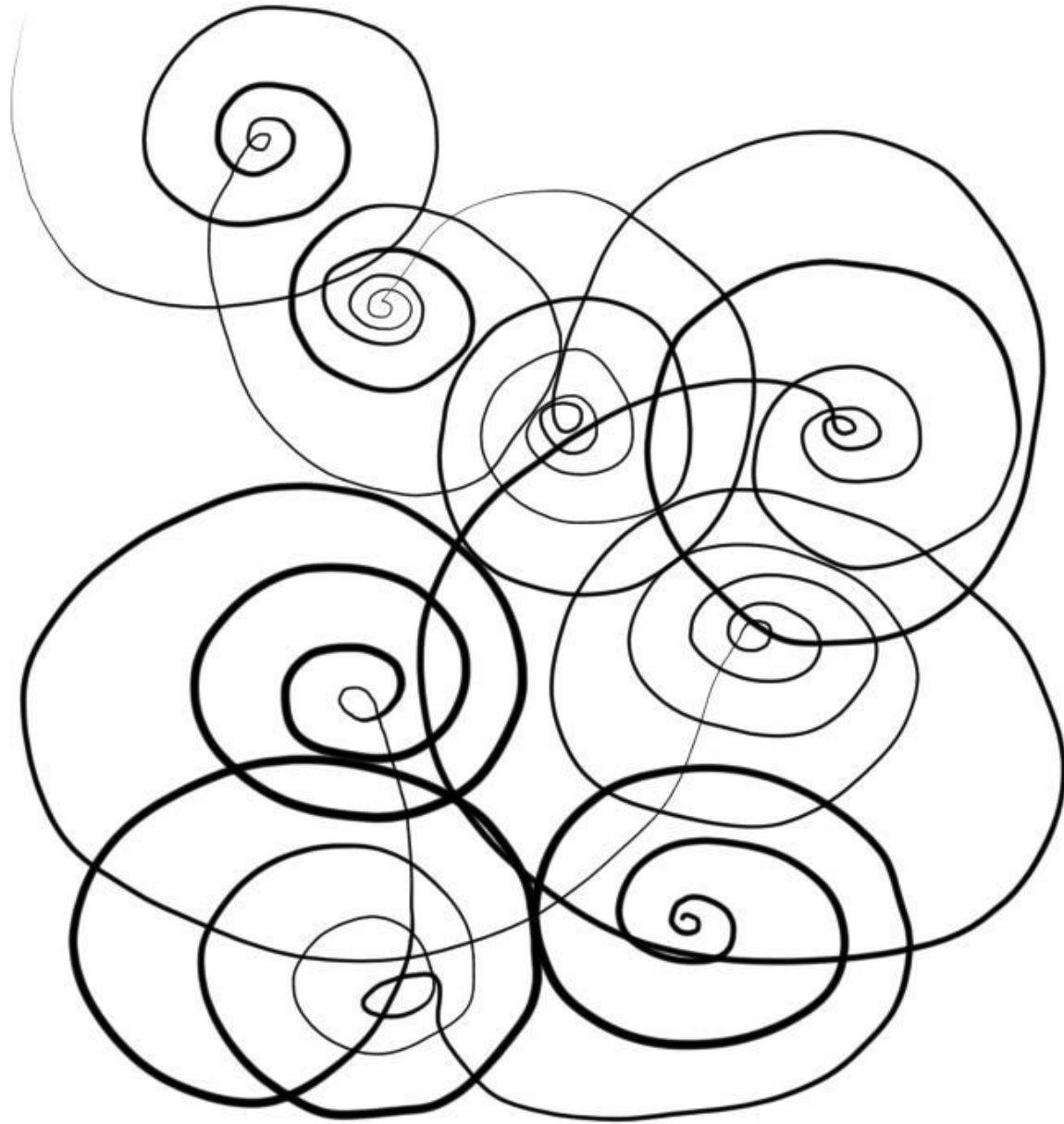


JavaScript Concurrency MVC
REST Design Patterns
DOM Dev Ops Event-Driven Programming
CSRF TDD OAuth Big Data
Performance Ruby UI Design
Memory Management Security
JSON API HTML Dynamic Languages
HTTP REST SSL CSS
Open Source Software

The CS cliff

credit: Godfrey Chan

Networking	Information Systems	Computer Vision	Animation
Operating System	Database	AI	Graphics
Data Structures	Algorithms	Statistics	Linear Algebra
Programming 101	Computer Architecture	Discrete Math	Calculus



Resources

BRACE YOURSELVES



FRUSTRATION IS COMING

" If I'm **frustrated** when coding, I'm teaching myself (unconsciously, of course) that *it's the correct way to feel when coding*, and inevitably I'm going to make less progress. "

Colin Jones: *Constant improvement does not require judgmental reflection*

<your favourite search engine>-fu

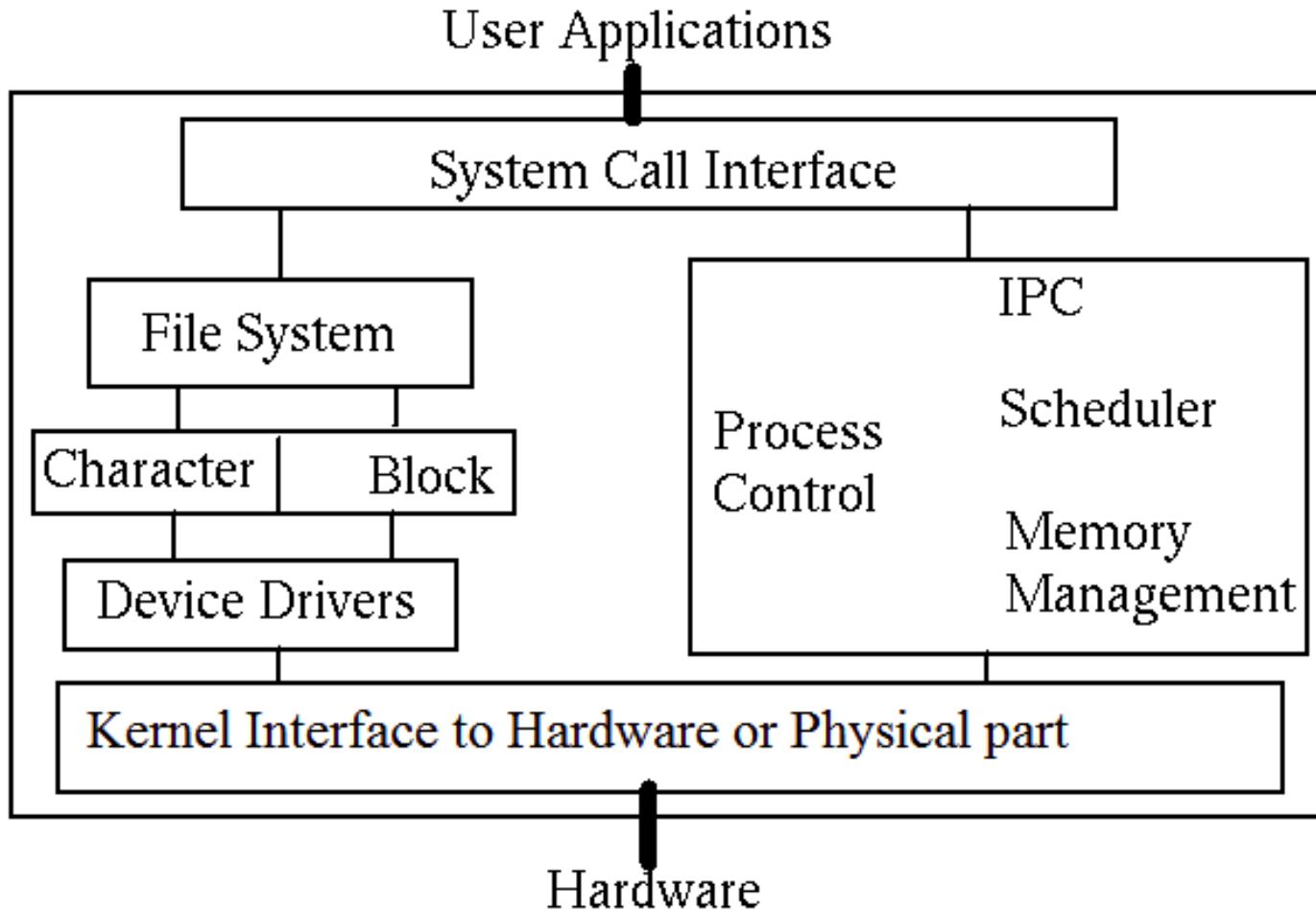
Teach to learn

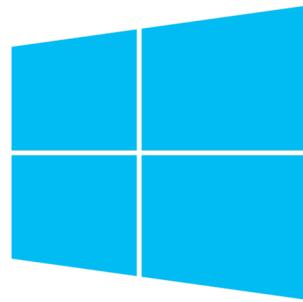
Your Dev Environment



ubuntu

Operating System





Windows

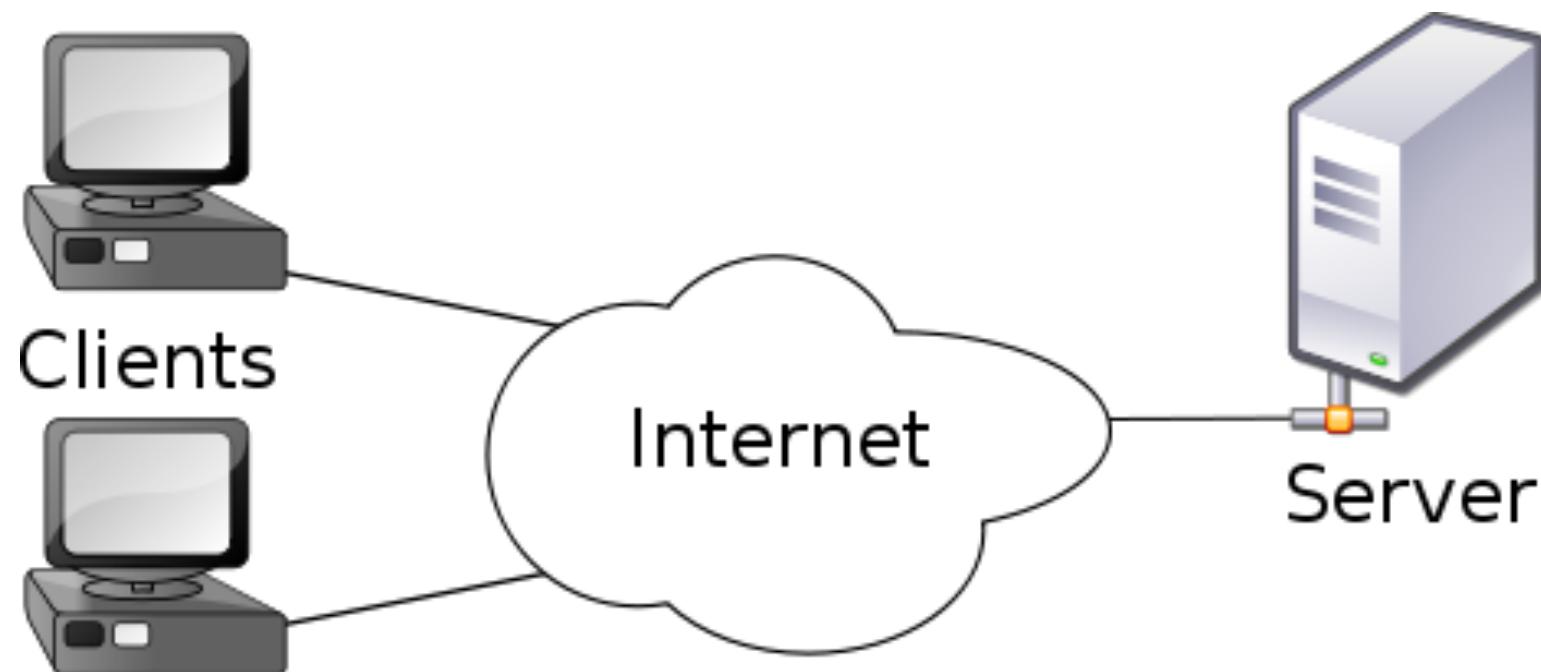




RTFM



curl



(domain name)

www.google.com



172.217.27.228

(IP address)

https://www.google.com.sg/webhp

(scheme)

(host)

(path)



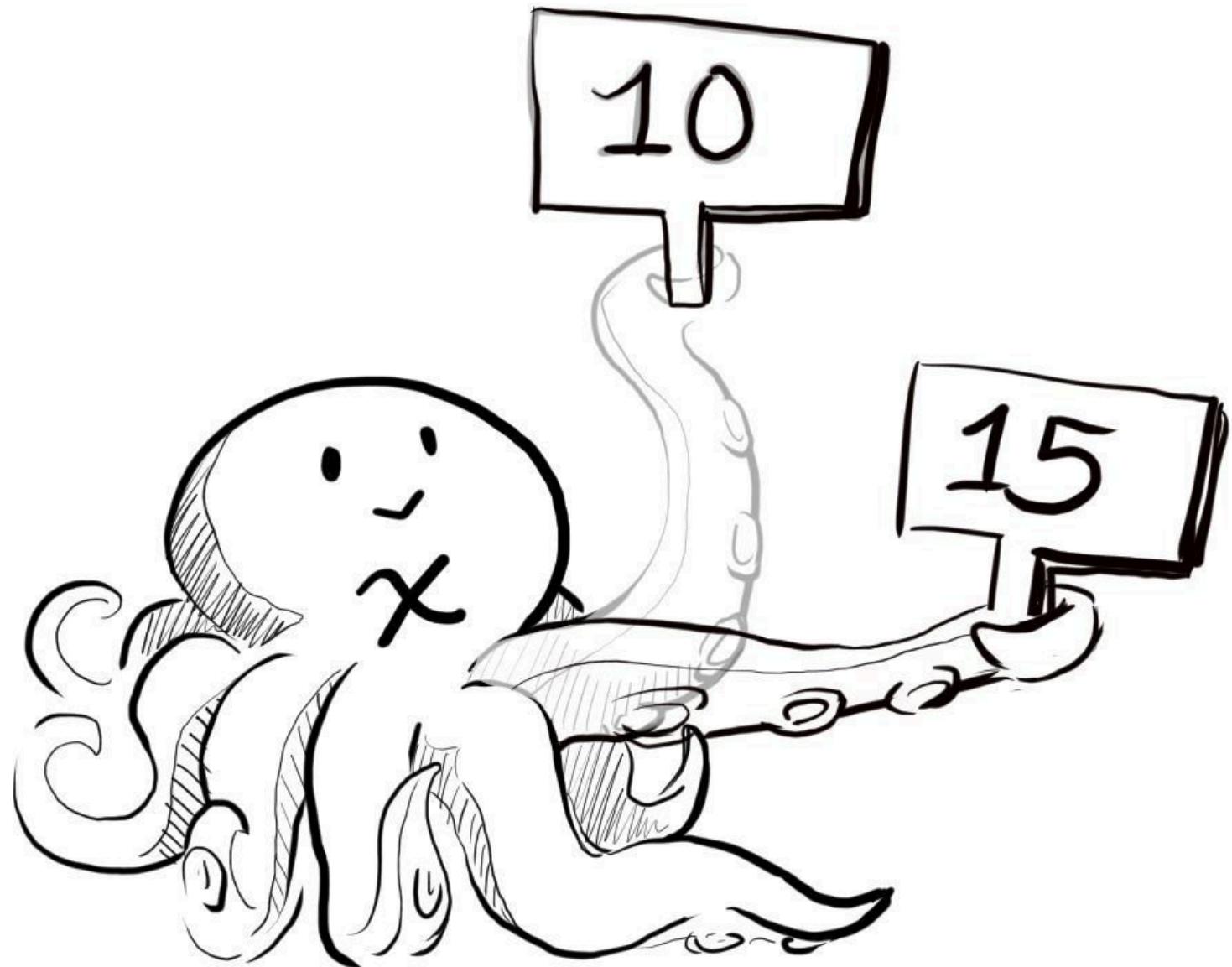
domain name



curl



ruby 101



$$x = 15$$



$x = \text{"Hello World"}$

$y = x$

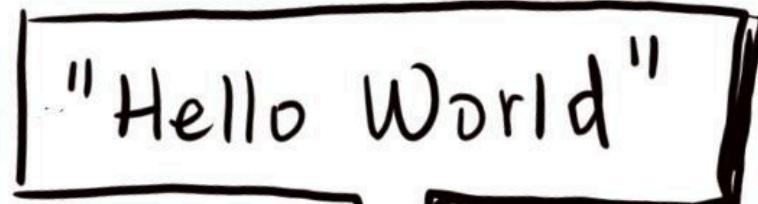
"Hello World".downcase

Memory Address 1



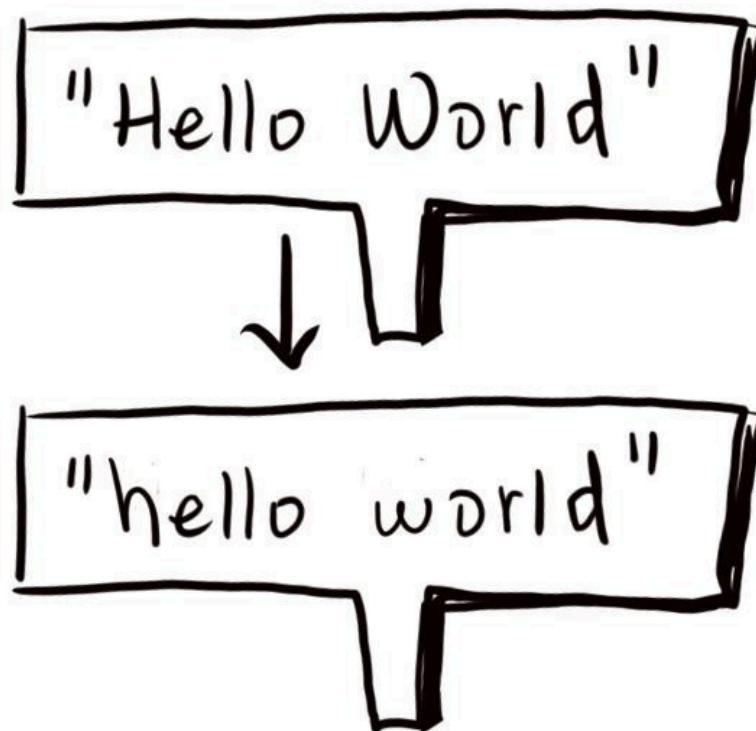
copy →

Memory Address 2

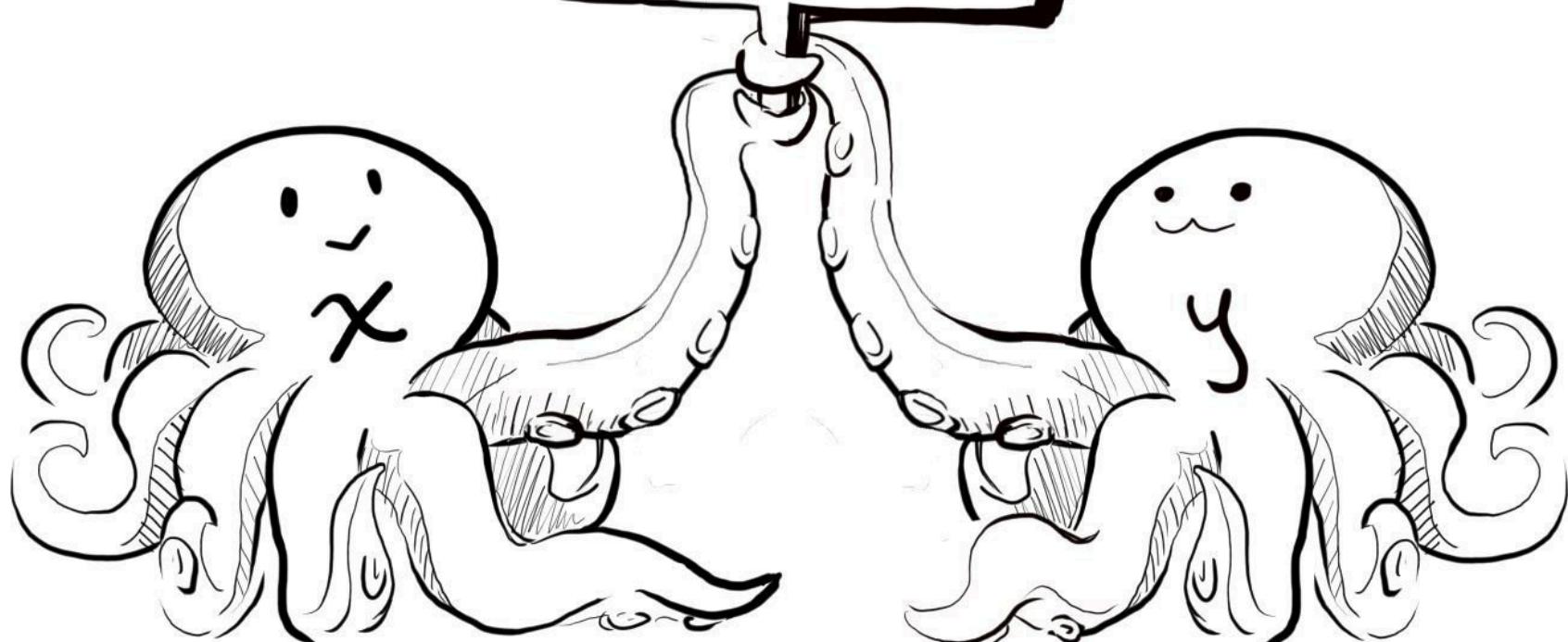
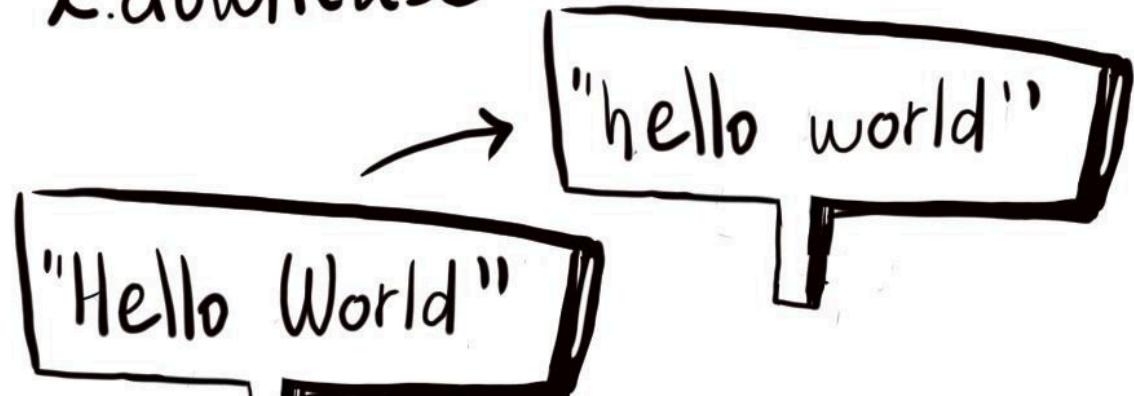


"Hello World".downcase!

Memory Address 1



$x.\text{downcase}$



$x = \text{"Hello World"}$

$y = x$

`x.downcase!`



`x = "Hello World"`

`y = x`

Class-based Object Oriented Programming

object instance



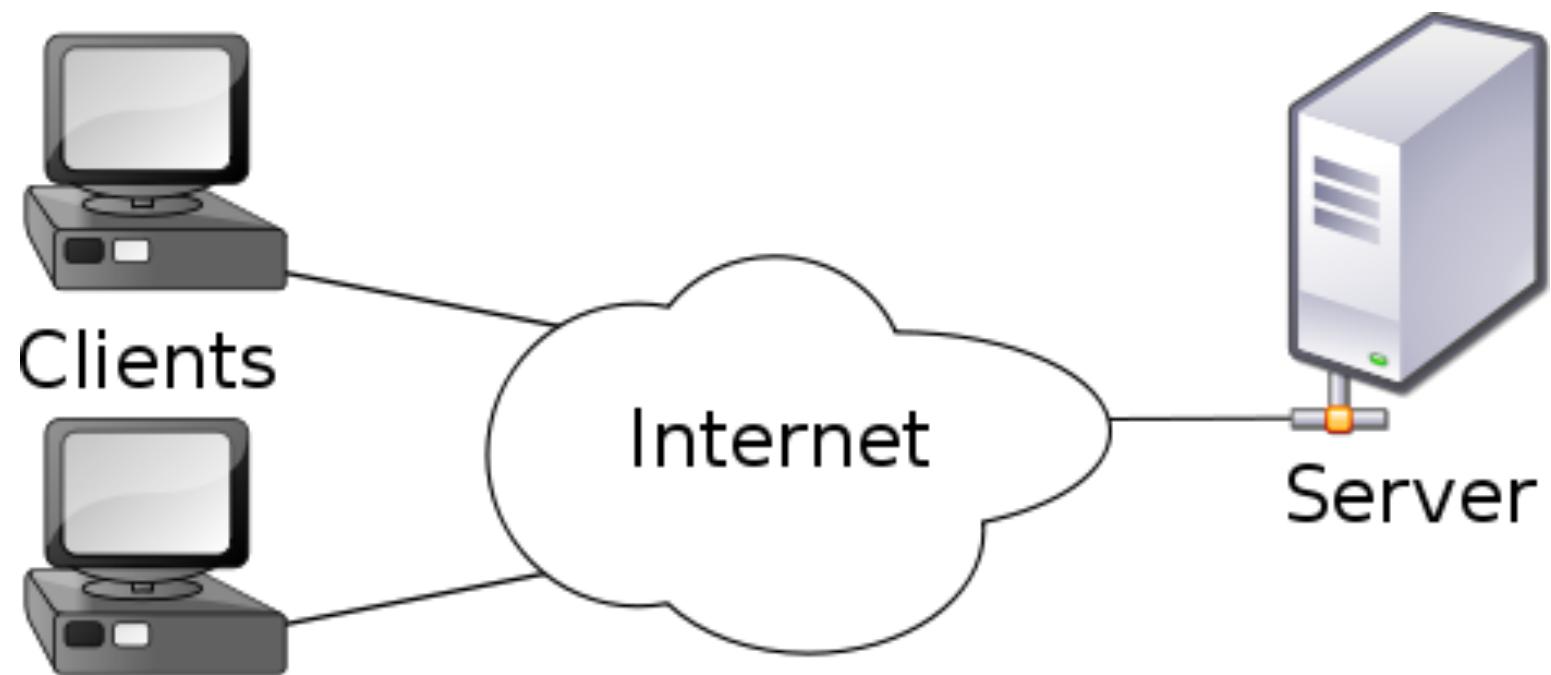
class



a simple web server

THE INTERNETZ

under 5 minutes



APPLICATION

HTTP

TRANSPORT

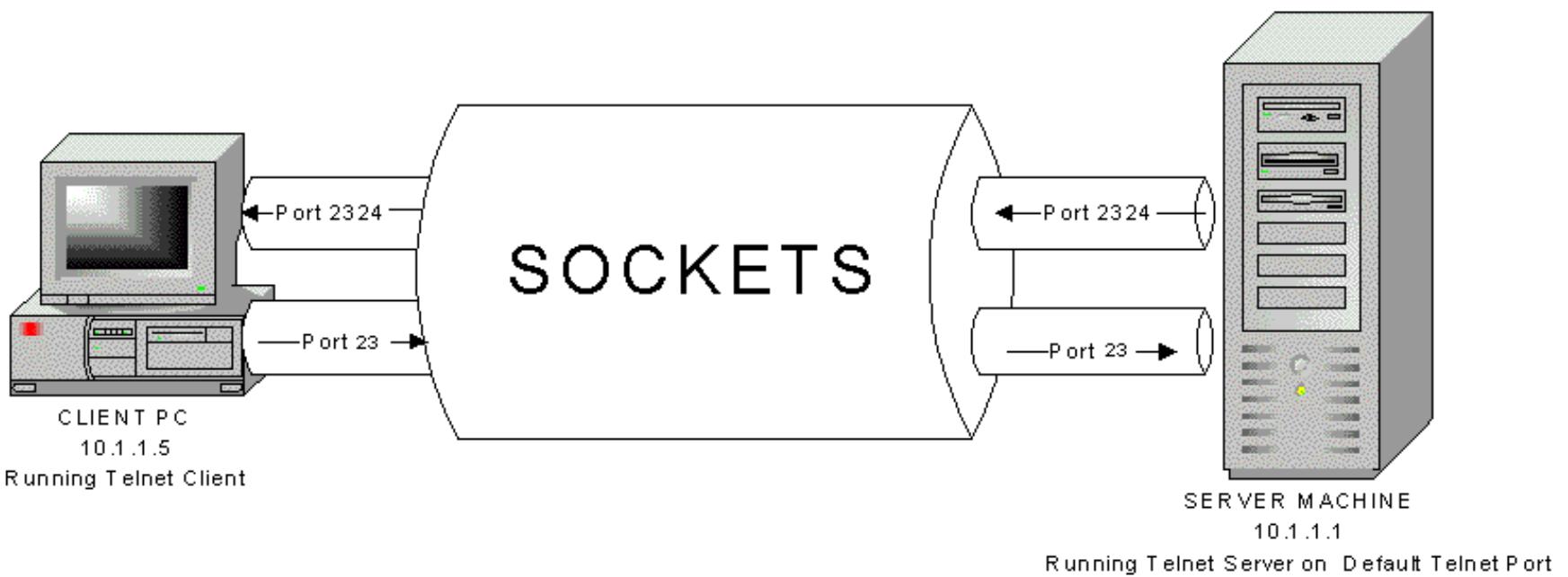
TCP

INTERNETWORK

IP

LINK & PHYSICAL

Ethernet
Wi-fi



Using Libraries

Ruby Standard Library (stdlib)

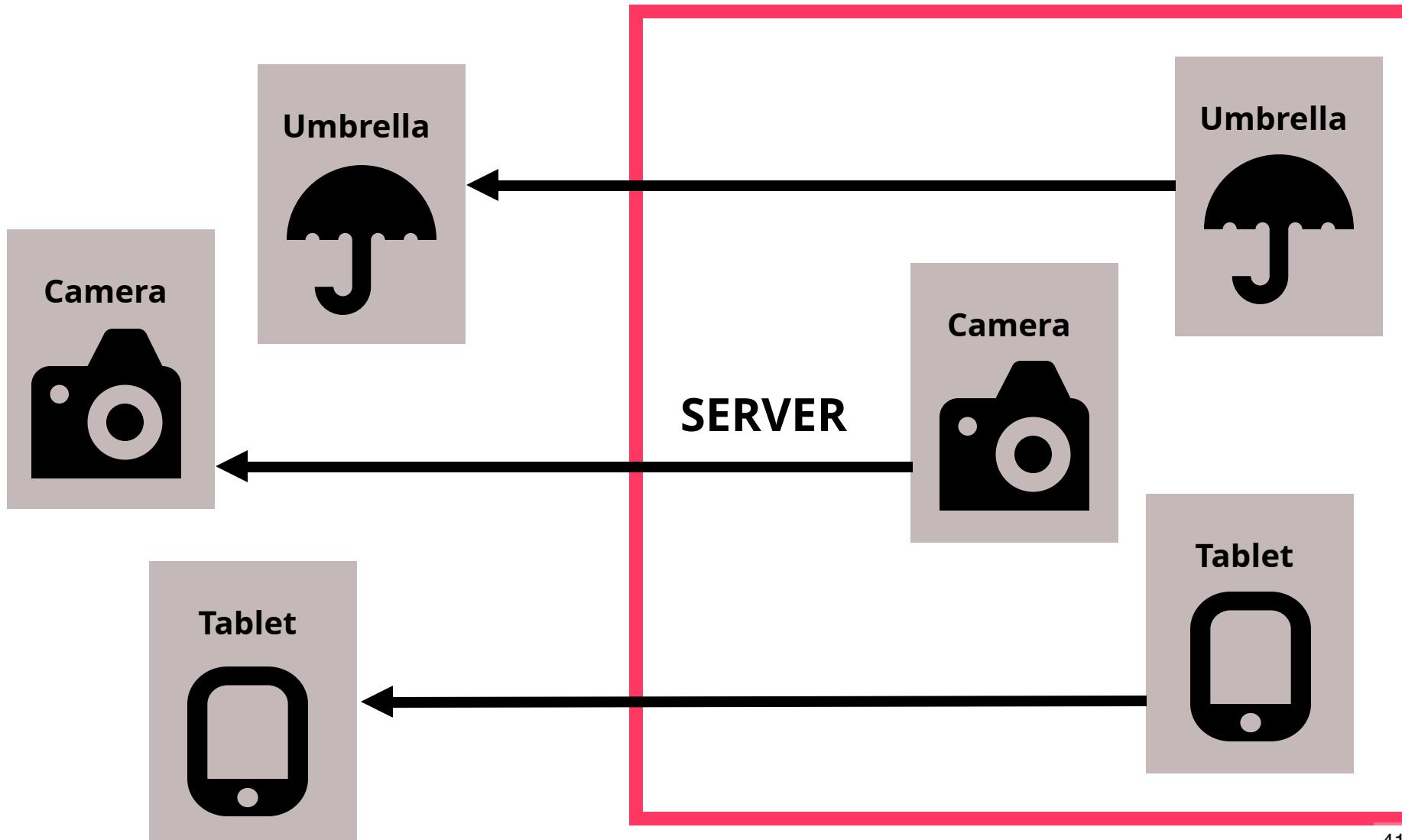


RTFM

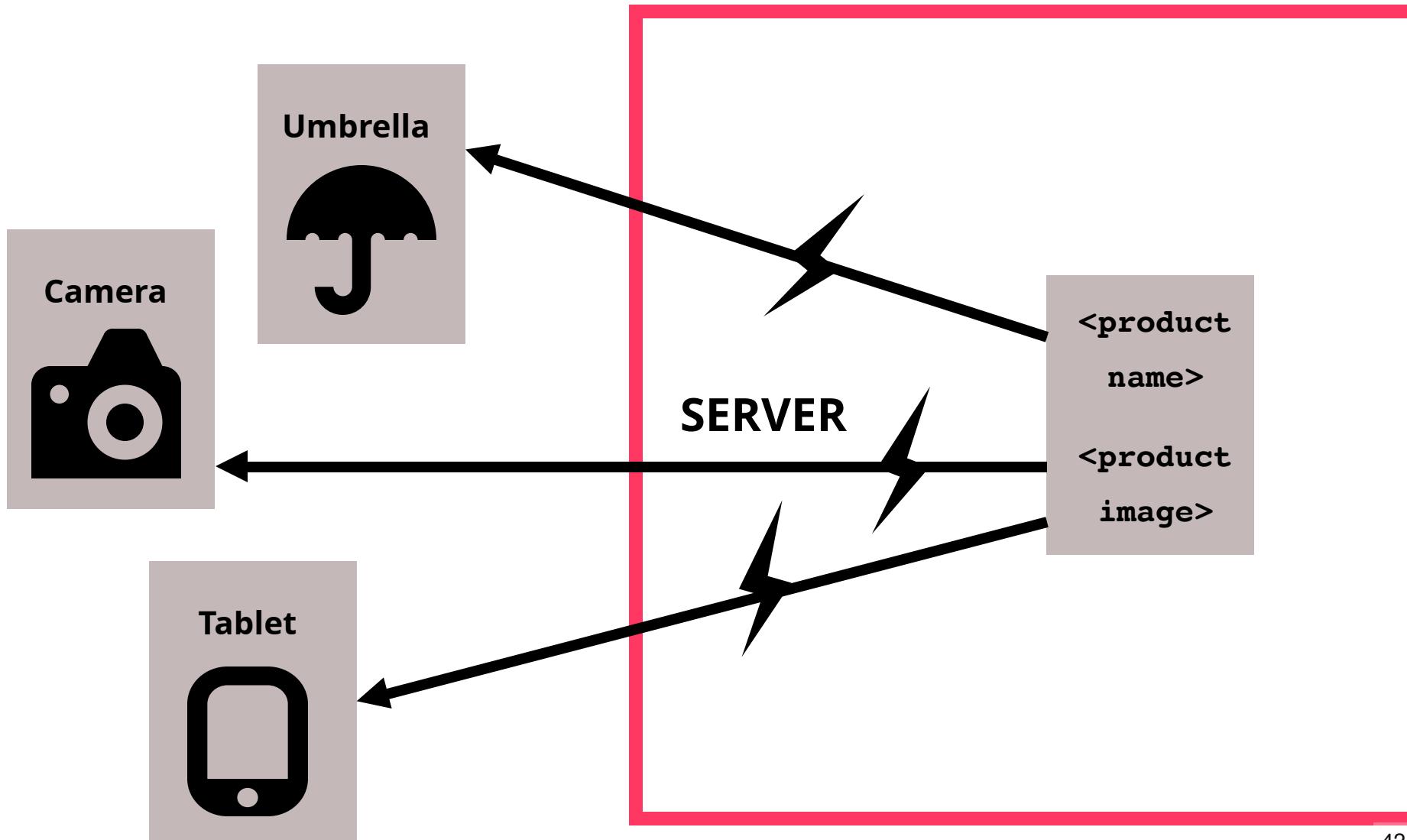
please

Serving Files

Static Content



Dynamic Content



REST



HTTP verbs

aka methods

GET

POST

PUT

DELETE

HTTP verb	Description	Typical Use
GET	Request a representation of the resource	Get a post/a list of posts
POST	Sends a body of data as a new resource for the server to store under the supplied URL	Create a new post
PUT	Sends a body of data to be stored under the supplied URL, replacing any existing resource	Update an existing post
DELETE	Tells the server to delete the specified resource	Delete a post

Request	Description
GET http://example.com/posts	Get a list of all posts
POST http://example.com/posts	Create a new post
GET http://example.com/posts/1	Get post with id 1
PUT http://example.com/posts/1	Update post with id 1
DELETE http://example.com/posts/1	Delete post with id

Storing Data

relational model

players table

name (varchar)	win_msg (varchar)	lose_msg (varchar)	wins (int)	losses (int)
Jun Qi	...	;(0	0
Kate	Aha!	Adoh!	0	0
Elisha	Cats!	Rats!	0	0
Wei-liang	Yesss	Noooo	0	0

IMPORTANT

Make sure you are executing all your terminal commands from the **exercises/simple_http_server** directory from now onwards. Or you will confuse yourself. Trust me

player record

name (varchar)	win_msg (varchar)	lose_msg (varchar)	wins (int)	losses (int)
Jun Qi	...	;)	0	0



ORM

Player object

```
class Player
  def win_game
    #...
  end

  def lose_game
    #...
  end
end
```

thank you!