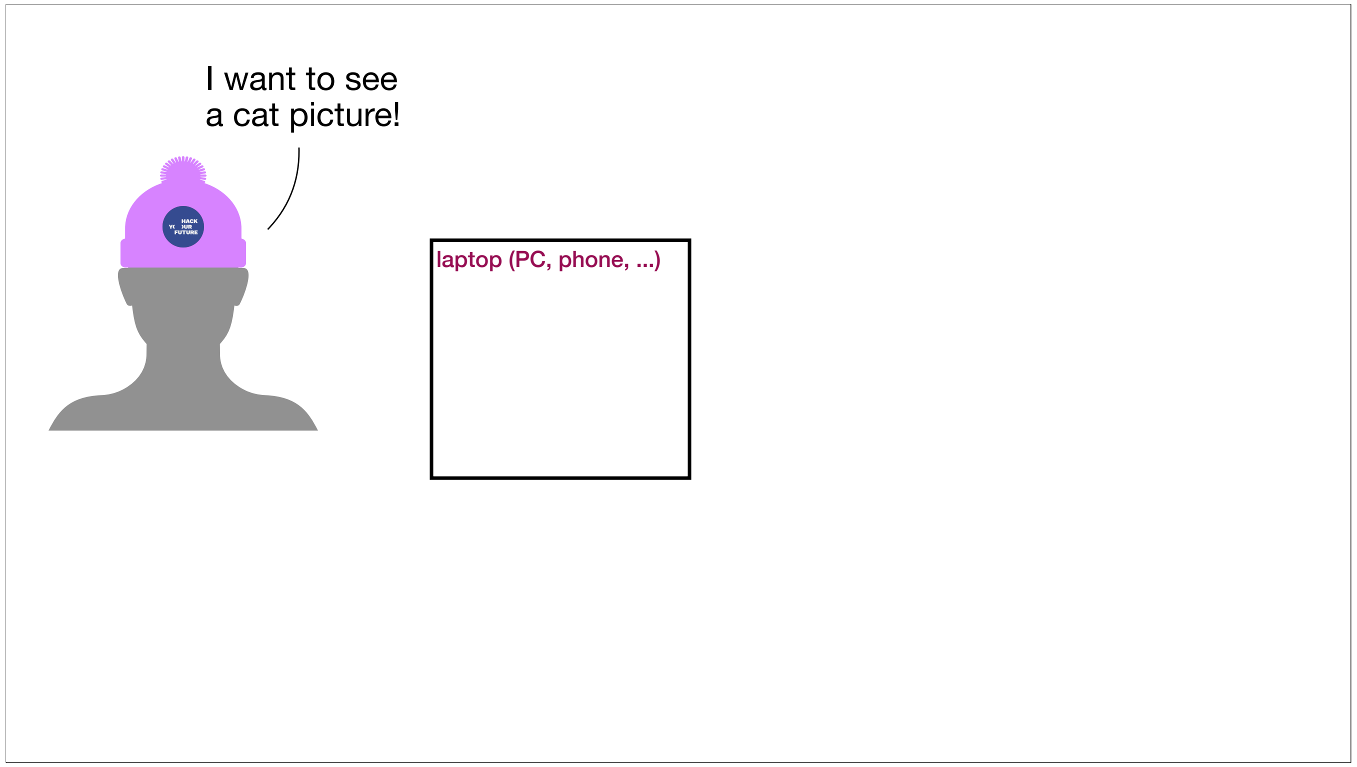
HACK
YOUR
FUTURE

Web Architecture 101

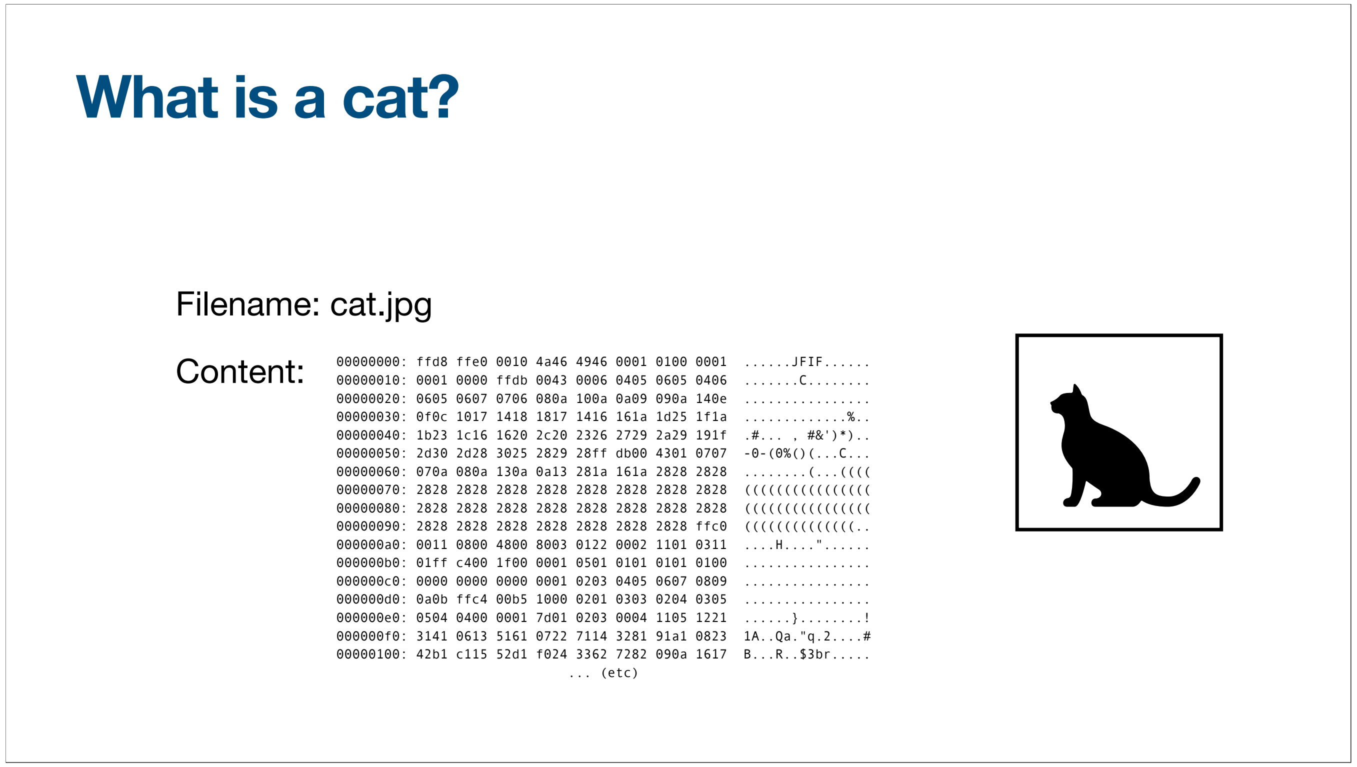
How web sites work

HTTP and Content-Type



You have your computer (laptop, phone, whatever), with a working Internet connection, and you want to see a picture of a cat. How do you do it?

(You already know the exact URL).



What is a cat?

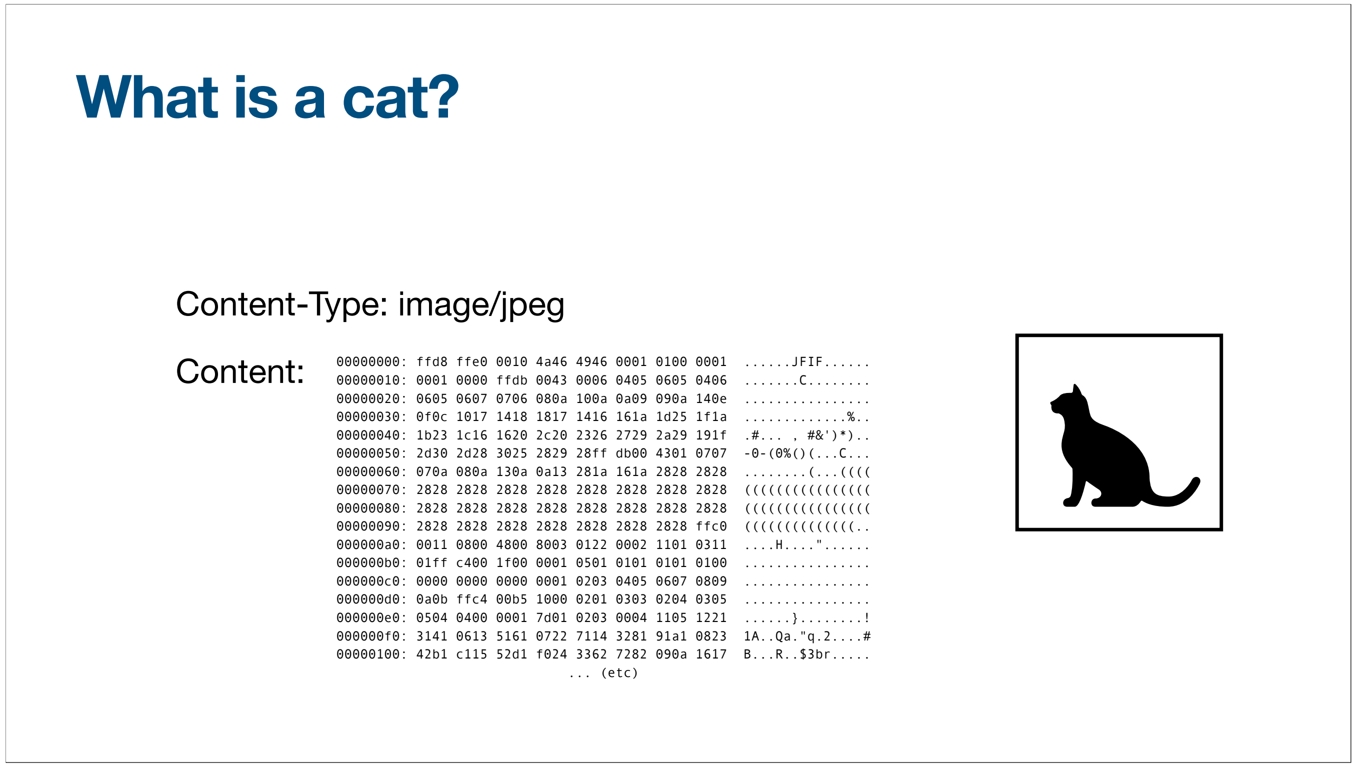
Filename: cat.jpg

Content:

```
00000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001  ......JFIF......
00000010: 0001 0000 ffdb 0043 0006 0405 0605 0406  .......C........
00000020: 0605 0607 0706 080a 100a 0a09 090a 140e  ................
00000030: 0f0c 1017 1418 1817 1416 161a 1d25 1f1a  .............%..
00000040: 1b23 1c16 1620 2c20 2326 2729 2a29 191f  .#... , #&')*)..
00000050: 2d30 2d28 3025 2829 28ff db00 4301 0707  -0-(0%()(...C...
00000060: 070a 080a 130a 0a13 281a 161a 2828 2828  ........(...((((
00000070: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000080: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000090: 2828 2828 2828 2828 2828 2828 2828 ffc0  ((((((((((((((..
000000a0: 0011 0800 4800 8003 0122 0002 1101 0311  ....H...."......
000000b0: 01ff c400 1f00 0001 0501 0101 0101 0100  ................
000000c0: 0000 0000 0000 0001 0203 0405 0607 0809  ................
000000d0: 0a0b ffc4 00b5 1000 0201 0303 0204 0305  ................
000000e0: 0504 0400 0001 7d01 0203 0004 1105 1221  ......}........!
000000f0: 3141 0613 5161 0722 7114 3281 91a1 0823  1A..Qa."q.2....#
00000100: 42b1 c115 52d1 f024 3362 7282 090a 1617  B...R..$3br......
                              ... (etc)
```

Well, what is a cat [picture]?

On your computer, a photographic picture is often saved as a JPEG file — something dot J P G.
Inside that file, there's ... stuff. Data. Bytes.



What is a cat?

Content-Type: image/jpeg

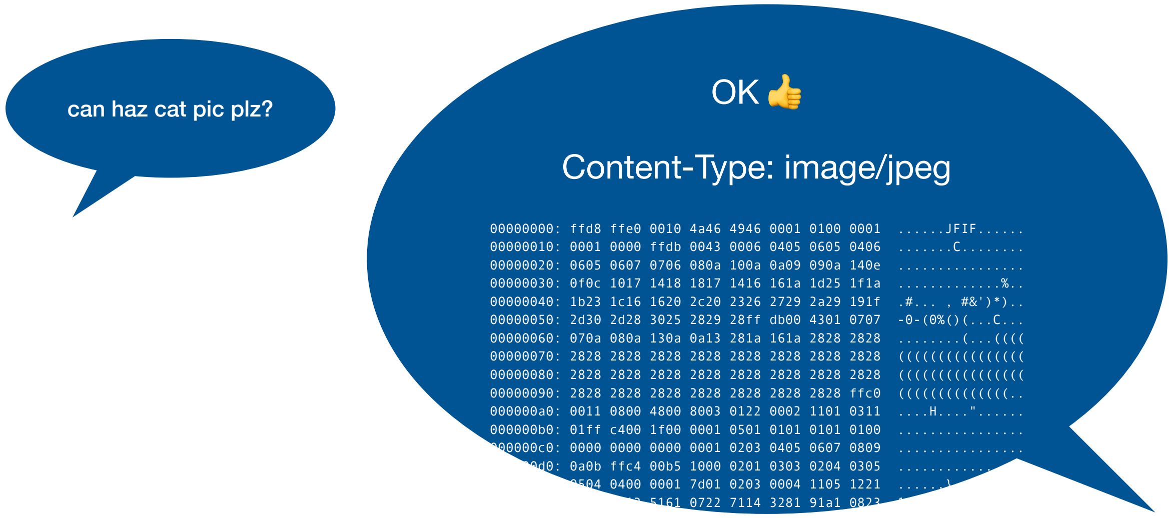
Content:

```
00000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001  ......JFIF......
00000010: 0001 0000 ffdb 0043 0006 0405 0605 0406  .......C........
00000020: 0605 0607 0706 080a 100a 0a09 090a 140e  ................
00000030: 0f0c 1017 1418 1817 1416 161a 1d25 1f1a  .............%..
00000040: 1b23 1c16 1620 2c20 2326 2729 2a29 191f  .#... , #&')*)..
00000050: 2d30 2d28 3025 2829 28ff db00 4301 0707  -0-(0%()(...C...
00000060: 070a 080a 130a 0a13 281a 161a 2828 2828  ........(...((((
00000070: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000080: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000090: 2828 2828 2828 2828 2828 2828 2828 ffc0  ((((((((((((((..
000000a0: 0011 0800 4800 8003 0122 0002 1101 0311  ....H...."......
000000b0: 01ff c400 1f00 0001 0501 0101 0101 0100  ................
000000c0: 0000 0000 0000 0001 0203 0405 0607 0809  ................
000000d0: 0a0b ffc4 00b5 1000 0201 0303 0204 0305  ................
000000e0: 0504 0400 0001 7d01 0203 0004 1105 1221  ......}........!
000000f0: 3141 0613 5161 0722 7114 3281 91a1 0823  1A..Qa."q.2....#
00000100: 42b1 c115 52d1 f024 3362 7282 090a 1617  B...R..$3br.....
                        ... (etc)
```

On the Internet, *filenames* aren't so important. Instead, we talk about "content types". A content type answers the question, "How should I interpret all of these bytes?".

The content type for JPEG images is "image/jpeg".

Cat picture, please!



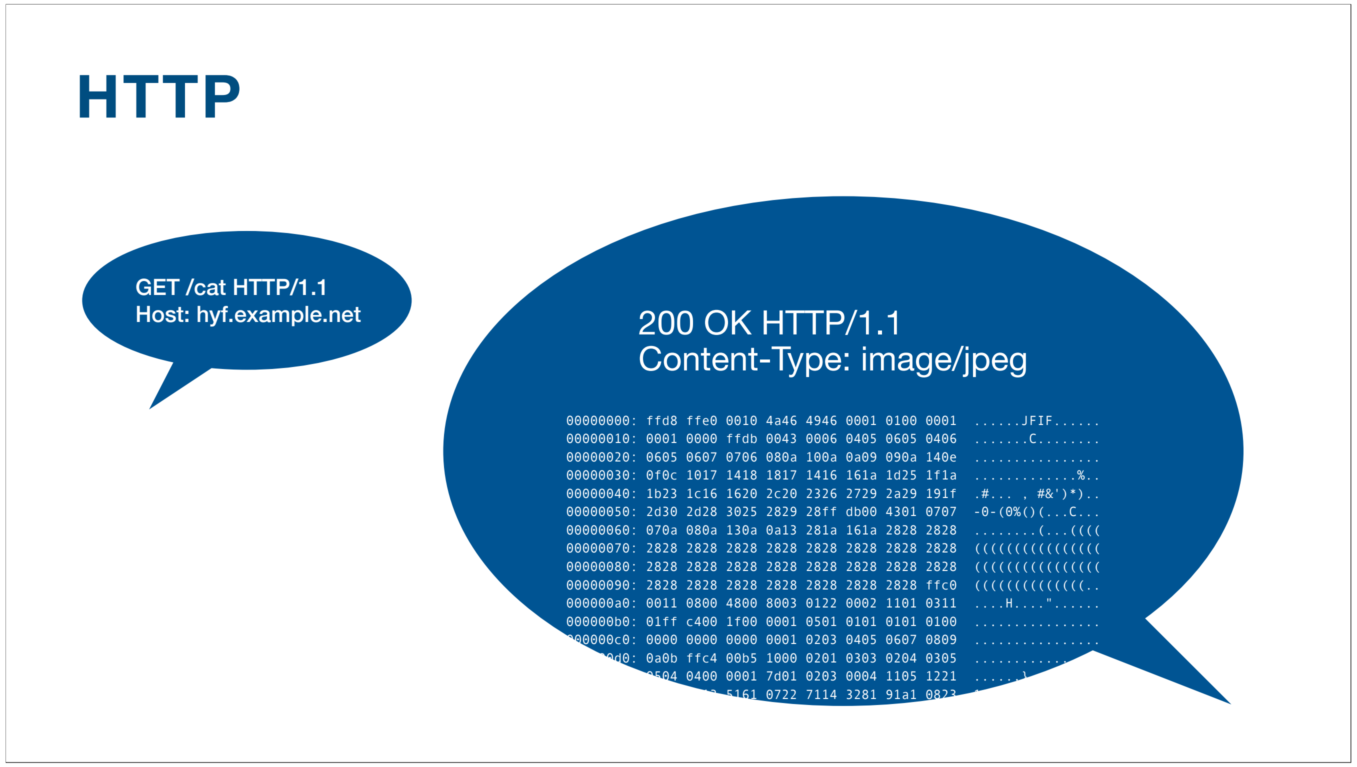
can haz cat pic plz?

OK 👍

Content-Type: image/jpeg

```
00000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001  ......JFIF......
00000010: 0001 0000 ffdb 0043 0006 0405 0605 0406  .......C........
00000020: 0605 0607 0706 080a 100a 0a09 090a 140e  ................
00000030: 0f0c 1017 1418 1817 1416 161a 1d25 1f1a  .............%..
00000040: 1b23 1c16 1620 2c20 2326 2729 2a29 191f  .#... , #&')*)..
00000050: 2d30 2d28 3025 2829 28ff db00 4301 0707  -0-(0%()(...C...
00000060: 070a 080a 130a 0a13 281a 161a 2828 2828  ........(...((((
00000070: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000080: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000090: 2828 2828 2828 2828 2828 2828 2828 ffc0  ((((((((((((((..
000000a0: 0011 0800 4800 8003 0122 0002 1101 0311  ....H...."......
000000b0: 01ff c400 1f00 0001 0501 0101 0101 0100  ................
000000c0: 0000 0000 0000 0001 0203 0405 0607 0809  ................
     0d0: 0a0b ffc4 00b5 1000 0201 0303 0204 0305  ...........
          0504 0400 0001 7d01 0203 0004 1105 1221  ......}
               5161 0722 7114 3281 91a1 0823
```

So if you ask for a picture of a cat pic, then the answer looks like this. OK, content-type, data.



HTTP

GET /cat HTTP/1.1
Host: hyf.example.net

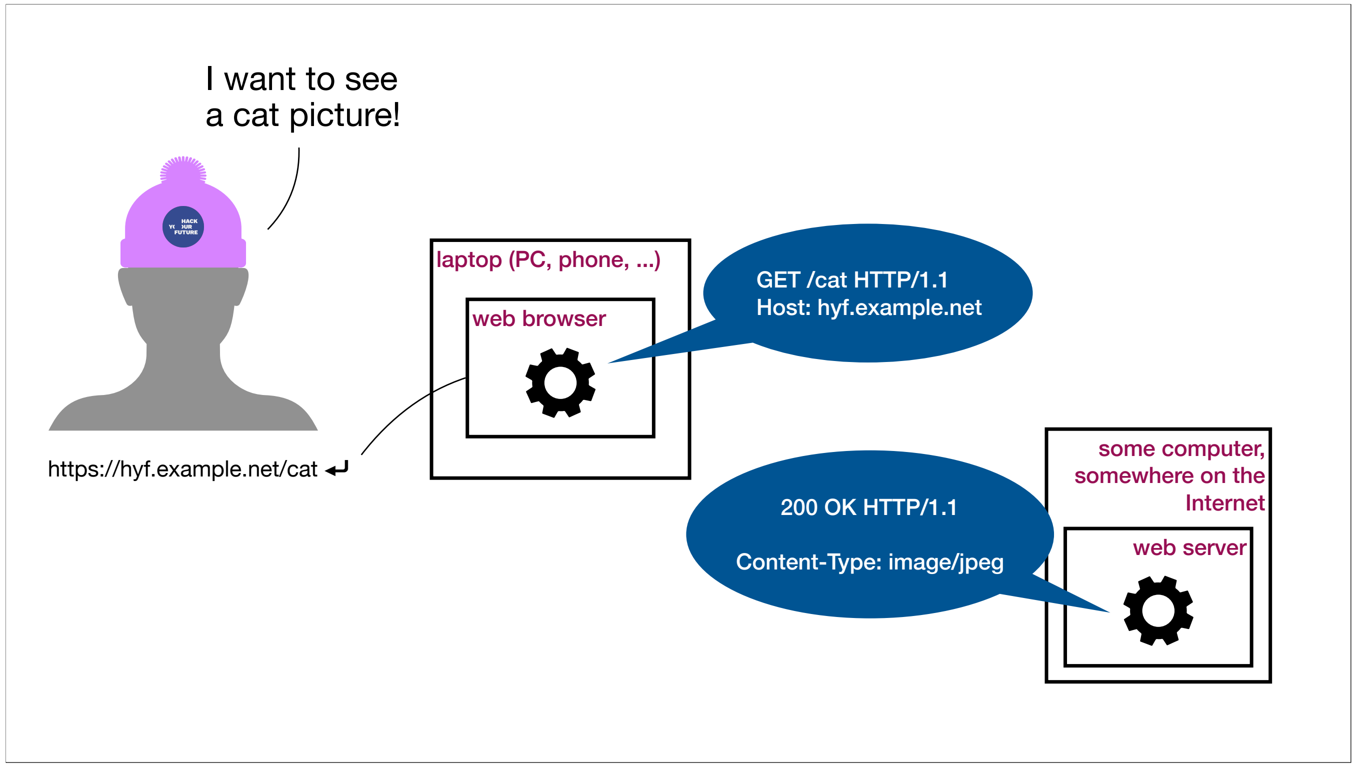
200 OK HTTP/1.1
Content-Type: image/jpeg

```
00000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001  ......JFIF......
00000010: 0001 0000 ffdb 0043 0006 0405 0605 0406  .......C........
00000020: 0605 0607 0706 080a 100a 0a09 090a 140e  ................
00000030: 0f0c 1017 1418 1817 1416 161a 1d25 1f1a  .............%..
00000040: 1b23 1c16 1620 2c20 2326 2729 2a29 191f  .#... , #&')*)..
00000050: 2d30 2d28 3025 2829 28ff db00 4301 0707  -0-(0%()(...C...
00000060: 070a 080a 130a 0a13 281a 161a 2828 2828  ........(...((((
00000070: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000080: 2828 2828 2828 2828 2828 2828 2828 2828  ((((((((((((((((
00000090: 2828 2828 2828 2828 2828 2828 2828 ffc0  ((((((((((((((..
000000a0: 0011 0800 4800 8003 0122 0002 1101 0311  ....H...."......
000000b0: 01ff c400 1f00 0001 0501 0101 0101 0100  ................
00000c0: 0000 0000 0000 0001 0203 0405 0607 0809  ................
   0d0: 0a0b ffc4 00b5 1000 0201 0303 0204 0305  ...........
        0504 0400 0001 7d01 0203 0004 1105 1221  ......}
             5161 0722 7114 3281 91a1 0823
```

To turn that into the language that computers actually use to communicate:

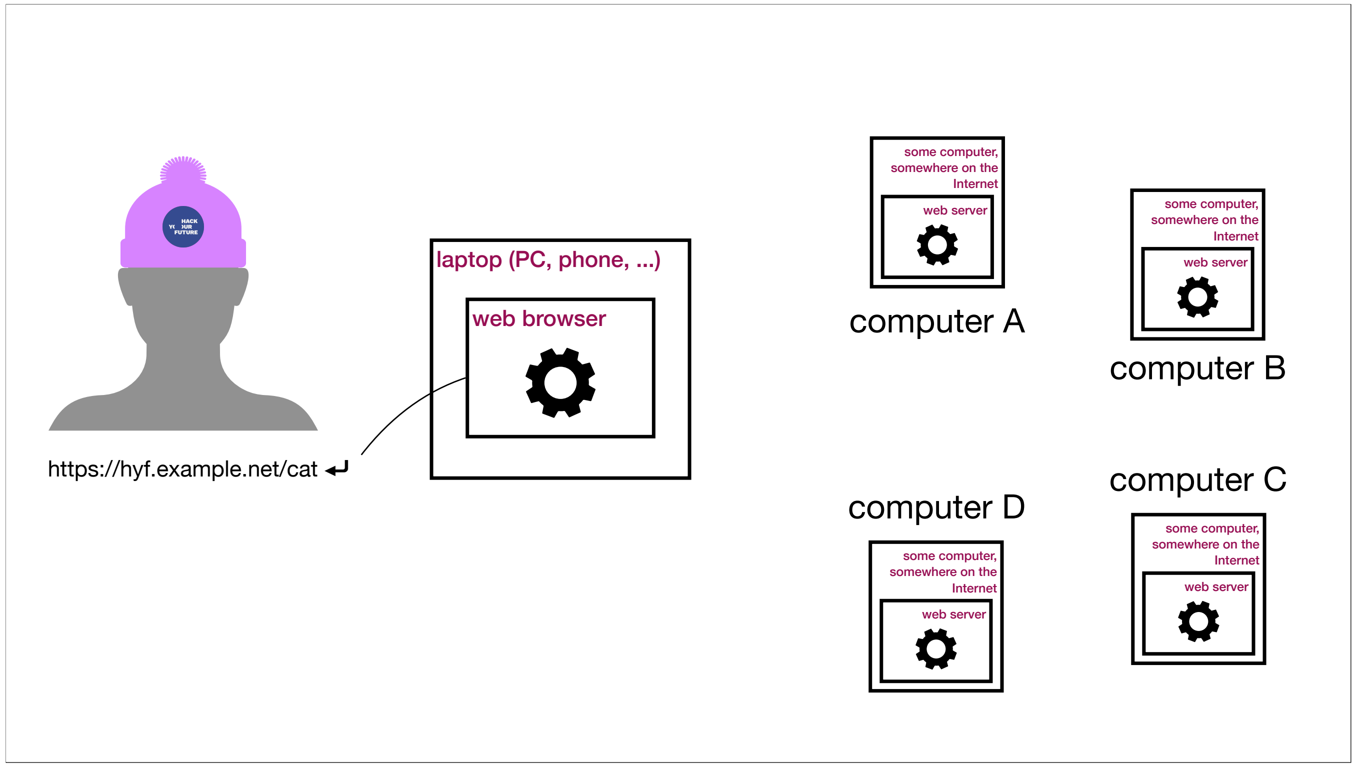
"Can haz cat pic plz?" becomes the pictured "GET" request.
"OK here you go" becomes a "200 OK" response.

This is HTTP. The somewhat-anachronously-named "Hypertext Transfer Protocol".



But who is having this conversation? Answer: your computer (or more specifically, the web browser software running on it) — and some other computer, out there on the Internet (or more specifically: the web *server software* running on it).

DNS



https://hyf.example.net/cat ↵

laptop (PC, phone, ...)

web browser

some computer, somewhere on the Internet

web server

computer A

some computer, somewhere on the Internet

web server

computer B

computer C

some computer, somewhere on the Internet

web server

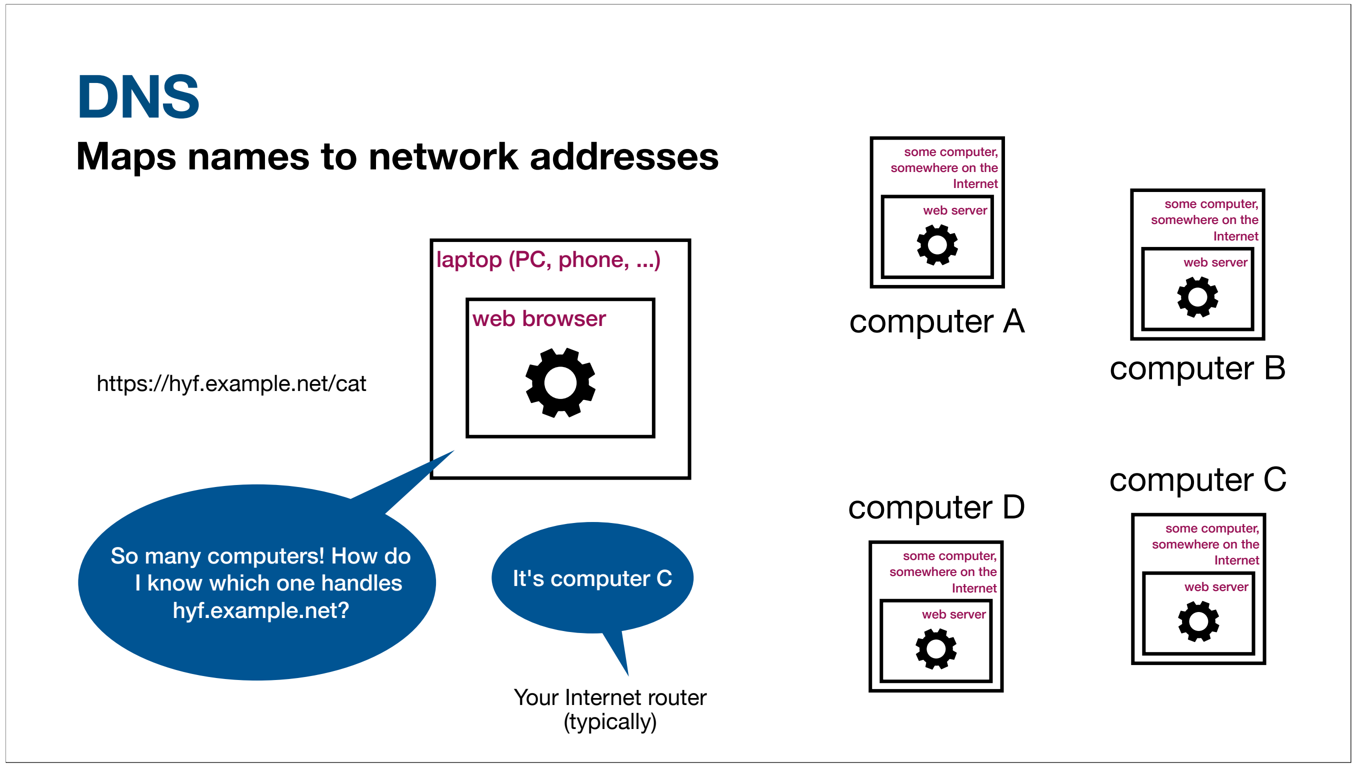
computer D

some computer, somewhere on the Internet

web server

The Internet is made up of at least 4 or 5 computers*. Which one should your web browser talk to?

* some say it's as many as 20.



DNS

Maps names to network addresses

laptop (PC, phone, ...)

web browser

https://hyf.example.net/cat

So many computers! How do I know which one handles hyf.example.net?

It's computer C

Your Internet router (typically)

some computer, somewhere on the Internet

web server

computer A

some computer, somewhere on the Internet

web server

computer B

computer D

some computer, somewhere on the Internet

web server

computer C

some computer, somewhere on the Internet

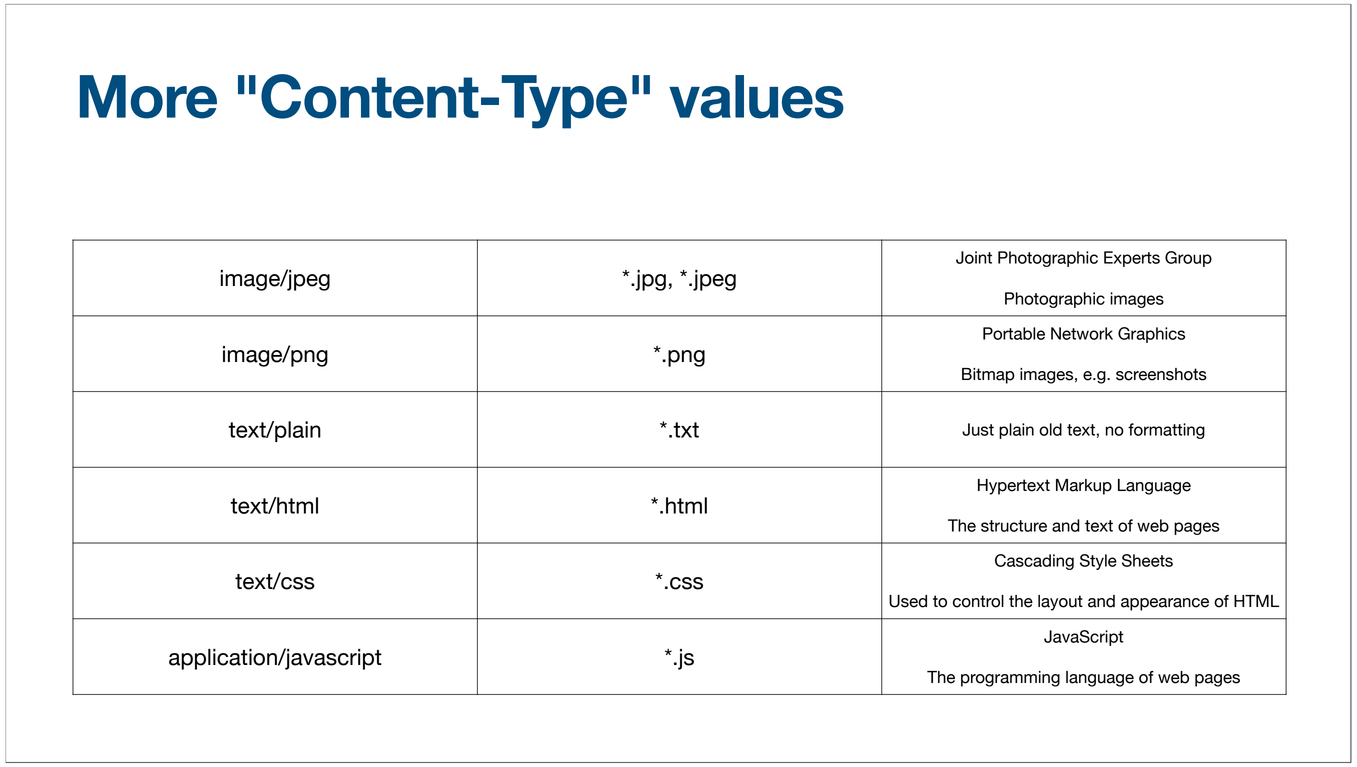
web server

To find out, your computer asks (typically) your Internet router "Which computer is hyf.example.net?" — since that's the "host" part of the URL in question. The router replies with the answer.

This is what we call DNS: the Domain Name System.

Content Types

also known as MIME types



More "Content-Type" values

image/jpeg	*.jpg, *.jpeg	Joint Photographic Experts Group Photographic images
image/png	*.png	Portable Network Graphics Bitmap images, e.g. screenshots
text/plain	*.txt	Just plain old text, no formatting
text/html	*.html	Hypertext Markup Language The structure and text of web pages
text/css	*.css	Cascading Style Sheets Used to control the layout and appearance of HTML
application/javascript	*.js	JavaScript The programming language of web pages

You don't need to memorise all of this right now. Just have a basic understanding of what Content Types are: a tiny piece of text (e.g. "text/html") which specifies what format some bit of data (the "content") is in.

More HTTP response codes

More HTTP response (status) codes

Just the most commonly-seen ones

- 200 OK
- 301/302/303/307/308 — various forms of redirection
- 404 Not Found
- 403 Forbidden
- 500 Server Error

More HTTP response (status) codes

5xx — server error

By far the most common:

- 500 Server Error

This is what you'll tend to see if the server software crashed while trying to handle your request.

Things to maybe include somewhere

Browser "developer tools"
The network tab

Server setups more complex than 1 server:
- server + database
- server + API request to inner server
- servers as "cattle" — stateless, not precious

- the concept of "deployment"
- test/staging