[301] Web 1

Tyler Caraza-Harter

Learning Objectives Today

Network basics

- IP addresses
- host/domain names
- client/server and request/response

HTTP basics

- URLs
- GET/POST/etc
- headers
- status codes

Requests modules

- downloading data with requests.get
- remote calls with requests.post

Learning Objectives Today

Motivation

Networking Basics

HTTP (Hypertext Transfer Protocol)

Requests Module

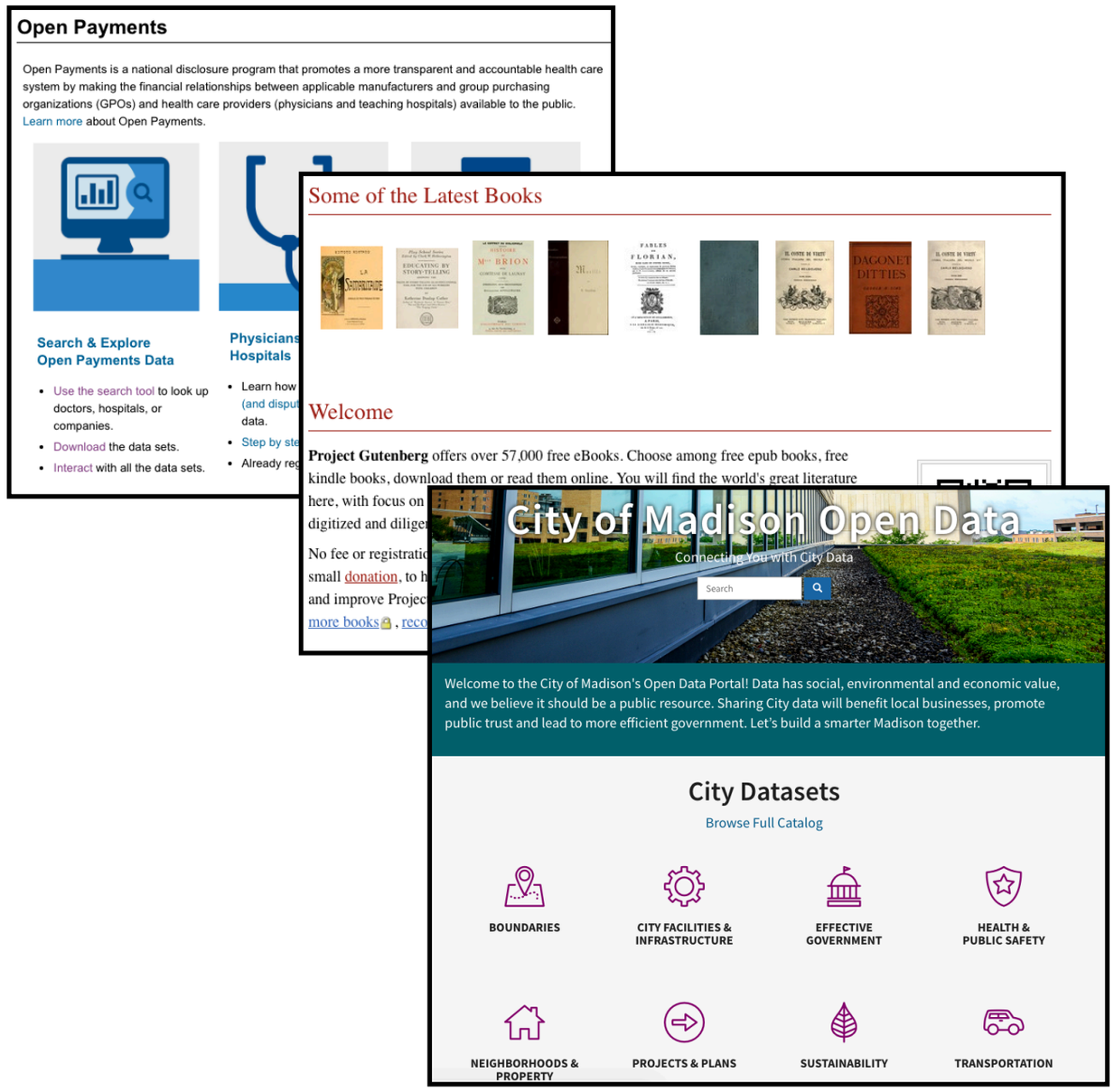
Data Science and the Internet

There are tons of online sources of data

- Examples: https://tyler.caraza-harter.com/cs301/spring19/datasets.html

Wide range of topics

- healthcare
- roads and city planning
- astronomy
- population
- business
- entertainment
- education
- etc



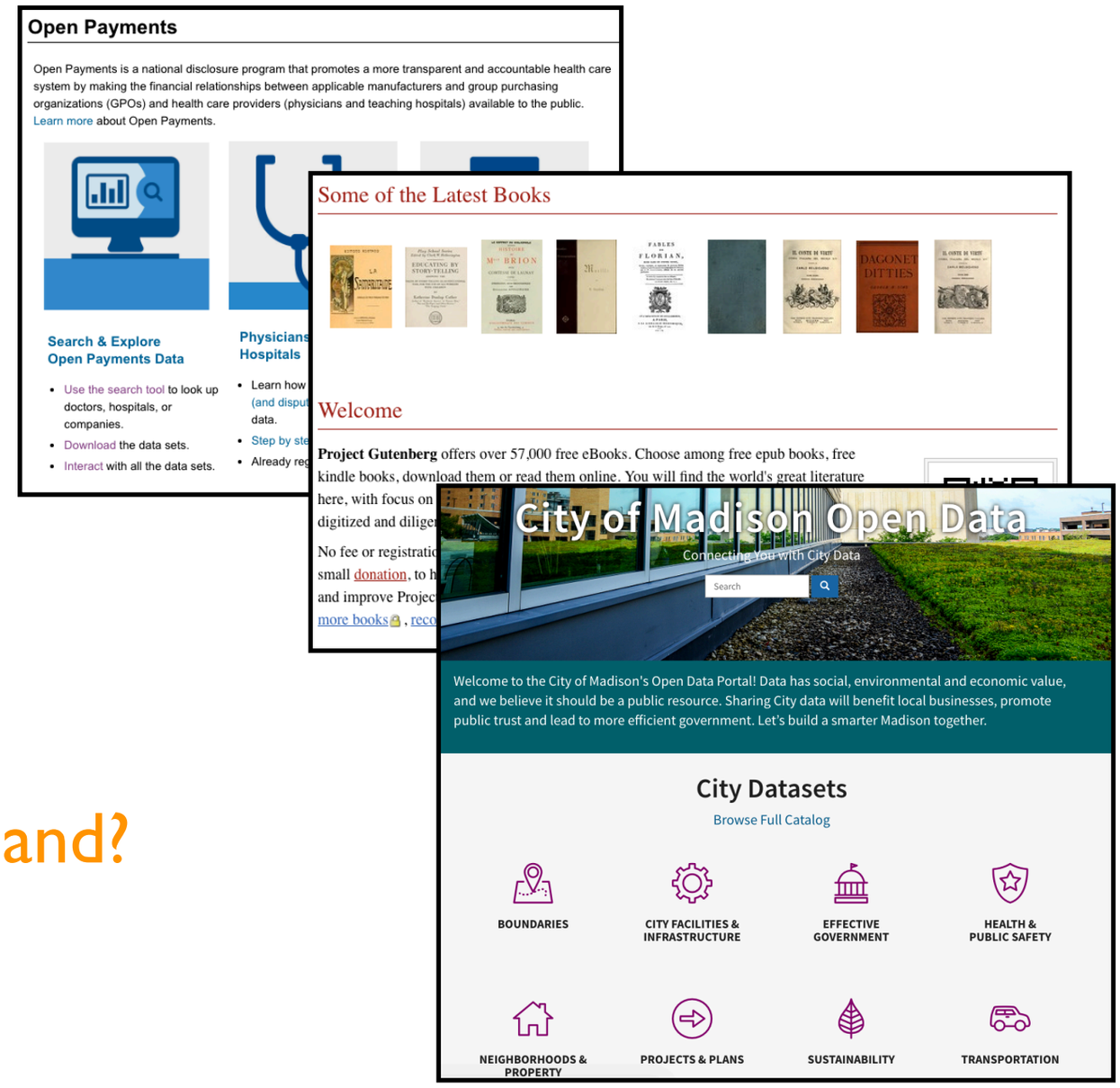
Data Science and the Internet

There are tons of online sources of data

- Examples: https://tyler.caraza-harter.com/cs301/spring19/datasets.html

Wide range of topics

- healthcare
- roads and city planning
- astronomy
- population
- business
- entertainment
- education
- etc

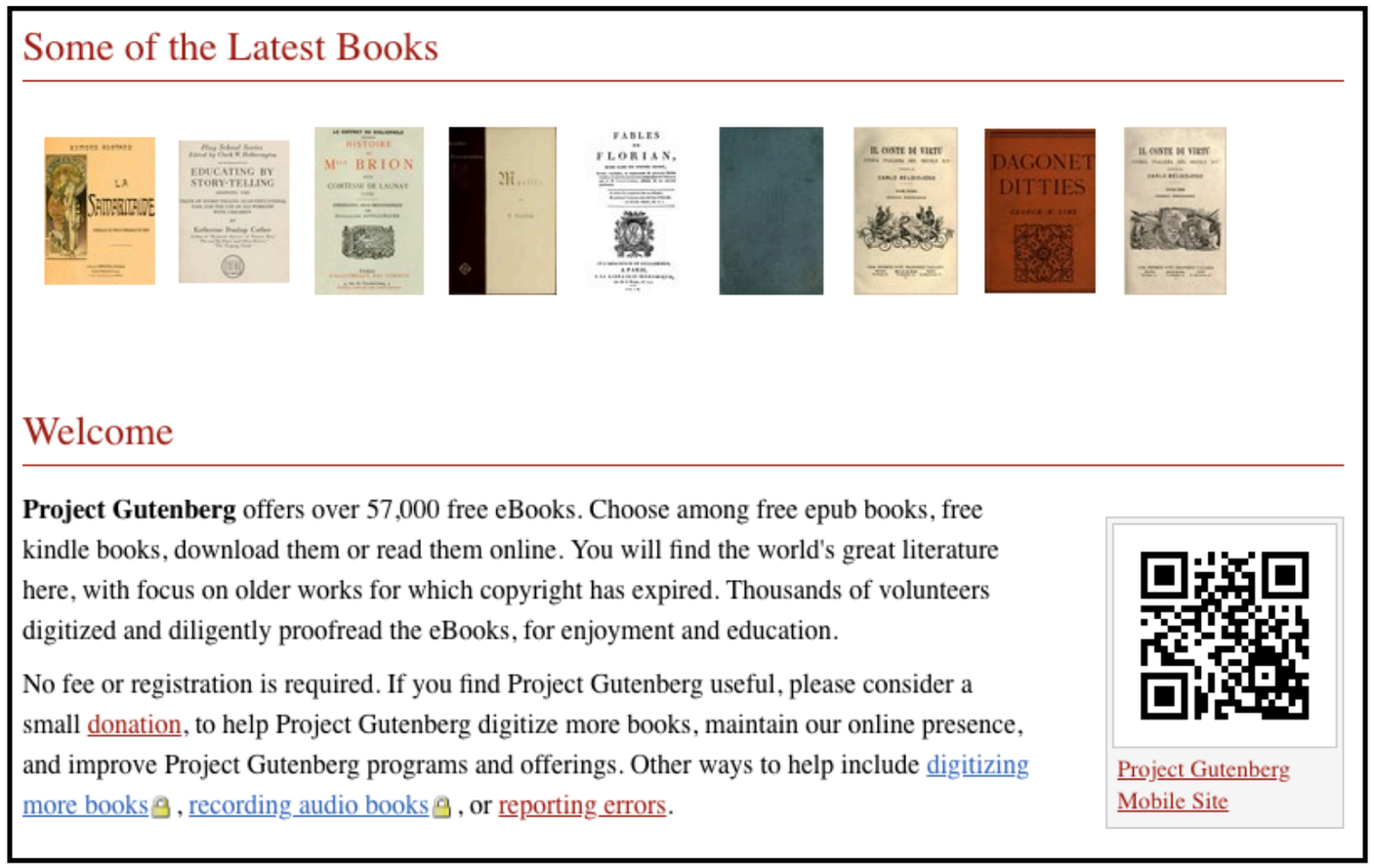


Why not just download data by hand?

Motivation 1: too much data

What if you're analyzing language trends over time?

- Dataset: Project Gutenberg has 57K free books
- Too much work to download one by one



Some of the Latest Books

Welcome

Project Gutenberg offers over 57,000 free eBooks. Choose among free epub books, free kindle books, download them or read them online. You will find the world's great literature here, with focus on older works for which copyright has expired. Thousands of volunteers digitized and diligently proofread the eBooks, for enjoyment and education.

No fee or registration is required. If you find Project Gutenberg useful, please consider a small donation, to help Project Gutenberg digitize more books, maintain our online presence, and improve Project Gutenberg programs and offerings. Other ways to help include digitizing more books, recording audio books, or reporting errors.

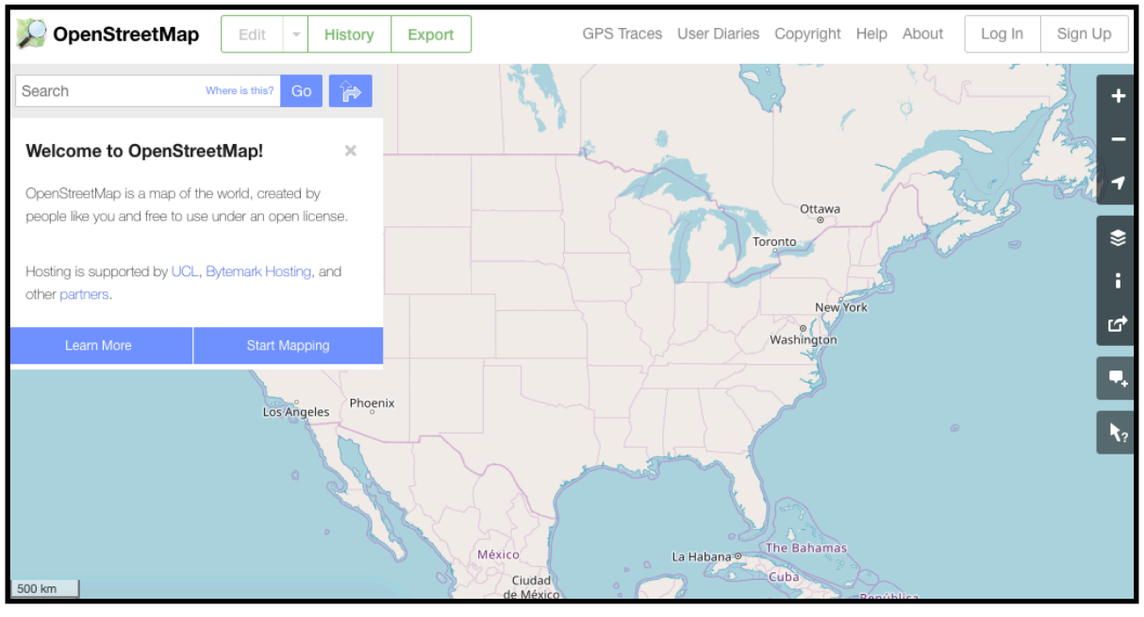
Project Gutenberg Mobile Site

Motivation 2: data doesn't always come in files

Many datasets are difficult to download complete

Instead, you can make function calls to servers
(we'll learn how) to grab specific data

- Dataset: OpenStreetMap
- You issue calls to get specific data:
  1. specify latitude/longitude rectangle
  2. specify structures of interest (e.g., bike paths)



Learning Objectives Today

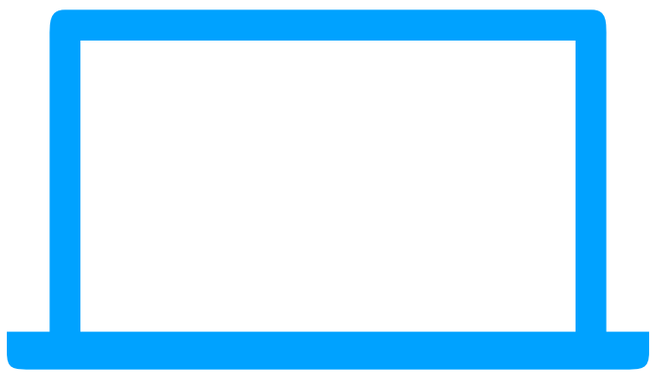
Motivation

Networking Basics

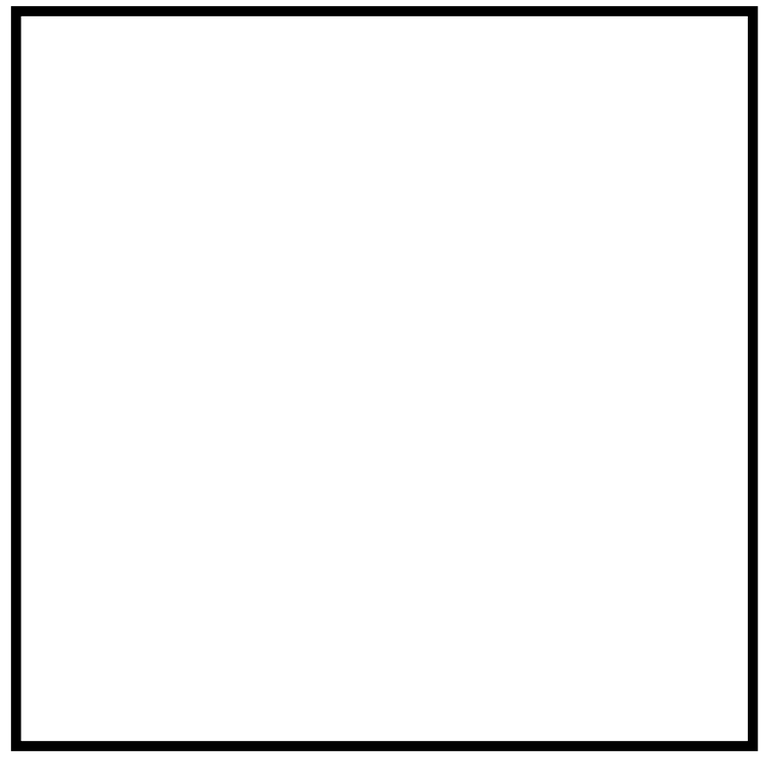
HTTP (Hypertext Transfer Protocol)

Requests Module

Networking Basics



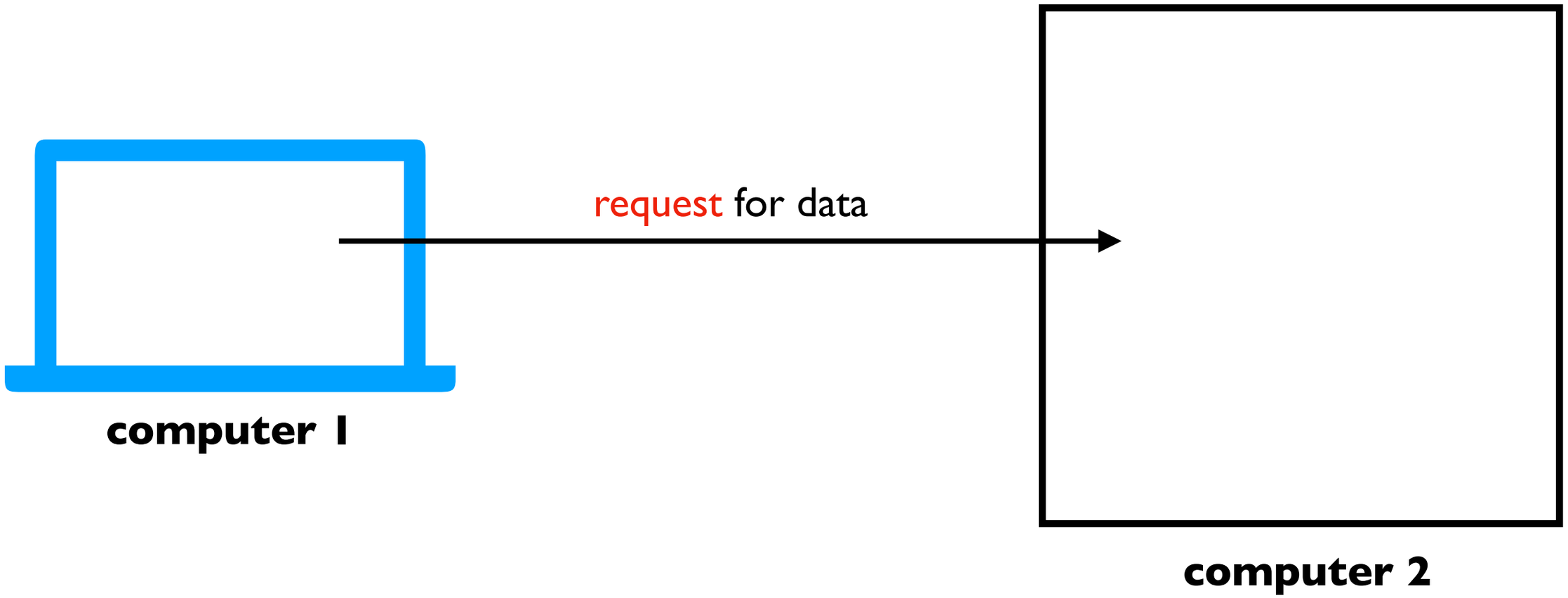
computer 1



computer 2

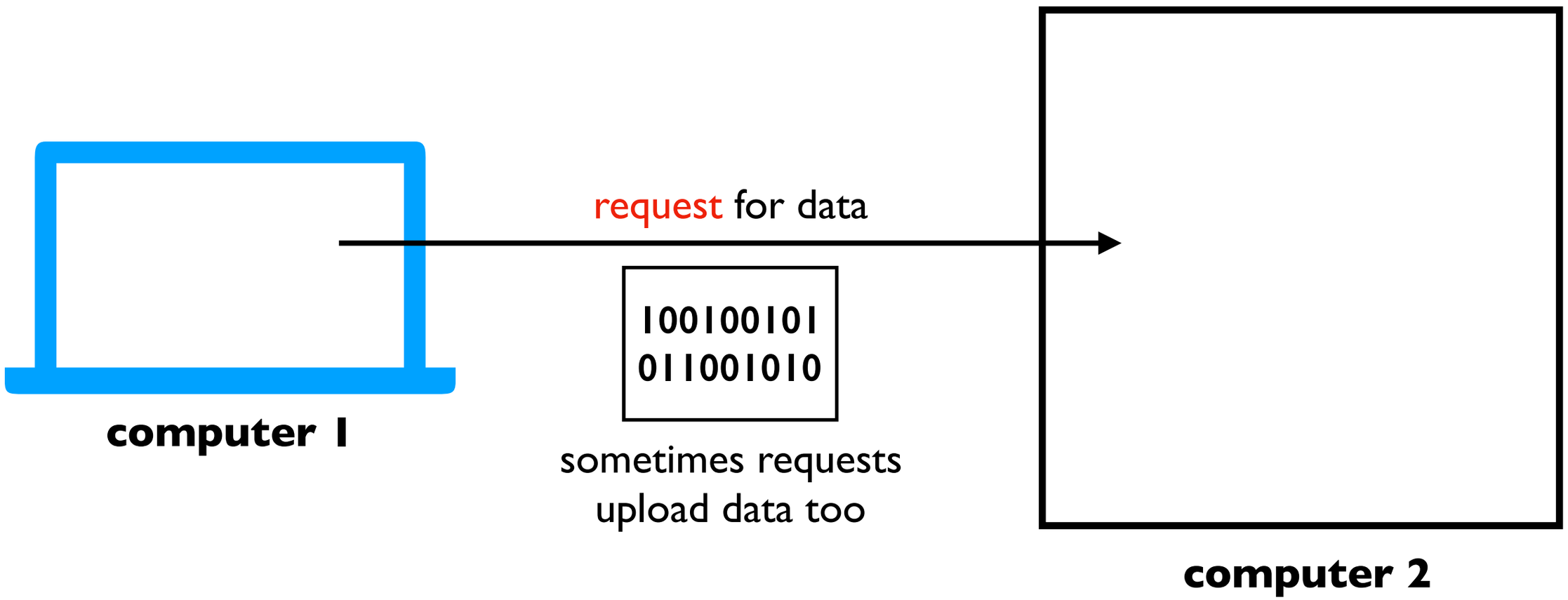
Computers communicate over a network (e.g., the Internet)
by sending messages to each other

Networking Basics



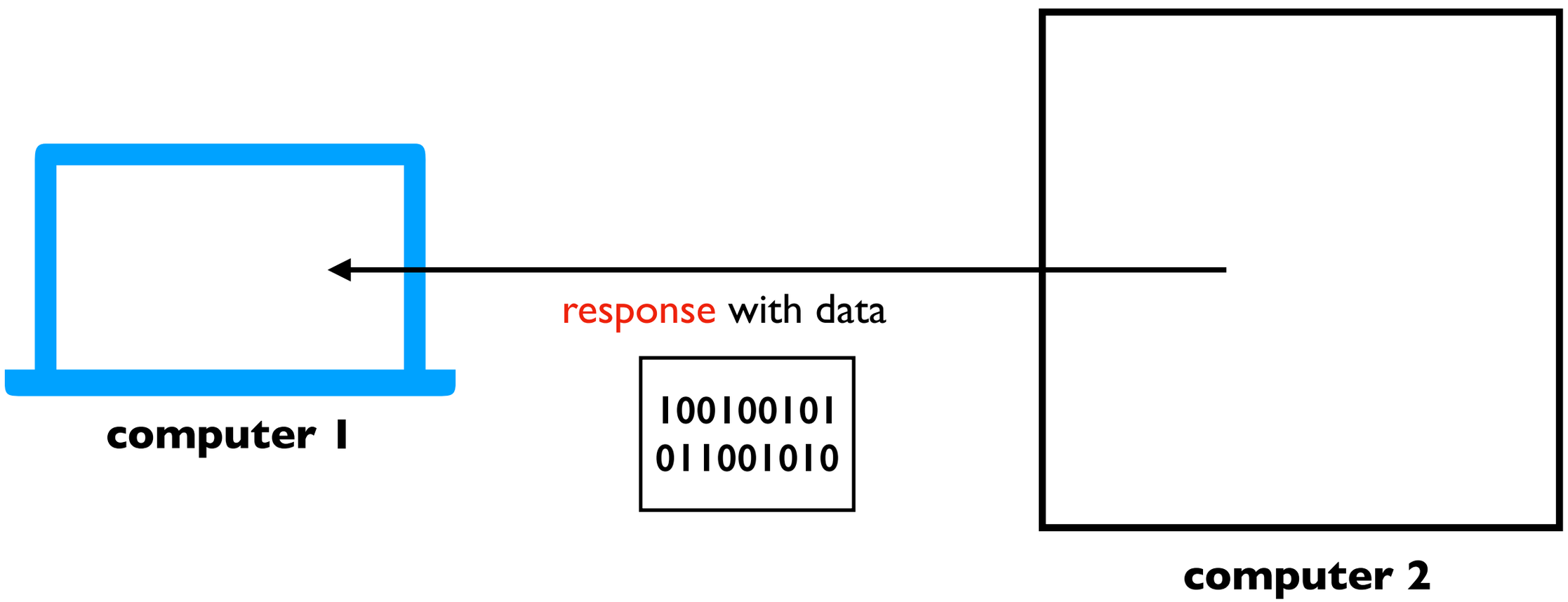
Computers communicate over a network (e.g., the Internet)
by sending messages to each other

Networking Basics



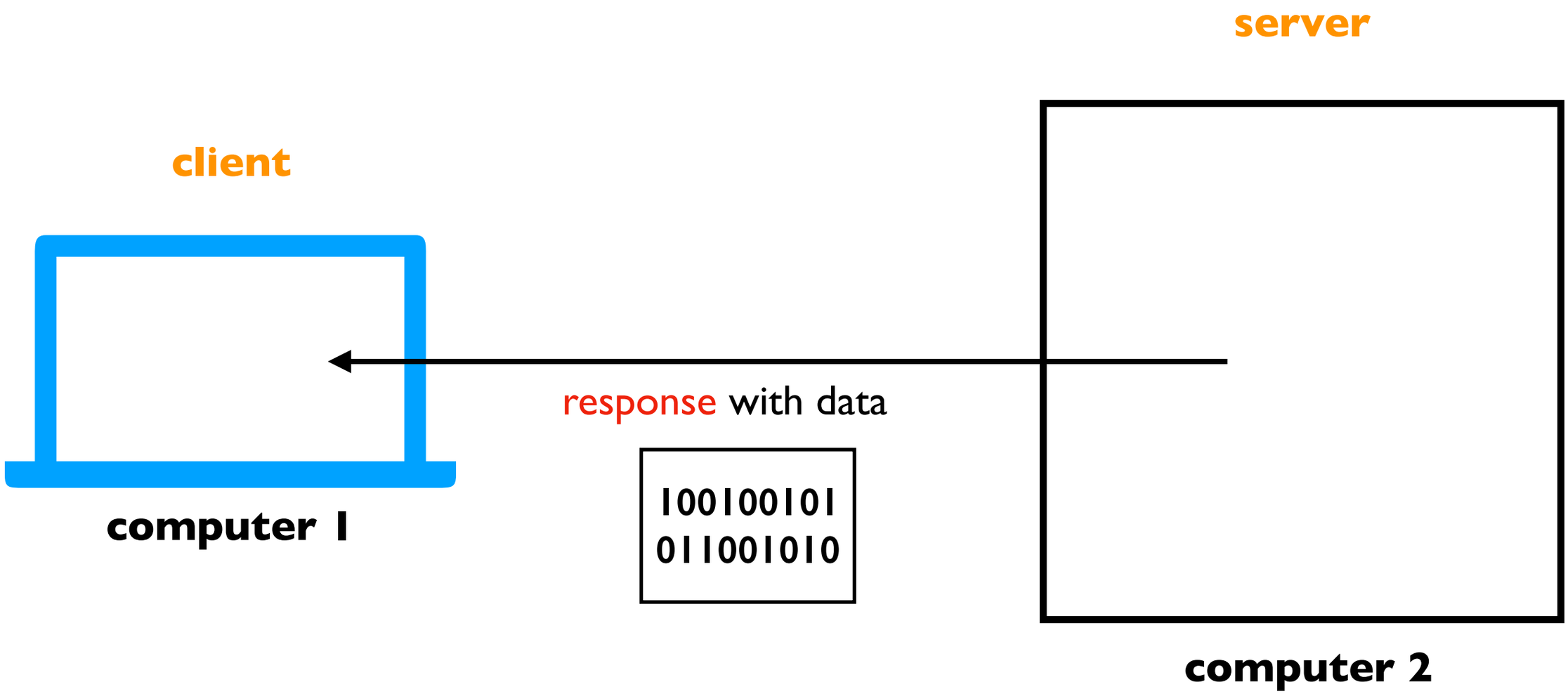
Computers communicate over a network (e.g., the Internet)
by sending messages to each other

Networking Basics



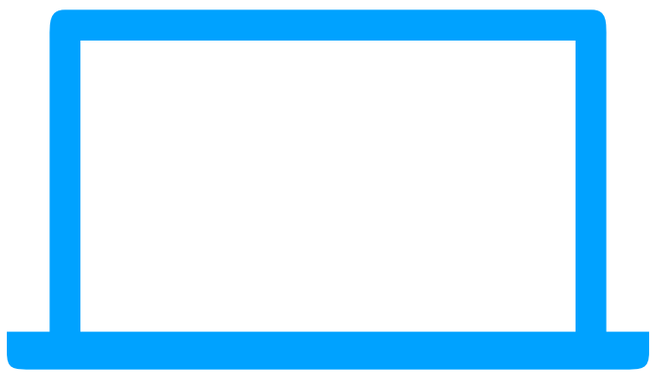
Computers communicate over a network (e.g., the Internet)
by sending messages to each other

Networking Basics

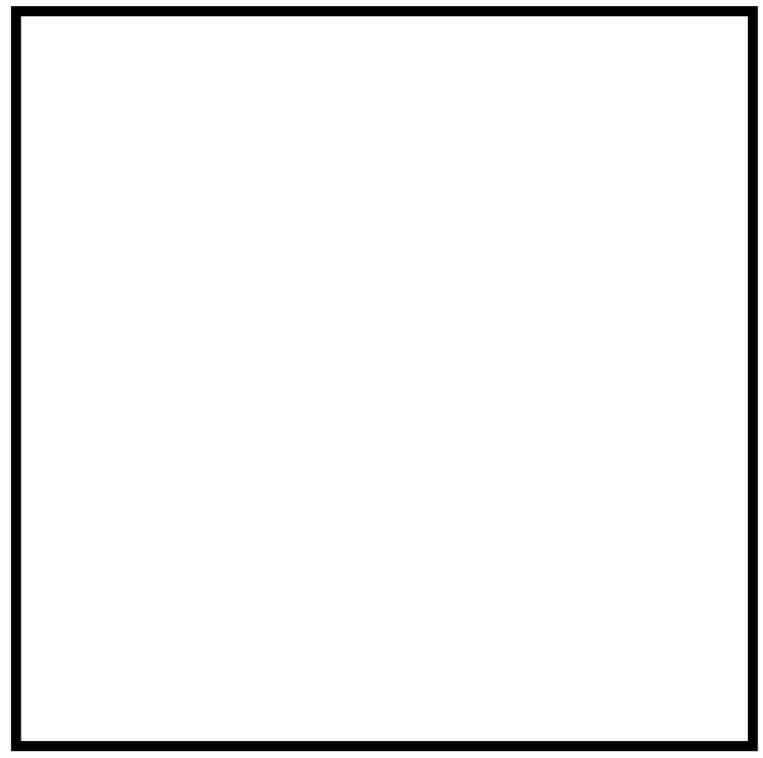


Computers communicate over a network (e.g., the Internet)
by sending messages to each other

Networking Basics



computer 1



computer 2

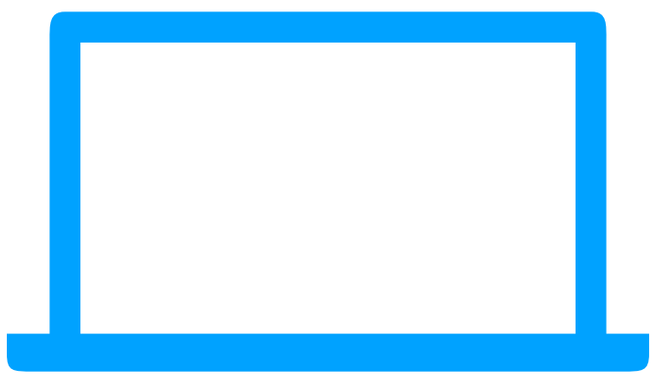
Challenge: there are millions of computers.

How do we indicate which machine should get our request?

How do we send a letter?

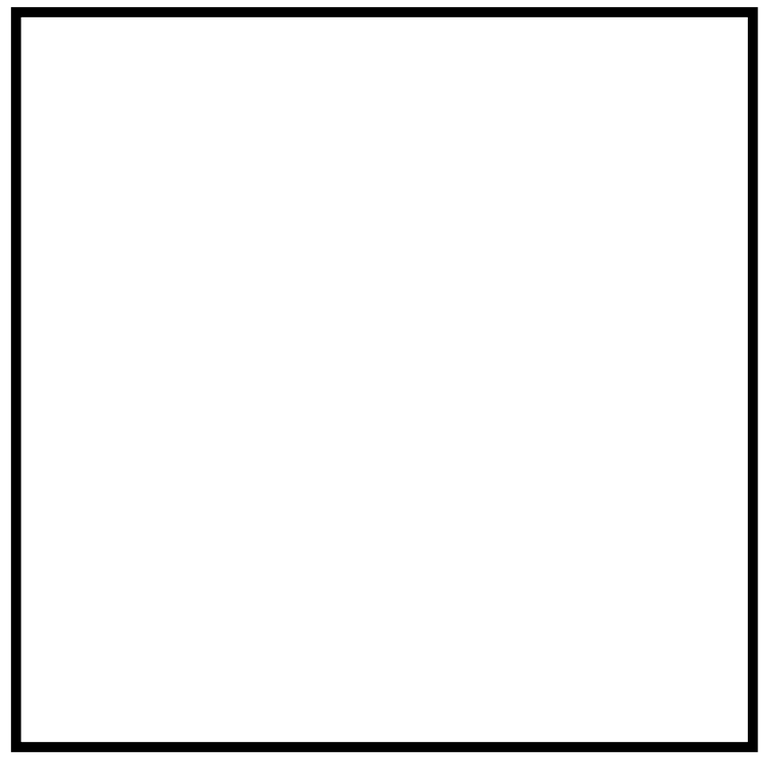
1. lookup friend's address in phone book
2. put address on the envelope
3. trust postal service to get letter to that address

Internet Protocol



computer 1

address: 18.216.110.65

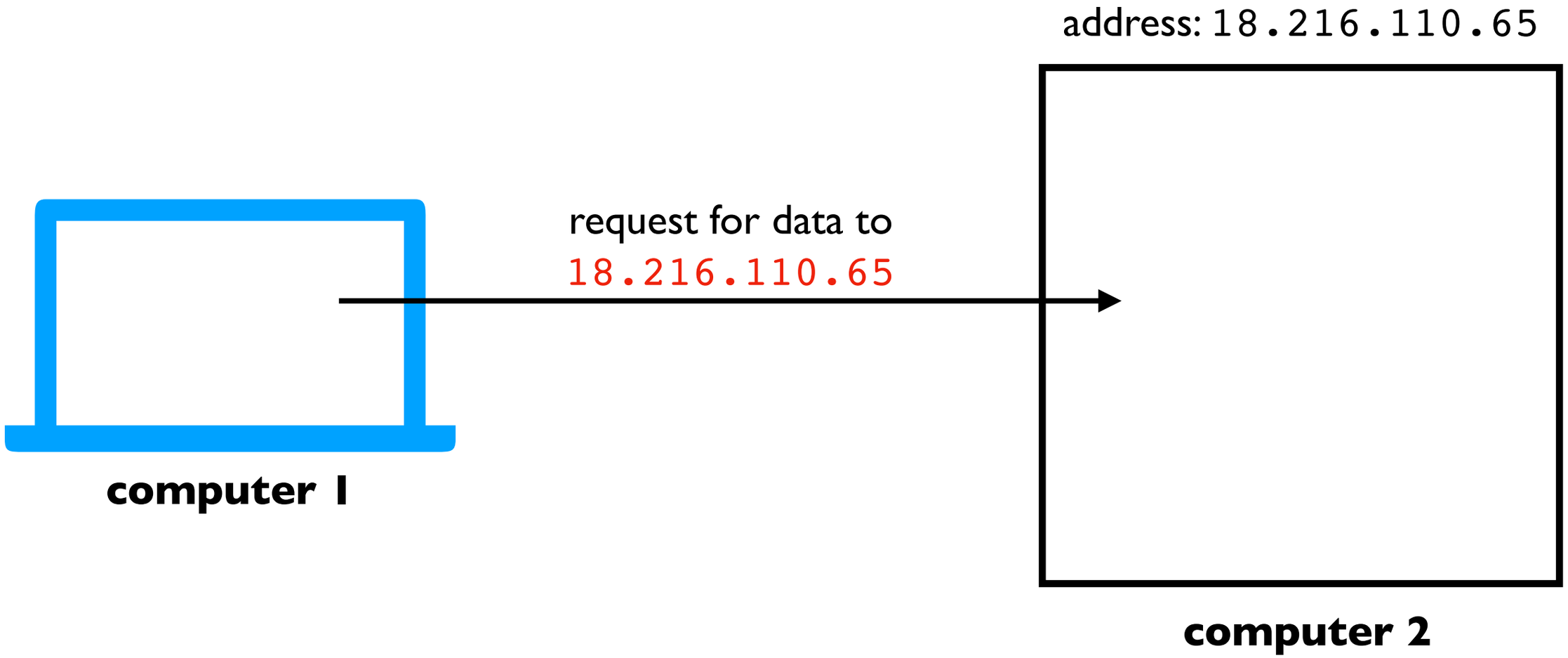


computer 2

Solution: every machine* has an IP address (Internet Protocol).
Requests are sent to a specific IP address.

*some machines have more multiple addresses

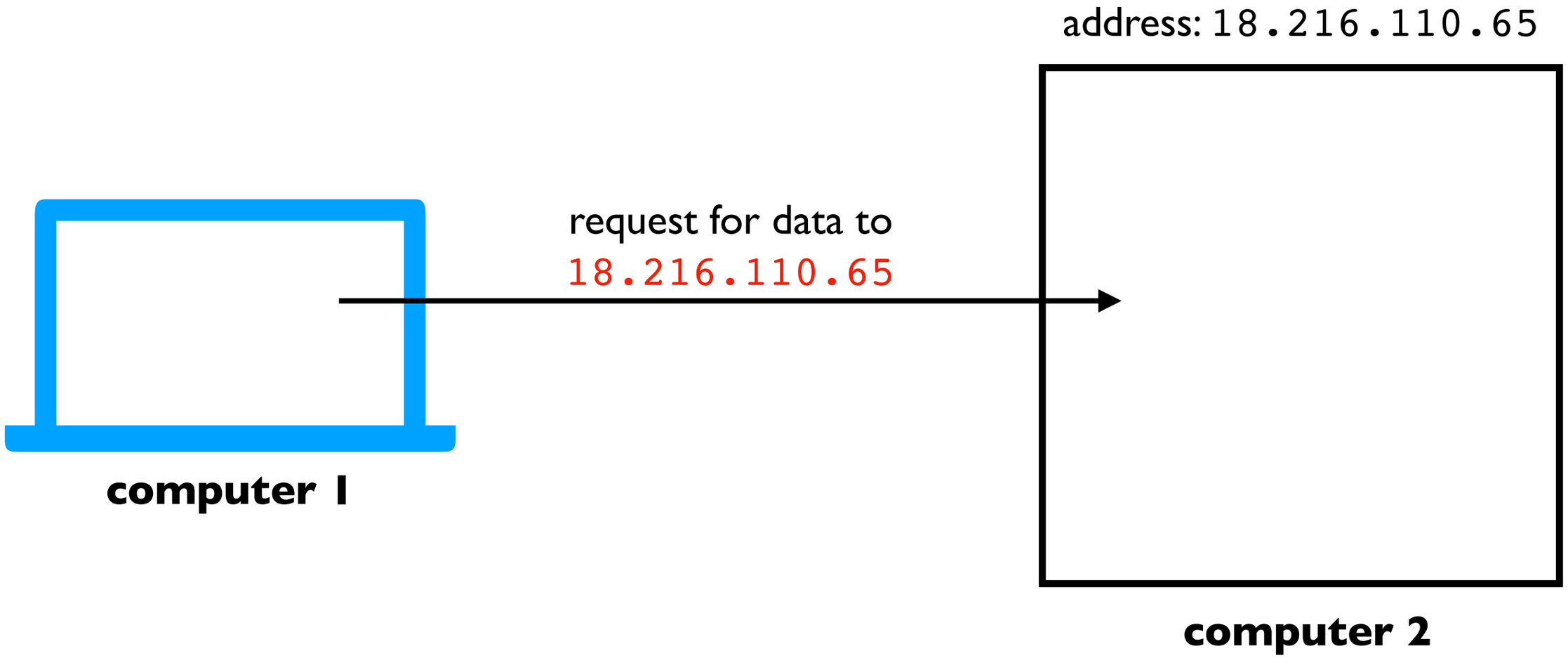
Internet Protocol



Solution: every machine* has an IP address (Internet Protocol).
Requests are sent to a specific IP address.

*some machines have more multiple addresses

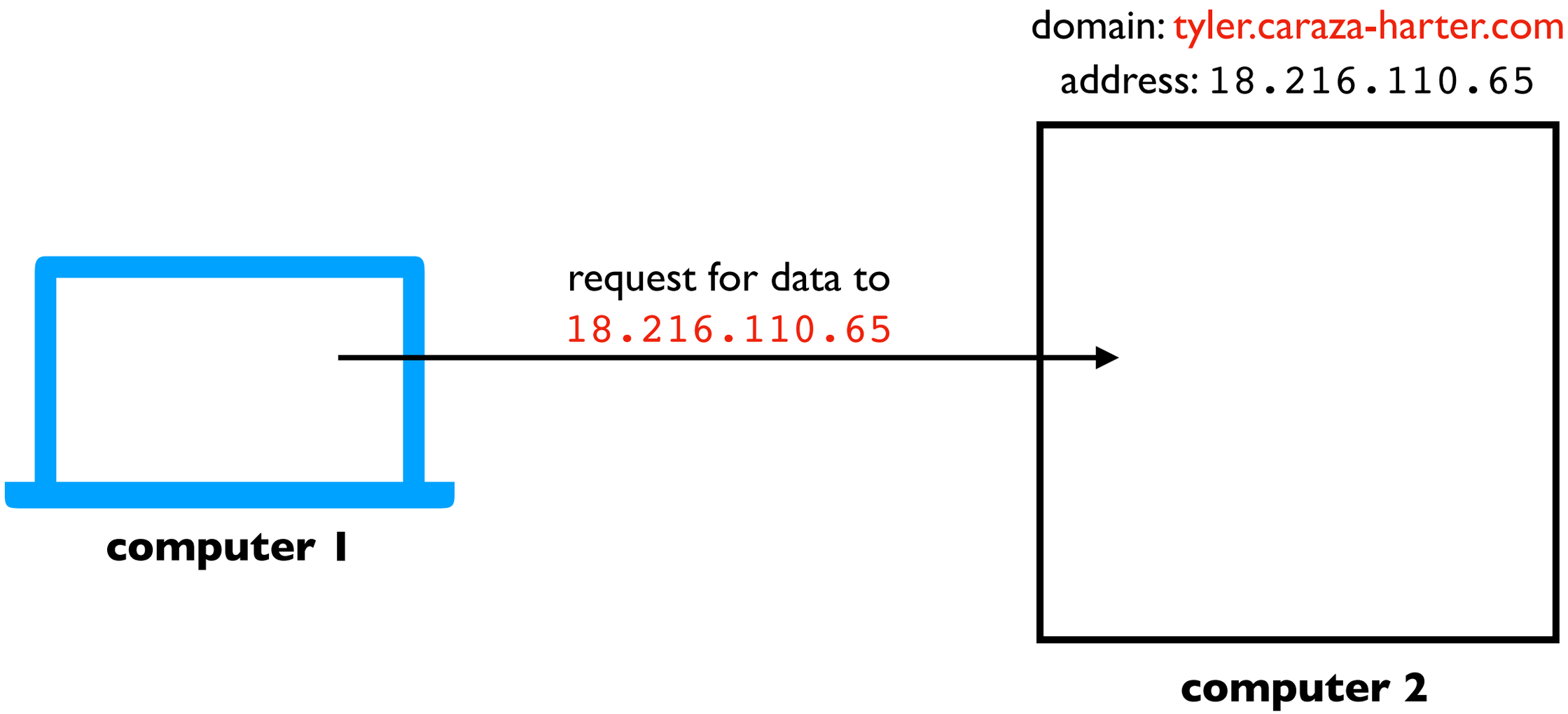
Internet Protocol



Challenge: it's hard to remember IP addresses.

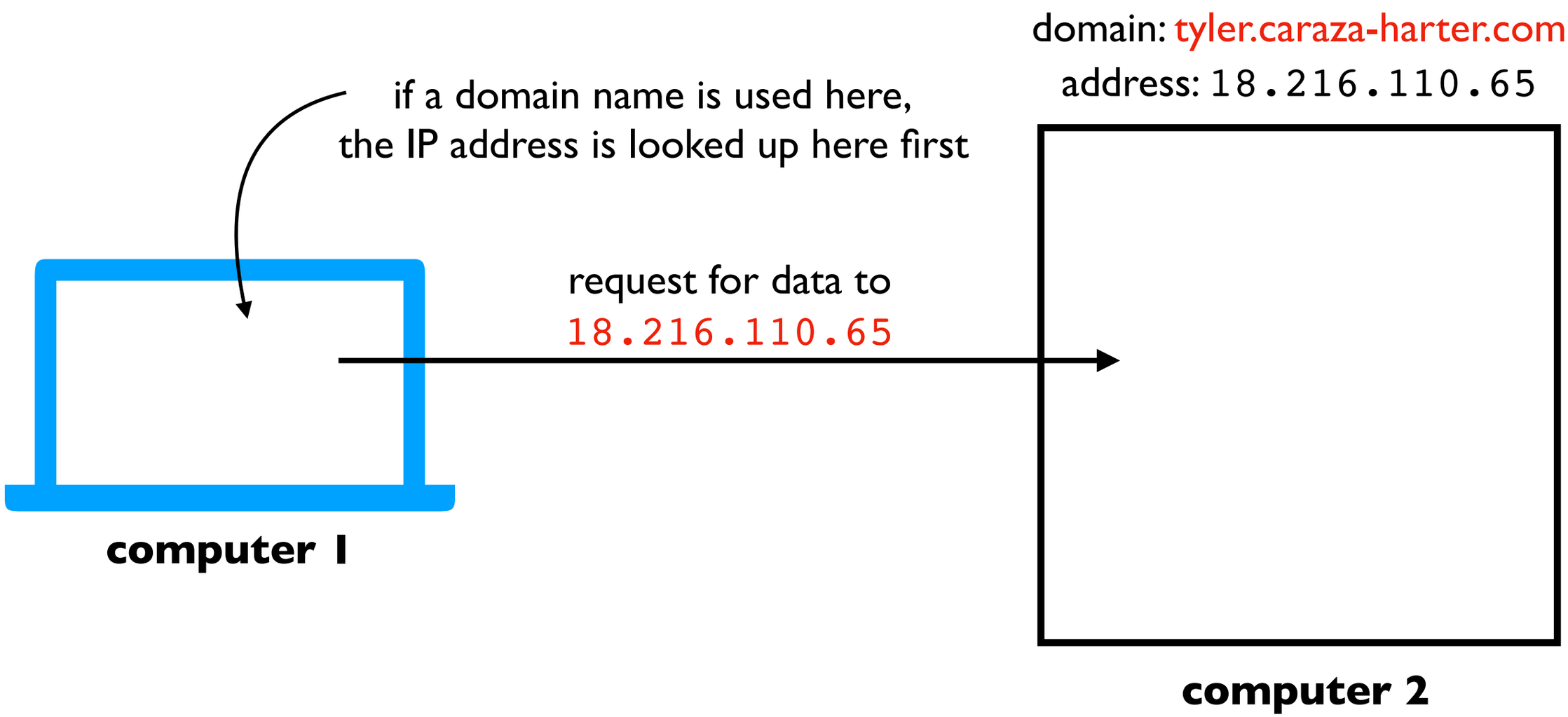
Imagine you had to type a number instead of www.google.com!

Domain Names



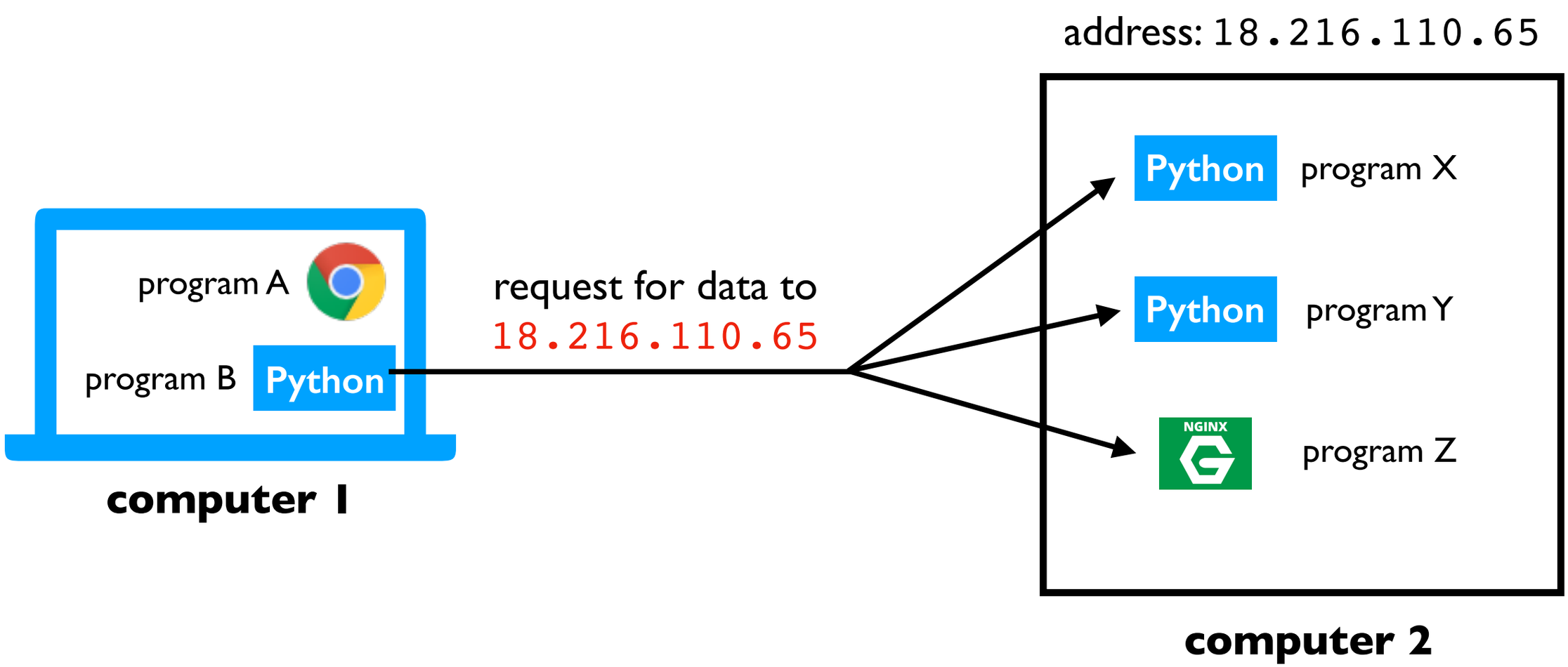
Solution: use "nicknames" (called domain names)
for IP addresses of machines that serve data

Domain Names



Solution: use "nicknames" (called domain names)
for IP addresses of machines that serve data

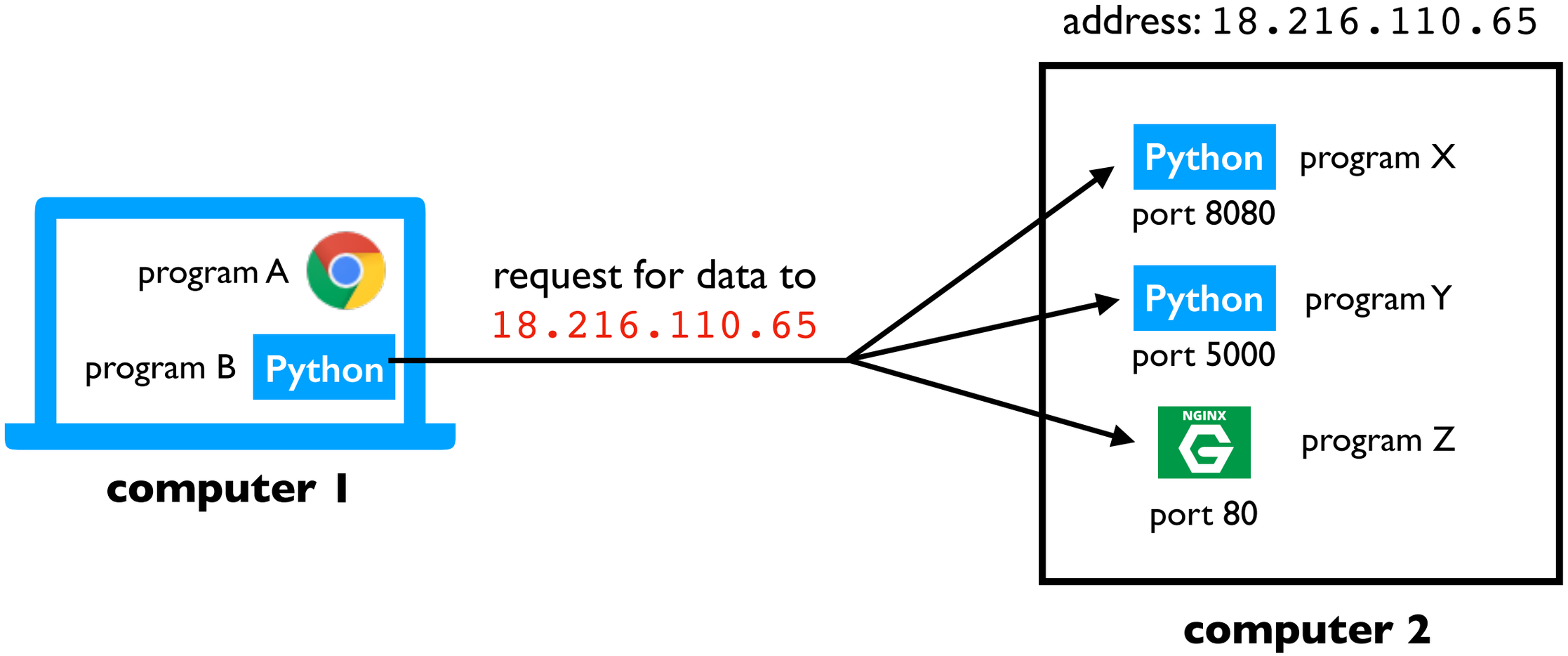
Port Numbers



Challenge: there may be multiple programs running on each computer.

How do we get the messages to the right program?

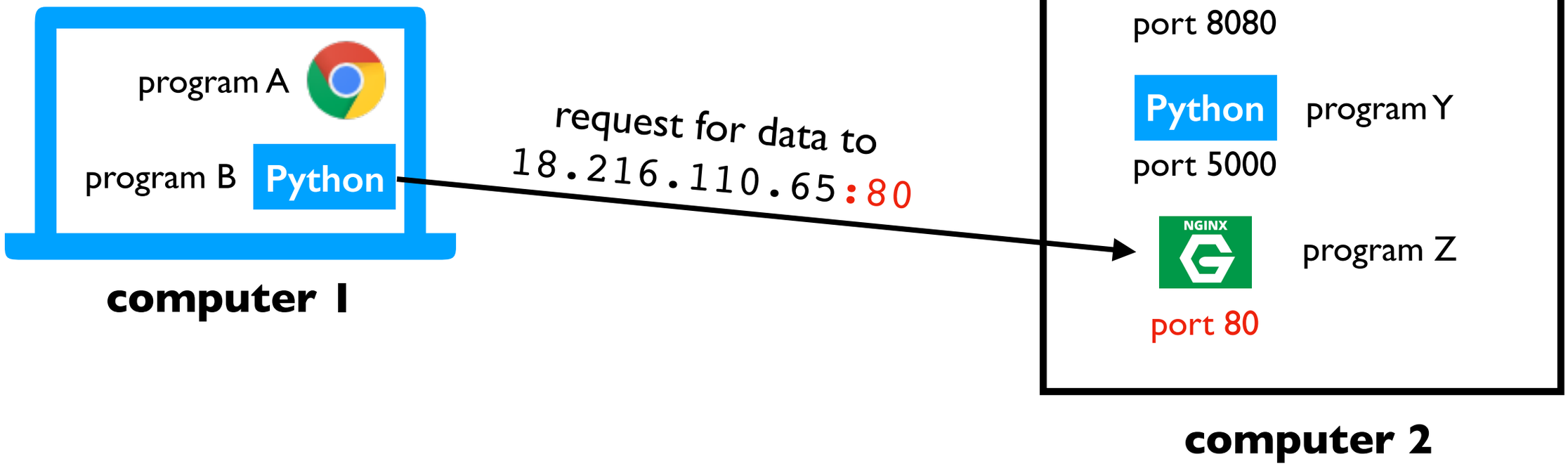
Port Numbers



Solution: give each program a unique ID (called a "port number")

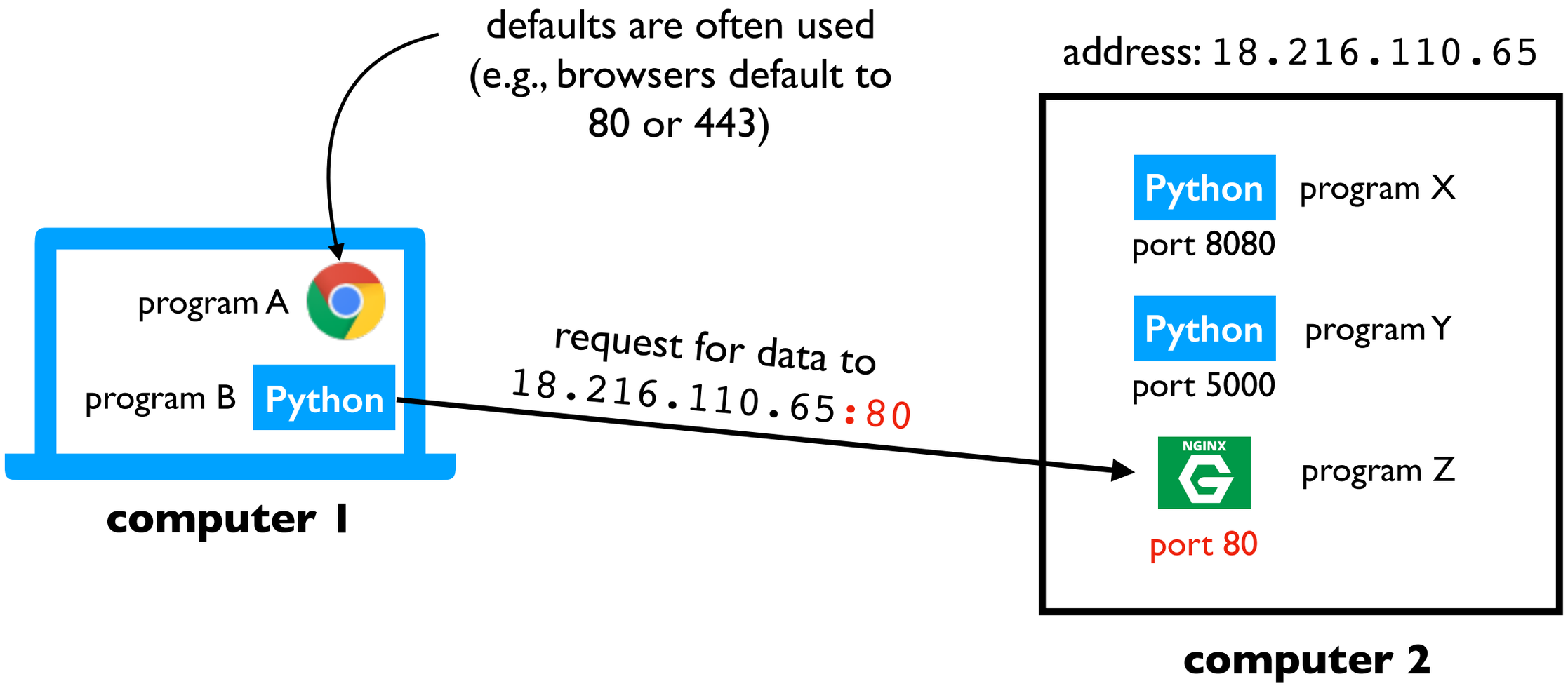
(like apartment numbers)

Port Numbers

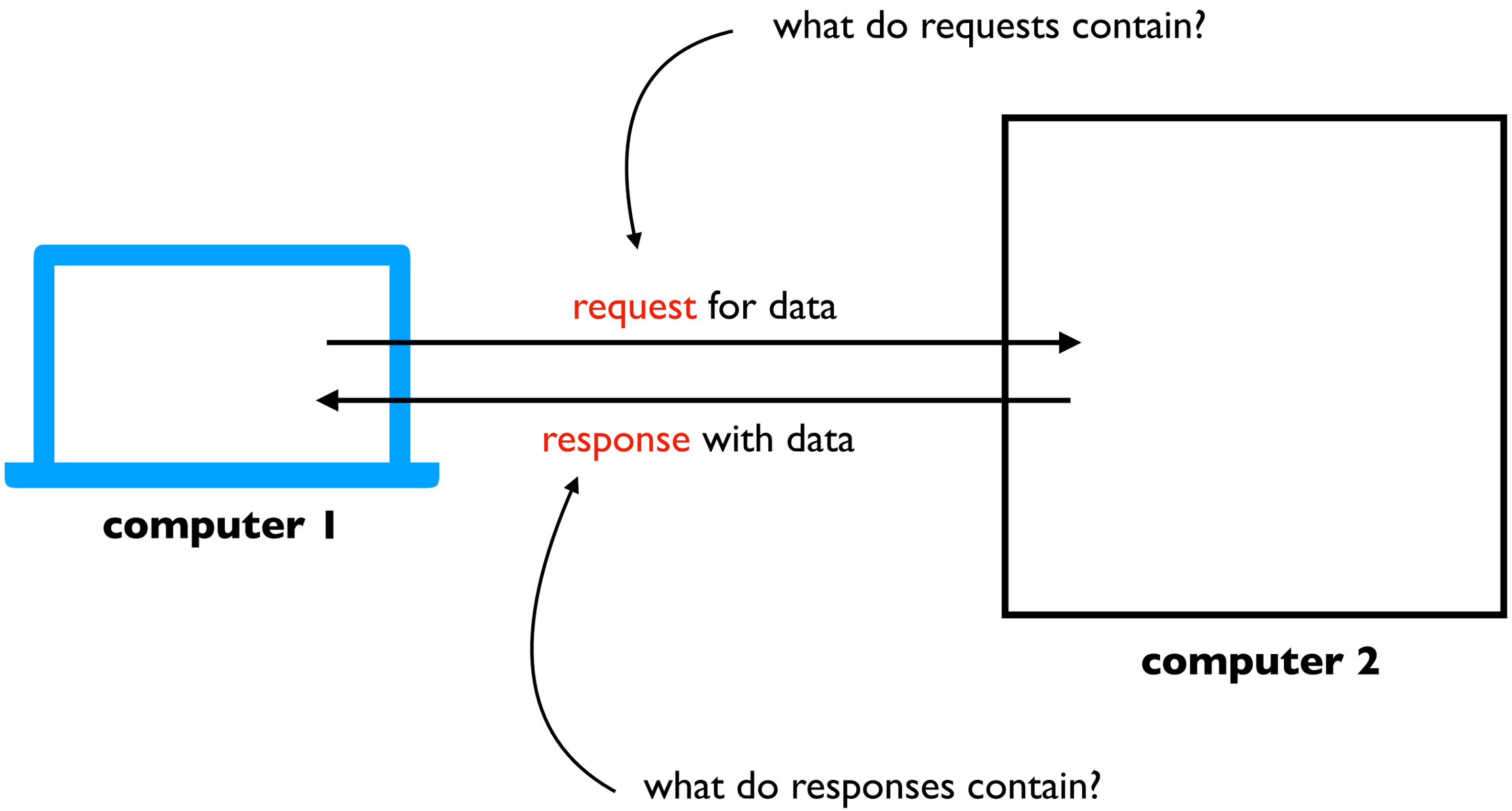


Solution: specify port number in request

Port Numbers

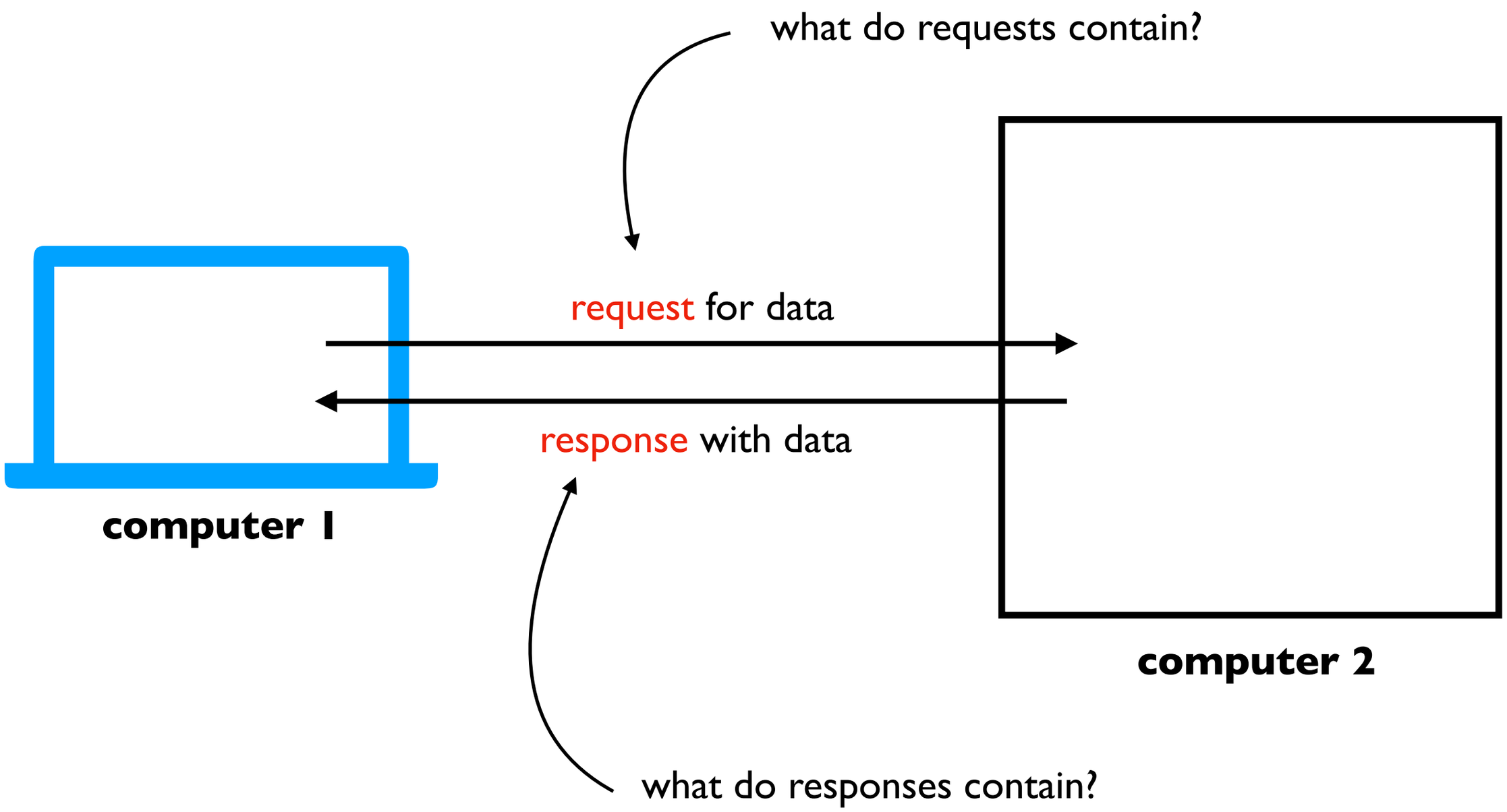


Solution: specify port number in request


what do requests contain?
request for data
response with data
what do responses contain?
computer 1
computer 2

depends on application! (video chat, web browsing, etc)

we'll only consider web applications for this semester



Learning Objectives Today

Motivation

Networking Basics

HTTP (Hypertext Transfer Protocol)

Requests Module

HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc

Note: we won't talk about HTTPS today, which is HTTP with encryption

HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc

HTTP

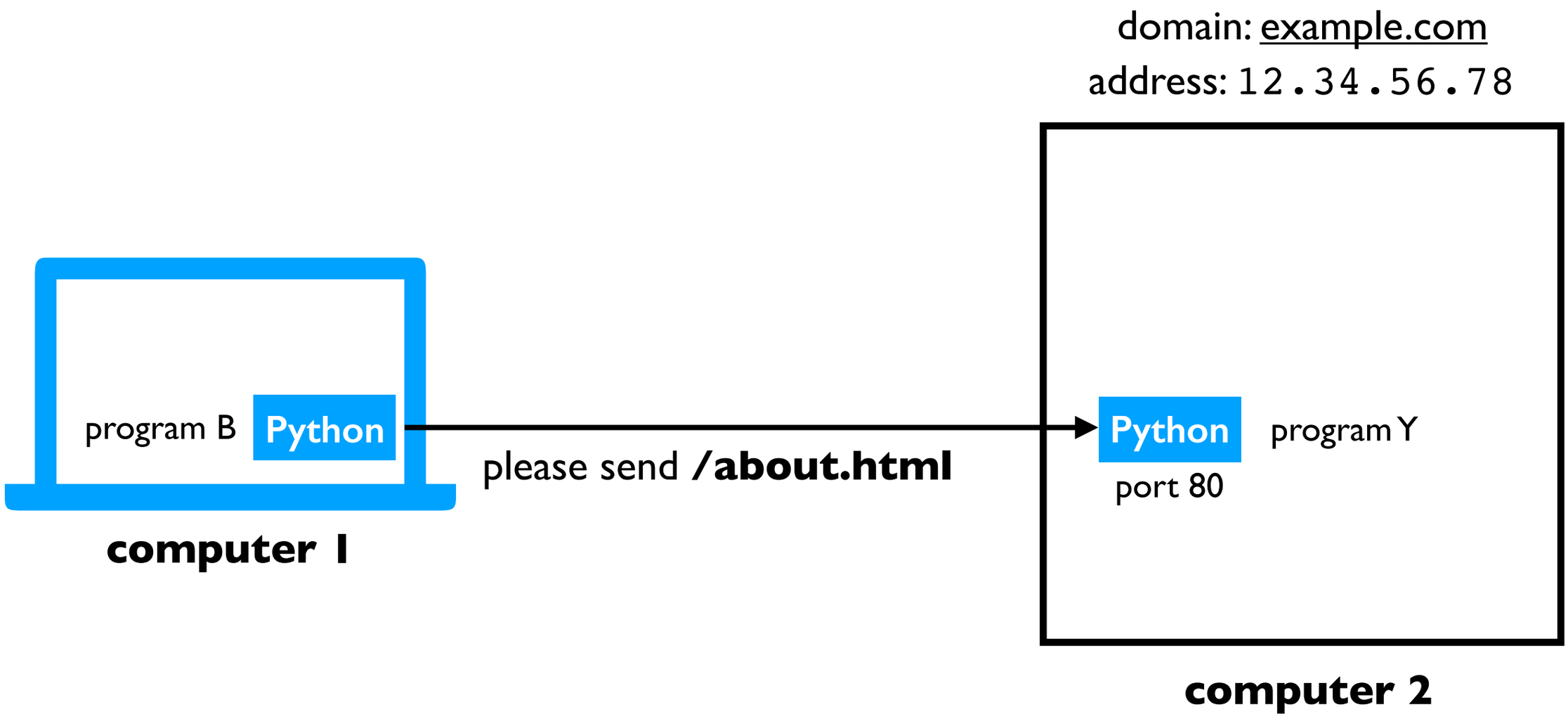
Protocol for communicating web data

- downloading a specific webpage, image, etc

HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc



HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc

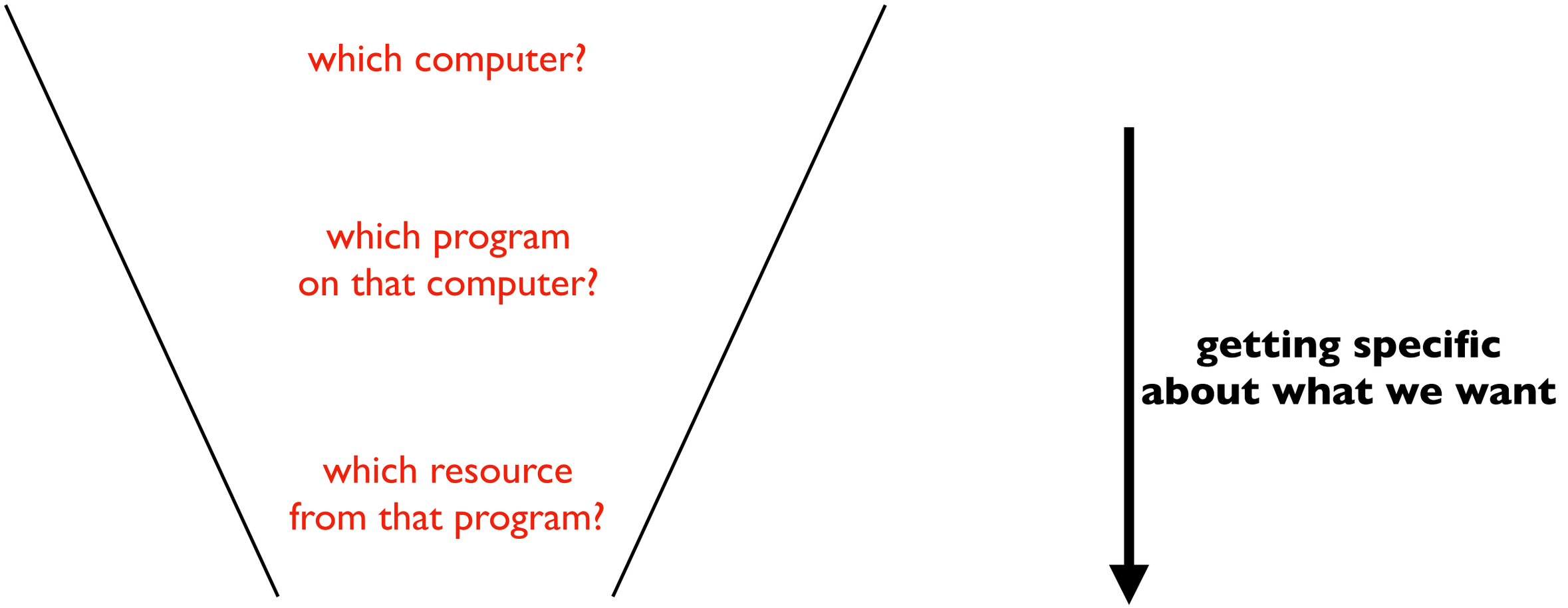
HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc

We need three things:

1. domain name
2. port number
3. resource (file name)



We need three things:

1. domain name
2. port number
3. resource (file name)

which computer?

which program
on that computer?

which resource
from that program?

**getting specific
about what we want**

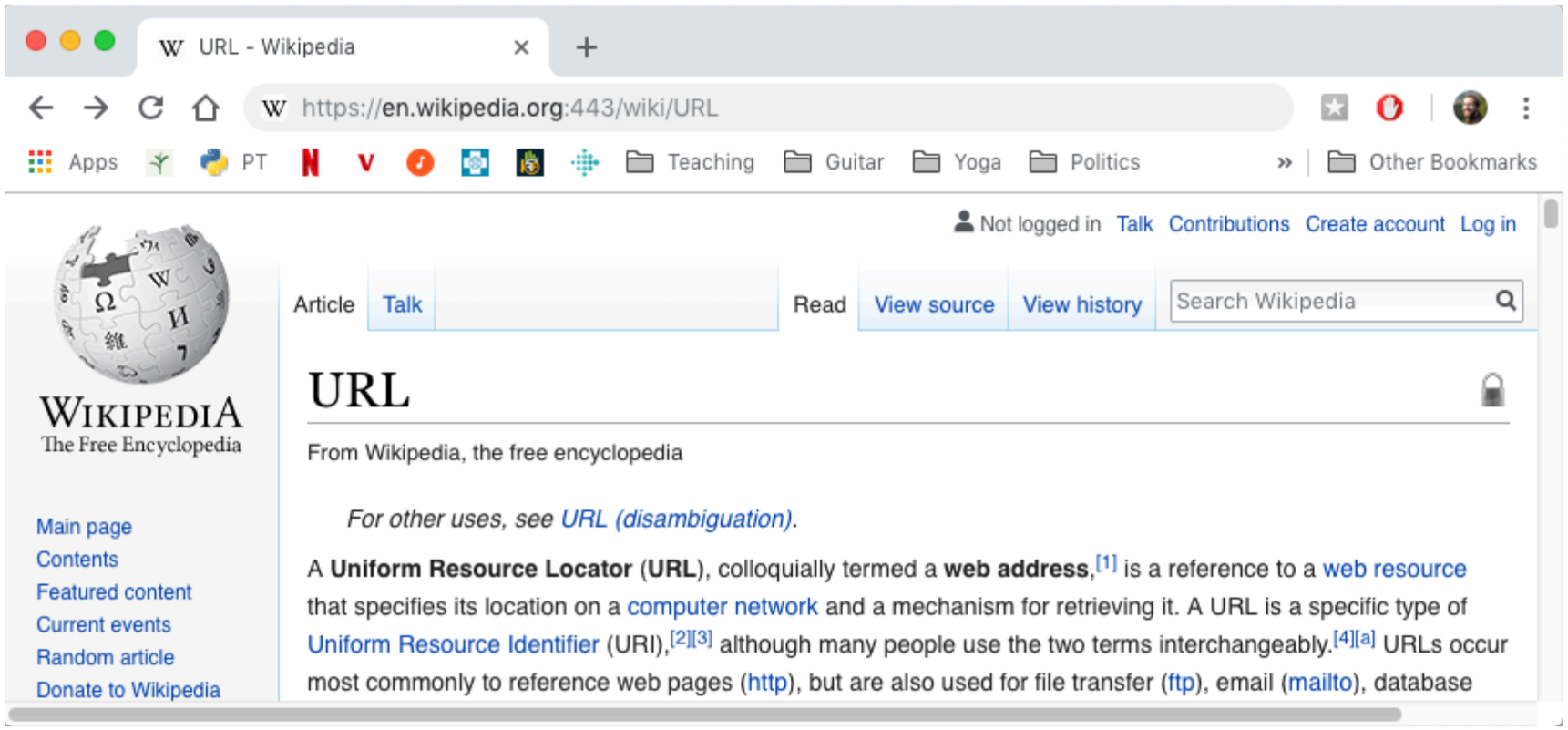
URL

We need three things:

1. domain name
2. port number
3. resource (file name)

URLs

https://en.wikipedia.org:443/wiki/URL



URL

We need three things:

1. domain name
2. port number
3. resource (file name)

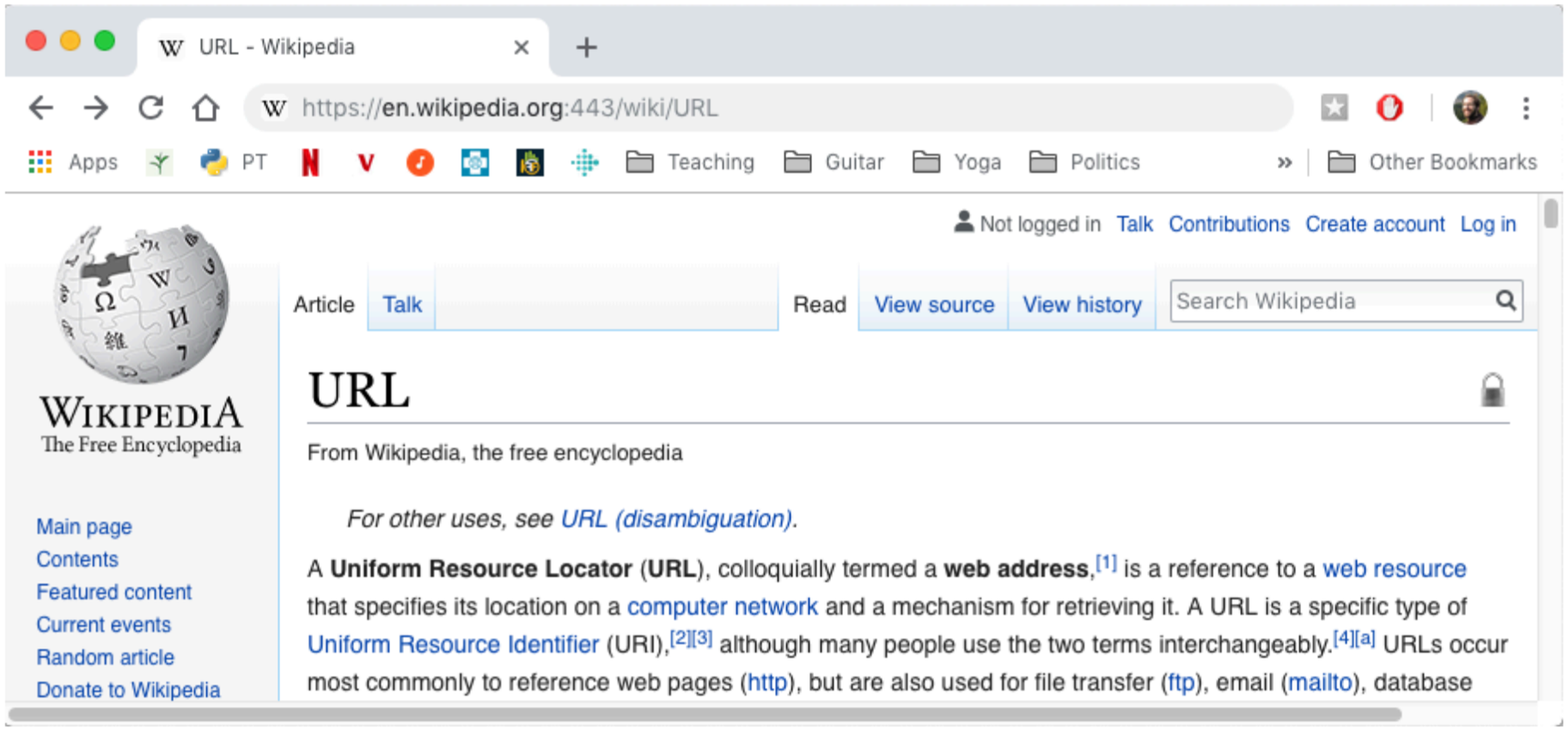
URLs

domain name

resource

`https://en.wikipedia.org:443/wiki/URL`

port



URL

We need three things:

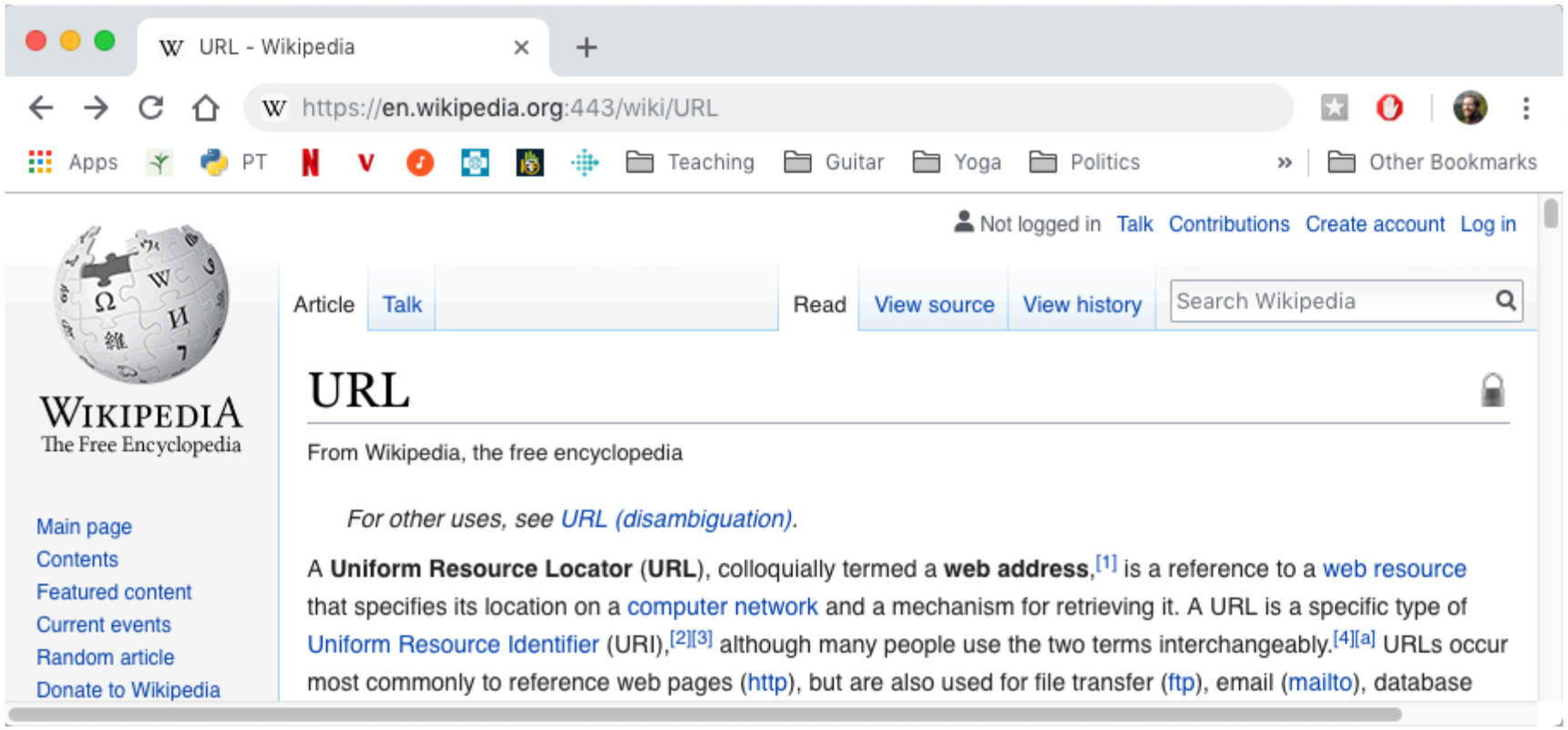
1. domain name
2. port number
3. resource (file name)

URLs

domain name resource

https://en.wikipedia.org/wiki/URL

port would have defaulted to 443 if not specified



URL

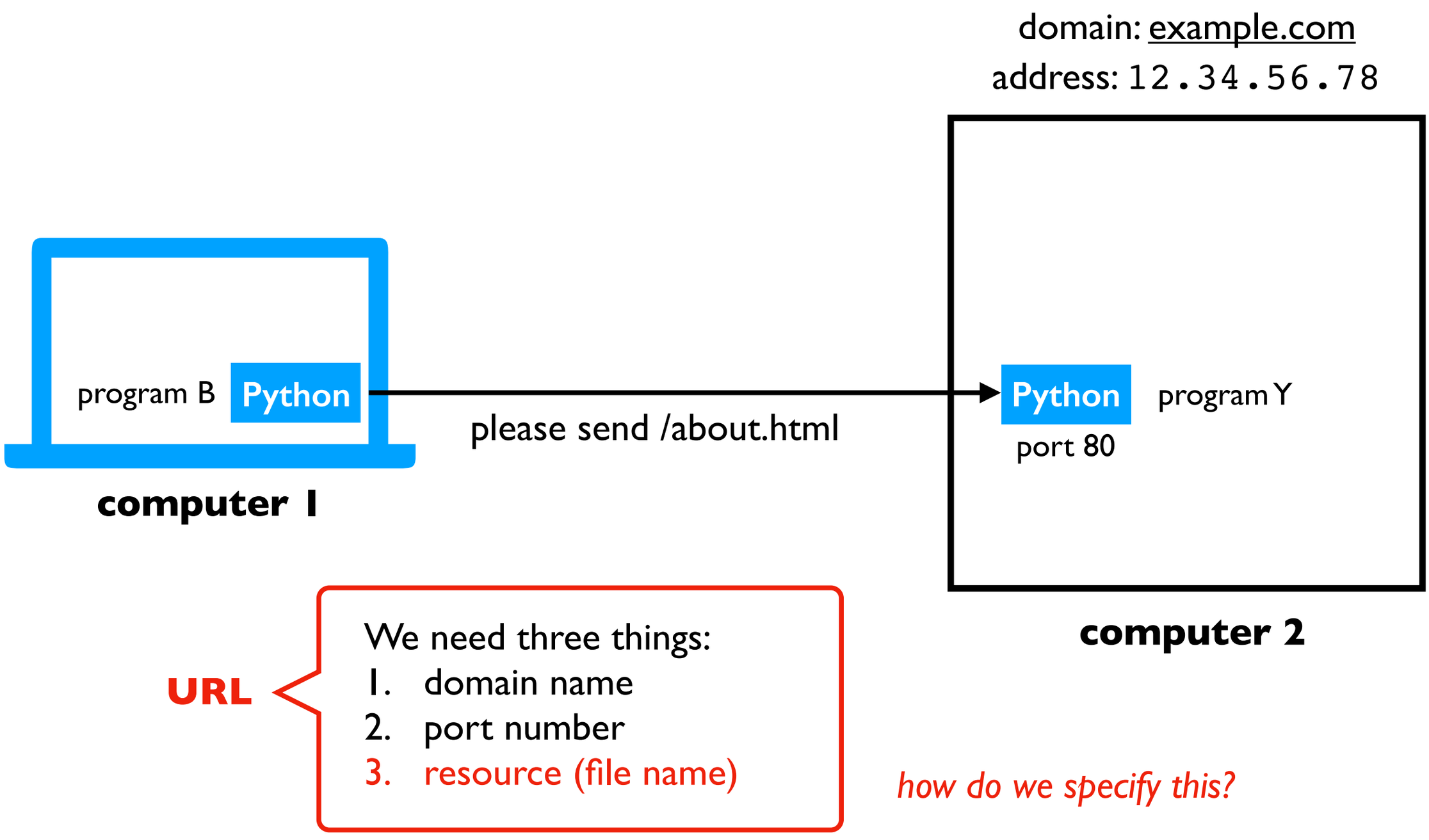
We need three things:

1. domain name
2. port number
3. resource (file name)

HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc



URL

We need three things:

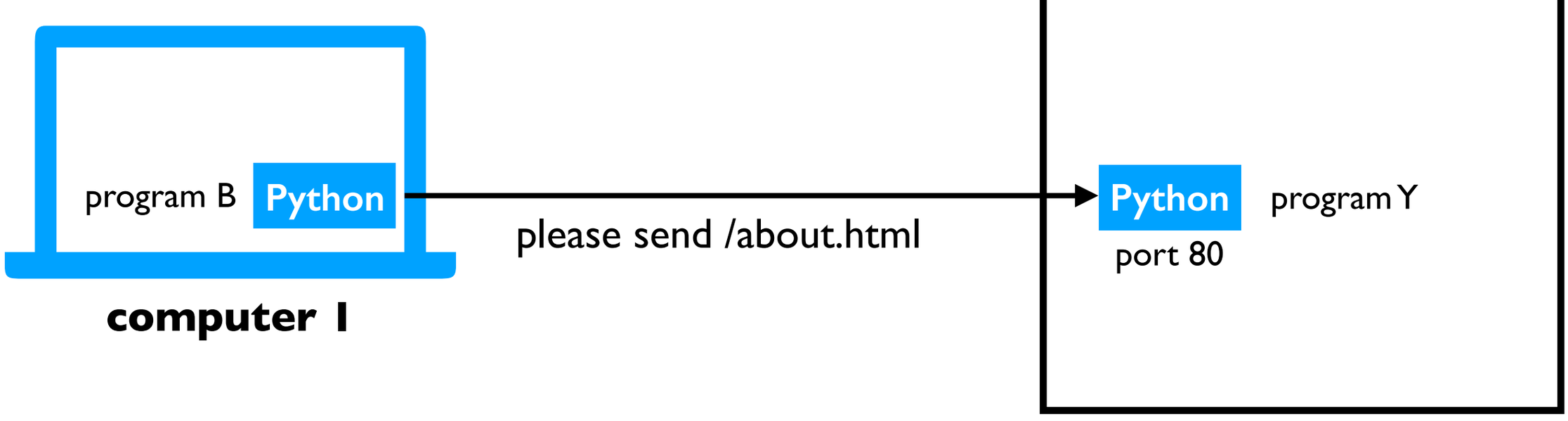
1. domain name
2. port number
3. resource (file name)

how do we specify this?

HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc



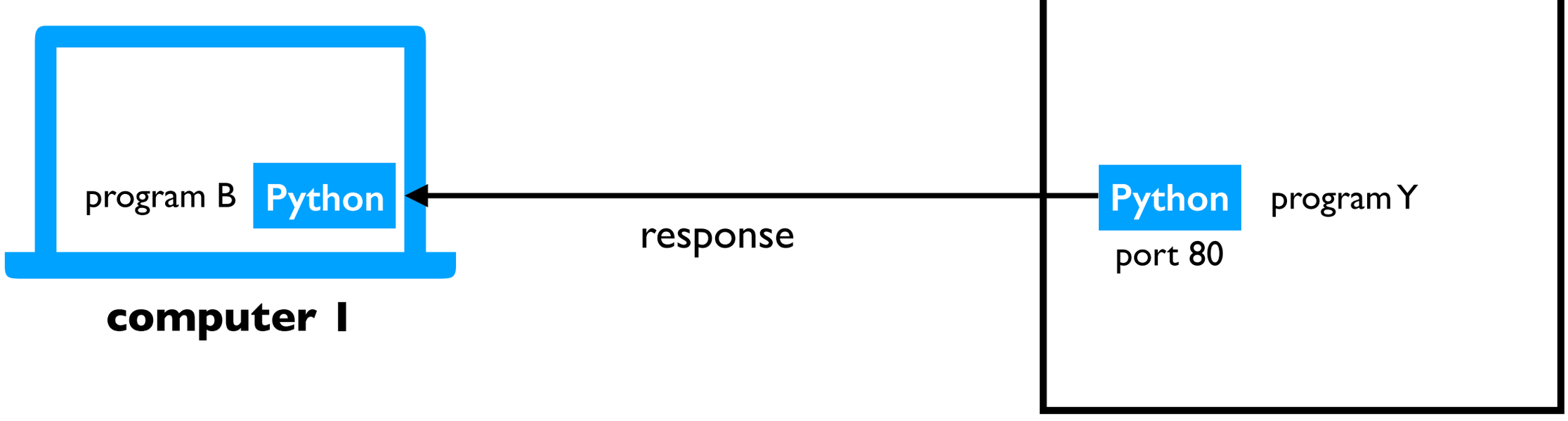
HTTP Request:

```
GET /about.html HTTP/1.1
Host: example.com
User-Agent: ...
Accept: */*
```

HTTP

Protocol for communicating web data

- downloading a specific webpage, image, etc



HTTP Response:

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/0.14.1 Python/3.6.6
Date: Sun, 11 Nov 2018 17:00:29 GMT

all the contents
```

Request and Response Headers

HTTP Request:

```
GET /about.html HTTP/1.1
Host: example.com
User-Agent: ...
Accept: */*
```

HTTP Response:

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/0.14.1 Python/3.6.6
Date: Sun, 11 Nov 2018 17:00:29 GMT

all the contents
```

There are **LOTS** of details here we don't care about right now

Request and Response Headers

we want the about.html page

HTTP Request:

```
GET /about.html HTTP/1.1
Host: example.com
User-Agent: ...
Accept: */*
```

HTTP Response:

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/0.14.1 Python/3.6.6
Date: Sun, 11 Nov 2018 17:00:29 GMT

all the contents
```

data in about.html

There are **LOTS** of details here we don't care about right now

Request and Response Headers

we want the about.html page

HTTP Request:

```
GET /about.html HTTP/1.1
Host: example.com
User-Agent: ...
Accept: */*
```

status code. 200 is good. 404, 500, others are various errors or other more complicated states

HTTP Response:

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/0.14.1 Python/3.6.6
Date: Sun, 11 Nov 2018 17:00:29 GMT

all the contents
```

data in about.html

There are **LOTS** of details here we don't care about right now

method. *GET* is simple download. *POST* means we are uploading data as part of our request. We won't talk about others today.

we want the about.html page

HTTP Request:

```
GET /about.html HTTP/1.1
Host: example.com
User-Agent: ...
Accept: */*
```

status code. 200 is good. 404, 500, others are various errors or other more complicated states

HTTP Response:

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/0.14.1 Python/3.6.6
Date: Sun, 11 Nov 2018 17:00:29 GMT

all the contents
```

data in about.html

There are **LOTS** of details here we don't care about right now

Learning Objectives Today

Motivation

Networking Basics

HTTP (Hypertext Transfer Protocol)

Requests Module

Requests module

Purpose

- easily send requests to a server and parse the response
- *"HTTP for Humans™"*

Installation

- install:

```
pip install requests
```

Using it

- just import:

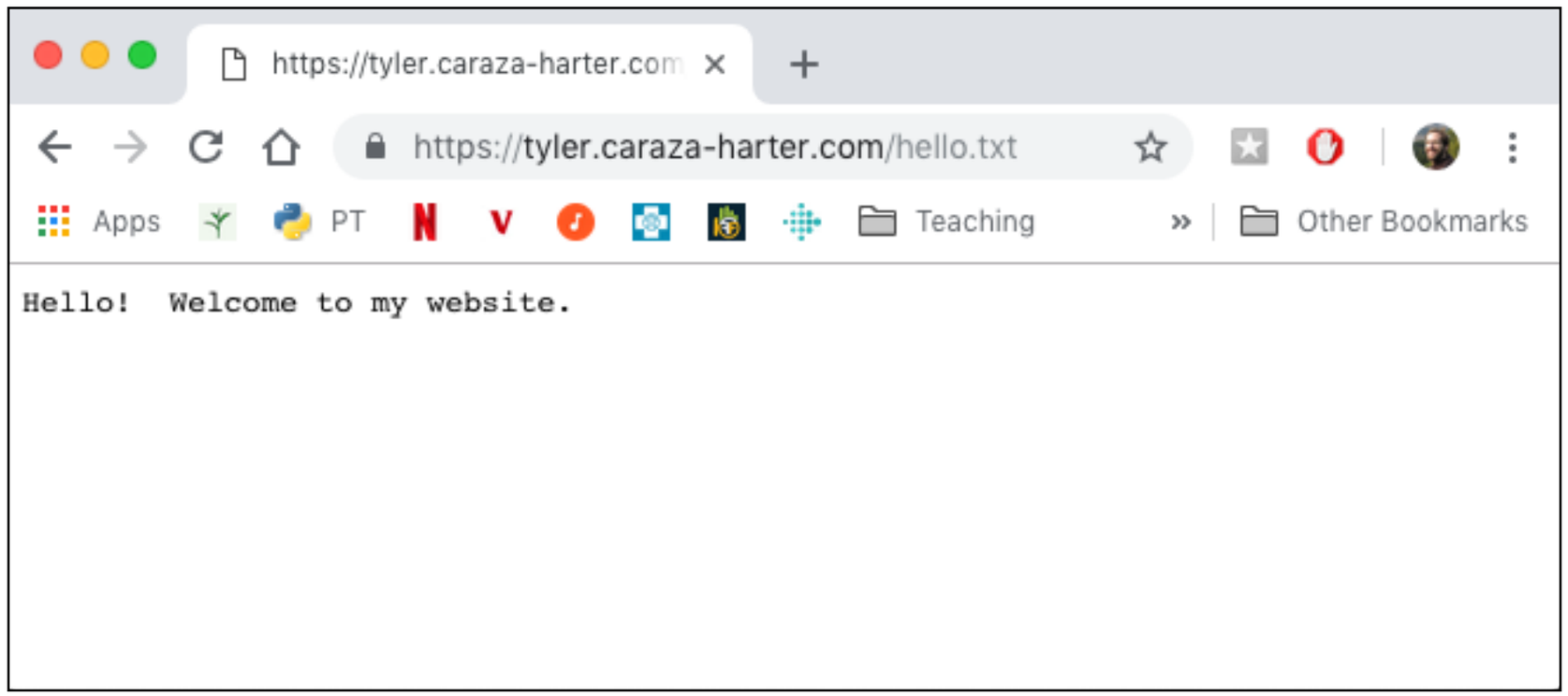
```
import requests
```

GET Request

```
import requests

url = "https://tyler.caraza-harter.com/hello.txt"

requests.get(url)
```



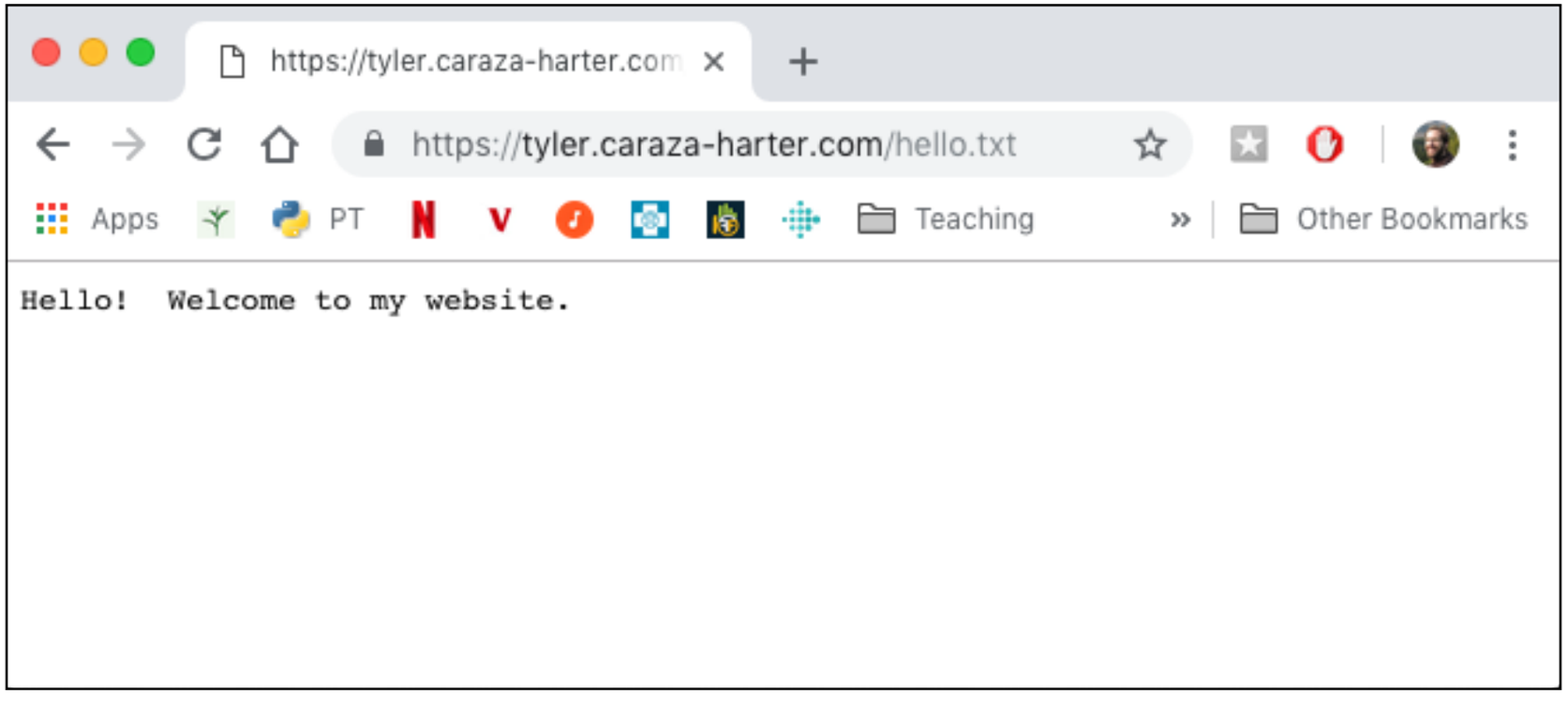
GET Request

```
import requests

url = "https://tyler.caraza-harter.com/hello.txt"

requests.get(url)
```

sends a **GET** request to tyler.caraza-harter.com, asking for the contents of the **/hello.txt** page



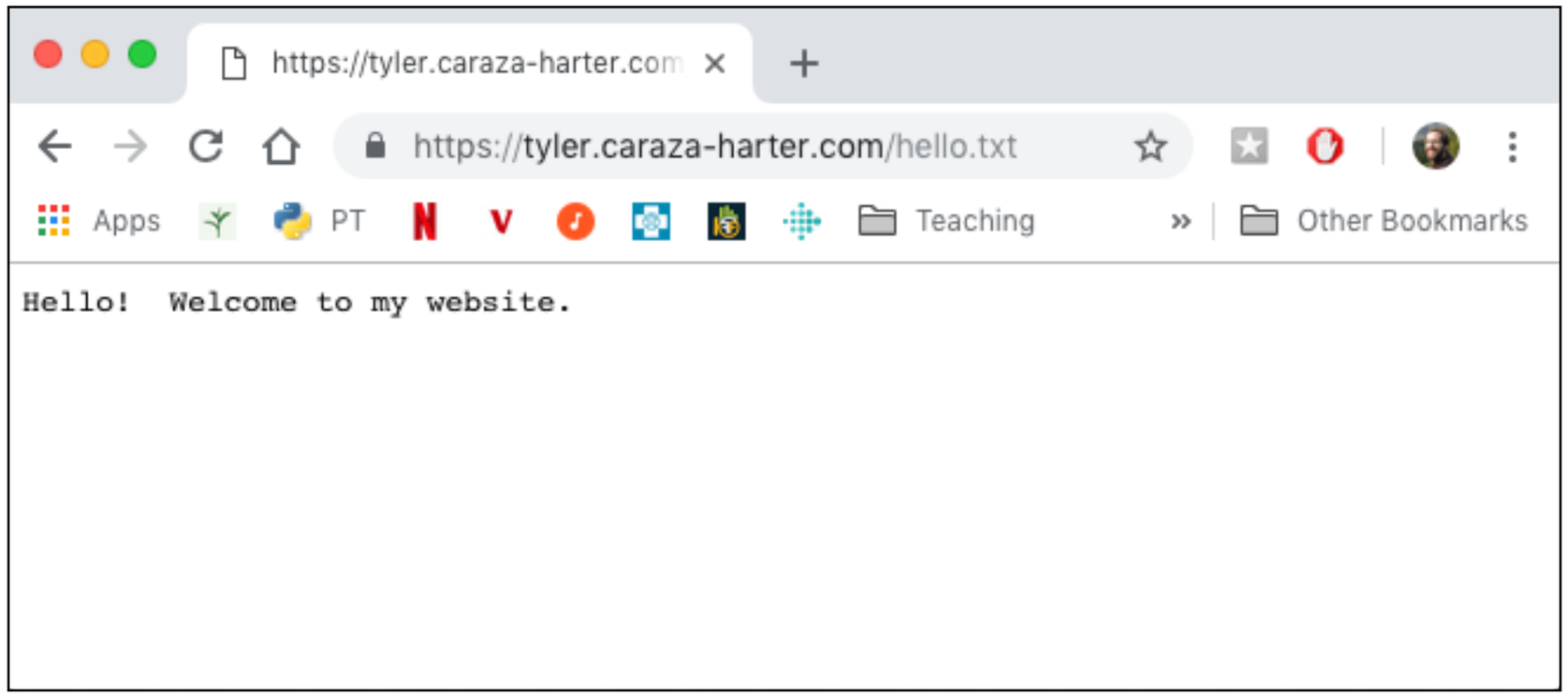
GET Request

```
import requests

url = "https://tyler.caraza-harter.com/hello.txt"

resp = requests.get(url)
```

put response from tyler.caraza-harter.com in the resp variable



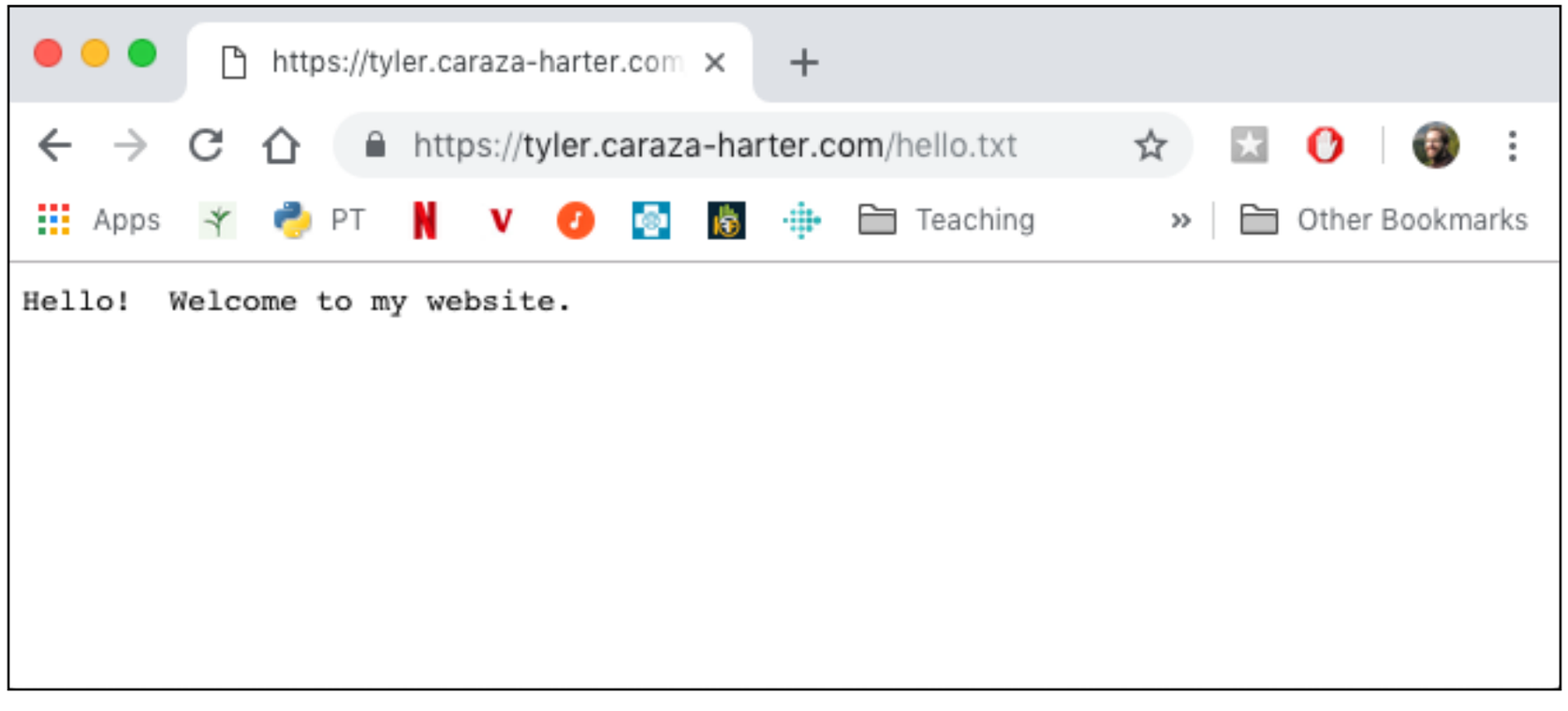
GET Request

```
import requests

url = "https://tyler.caraza-harter.com/hello.txt"

resp = requests.get(url)

# make sure we got 200 (success) back
assert(resp.status_code == 200)
```



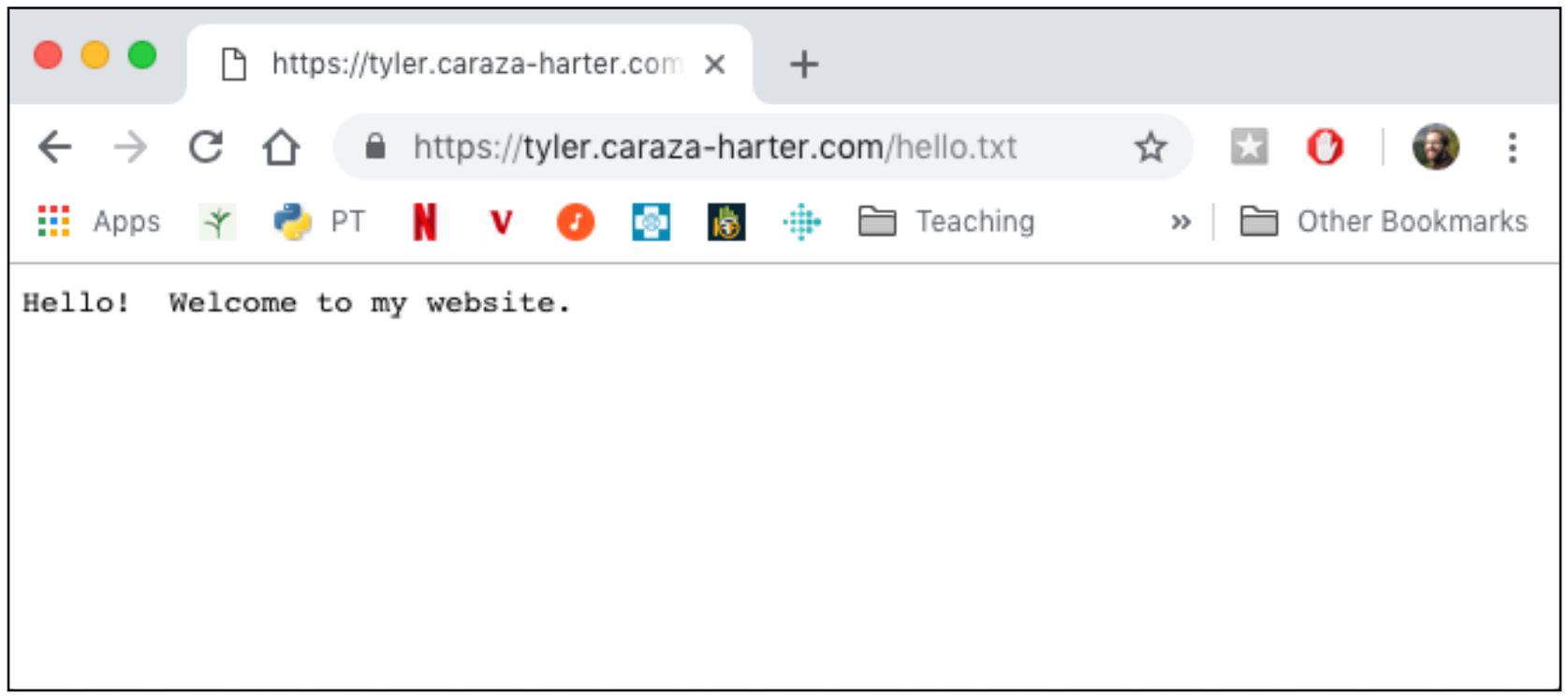
GET Request

```
import requests

url = "https://tyler.caraza-harter.com/hello.txt"

resp = requests.get(url)

resp.raise_for_status() # shortcut
```



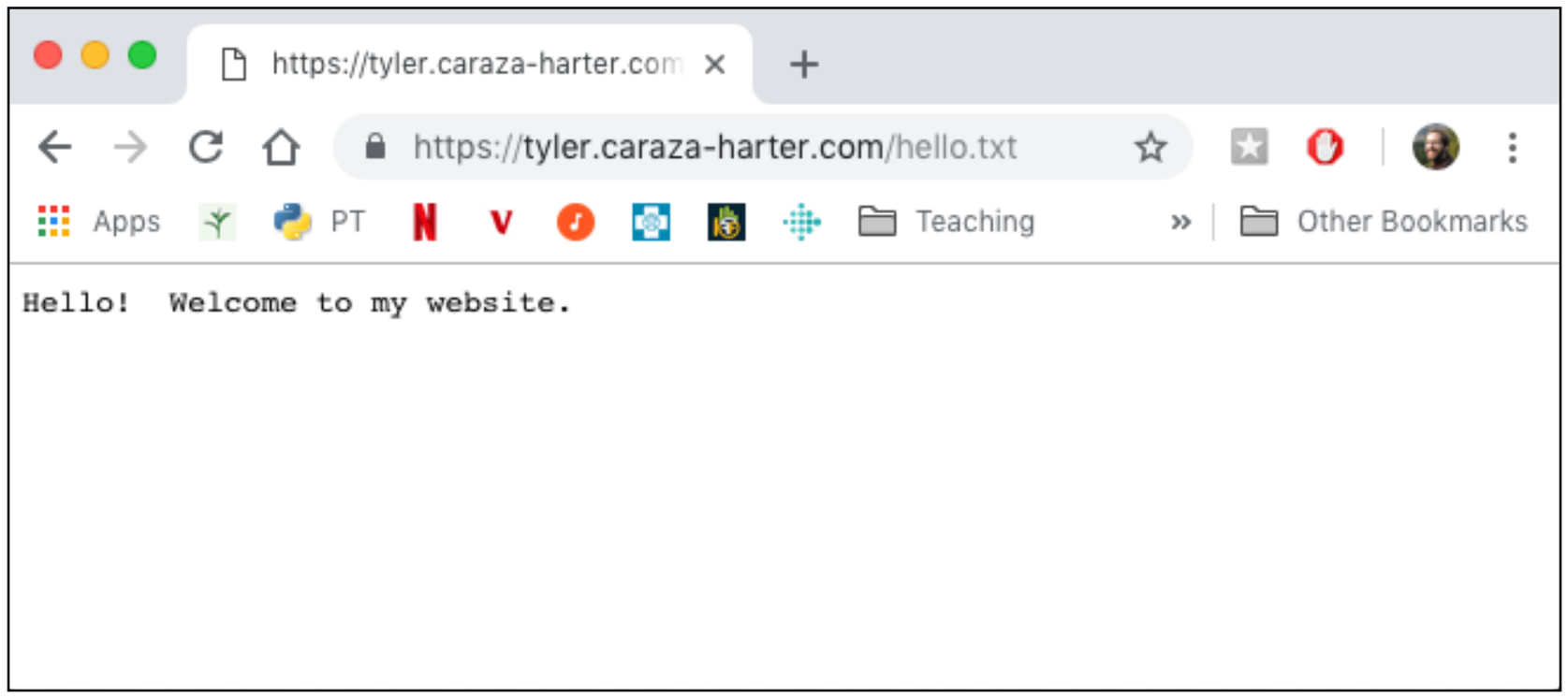
GET Request

```
import requests

url = "https://tyler.caraza-harter.com/hello.txt"

resp = requests.get(url)

resp.raise_for_status() # shortcut
print(resp.text) # "Hello! Welcome to my website."
```

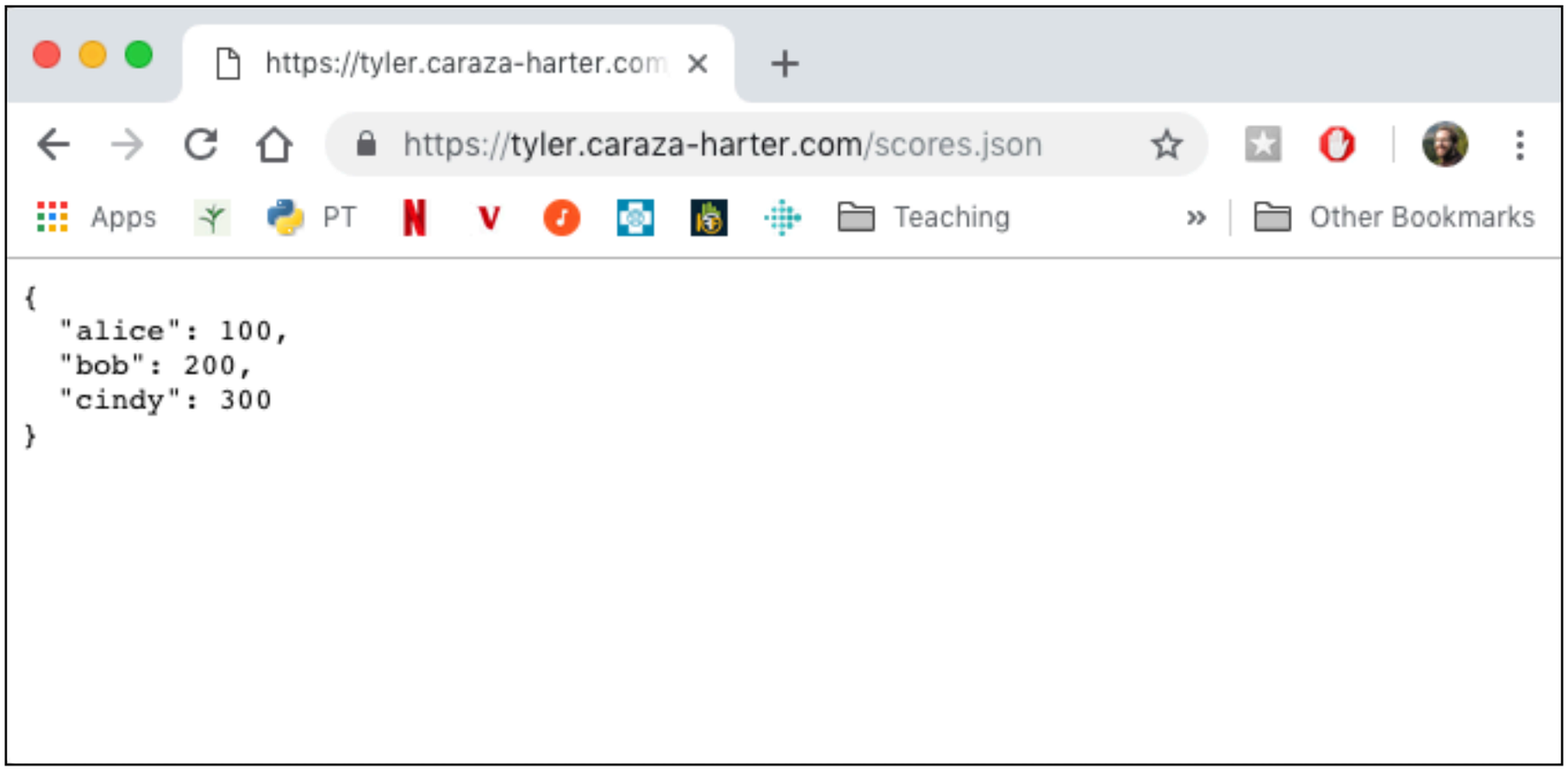


JSON Responses

```
import requests, json

url = "https://tyler.caraza-harter.com/scores.json"
resp = requests.get(url)

scores = json.loads(resp.text)
```

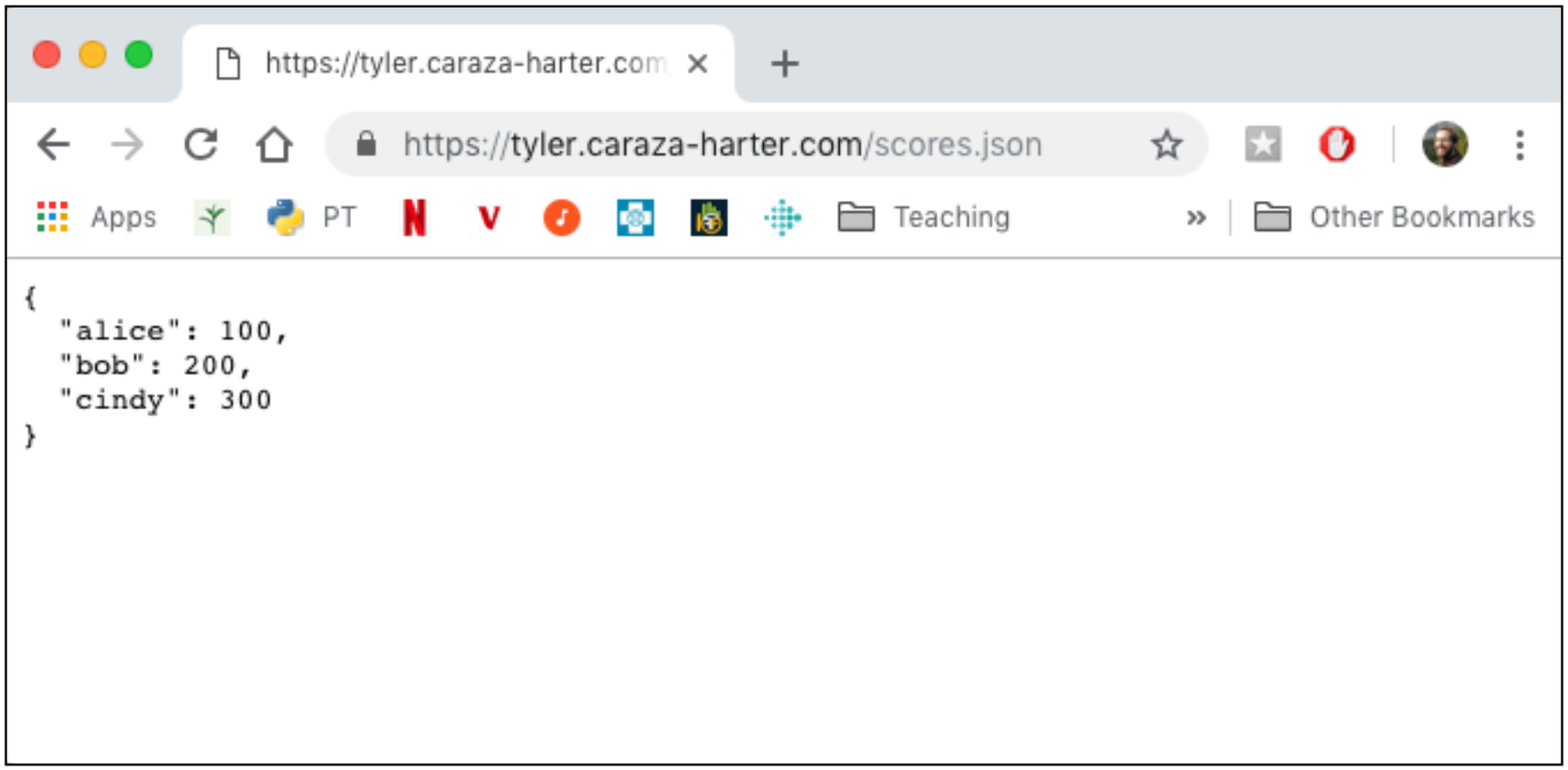


JSON Responses

```
import requests, json

url = "https://tyler.caraza-harter.com/scores.json"
resp = requests.get(url)

scores = json.loads(resp.text)
scores = resp.json() # shortcut
```



Demo 1: reddit bot

Goal: fetch titles from a subreddit

```python
r = requests.get("https://www.reddit.com/r/UWMadison.json")
r.raise_for_status()
page = r.json()
for child in page["data"]["children"]:
    print(child["data"]["title"])
```

```
[Mod Post] /r/UWMadison feedback thread
Any other aquariums on campus besides the one in Birge Hall?
Is there any way to get an Access mental health appointment within a week?
Intermediate/Advanced 3-4 Credit L+S Class Recommendation
Looking for an artist/band to play a house show
Lost my wallet
Looking for Fall2020 semester short term lease
Odds I get into Madison
Looking for an easy study abroad summer program
When would we know which sections Professors are teaching
Does anyone have experience in MS Biology programs?
Question
Are you or anyone you know doing exciting research on environmental issues?
```

Let's not all hit reddit at once (feel free to use these snapshots):

https://tyler.caraza-harter.com/cs301/fall19/materials/code/lec-31/python.json

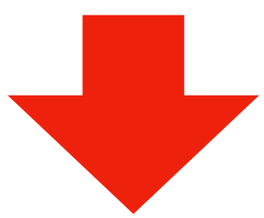
https://tyler.caraza-harter.com/cs301/fall19/materials/code/lec-31/UWMadison.json

Demo 2: Madison bus alerts

Goal: get text of all outstanding alerts

```python
r = requests.get("http://transitdata.cityofmadison.com/GTFS-RealTime/TrapezeRealTimeFeed.json")
d = r.json()

for row in d["entity"]:
    if row["alert"] != None:
        print(row["alert"]["description_text"]["translation"][0]["text"])
```



```
Trips temporarily stop on the west side of N Mills, north of W Johnson-thru Nov 12
Trips skip stops along Lien, between E Washington and Thierer-thru Nov 17 @ https://bit.ly/2xuxUUD
Trips skip stops along Dempsey, Davies and Buckeye, between Cottage Grove & USH 51-thru Nov 15 @ https://bit.ly/2QQnr
f7
Trips serve stop along W Johnson at Mills, between Charter & Lake-thru Nov @ https://bit.ly/2I6q5fu
Trips skip stops along Packers & First, between Commercial & E Washington-thru Nov 13
Trips temporarily stop on the west side of N Sherman, north of Roxbury-thru Jul 2020
Trips skip some stops west of Park & south of University (via Mills)-thru 2020 @ https://bit.ly/2Z62YdU
Trips skip stops along Broadway, between Bridge & Hoboken-thru Nov
```

Let's not all hit Madison at once (feel free to use this snapshot):

https://tyler.caraza-harter.com/cs301/fall19/materials/code/lec-31/TrapezeRealTimeFeed.json

Demo 3: State Populations

Goal: fetch population data for all states and provide summary stats

Input:

- List of state files: https://tyler.caraza-harter.com/cs301/fall19/materials/code/lec-31/data/state_files.txt
- The 50 JSON files

Output:

- Stats about population: mean, max, min, etc

Bonus! "cache" results to make reruns of notebook faster

```
In [19]: df.describe().astype(int)
```

Out[19]:

	2000	2010	2015
count	50	50	50
mean	5616996	6162876	6364951
std	6185579	6848235	7152085
min	493782	563626	584304
25%	1735533	1833004	1857308
50%	4026890	4436369	4530803
75%	6281944	6680312	6986155
max	33871648	37253956	38792291