# Web Technologies Overview

# Web is a complex environment…

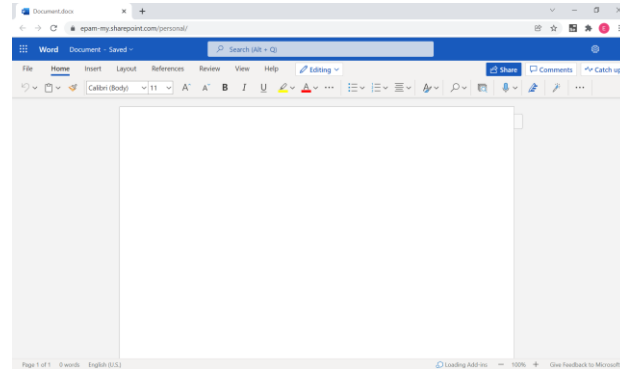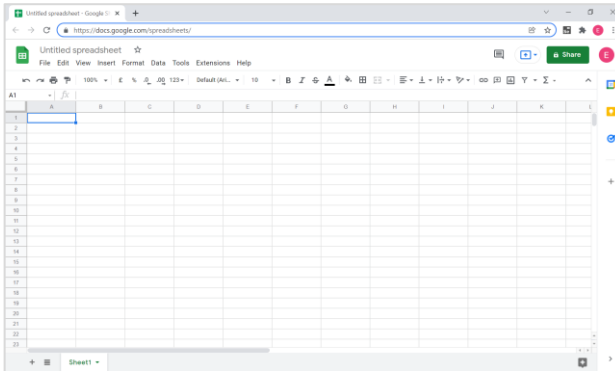| | |
|---|---|
| Web applications | Programming languages |
| Web services | Markup languages |
| Web servers | Data formats |
| Application servers | Network technologies |
| Databases | And so on… |

A **web application** (or web app) is application software that runs on a web server ☺.

It usually requires a network connection and a web browser (as a client software), but there may be exceptions.

# Web application vs web site : STOP IT!

For decades there are almost no significant technical differences between web applications and web sites.

If it still bothers you, you may imagine that a web application is something complex with enhanced interactivity, and a web site is something simple with mostly "read only" capabilities.
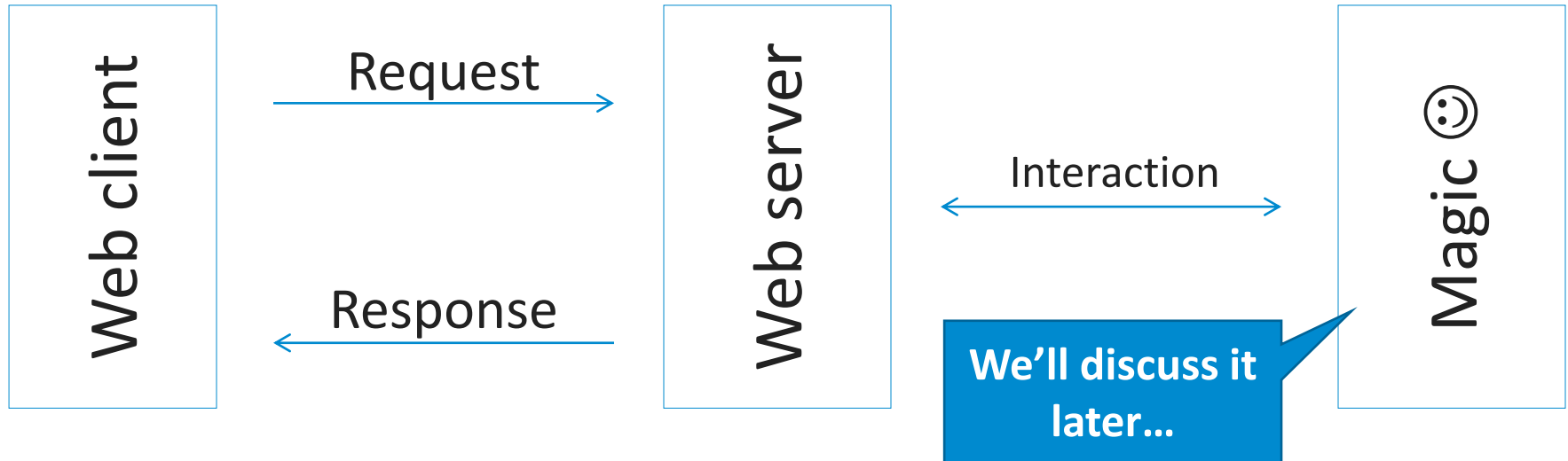
A **web service** is service offered by an electronic device to another electronic device, communicating with each other via the web.

Usually, it's just a "small one-purpose application" (that provides something like "currency exchange rates", "stock exchange data", "weather data", etc…)

```
GET https://data.fixer.io/api/latest

{
  "base": USD,
  "date": "2018-02-13",
  "rates": {
      "CAD": 1.260046,
      "CHF": 0.933058,
      "EUR": 0.806942,
      "GBP": 0.719154,
      [170 world currencies]
  }
}
```
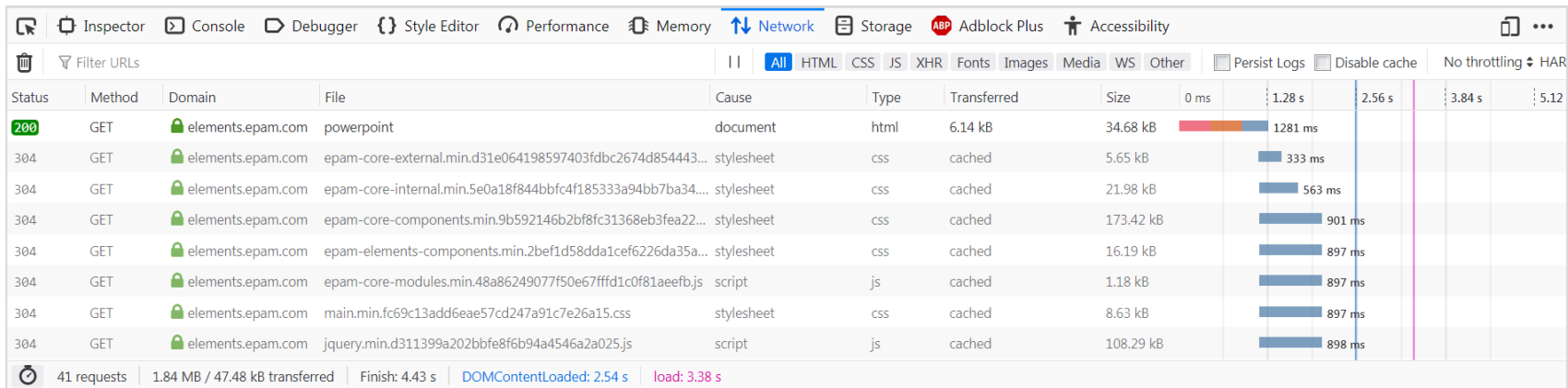
A **web server** is computer software and underlying hardware that processes requests via HTTP.

Web client → Request → Web server

Web server ← Response ← Web client

Web server ↔ Interaction ↔ Magic ☺

We'll discuss it later…

# Web client communication with web server

Usually web-client sends hundreds of requests to dozens of web-servers to compose and display a single web page.

… so, what is a web client?

A **web client** is client side (user side) of the web. In most cases it's just a web browser, but it any be any software capable to perform http-requests.
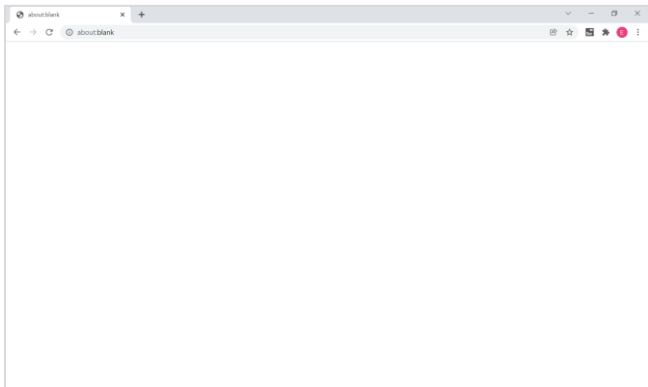


wget http://apache.org/httpd/latest.gz

curl -o file https://stackoverflow.com

# Web server typical functions

Requests processing and responses generation

Forwarding requests to application server (and responses handling)

Handling of indirect access to file system

Access permissions control

Traffic encryption

Load balancing and bandwidth control

Virtual hosts management and requests routing

Errors handling

Logging

# Web server typical functions: see also

Another good source for web server function understanding is the list of HTTP response codes:

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

# Most common web servers (and application severs)

See the latest data here:

https://w3techs.com/technologies/overview/web_server

For years the most common web-servers are: Apache httpd, Nginx, Cloudflare Server.

So, web servers generate responses (returns some content)…

Content

Static

Something that doesn't need any additional processing

May be stored in cache and re-used

Dynamic

Something that has to be created once a request is received
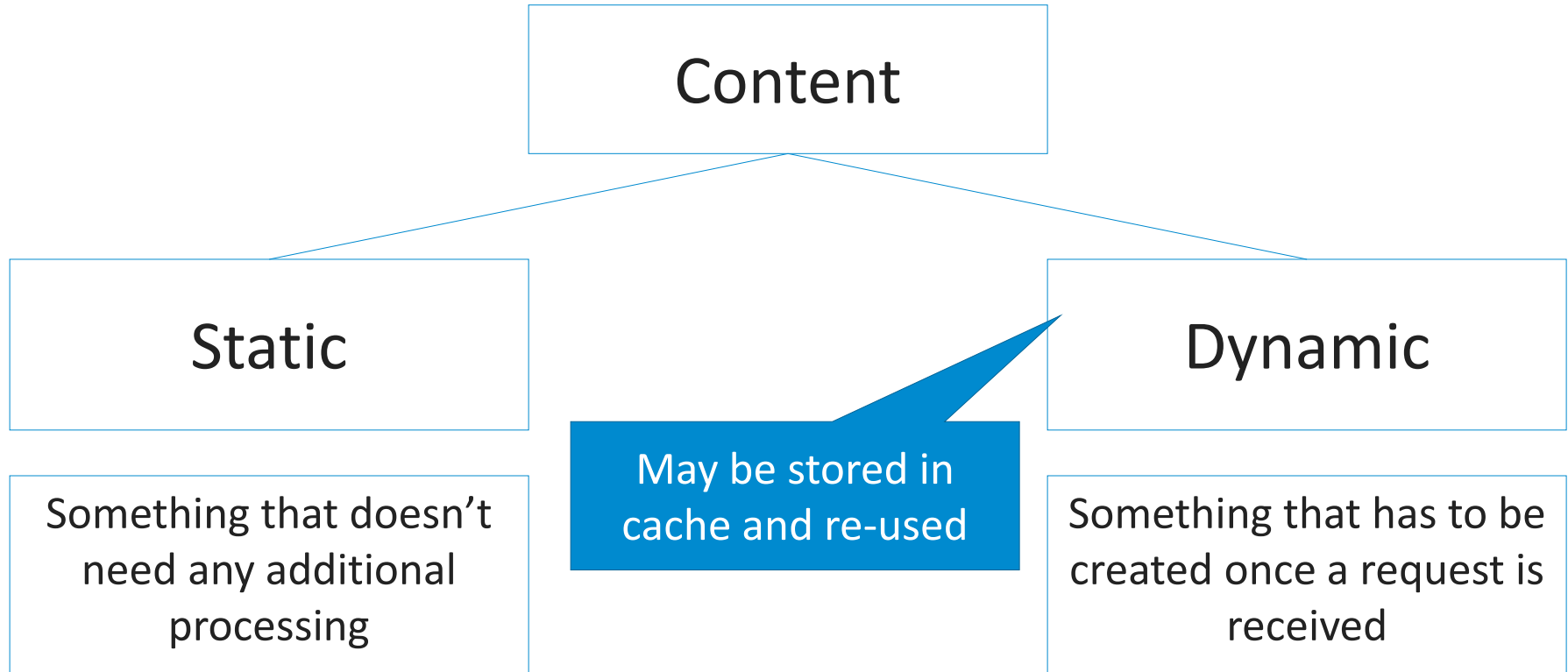
And one more extremely useful web servers' feature

**URL rewriting** is a process of internal URL transformation.

**www.site.com/catalog/notebooks/hp/new/**

VS

www.site.com/index.php?page=catalog&category=notebooks&vendor=hp&mode=new

Details: http://httpd.apache.org/docs/current/mod/mod_rewrite.html

# Two simple samples: URL rewriting and access control

```
RewriteEngine On
RewriteBase /
RewriteRule .* index.php?url=$0 [QSA,L]
```

```
<Files "config.php">
Order allow,deny
Deny from All
</Files>
```

These lines in ".htaccess" file will change any URL request to the "index.php" file request with the initial URL as the parameter.

These lines in ".htaccess" file will deny access to "config.php" file.

An **application server** is computer software that provides applications with rich API and isolates them from OS and/or hardware.

In most cases runtime environments like Java Runtime Environment (JRE), .NET Framework, PHP, and so on may be considered "application servers".

| Application |
|---|

| Application Server |
|---|

| Operating System |
|---|

| Hardware |
|---|
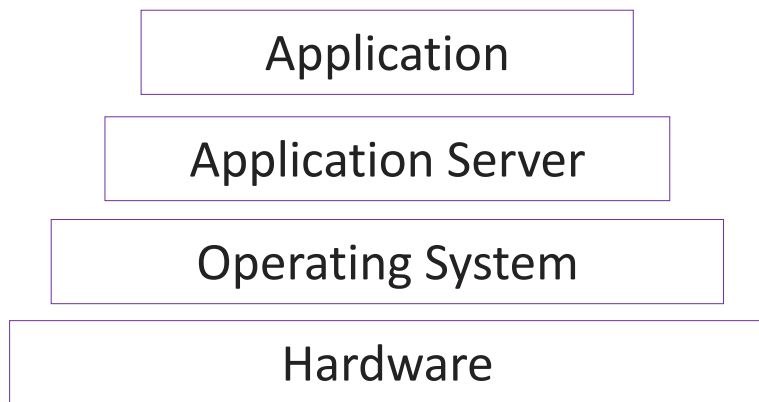
# Application server typical functions

Applications isolation from OS and/or hardware

Provision of API for typical operations

Resources management

Logging

Performance optimization

Load balancing

Security management

Errors handling

*{Any specific function depending on particular software}*

A **database** is an organized collection of structured information, or data, typically controlled by a database management system (DBMS).

| Web client | Request → | Web server | ↔ | Application server |
|---|---|---|---|---|
| | ← Response | | | ↕ |
| | | | ↔ | DBMS |
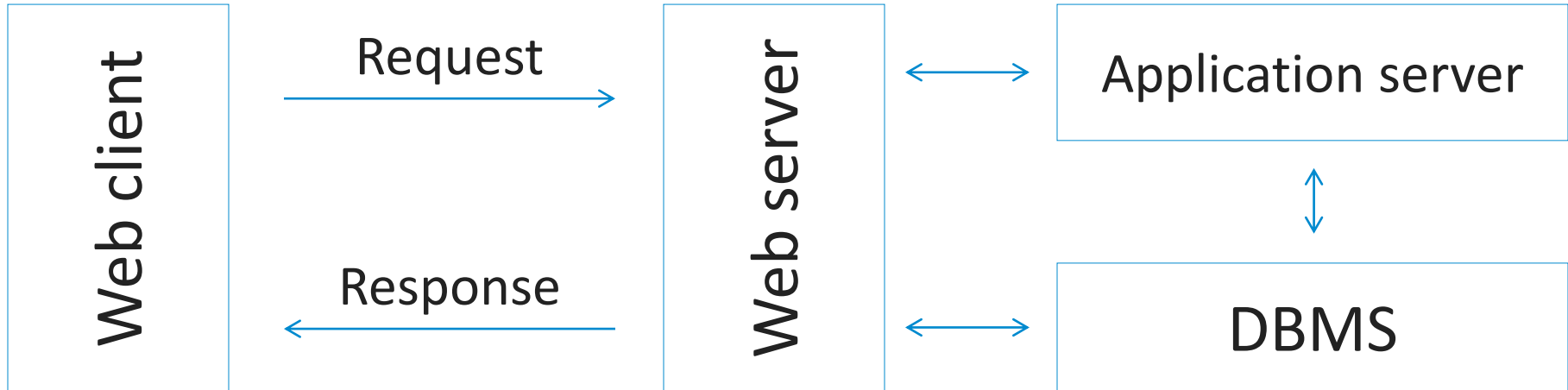
# So, the most common communication schema is…
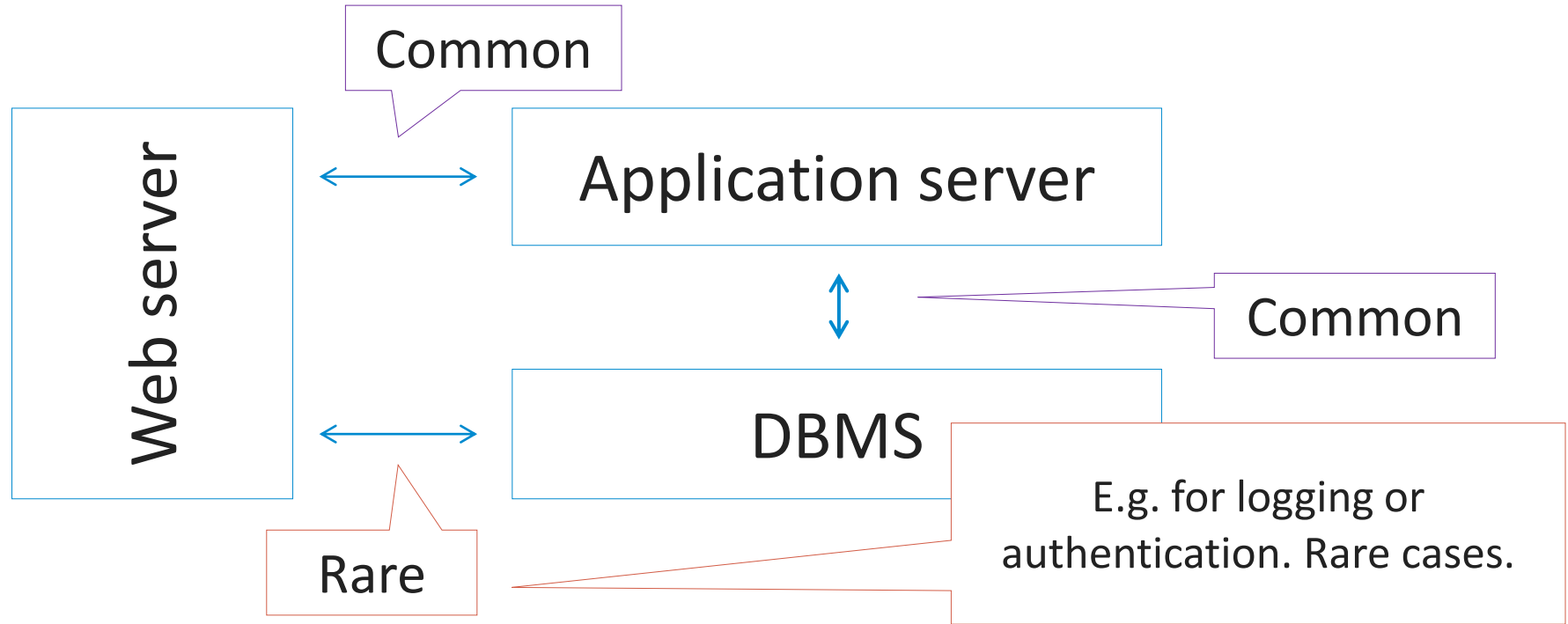
# Just a quick question: can this happen?
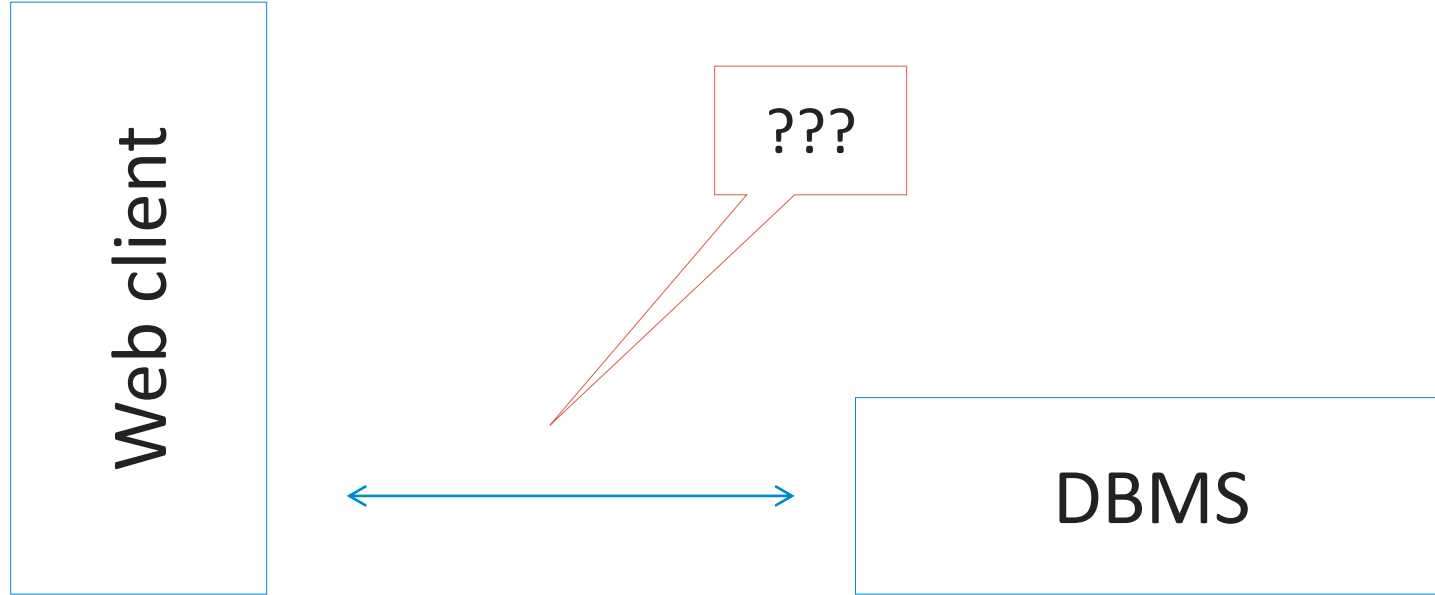
# Programming languages (most commonly used for web development)

| Java |
| --- |

| PHP |
| --- |

| Python |
| --- |

| JavaScript |
| --- |

| ASP |
| --- |

| Perl |
| --- |

| Ruby |
| --- |

| ActionScript |
| --- |

Server-side

Client-side
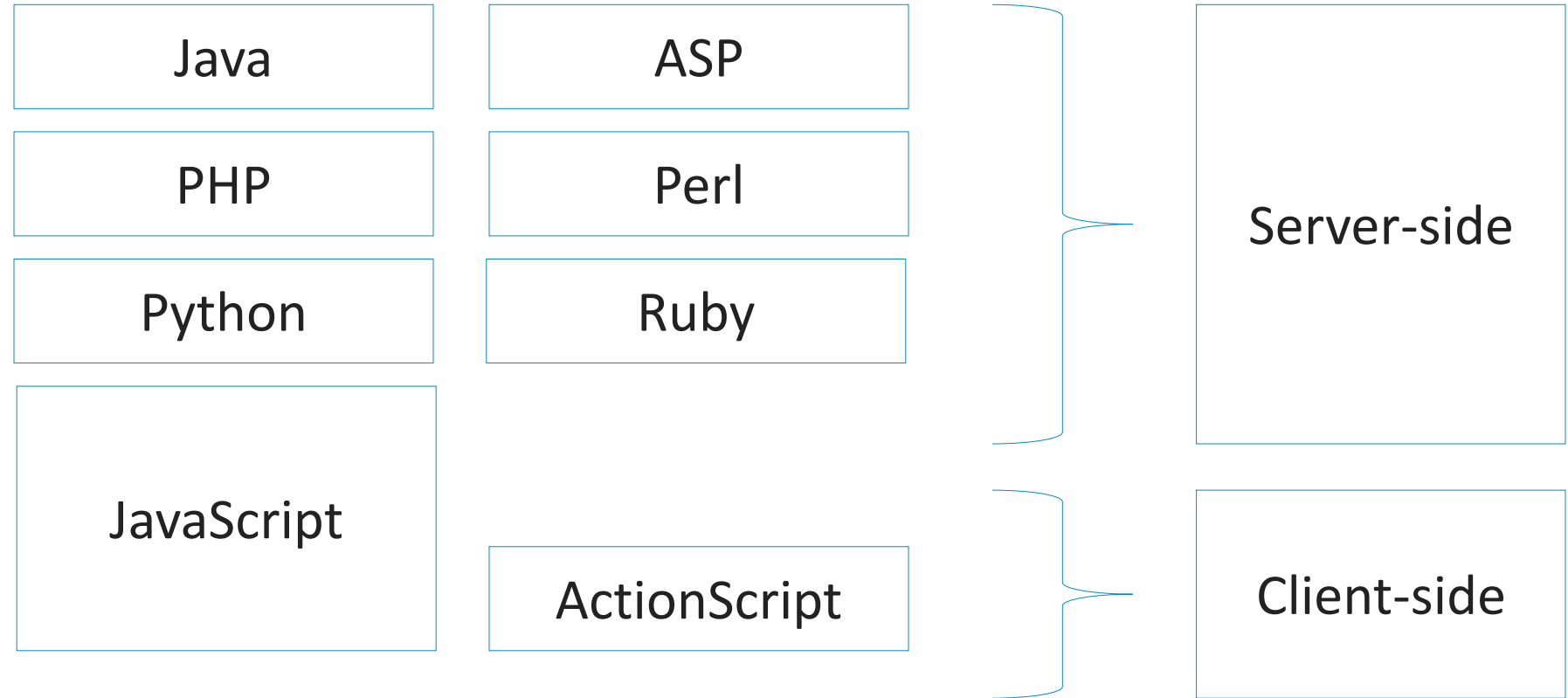
# Markup / design languages (most commonly used for web development)

HTML

CSS

# Data formats

XML

DTD

WSDL

SOAP

JSON

RSS

# Network technologies

| Protocol data unit | Layer | Function |
|---|---|---|
| Data | 7. Application | High-level APIs, including resource sharing, remote file access |
| (We are here ☺) | 6. Presentation | Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption |
| | 5. Session | Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes |
| Segment, Datagram | 4. Transport | Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing |
| Packet | 3. Network | Structuring and managing a multi-node network, including addressing, routing and traffic control |
| Frame | 2. Data link | Reliable transmission of data frames between two nodes connected by a physical layer |
| Bit, Symbol | 1. Physical | Transmission and reception of raw bit streams over a physical medium |

# Finally: what is where?

| Stored in database | Stored in file system | Generated per request |
|---|---|---|
| Application / site structure | Scripts | Pages / pages' parts |
| Texts, numbers, etc. | Multimedia | Texts, numbers, etc. |
| Some settings | Some settings | Multimedia |

## Are there any good books / links / courses?

Always prefer official documentation. Always!

Google it! In ENGLISH (never in any other language).

Prefer up-to-date articles / manuals (many books are outdated).

Start with:
**php.net**
**stackoverflow.com**
**w3schools.com**

And then:
**developer.mozilla.org**
**rs.school**
**web.dev**

# Web Technologies Overview

**Disclaimer:** вы смотрите просто запись лекции, это **НЕ** специально подготовленный видеокурс!