

Python & the Web

April 27, 2021

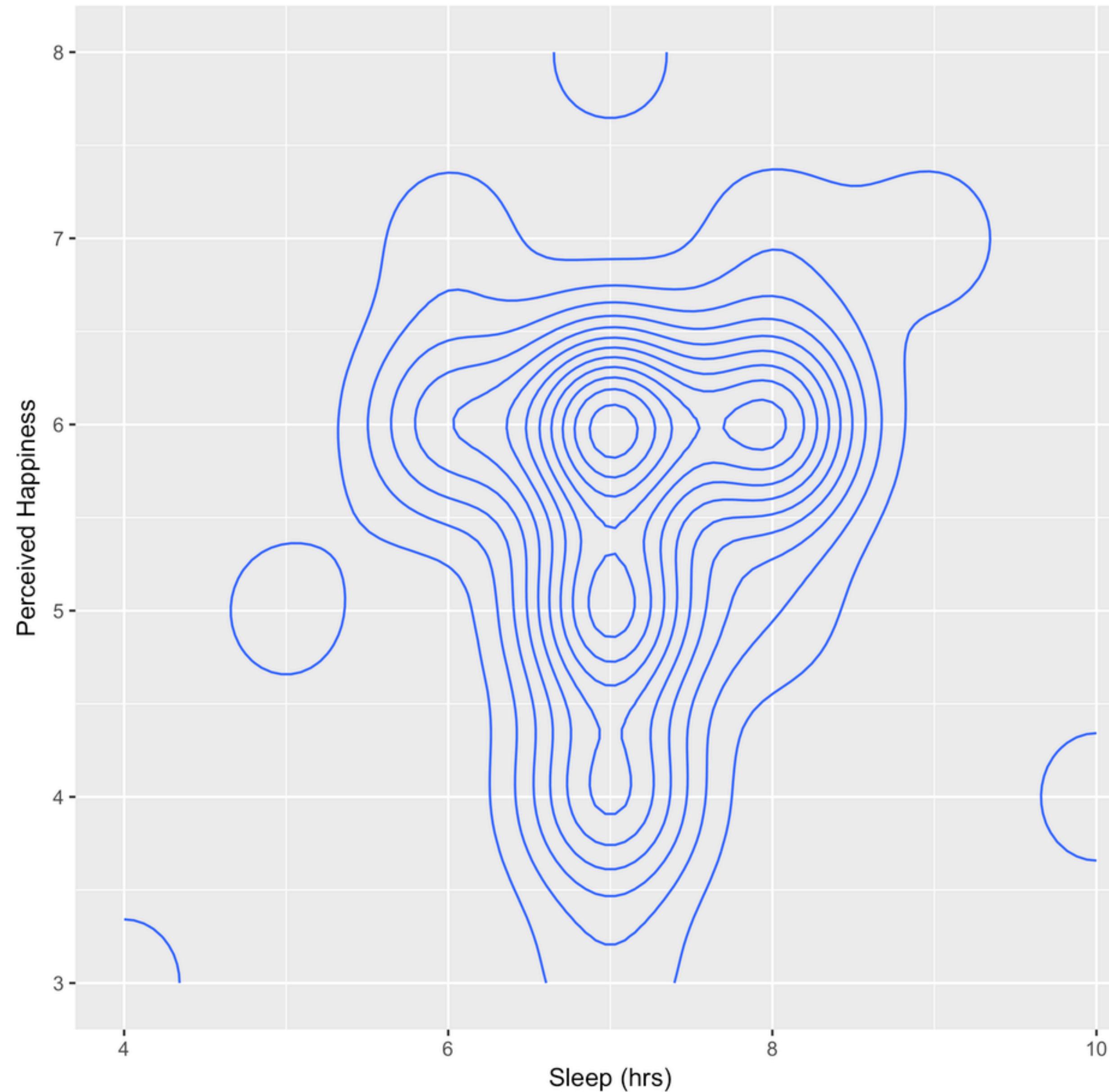
Dear Data



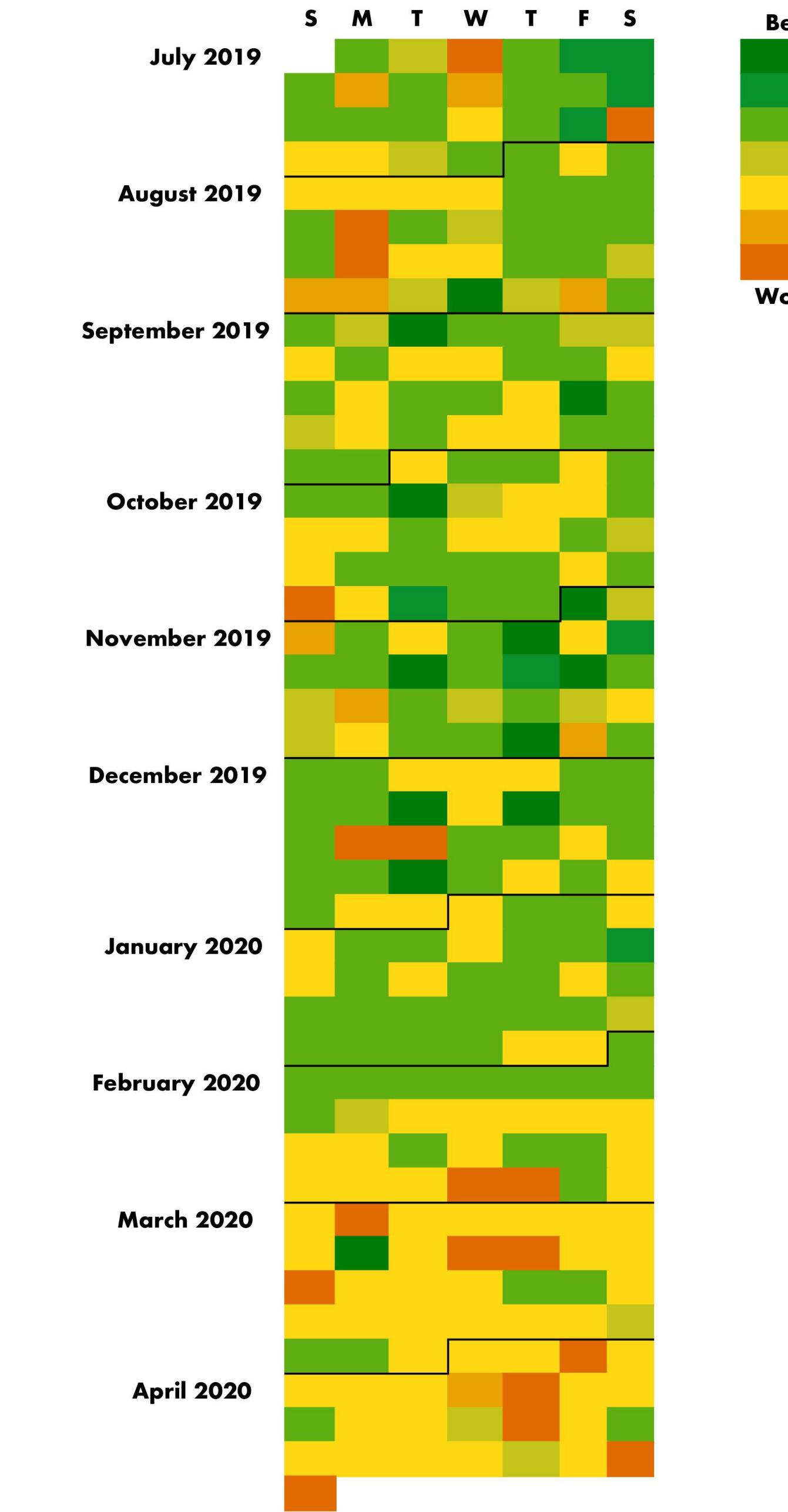
This graph shows the length of daily delays on one woman's 40-minute commute between Munich and a town in the German countryside during 2018. The commuter knitted two rows each day. Gray for delays under five minutes, pink for up to 30 minutes, and red for a delay of more than a half-hour or delays in both directions.

Credit: [The New York Times](#)

Hours of Sleep vs. Perceived Happiness



Credit: [u/avgsuperhero](#)

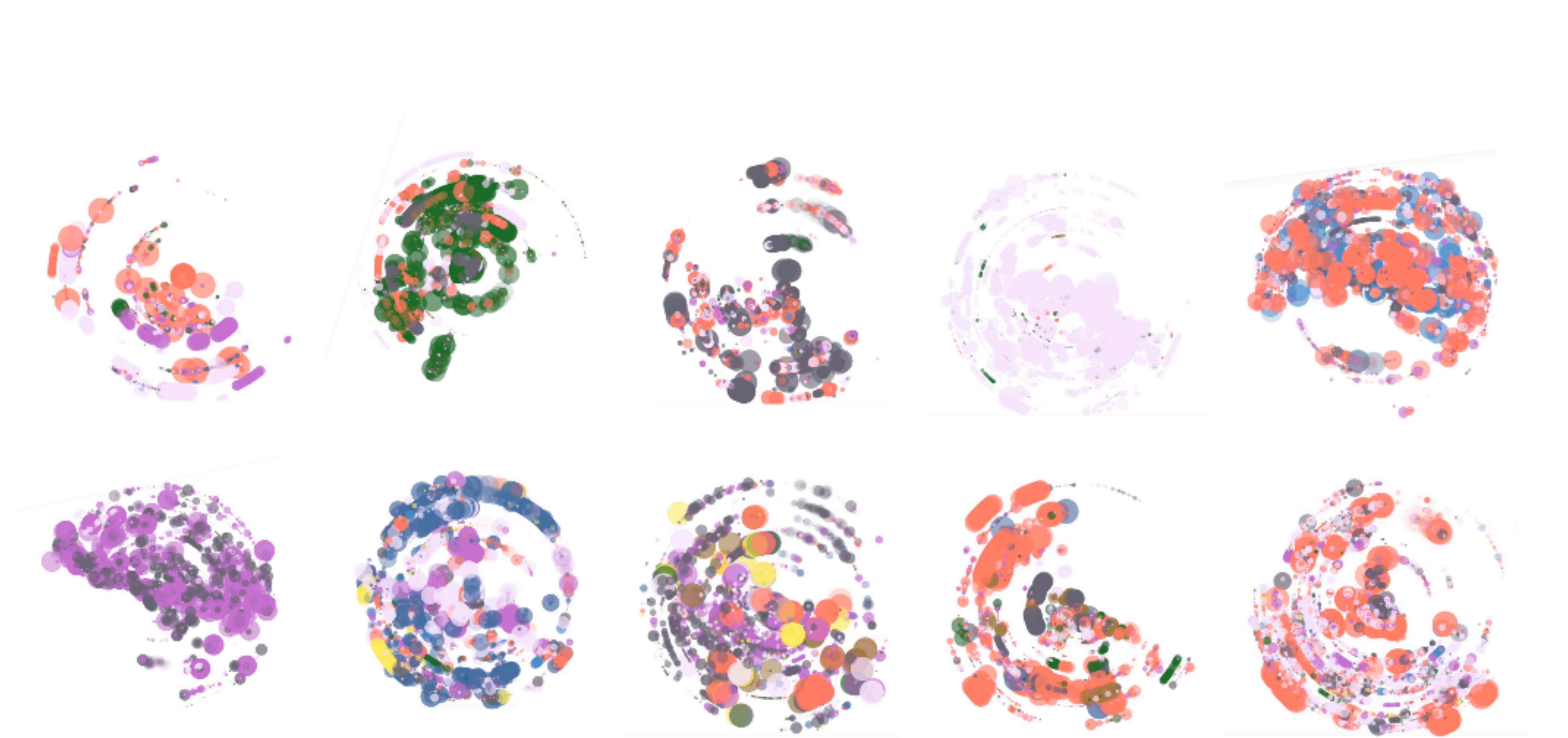


Credit: [u/joebidengarlicbread](#)

More mood visualizations:
[Artistic, aggregate visualizations](#)
[Using Spotify listening data as a proxy for mood](#)



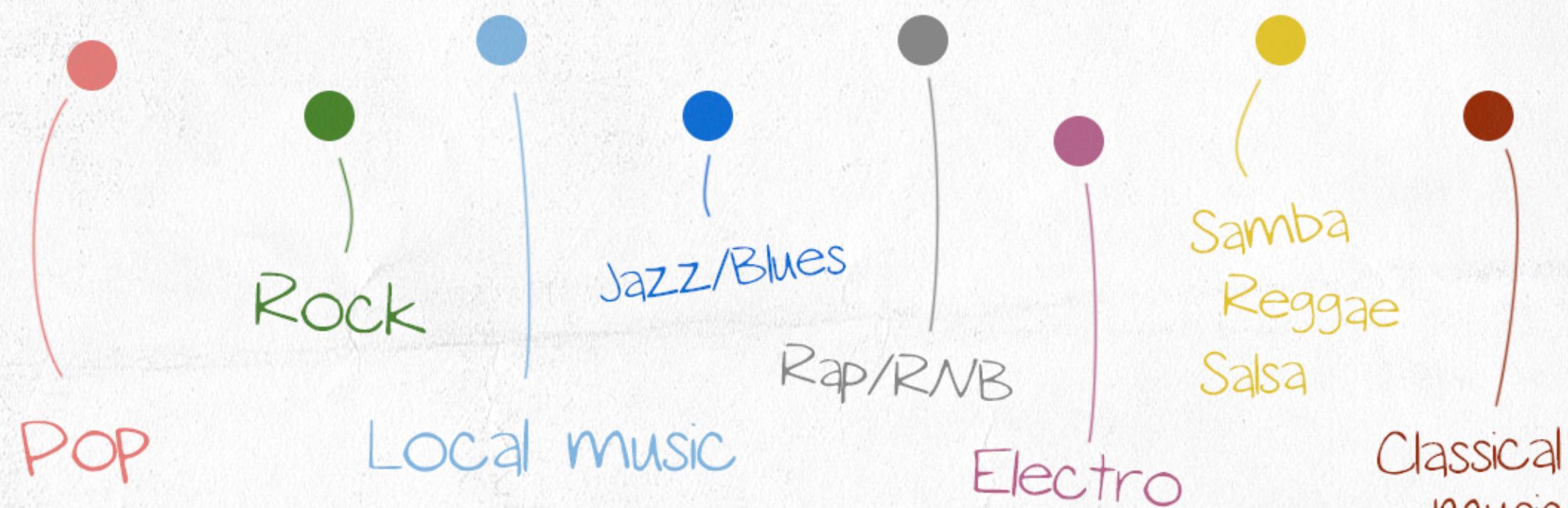
<https://www.robgreenfield.org/trashme/>



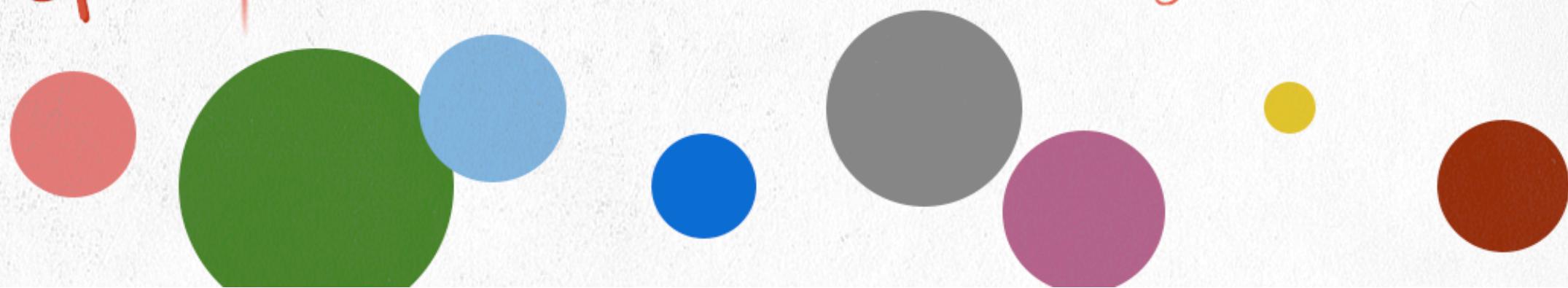
Step1 | 1 listen = 1 bubble



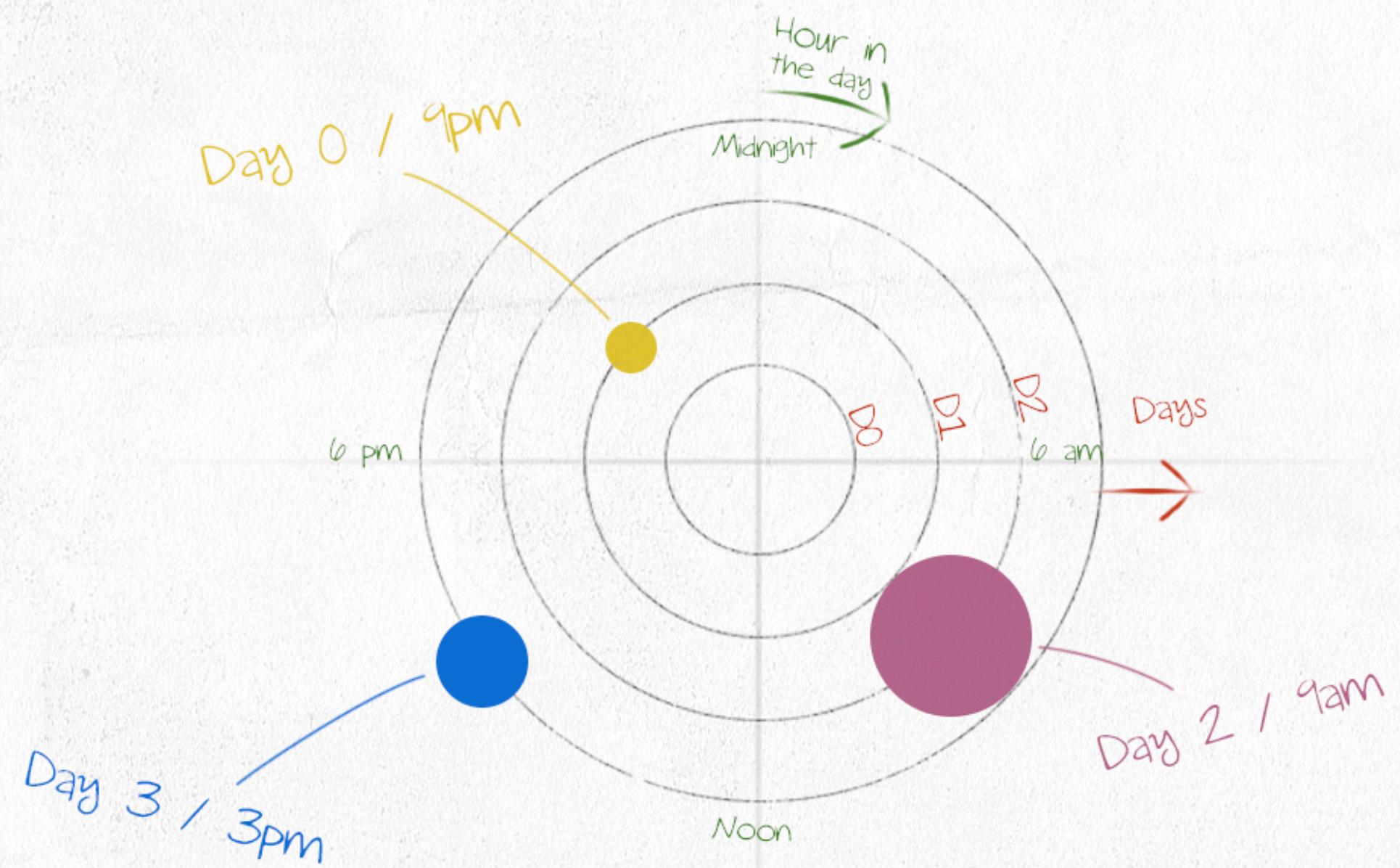
Step2 | Color the bubbles according to genre



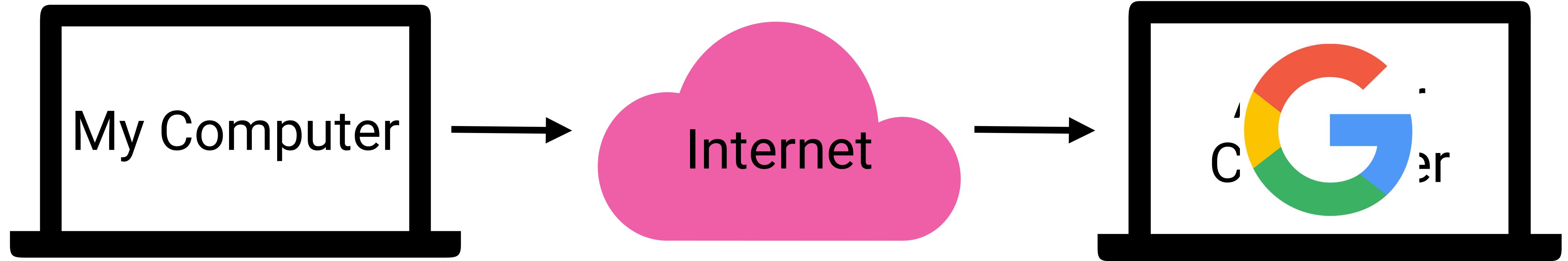
Step3 | Size the bubbles according to stickiness

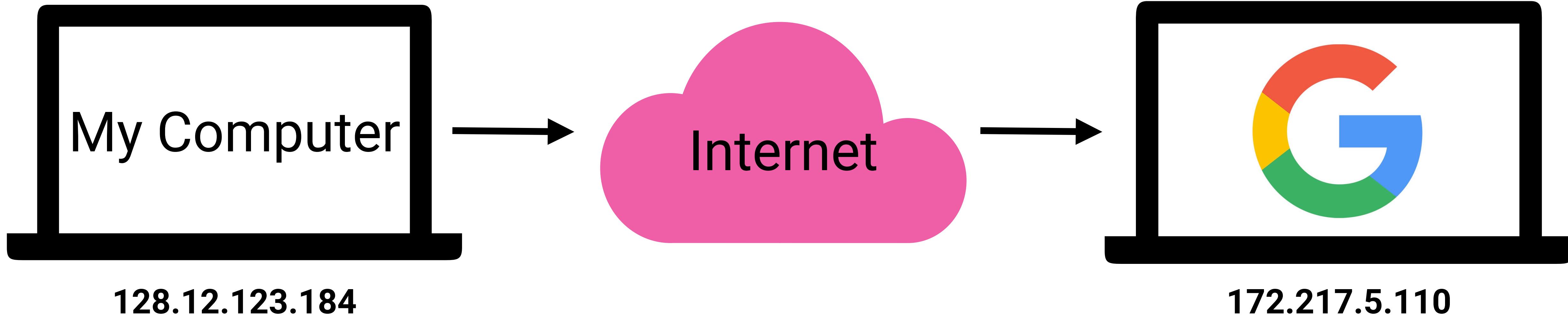


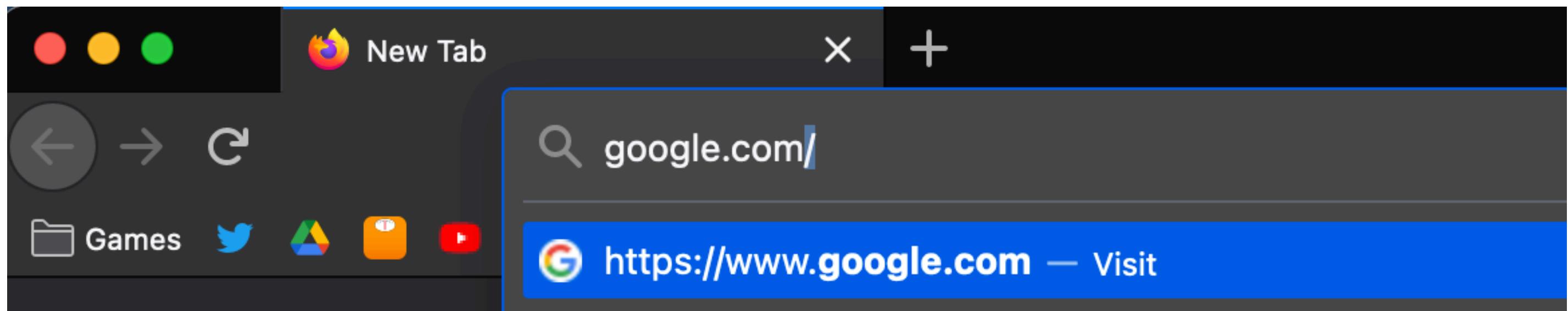
Step4 | Place the bubbles



How the Internet Works





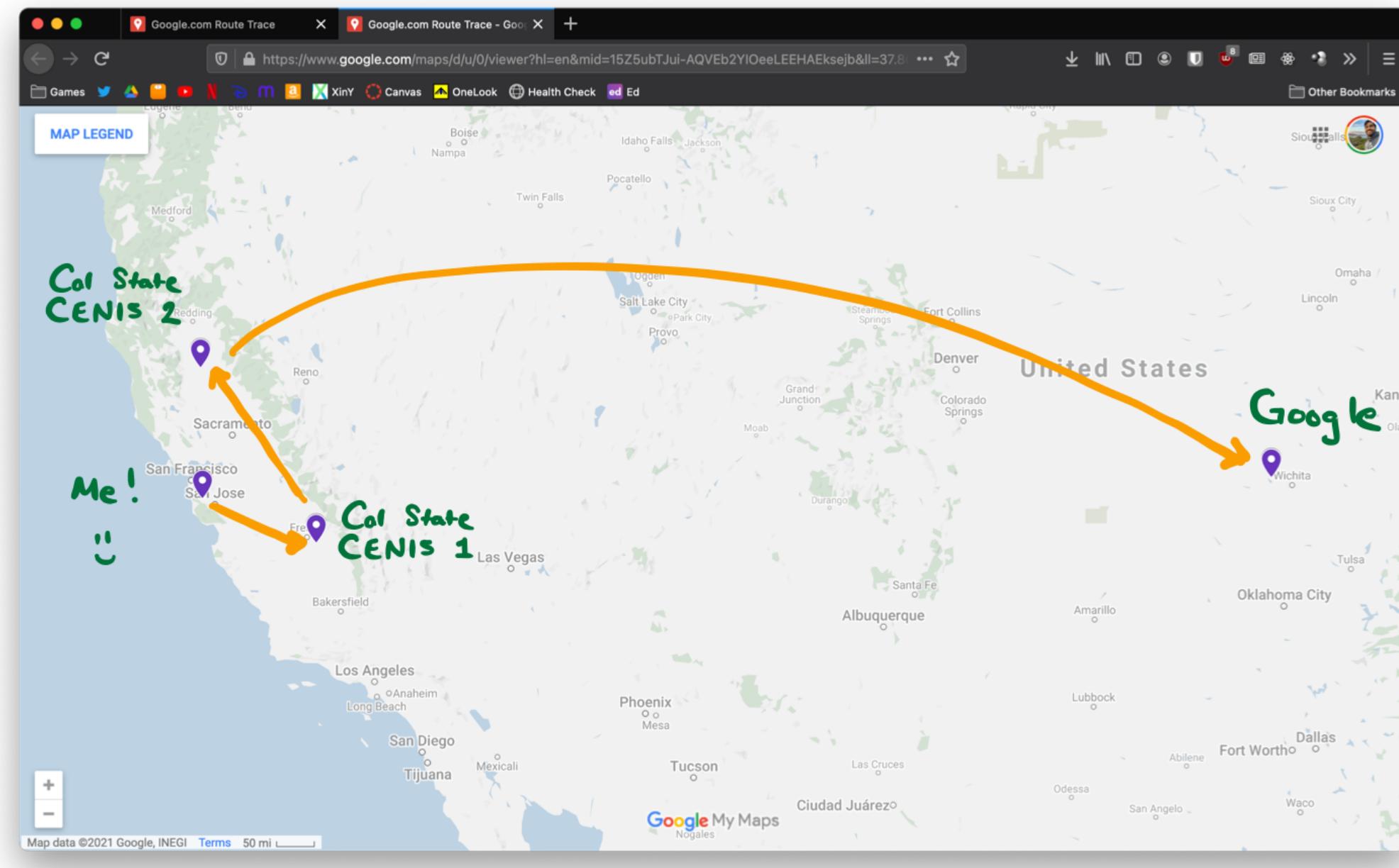


172.217.5.110

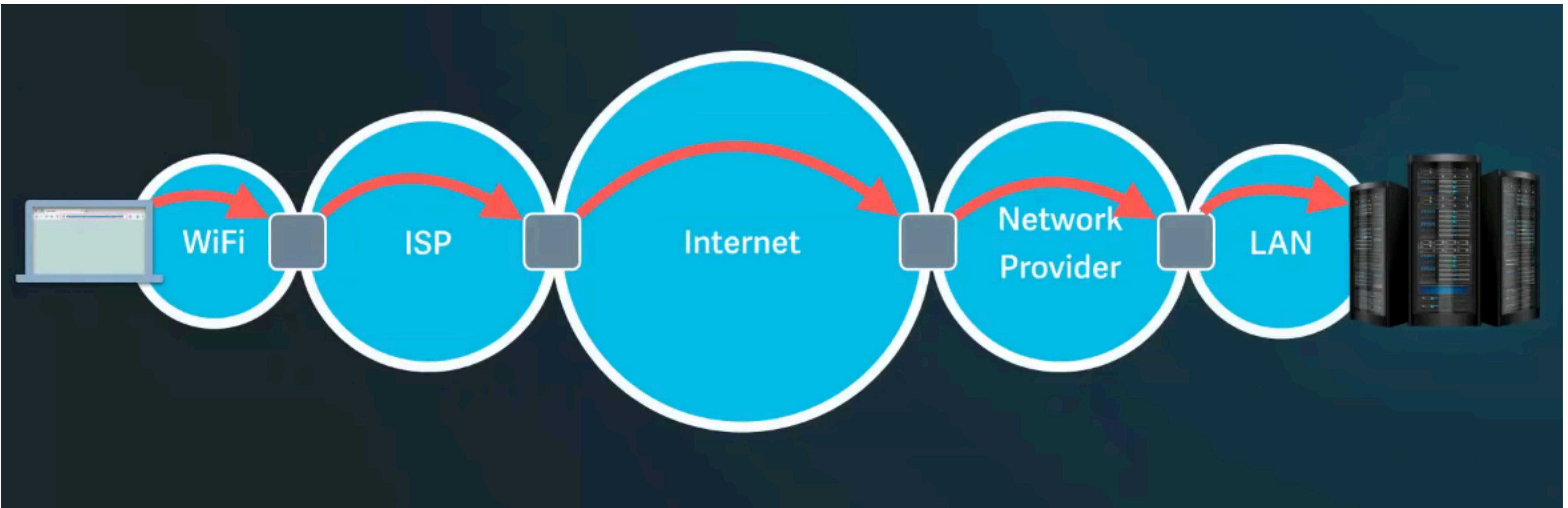
I enter google.com into
the browser bar

"DNS"

That name gets
converted into the
computer's IP address



The request actually
goes to Google



```
$ traceroute google.com
traceroute to google.com (172.217.5.110), 64 hops max, 52 byte packets
 1 sc-srtr-vl3566.sunet (128.12.72.1)  1.239 ms  0.676 ms  0.668 ms
 2 res-press-mx240-temp-rtr.mgmt.sunet (192.168.190.164)  2.018 ms  1.964 ms  1.242 ms
 3 xb-nw-rtr-vlan11.sunet (171.64.0.193)  0.947 ms  1.031 ms  0.936 ms
 4 dc-svl-rtr-vl3.sunet (171.66.255.190)  1.860 ms
  dc-svl-rtr-vl2.sunet (171.64.255.190)  1.716 ms
  dc-svl-rtr-vl3.sunet (171.66.255.190)  1.288 ms
 5 dc-svl-agg4--stanford-100ge.cenic.net (137.164.23.144)  2.958 ms  2.271 ms  2.171 ms
 6 dc-lax-agg8--svl-agg8--100ge--2.cenic.net (137.164.11.20)  9.889 ms  9.768 ms  9.540 ms
 7 dc-lax-agg10--lax-agg8-300g.cenic.net (137.164.11.70)  9.157 ms  9.014 ms  9.122 ms
 8 74.125.49.165 (74.125.49.165)  9.255 ms
    72.14.222.56 (72.14.222.56)  9.481 ms  9.103 ms
 9 108.170.247.147 (108.170.247.147)  14.406 ms  10.177 ms *
10 108.170.230.131 (108.170.230.131)  10.513 ms  10.146 ms *
11 172.253.70.149 (172.253.70.149)  10.123 ms
  172.253.70.151 (172.253.70.151)  12.034 ms
  172.253.70.149 (172.253.70.149)  14.035 ms
12 * 142.250.210.184 (142.250.210.184)  13.104 ms
  72.14.237.161 (72.14.237.161)  10.617 ms
13 108.170.242.225 (108.170.242.225)  10.704 ms
  172.253.70.153 (172.253.70.153)  10.751 ms
  172.253.70.149 (172.253.70.149)  9.995 ms
14 108.170.236.63 (108.170.236.63)  9.821 ms
  108.170.236.61 (108.170.236.61)  11.795 ms
  142.250.210.184 (142.250.210.184)  12.378 ms
15 sfo03s07-in-f110.1e100.net (172.217.5.110)  12.449 ms  10.265 ms  10.364 ms
```



 CENIC

ABOUT NETWORK COMMUNITY NEWS EVENTS PUBLICATIONS BLOG

Established in 1997, this nonprofit organization operates the California Research and Education Network (CalREN), a high-capacity computer network with more than 8,000 miles of optical fiber. The network serves over 20 million users across California, including the vast majority of K-20 students together with educators, researchers, and individuals at other vital public-serving institutions.

CENIC connects California to the world — advancing education and research statewide by providing a world-class network essential for innovation, collaboration, and economic growth.



```
$ traceroute google.com
traceroute to google.com (172.217.5.110), 64 hops max, 52 byte packets
 1 sc-srtr-vl3566.sunet (128.12.72.1)  1.239 ms  0.676 ms  0.668 ms
 2 res-press-mx240-temp-rtr.mgmt.sunet (192.168.190.164)  2.018 ms  1.964 ms  1.242 ms
 3 xb-nw-rtr-vlan11.sunet (171.64.0.193)  0.947 ms  1.031 ms  0.936 ms
 4 dc-svl-rtr-vl3.sunet (171.66.255.190)  1.860 ms
dc-svl-rtr-vl2.sunet (171.64.255.190)  1.716 ms
dc-svl-rtr-vl3.sunet (171.66.255.190)  1.288 ms
 5 dc-svl-agg4--stanford-100ge.cenic.net (137.164.23.144)  2.958 ms  2.271 ms  2.171 ms
 6 dc-lax-agg8--svl-agg8--100ge--2.cenic.net (137.164.11.20)  9.889 ms  9.768 ms  9.540 ms
 7 dc-lax-agg10--lax-agg8-300g.cenic.net (137.164.11.70)  9.157 ms  9.014 ms  9.122 ms
 8 74.125.49.165 (74.125.49.165)  9.255 ms
72.14.222.56 (72.14.222.56)  9.481 ms  9.103 ms
 9 108.170.247.147 (108.170.247.147)  14.406 ms  10.177 ms *
10 108.170.230.131 (108.170.230.131)  10.513 ms  10.146 ms *
11 172.253.70.149 (172.253.70.149)  10.123 ms
172.253.70.151 (172.253.70.151)  12.034 ms
172.253.70.149 (172.253.70.149)  14.035 ms
12 * 142.250.210.184 (142.250.210.184)  13.104 ms
72.14.237.161 (72.14.237.161)  10.617 ms
13 108.170.242.225 (108.170.242.225)  10.704 ms
172.253.70.153 (172.253.70.153)  10.751 ms
172.253.70.149 (172.253.70.149)  9.995 ms
14 108.170.236.63 (108.170.236.63)  9.821 ms
108.170.236.61 (108.170.236.61)  11.795 ms
142.250.210.184 (142.250.210.184)  12.378 ms
15 sfo03s07-in-f110.1e100.net (172.217.5.110)  12.449 ms  10.265 ms  10.364 ms
```



CENIC



Week 5 - Google Drive | Lesson Plan - Python and the W | the duolingo owl - Griefbacon | IP Address Lookup for 137.164.23.144 | +

https://whatismyipaddress.com/ip/137.164.23.144

Games Twitter Google YouTube Netflix M a XinY Canvas OneLook Health Check Ed Ed Other Bookmarks

What's MyIPAddress.com

Enter Keywords or IP Address... Search

ABOUT PRESS BLOG CONTACT

MY IP IP LOOKUP HIDE MY IP VPNS TOOLS LEARN

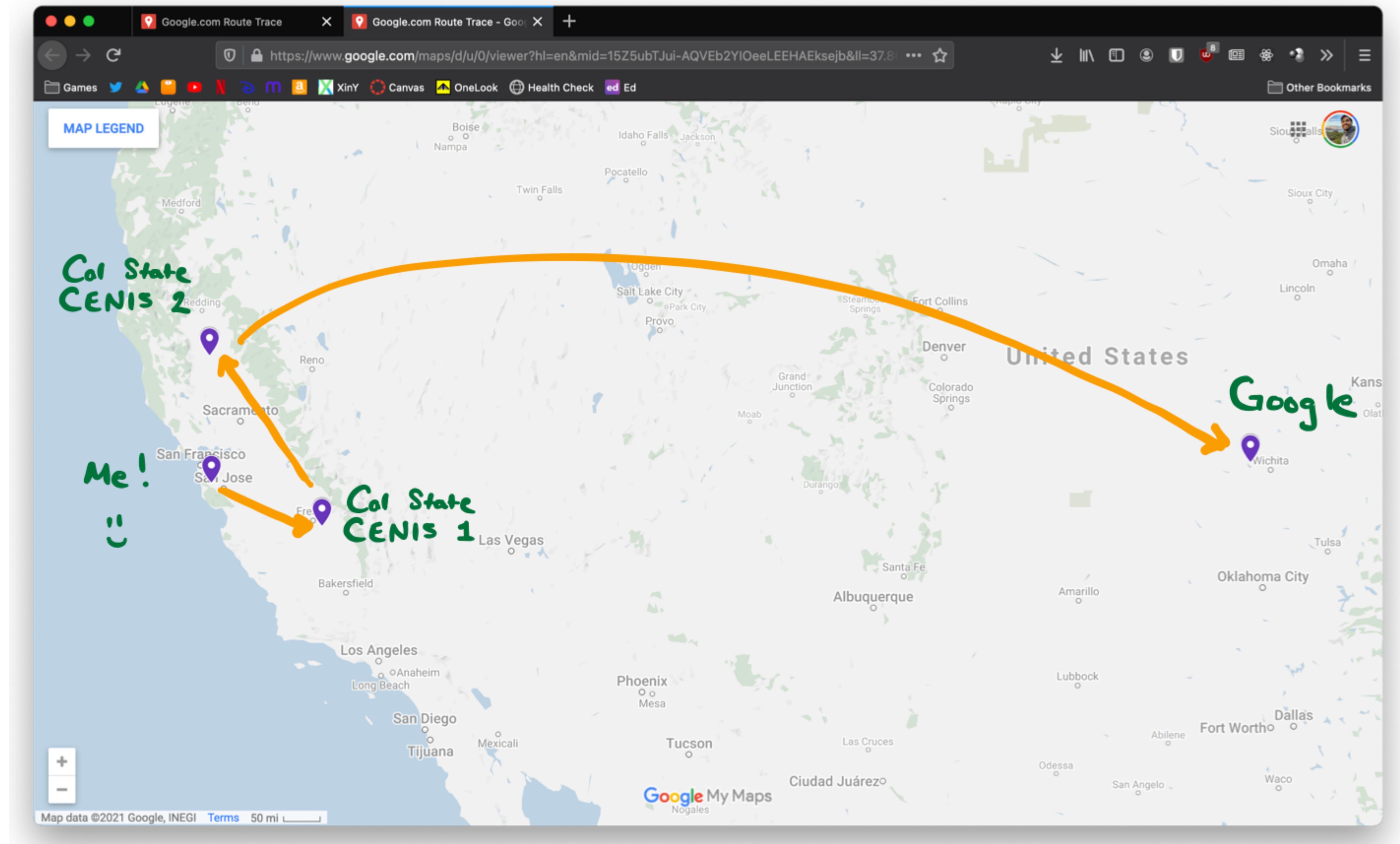
IP Details For: 137.164.23.144

Decimal: 2309232528
Hostname: dc-svl-agg4--stanford-100ge.cenic.net
ASN: 2152
ISP: California State University, Office of the Chancel
Organization: California State University, Office of the Chancel
Services: None detected
Type: Broadband
Assignment: Likely Static IP
Continent: North America
Country: United States
State/Region: California
City: Parlier

Latitude: 36.6089 (36° 36' 32.04" N)
Longitude: -119.5416 (119° 32' 29.76" W)
Postal Code: 93648

CLICK TO CHECK BLACKLIST STATUS





What's actually being sent?

GET / HTTP/2

Host: www.google.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:88.0) Gecko/20100101 Firefox/88.0

Accept: text/html, application/
xhtml+xml, application/xml;q=0.9, image/webp, */
*;q=0.8

Accept-Language: en-US, en; q=0.5

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0

TE: Trailers

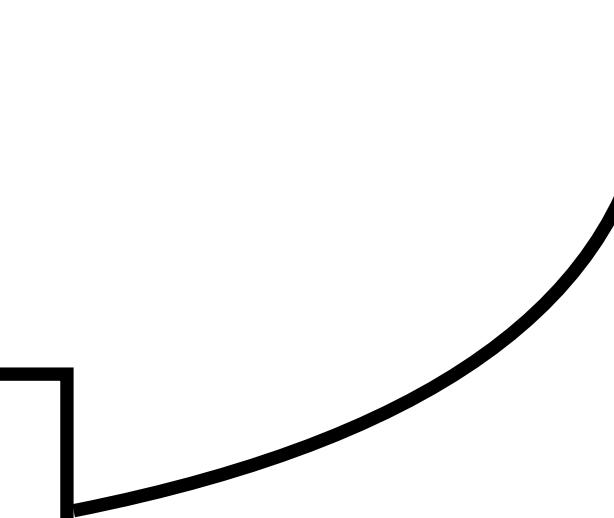
Path

Headers

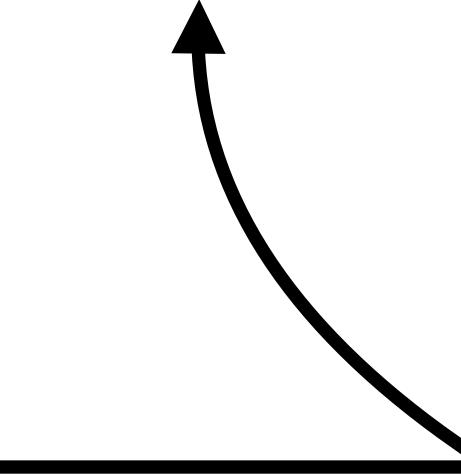
Making Requests with Python

```
import requests  
r = requests.get(url)
```

Sends a GET request for
that URL, with no headers



```
r = requests.get(  
    url,  
    headers={ 'Accept' : 'application/json' }  
)
```

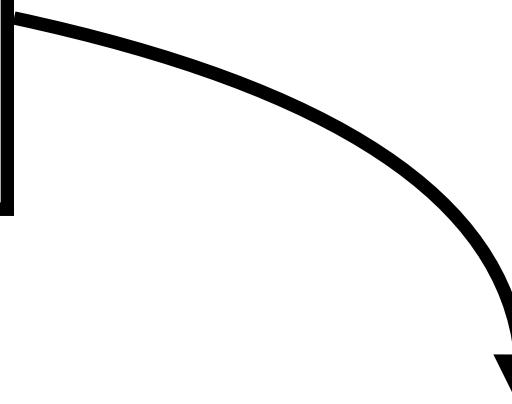


You can specify the headers
as a dictionary if you want

```
r = requests.get(  
    'https://www.google.com/search',  
    params={ 'q': 'unicorns' }  
)
```

This queries [google.com/search?q=unicorns](https://www.google.com/search?q=unicorns)

There are other types of requests—this is a POST request



```
r = requests.post(  
    url,  
    data={'key1': 'value1'}  
)
```

Interpreting the Response

Data on the Web

HTML

Built for web browsers—most of the web pages you browse are probably written in HTML.

JSON

A way to represent objects as strings, not typically displayed to users.

Text

Most webpages these days are not written in raw text, but some servers may respond with raw, unformatted text.

Other

Other files on the web (images, PDFs, etc.) have their own formatting.

JSON

Lots of things on the web use JSON!

```
{  
  "id":317,  
  "type":"general",  
  "setup":"Why are oranges the smartest fruit?",  
  "punchline":"Because they are made to concentrate."  
}
```

```
import requests
r = requests.get(
    'https://official-joke-api.appspot.com/random_joke'
)
This is raw bytes data...
r.content
# => b'{"id":86,"type":"general","setup":"Did you hear about the bread
factory burning down?","punchline":"They say the business is toast."}'

...this is a string
r.text
# => '{"id":86,"type":"general","setup":"Did you hear about the bread
factory burning down?","punchline":"They say the business is toast."}'

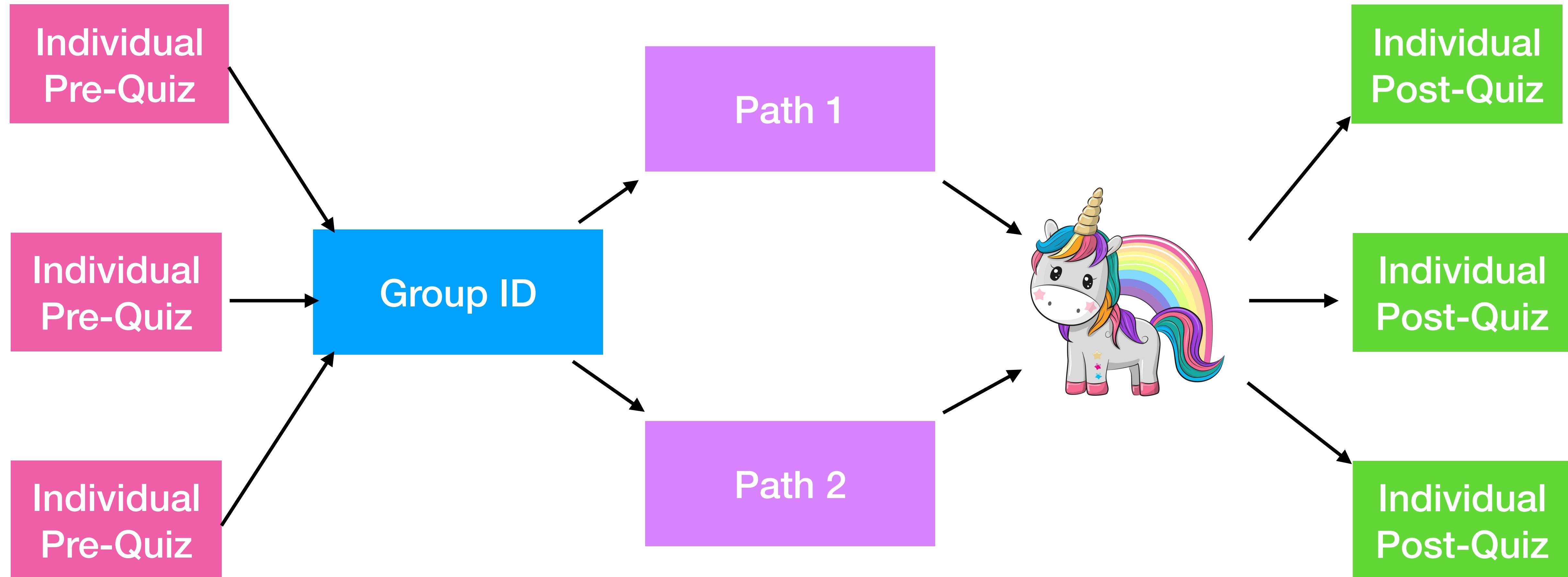
r.json()
# => {'id': 86, 'type': 'general', 'setup': 'Did you hear about the
bread factory burning down?', 'punchline': 'They say the business is
toast.'}
...and this is a dictionary
```

Exploring an API

What's an API?

- API = Application Programming Interface
 - A way for applications (computer programs) to talk to each other
 - Exposes an interface that's not usually user-friendly, but *is* programmer-friendly so that other programs can hook into it.
- For example, the random jokes API: https://official-joke-api.appspot.com/random_joke

Now, let's explore interacting with an API!



Due EOW 8

Building a Server with Flask

Receiving Request Payloads

GET / HTTP/2

Host: www.google.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:88.0) Gecko/20100101 Firefox/88.0

Accept: text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, */*; q=0.8

Accept-Language: en-US, en; q=0.5

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0

TE: Trailers

```
app.py
```

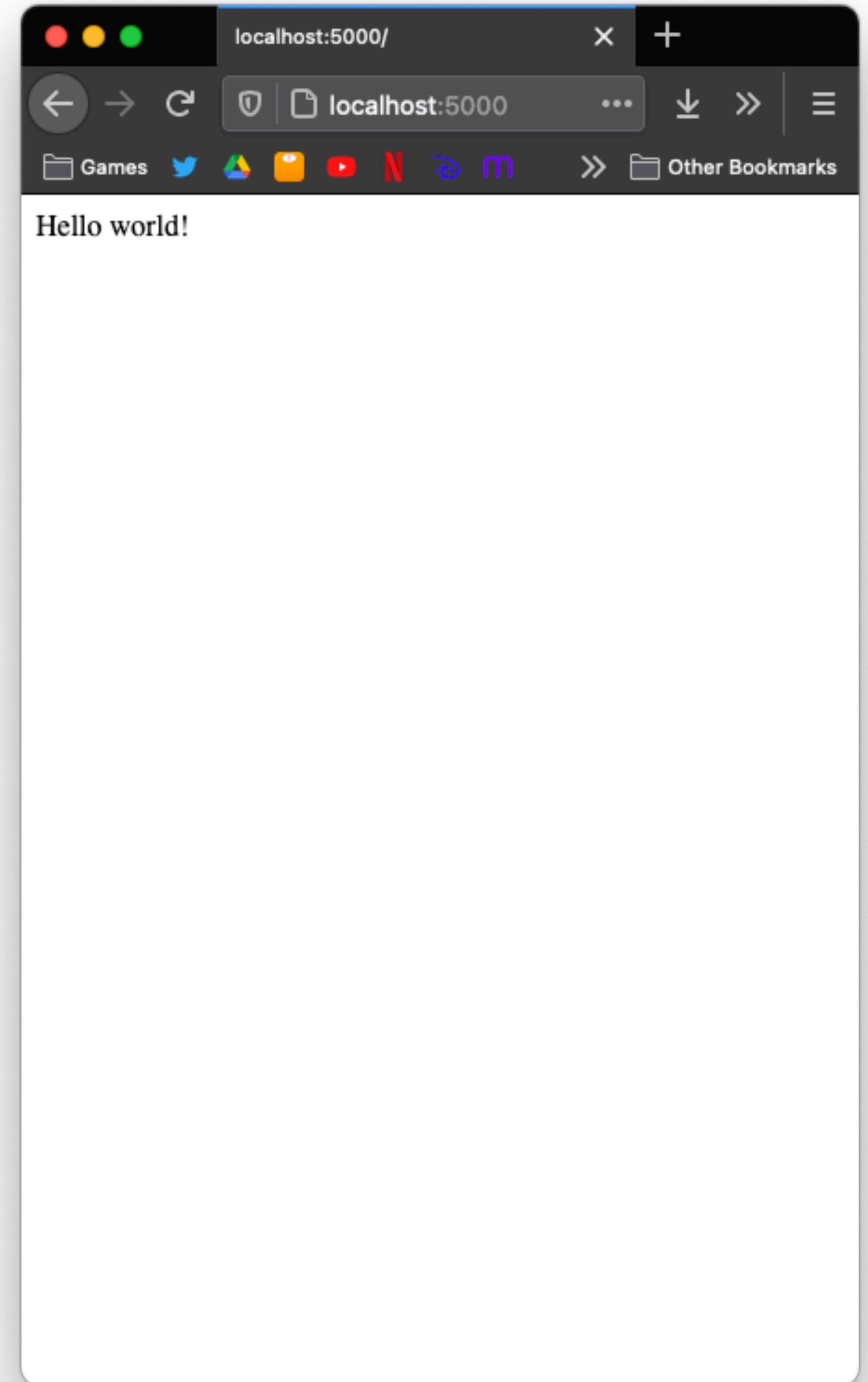
```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/', methods=['GET'])
def index():
    return 'Hello world!'
```

```
$ flask run
```

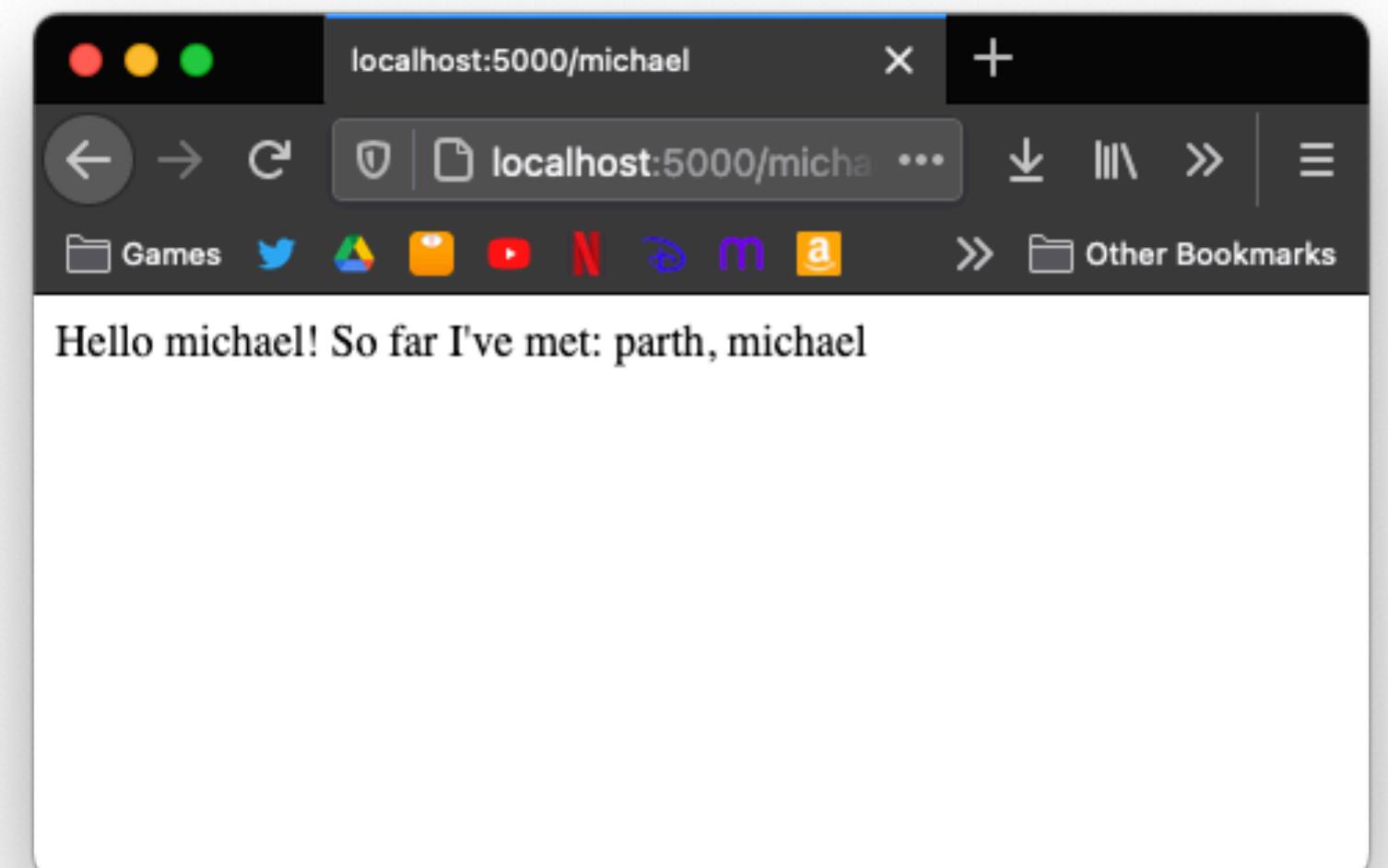
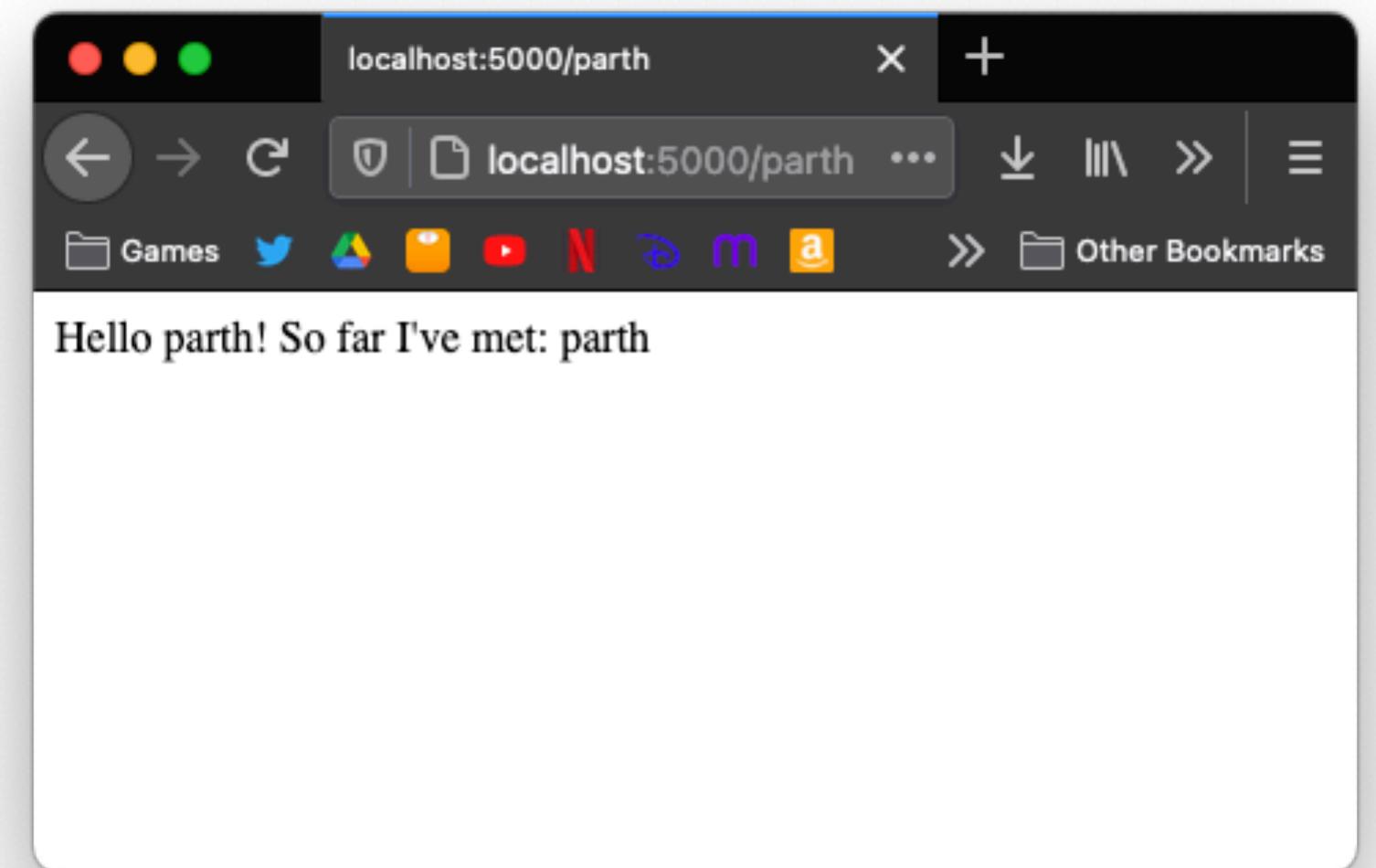
```
* Serving Flask app "app.py" (lazy loading)
* Environment: development
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press
CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 460-241-179
```



```
from flask import Flask
```

```
app = Flask(__name__)
history = []
```

```
@app.route('/<name>', methods=['GET'])
def index(name):
    history.append(name)
    return (
        f"Hello {name}! So far I've met: "
        f"'{', '.join(history)}'"
    )
```



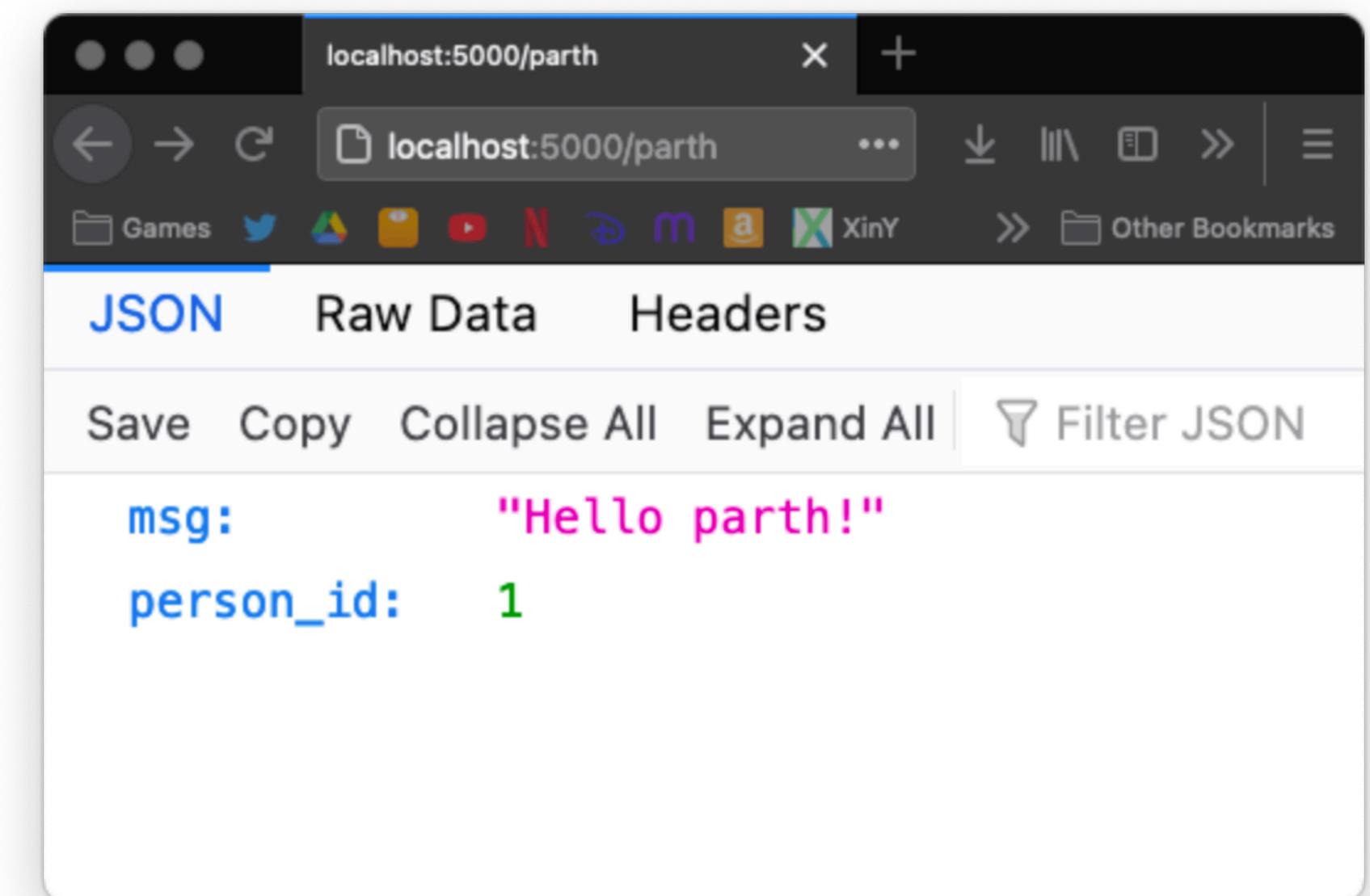
```

from flask import Flask

app = Flask(__name__)
history = []

@app.route('/<name>', methods=['GET'])
def index(name):
    history.append(name)
    return {
        'msg': f"Hello {name}!",
        'person_id': len(history)
    }

```

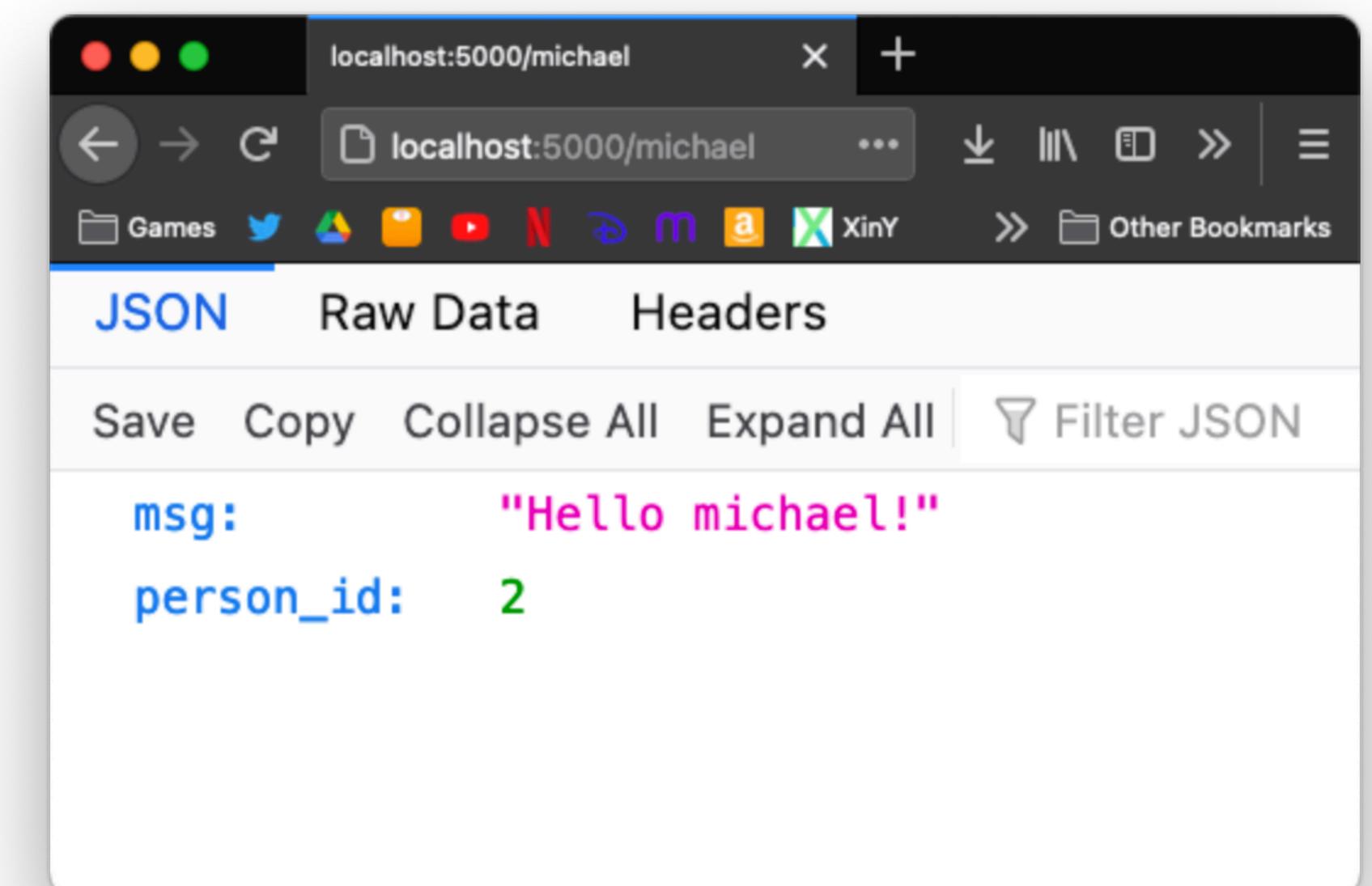


localhost:5000/parth

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

msg:	"Hello parth!"
person_id:	1



localhost:5000/michael

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

msg:	"Hello michael!"
person_id:	2