Programowanie aplikacji WWW
Web application programming

Grzegorz Ostrek

g.ostrek@mini.pw.edu.pl

Wydział Matematyki i Nauk Informacyjnych
00-662 Warszawa
ul. Koszykowa 75

https://pages.mini.pw.edu.pl/~ostrekg/WAP24Z/

HTML, XHTML, HTML5

- XML is a subset of SGML, optimized version
- HTML is a document of SGML that follows the DTD standard, and it can also be said to be an example of SGML
- XHTML is an HTML document that follows the XML standard.
- HTML5 is the latest HTML standard of HTML. But it is not based on SGML, so it does not follow the DTD standard

Standard Generalized Markup Language

Sounds Great Maybe Later

https://www.w3.org/TR/html5-diff/

"HTML5 Differences from HTML4" describes the differences of the HTML5 specification from those of HTML4

https://web.archive.org/web/20210211095022/https://blog.whatwg.org/xhtml5-in-a-nutshell

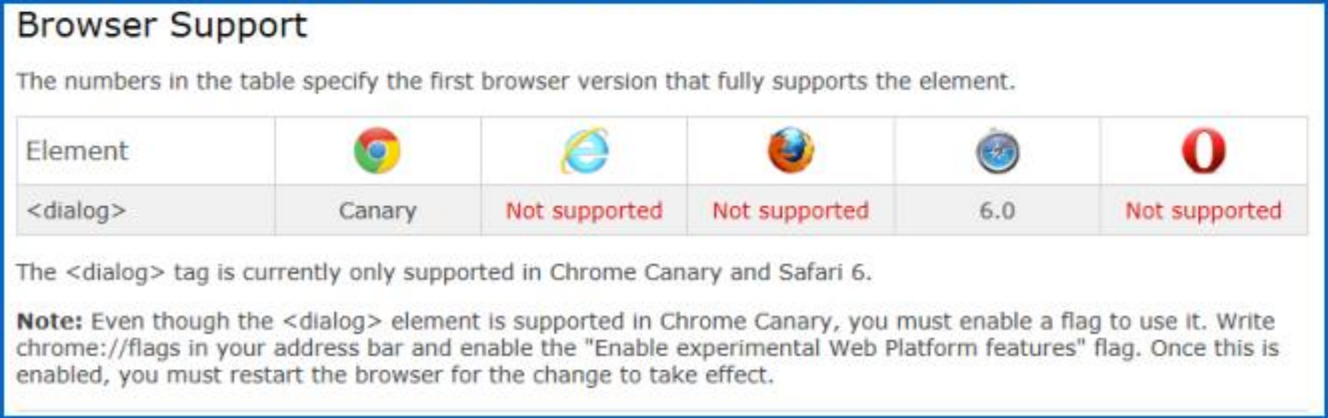
HTML5

New HTML5 features:

- Image aspect: Canvas
- Location: Geolocation
- Storage aspect: Local Storage, Session Storage
- Multitasking: Web Worker
- Persistent connection: WebSocket
- Offline response to requests: Service-Worker
- semantic tags: header, nav， article， section， footer
- Form control: calendar, time, color, email, search, date

HTML5 tags no longer present in the standard

- Some tags are no longer present in W3ORG HTML5 standard, but might have been already considered by browser vendors
- Hence, they may be listed at W3ORG schools website



Browser Support

The numbers in the table specify the first browser version that fully supports the element.

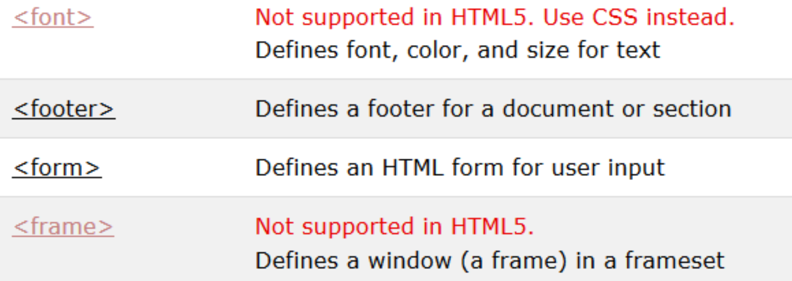
Element					O
<dialog>	Canary	Not supported	Not supported	6.0	Not supported

The <dialog> tag is currently only supported in Chrome Canary and Safari 6.

Note: Even though the <dialog> element is supported in Chrome Canary, you must enable a flag to use it. Write chrome://flags in your address bar and enable the "Enable experimental Web Platform features" flag. Once this is enabled, you must restart the browser for the change to take effect.

Source and further details at: http://www.w3schools.com/html5, http://www.w3schools.com/tags/default.asp
See http://www.w3.org/TR/html5/index.html#elements-1 for a current listing of tags

https://www.w3schools.com/TAGS/default.ASP



<font>	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<footer>	Defines a footer for a document or section
<form>	Defines an HTML form for user input
<frame>	Not supported in HTML5. Defines a window (a frame) in a frameset

HTML5 elements

HTML5 has a total of 12 formatting tags:

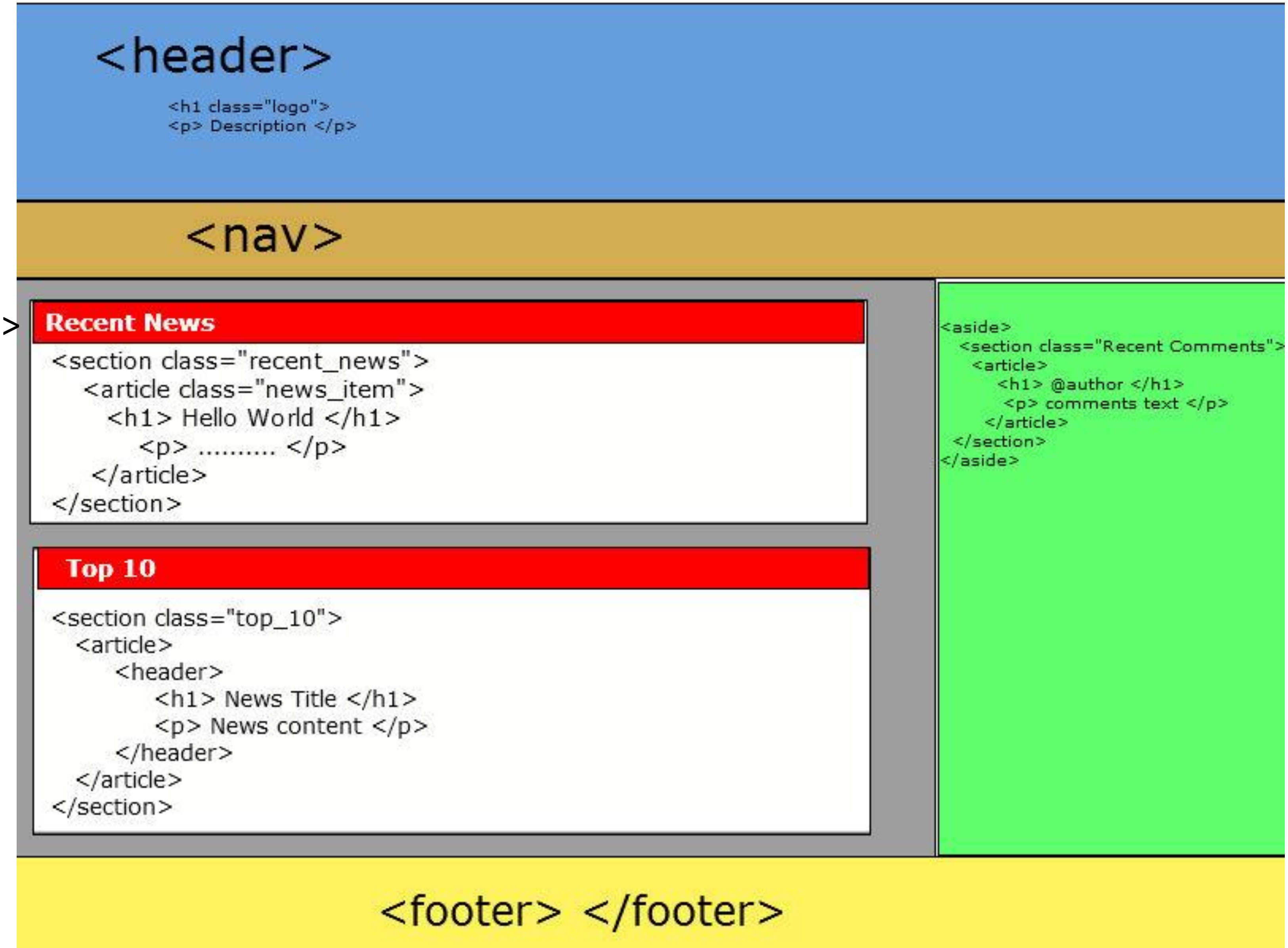
<b> making the text bold.
<del> Used to display the deleted text.
<em> text italic with added semantics importance.
<i> Used to make the text italic.
<ins> Used for displaying the added text.
<mark> Used for highlighting the text.
<small> Used to decrease the font size of the text by one unit.
<strong> - Used for telling the browser that the text is important.
<sub> - Used for making the text a subscript.
<sup> - Used to make the text a superscript.
<u> - Used for underlining the text.

<blockquote cite="http://www...> ... </blockquote>
<q> short quotation </q>

Following tags were removed in the HTML5:

<acronym>
<applet>
<basefont>
<big>
<bigcenter>
<center>
<dir>
<frame>
<frameset>
<font>
<noframes>
<strike>
<tt>

Semantic HTML



<main>

```
<header>
        <h1 class="logo">
        <p> Description </p>
```

```
<nav>
```

Recent News

```
<section class="recent_news">
   <article class="news_item">
     <h1> Hello World </h1>
        <p> .......... </p>
    </article>
</section>
```

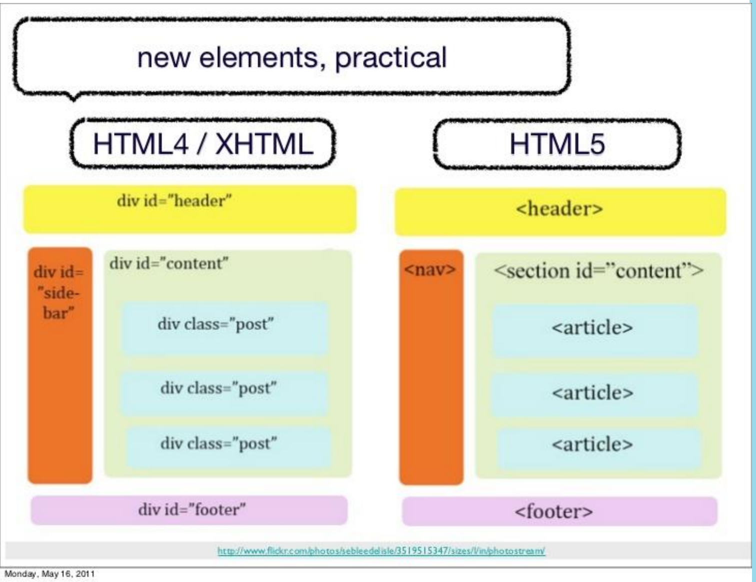
Top 10

```
<section class="top_10">
  <article>
      <header>
          <h1> News Title </h1>
          <p> News content </p>
      </header>
  </article>
</section>
```

```
<aside>
  <section class="Recent Comments">
    <article>
       <h1> @author </h1>
        <p> comments text </p>
     </article>
  </section>
</aside>
```

```
<footer> </footer>
```

Semantic HTML cont.



new elements, practical

HTML4 / XHTML

div id="header"

div id="side-bar"

div id="content"

div class="post"

div class="post"

div class="post"

div id="footer"

HTML5

<header>

<nav>

<section id="content">

<article>

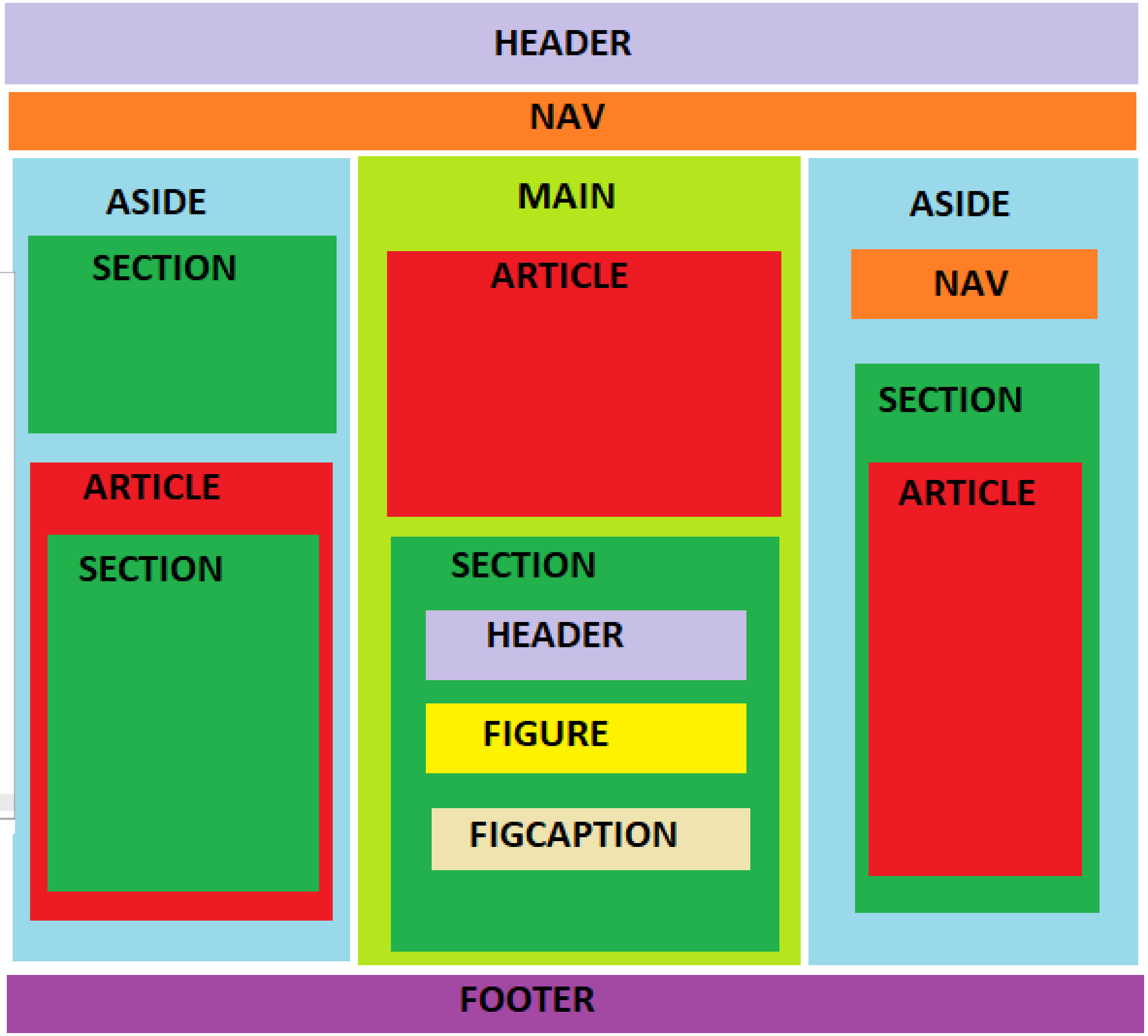
<article>

<article>

<footer>

http://www.flickr.com/photos/sebleedelisle/3519515347/sizes/l/in/photostream/

Monday, May 16, 2011



HEADER

NAV

ASIDE

SECTION

ARTICLE

SECTION

MAIN

ARTICLE

SECTION

HEADER

FIGURE

FIGCAPTION

ASIDE

NAV

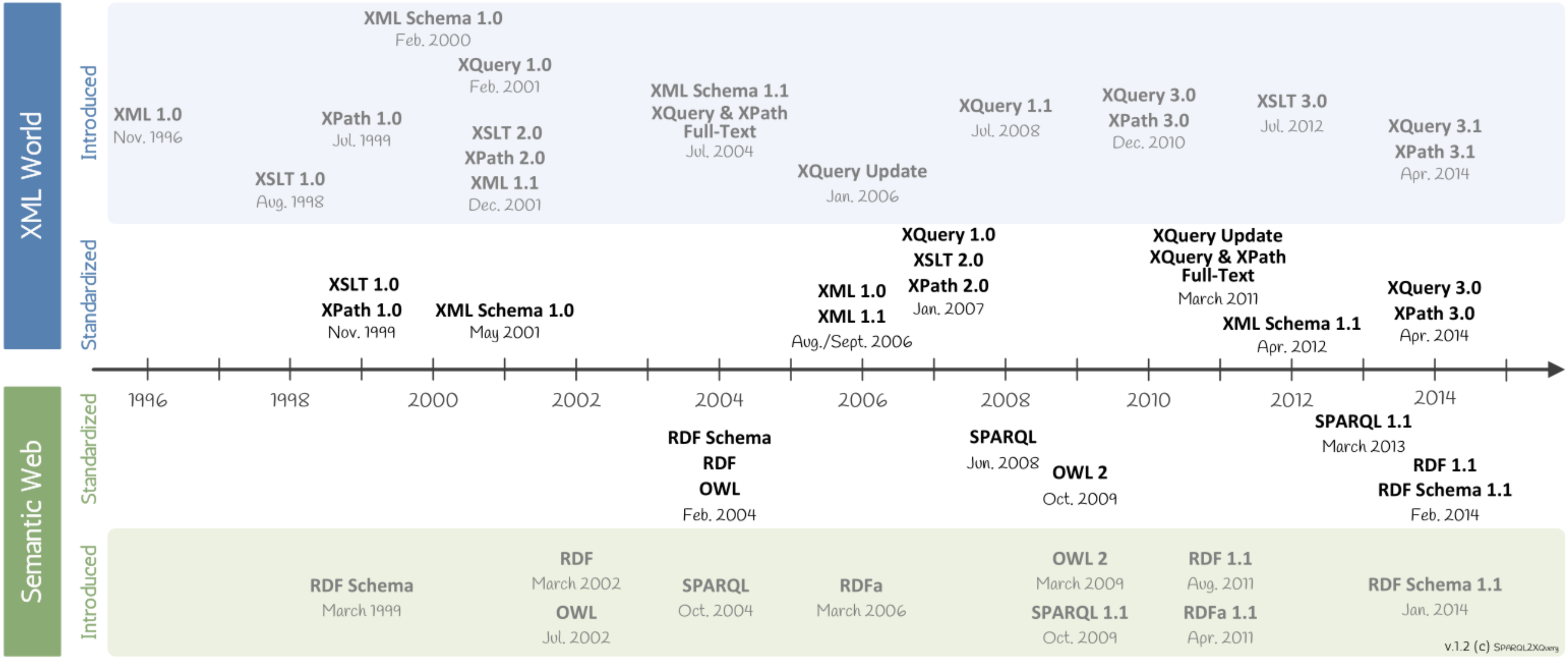
SECTION

ARTICLE

FOOTER

- Structured layout
- *semantic element* contain proper information

SemanticWeb



XML World

Introduced

XML 1.0
Nov. 1996

XSLT 1.0
Aug. 1998

XPath 1.0
Jul. 1999

XML Schema 1.0
Feb. 2000

XQuery 1.0
Feb. 2001

XSLT 2.0
XPath 2.0
XML 1.1
Dec. 2001

XML Schema 1.1
XQuery & XPath
Full-Text
Jul. 2004

XQuery Update
Jan. 2006

XQuery 1.1
Jul. 2008

XQuery 3.0
XPath 3.0
Dec. 2010

XSLT 3.0
Jul. 2012

XQuery 3.1
XPath 3.1
Apr. 2014

Standardized

XSLT 1.0
XPath 1.0
Nov. 1999

XML Schema 1.0
May 2001

XML 1.0
XML 1.1
Aug./Sept. 2006

XQuery 1.0
XSLT 2.0
XPath 2.0
Jan. 2007

XQuery Update
XQuery & XPath
Full-Text
March 2011

XML Schema 1.1
Apr. 2012

XQuery 3.0
XPath 3.0
Apr. 2014

1996 1998 2000 2002 2004 2006 2008 2010 2012 2014

Semantic Web

Standardized

RDF Schema
RDF
OWL
Feb. 2004

SPARQL
Jun. 2008

OWL 2
Oct. 2009

SPARQL 1.1
March 2013

RDF 1.1
RDF Schema 1.1
Feb. 2014

Introduced

RDF Schema
March 1999

RDF
March 2002

OWL
Jul. 2002

SPARQL
Oct. 2004

RDFa
March 2006

OWL 2
March 2009

SPARQL 1.1
Oct. 2009

RDF 1.1
Aug. 2011

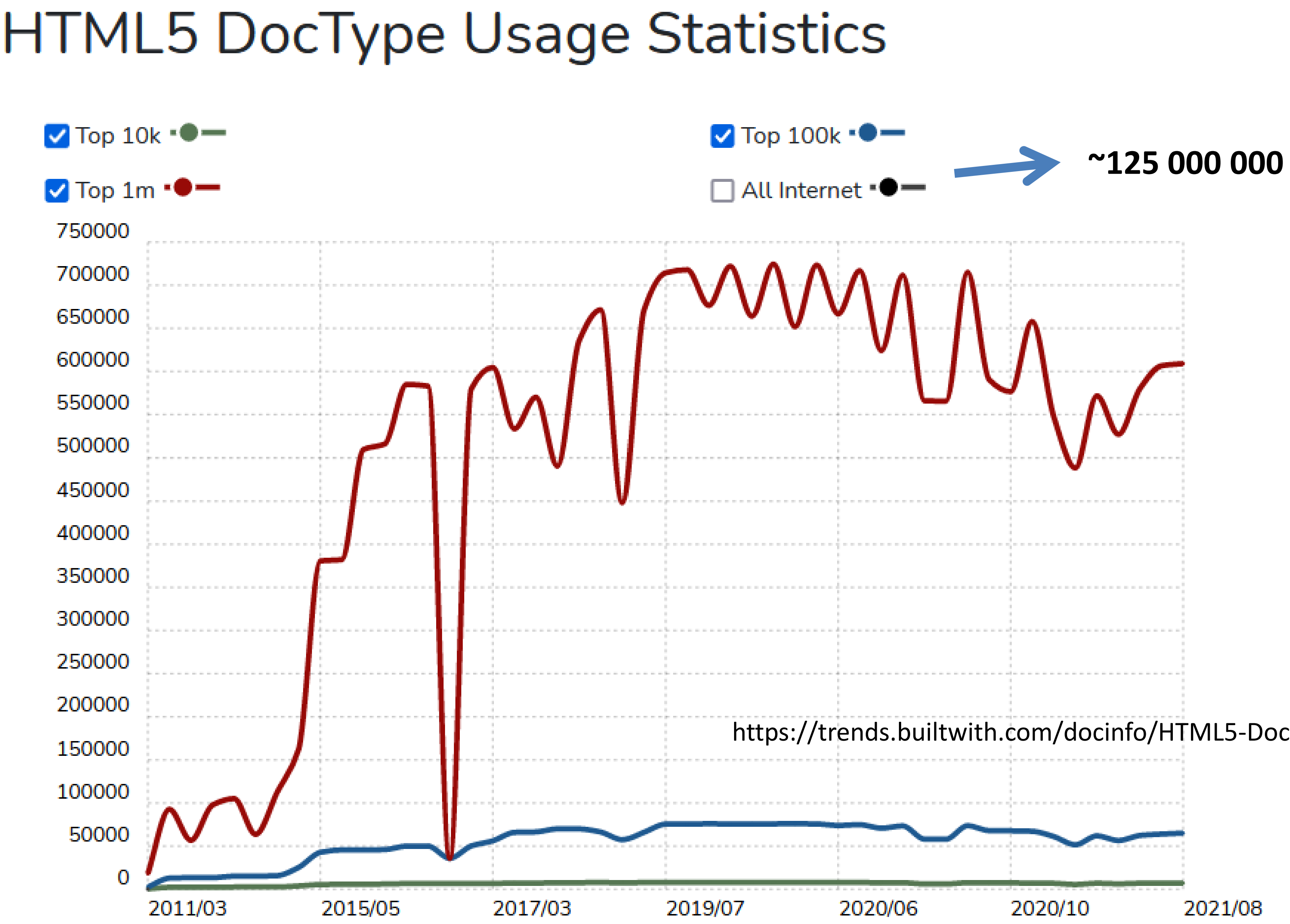
RDFa 1.1
Apr. 2011

RDF Schema 1.1
Jan. 2014

v.1.2 (c) SPARQL2XQuery

https://www.nbikakis.com/XMLSemanticWebW3CTimeline.pdf

HTML5 statistics



HTML5 DocType Usage Statistics

☑ Top 10k ●

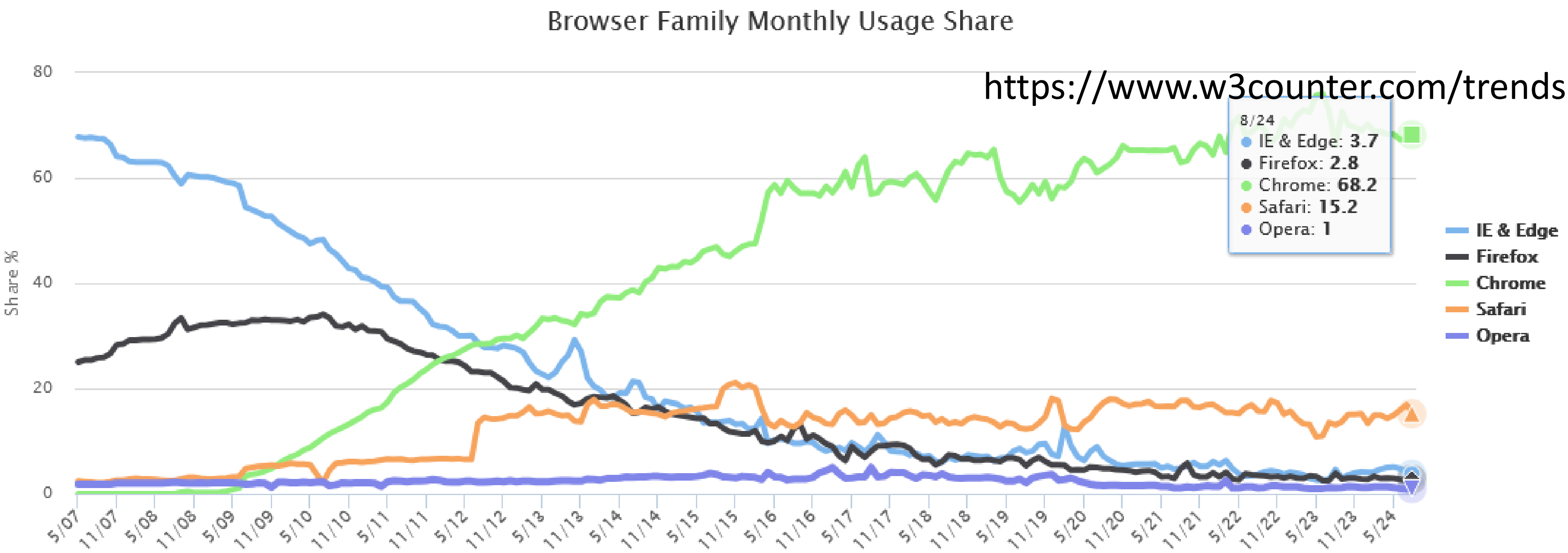
☑ Top 1m ●

☑ Top 100k ●

□ All Internet ●

~125 000 000

https://trends.builtwith.com/docinfo/HTML5-DocType



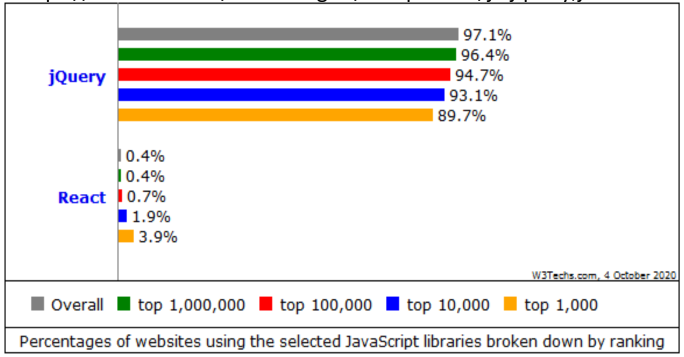
https://www.w3counter.com/trends

Browsers	Percentage Market Share	Polska (09.2020) (Net Applications/gs.statcounter.com)
Chrome	66.3%	69,9%/72.03%
Safari	16.76%	3,6%/4.11%
Firefox	4.08%	7,2%/10.72%
Samsung Internet	3.24%	_ /4.34%
Edge	2.61%	8,8%+3,9%(IE)/ 2,49%
Opera	2.06%	_ /5%

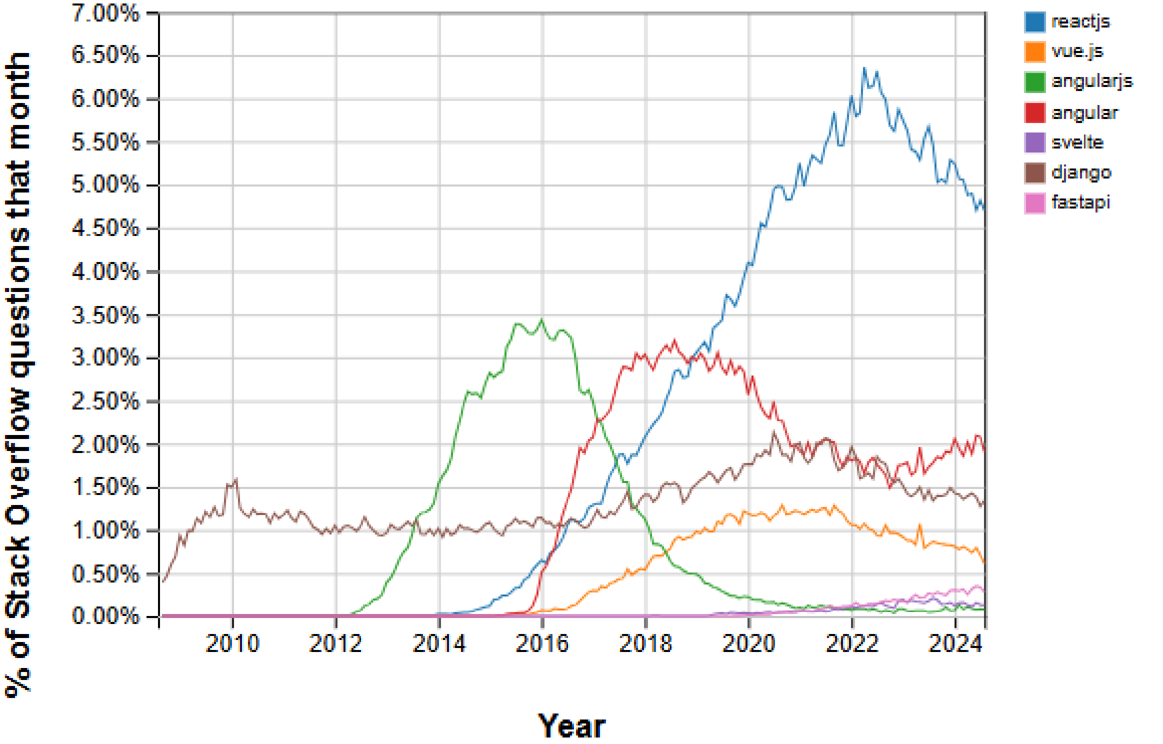
Browser Market Share Worldwide - September 2020
https://gs.statcounter.com/

- SPA single-page application
- PWA Progressive Web Apps

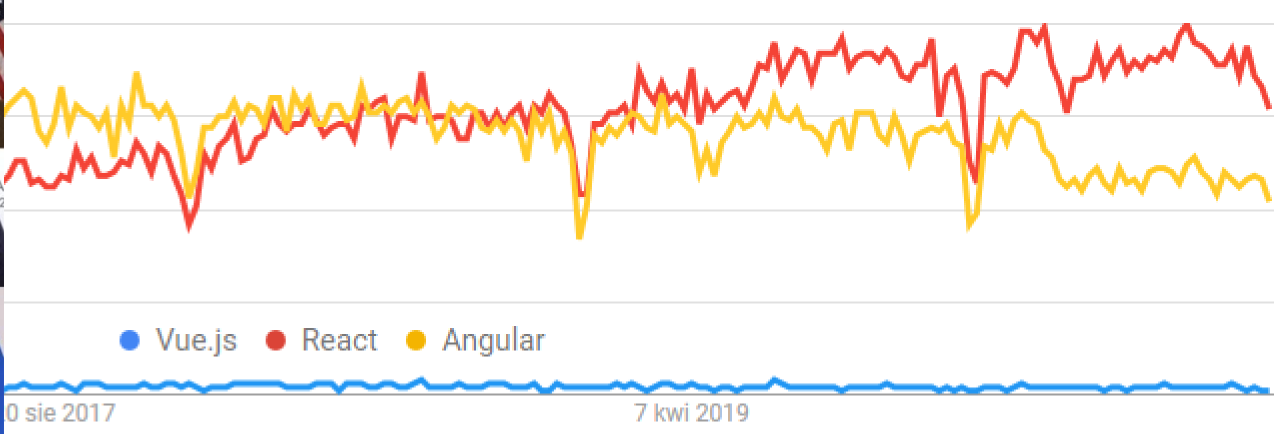
https://w3techs.com/technologies/comparison/js-jquery,js-react



Percentages of websites using the selected JavaScript libraries broken down by ranking



https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangularjs%2Cangular%2Csvelte%2Cdjango%2Cfastapi

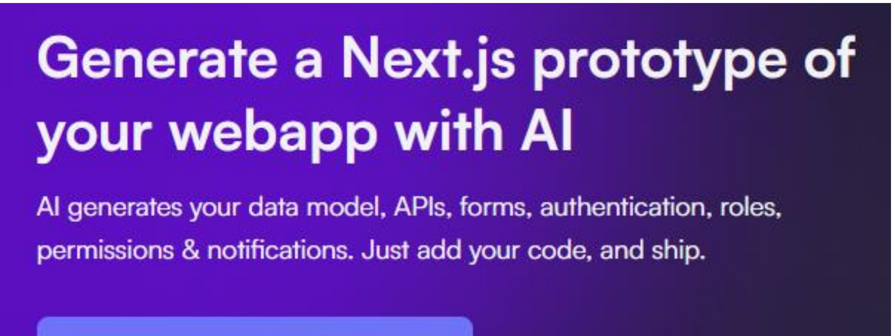


https://trends.google.com/trends/explore?cat=31&date=2016-01-01%202020-10-04&q=Vue.js,React,Angular

Web X.0

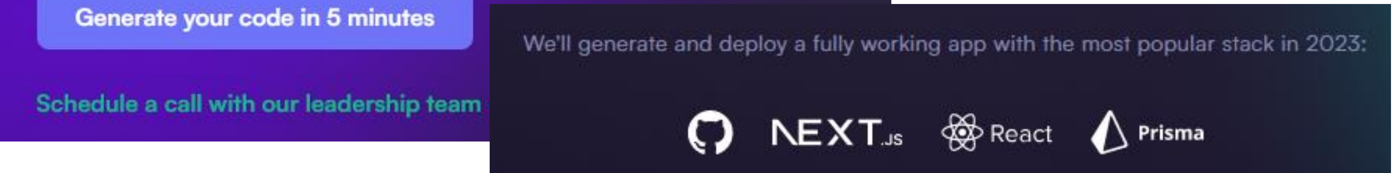
- Web 1.0 – static 1990
- Web 2.0 – dynamic content: forums, blogs, 2000
- Web 3.0 – Semantic Web, AI, mobile … 2008
- Web 4.0 – Active Web, IoT, VR… 20XX

https://www.roq.ai/

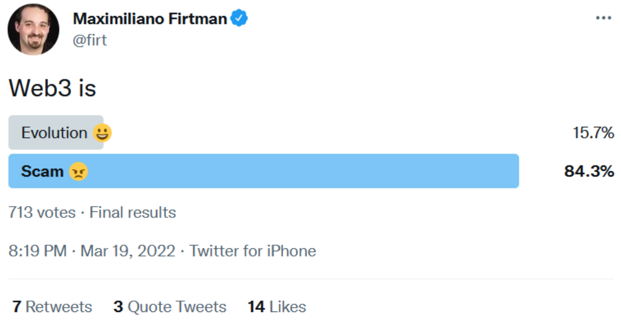


Generate a Next.js prototype of your webapp with AI

AI generates your data model, APIs, forms, authentication, roles, permissions & notifications. Just add your code, and ship.



We'll generate and deploy a fully working app with the most popular stack in 2023:



Maximiliano Firtman
@firt

Web3 is

Evolution 😃	15.7%
Scam 😠	84.3%

713 votes · Final results

8:19 PM · Mar 19, 2022 · Twitter for iPhone

7 Retweets 3 Quote Tweets 14 Likes

HTTP Request

- URI , URL

URI   = scheme ":" hier-part ["?" query] ["#" fragment]

// - start host name known by DNS

/ - denotes local resource path

query : „?” followed "key=value„

authority : username:password@hostname:port

When authority is not present, the path cannot begin with two slash   characters ("//")

https://tools.ietf.org/html/rfc3986#section-3.3

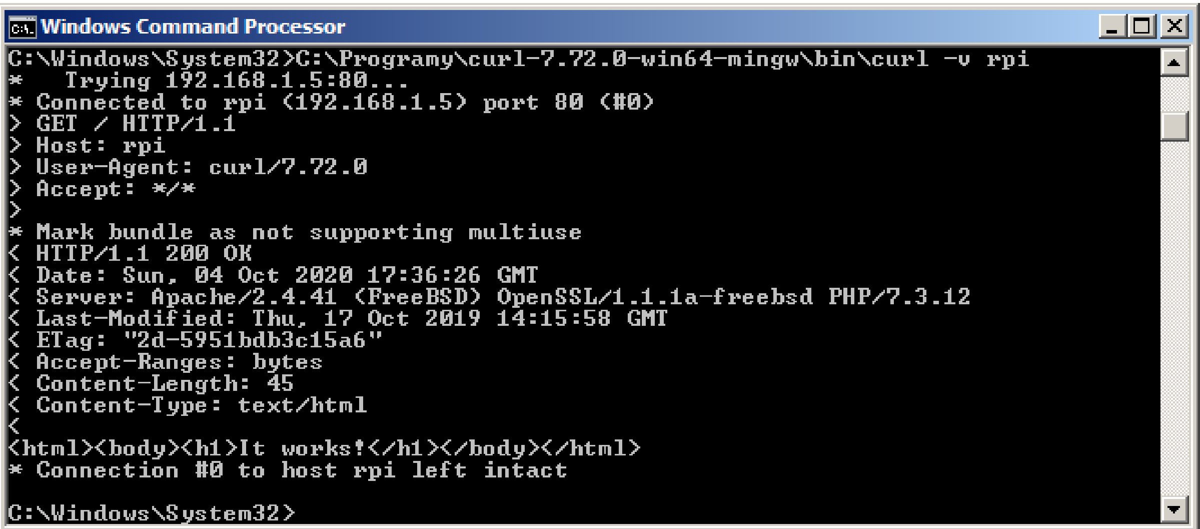
```
The following are two example URIs and their component parts:

         foo://example.com:8042/over/there?name=ferret#nose
         \_/   \______________/\_________/ \_________/ \__/
          |           |            |            |        |
       scheme     authority       path        query   fragment
          |   _____________________|__
         / \ /                        \
         urn:example:animal:ferret:nose
```

http:com/example/www/plik – considered alternative approach without ’//’

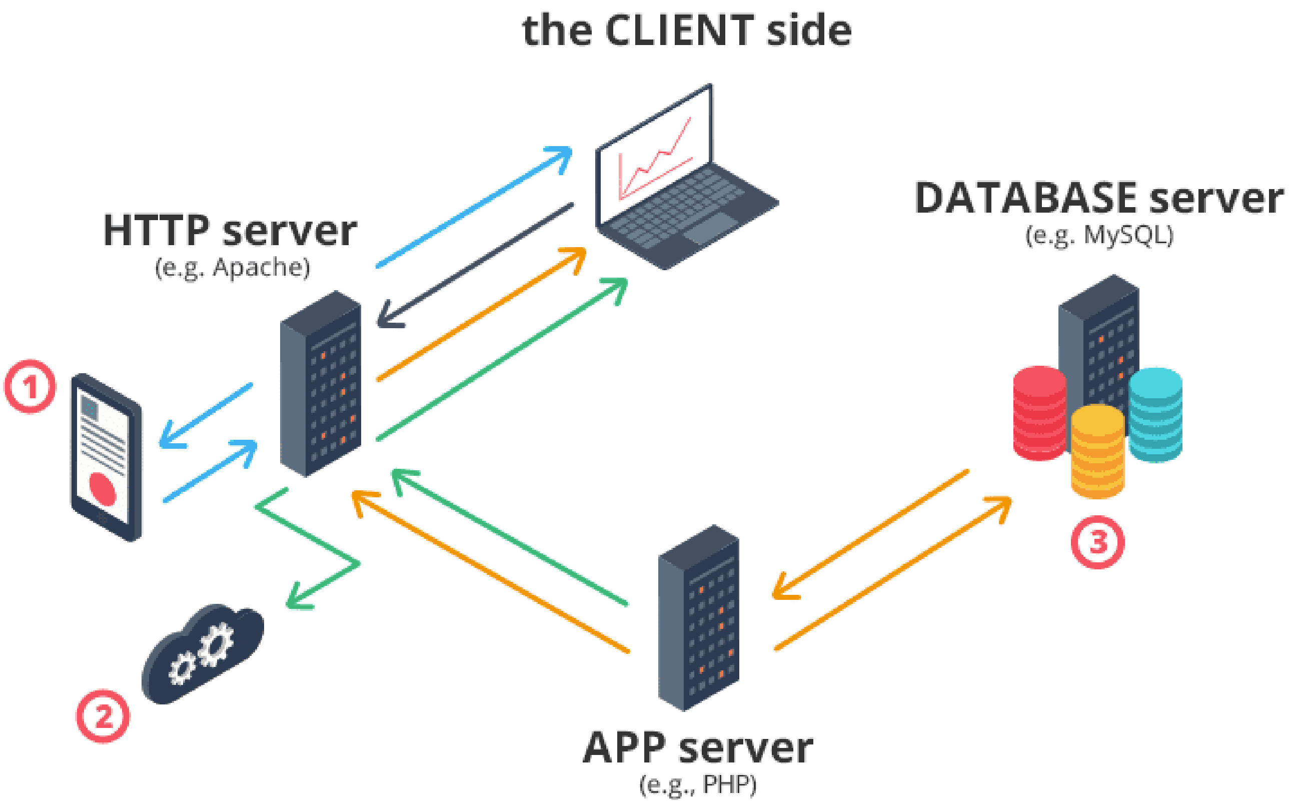
https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Identifying_resources_on_the_Web

scheme:

data	Data URIs data:[<mediatype>][;base64],<data> mediatype=MIME
file	Host-specific file names
ftp	File Transfer Protocol
http/https	Hyper text transfer protocol (Secure)
javascript	URL-embedded JavaScript code
mailto	Electronic mail address
ssh	Secure shell
tel	telephone
urn	Uniform Resource Names
view-source	Source code of the resource
ws/wss	(Encrypted) WebSocket connections



```
C:\Windows\System32>C:\Programy\curl-7.72.0-win64-mingw\bin\curl -v rpi
*   Trying 192.168.1.5:80...
* Connected to rpi (192.168.1.5) port 80 (#0)
> GET / HTTP/1.1
> Host: rpi
> User-Agent: curl/7.72.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sun, 04 Oct 2020 17:36:26 GMT
< Server: Apache/2.4.41 (FreeBSD) OpenSSL/1.1.1a-freebsd PHP/7.3.12
< Last-Modified: Thu, 17 Oct 2019 14:15:58 GMT
< ETag: "2d-5951bdb3c15a6"
< Accept-Ranges: bytes
< Content-Length: 45
< Content-Type: text/html
<
<html><body><h1>It works!</h1></body></html>
* Connection #0 to host rpi left intact

C:\Windows\System32>
```



🔵 Simple html or graphic

🟢 Complex file request

🟠 Database request

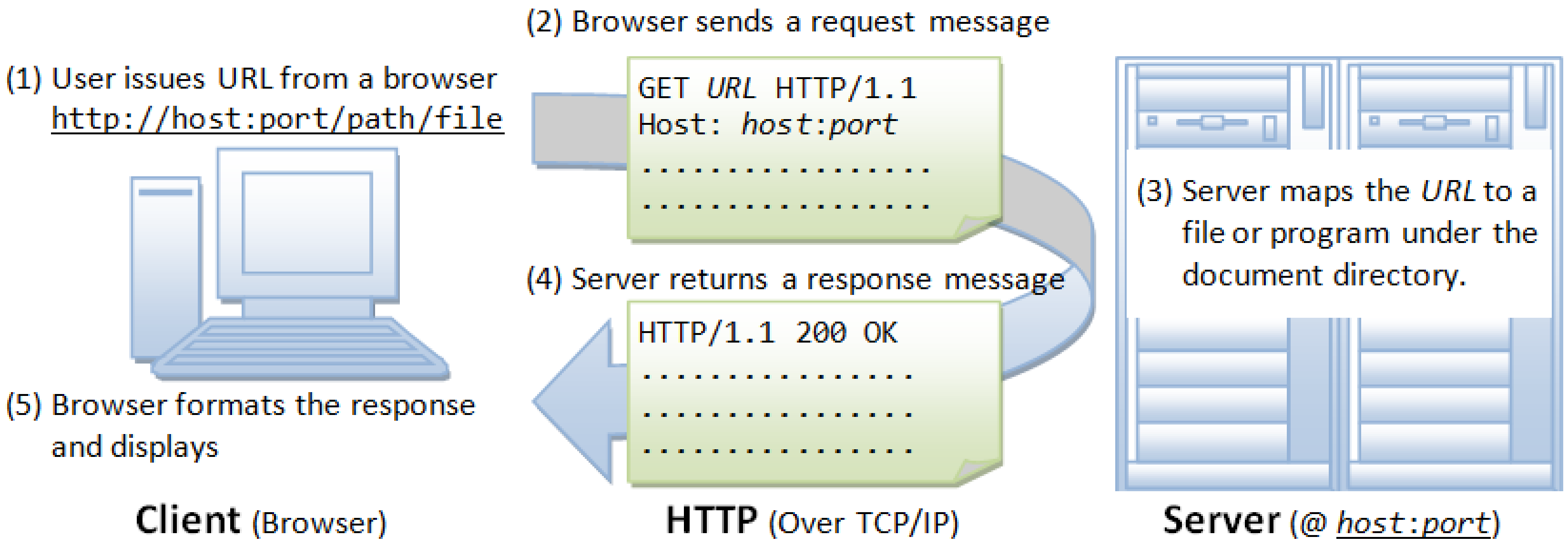
① Is it simple html or graphic file request?
HTTP server can process and send back

② Is it complex file request? HTTP server will send it to the app server
Request is interpreted, executed and sent back to the server. (app server)

③ Is it required to pull, edit, delete or save data?
Server-side script gathers, handles the request and send it back.

https://www.inception.education/technology-stack-for-web-application-development/

Request - Response



(1) User issues URL from a browser
http://host:port/path/file

(2) Browser sends a request message

```
GET URL HTTP/1.1
Host: host:port
.................
.................
```

(3) Server maps the *URL* to a file or program under the document directory.

(4) Server returns a response message

```
HTTP/1.1 200 OK
................
................
................
```

(5) Browser formats the response and displays

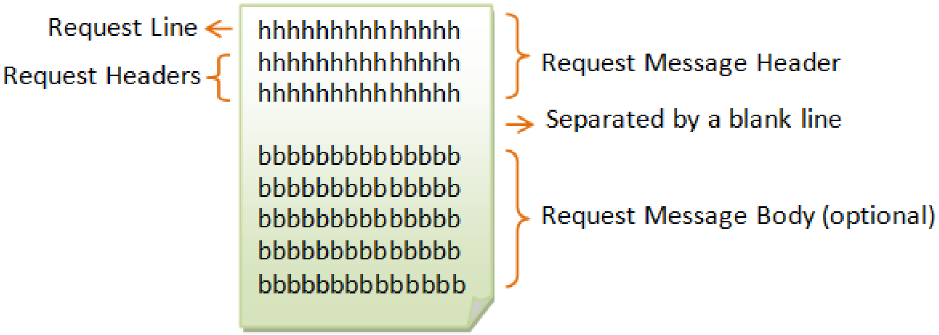
Client (Browser)

HTTP (Over TCP/IP)

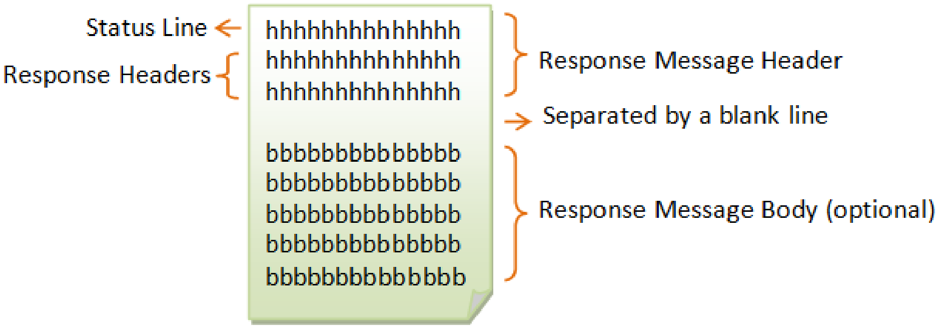
Server (@ *host:port*)

https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

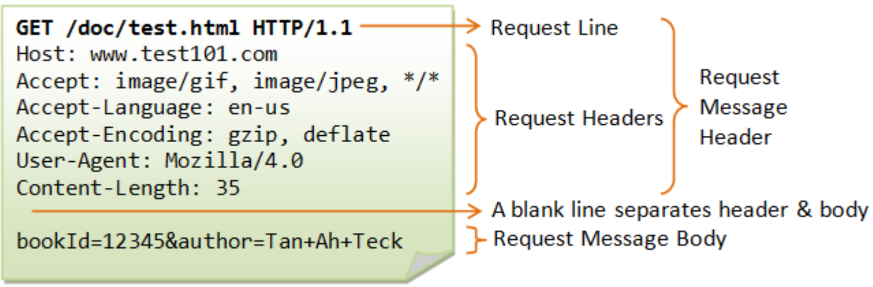
Request – Response cont



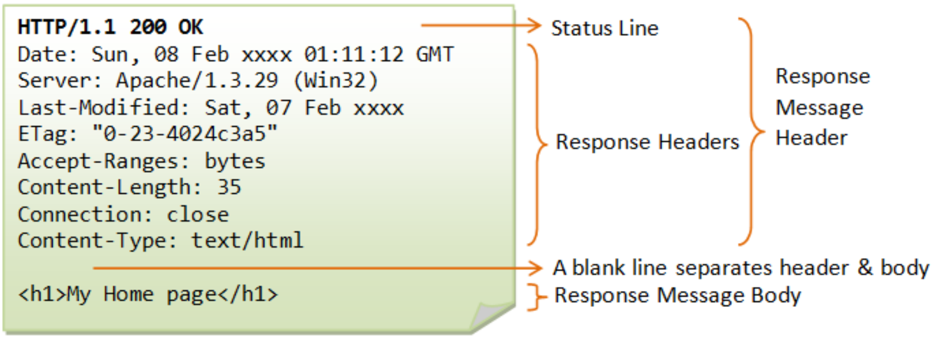
HTTP Request Message



HTTP Response Message



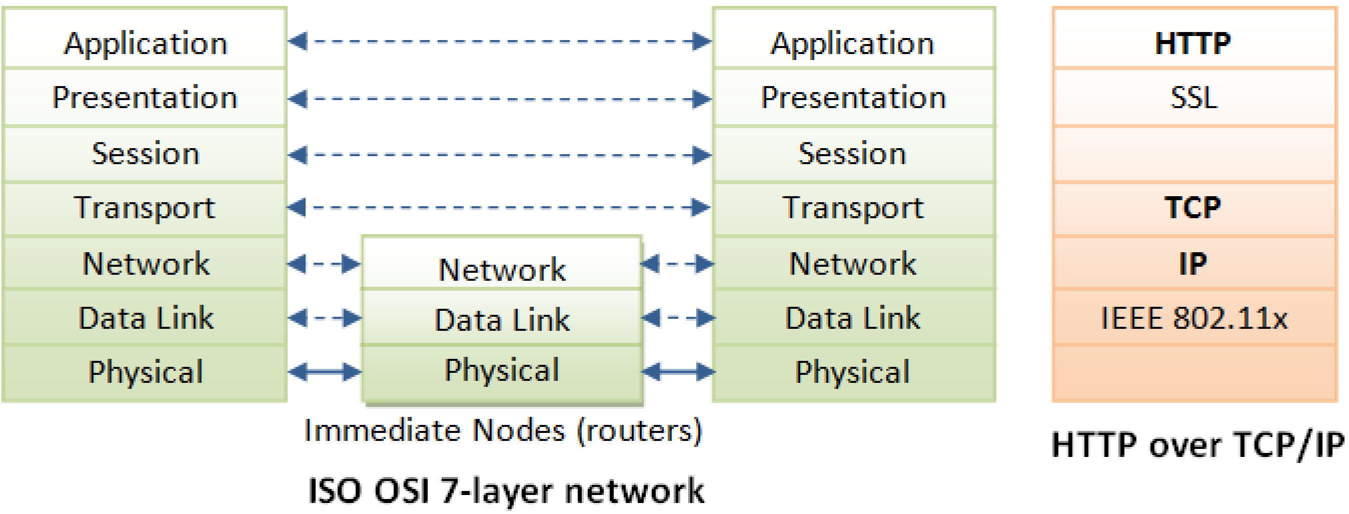
```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

HTTP



Application
Presentation
Session
Transport
Network
Data Link
Physical

Network
Data Link
Physical

Immediate Nodes (routers)

Application
Presentation
Session
Transport
Network
Data Link
Physical

ISO OSI 7-layer network

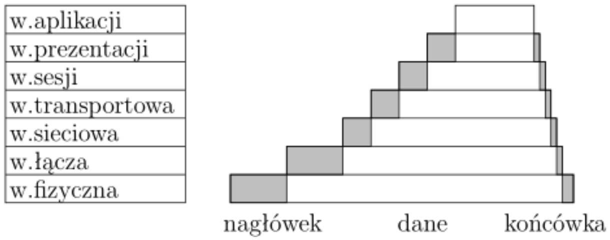
HTTP
SSL

TCP
IP
IEEE 802.11x

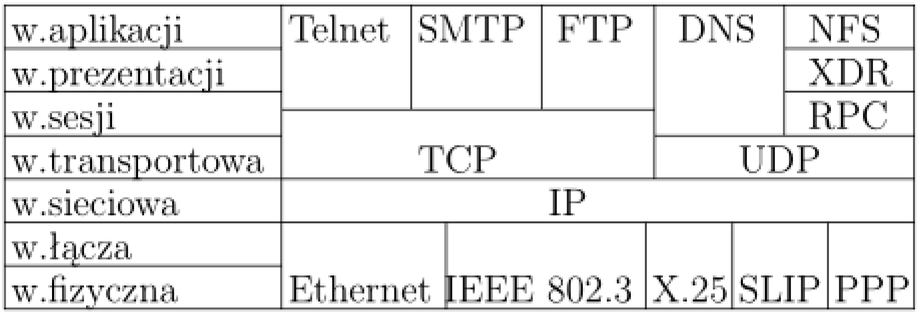
HTTP over TCP/IP

https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

OSI packet encapsulation



Model ISO - OSI



w.aplikacji	Telnet	SMTP	FTP	DNS	NFS
w.prezentacji					XDR
w.sesji					RPC
w.transportowa	TCP			UDP	
w.sieciowa	IP				
w.łącza	Ethernet	IEEE 802.3	X.25	SLIP	PPP
w.fizyczna					

HTTP cont.

Year	HTTP Version	
1991	0.9	
1996	1.0	https://tools.ietf.org/html/rfc1945
1997	1.1	https://www.ietf.org/rfc/rfc2616.txt
		https://www.rfc-editor.org/rfc/rfc2068.html
		AKTUALNE https://tools.ietf.org/html/rfc7230 DO rfc7237
2015	2.0	https://datatracker.ietf.org/doc/draft-thomson-httpbis-http2bis/
2018	3.0	https://datatracker.ietf.org/doc/draft-ietf-quic-http/

- HTTP1 & 2 - TCP
- HTTP3 - UDP

http header live – firefox addon

HTTP - names

hostname

-/etc/hosts C:\Windows\System32\drivers\etc\hosts

-Local caching resolver

-/etc/nscd.conf

-/etc/nsswitch.conf

-DNS

```
nslookup 185.189.125.85
nslookup mini.pw.edu.pl.     #mind the dot!
dig mini.pw.edu.pl.
dig @sextus.mini.pw.edu.pl -x 194.29.160.35
drill ntp.xs4all.nl @127.0.0.1
```

```
less /etc/services

#                 77/tcp     any private RJE service
#                 77/udp     any private RJE service
vettcp            78/tcp
vettcp            78/udp
finger            79/tcp
finger            79/udp
http              80/sctp    www www-http #World Wide Web HTTP
http              80/tcp     www www-http #World Wide Web HTTP
http              80/udp     www www-http #World Wide Web HTTP
hosts2-ns         81/tcp     #HOSTS2 Name Server
hosts2-ns         81/udp     #HOSTS2 Name Server
xfer              82/tcp     #XFER Utility
xfer              82/udp     #XFER Utility
mit-ml-dev        83/tcp     #MIT ML Device
mit-ml-dev        83/udp     #MIT ML Device
…
mobilip-mn       435/udp
https            443/tcp     # http protocol over TLS/SSL
https            443/udp     # http protocol over TLS/SSL
https            443/sctp    # http protocol over TLS/SSL
snpp             444/tcp     # Simple Network Paging Protocol
snpp             444/udp
```

Special-Use Domain Names	
home.arpa.	[RFC8375]
example.	[RFC6761]
example.com.	[RFC6761]
example.net.	[RFC6761]
example.org.	[RFC6761]
invalid.	[RFC6761]
ipv4only.arpa.	[RFC8880]
local.	[RFC6762]
localhost.	[RFC6761]
onion.	[RFC7686]
test.	[RFC6761]

https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml

```
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --permanent --add-port=443/tcp
sudo firewall-cmd --reload
```

start

```
sudo systemctl start httpd
```

Be sure that Apache starts at boot:

```
sudo systemctl enable httpd
```

HTTP status code

- The status code is a 3-digit number:
- 1xx (Informational): Request received, server is continuing the process.
- 2xx (Success): The request was successfully received, understood, accepted and serviced.
- 3xx (Redirection): Further action must be taken in order to complete the request.
- 4xx (Client Error): The request contains bad syntax or cannot be understood.
- 5xx (Server Error): The server failed to fulfill an apparently valid request.

http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
https://httpstatuses.com/

Request Methods

	https://tools.ietf.org/html/rfc7231#page-21
GET	The GET method requests a representation of the specified resource. Requests using GET should only retrieve data. *Consecutive request may result in different data returned*
HEAD	The HEAD method asks for a response identical to that of a GET request, but without the response body.
POST	The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.
PUT	The PUT method replaces all current representations of the target resource with the request payload.
DELETE	The DELETE method deletes the specified resource.
CONNECT	The CONNECT method establishes a tunnel to the server identified by the target resource.
OPTIONS	The OPTIONS method is used to describe the communication options for the target resource.
TRACE	The TRACE method performs a message loop-back test along the path to the target resource.
PATCH	The PATCH method is used to apply partial modifications to a resource.

Headers

CLIENT (request)	SERVER (response)	example
Host		
	ETag	HEAD
Accept	Content-Type	application/json; text/html+xml
Cookie	Set-Cookie	
	Content-Encoding	zip,gzip
	Content-Length	
Accept-Language		fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5
Referer		
Connection	Connection	Keep-alive/close
	Location	HTTP->HTTPS
	Server	
	X-Powered-By	
	Access-Control-Allow-Origin	

Objects , MIME

Multipurpose Internet Mail Extensions

Media Type Specifications

- image/png
- application/rss+xml
- video/mp4; codecs="avc1.640028"
- application/vnd.google-earth.kmz
- Currently, there are nine top-level types, which are: "application", "audio", "example", "image", "message", "model", "multipart", "text" and "video".

elements having attributes like :
type, enctype, formenctype and accept

[type]/[tree.][subtype][+suffix][;parameters]

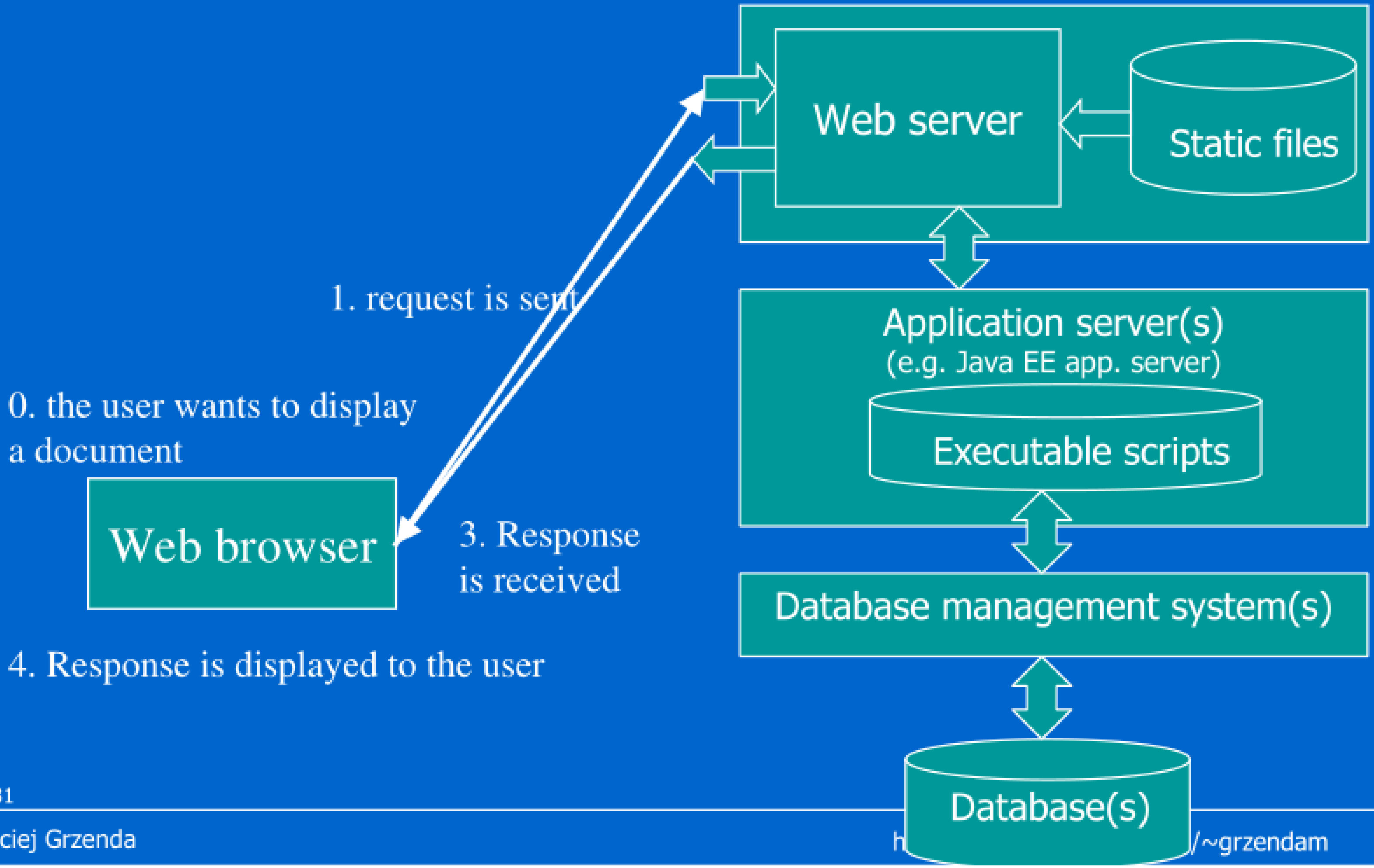
- vnd - vendor specific
- .docx Microsoft Word (OpenXML)
  application/vnd.openxmlformats-officedocument.wordprocessingml.document
- .odt OpenDocument text document
  application/vnd.oasis.opendocument.tex

https://tools.ietf.org/html/rfc6838 https://tools.ietf.org/html/rfc2045
https://www.iana.org/assignments/media-types/media-types.xhtml

Server-side scripts

- Allow to generate dynamic content (e.g. display the status of the order you made in an internet shop)
- Can be treated as programs that generate web content, in most cases HTML documents
- In most cases access databases to query and update their content
- Typical server-side technologies:
  - Microsoft: ASP,ASP.NET
  - Java-based: JSP, JavaServlets
  - PHP
  - CGI (historically first, just starts the executable file that produces the web content)
  - …

Dynamic documents: a wider perspective



Web server

Static files

Application server(s)
(e.g. Java EE app. server)

Executable scripts

Database management system(s)

Database(s)

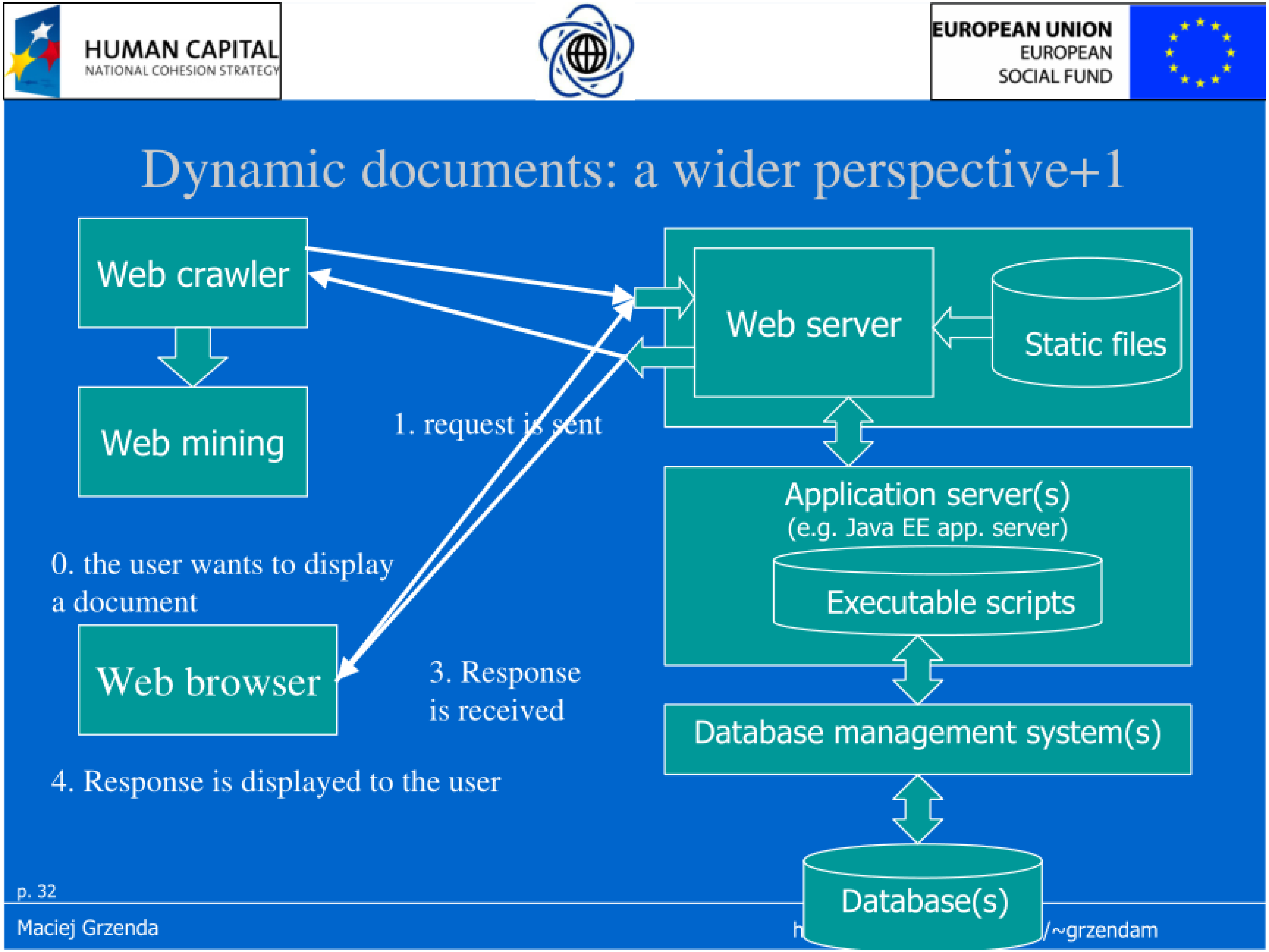
1. request is sent

0. the user wants to display a document

Web browser

3. Response is received

4. Response is displayed to the user



Dynamic documents: a wider perspective+1

Web crawler

Web mining

Web server

Static files

1. request is sent

0. the user wants to display
a document

Web browser

3. Response
is received

4. Response is displayed to the user

Application server(s)
(e.g. Java EE app. server)

Executable scripts

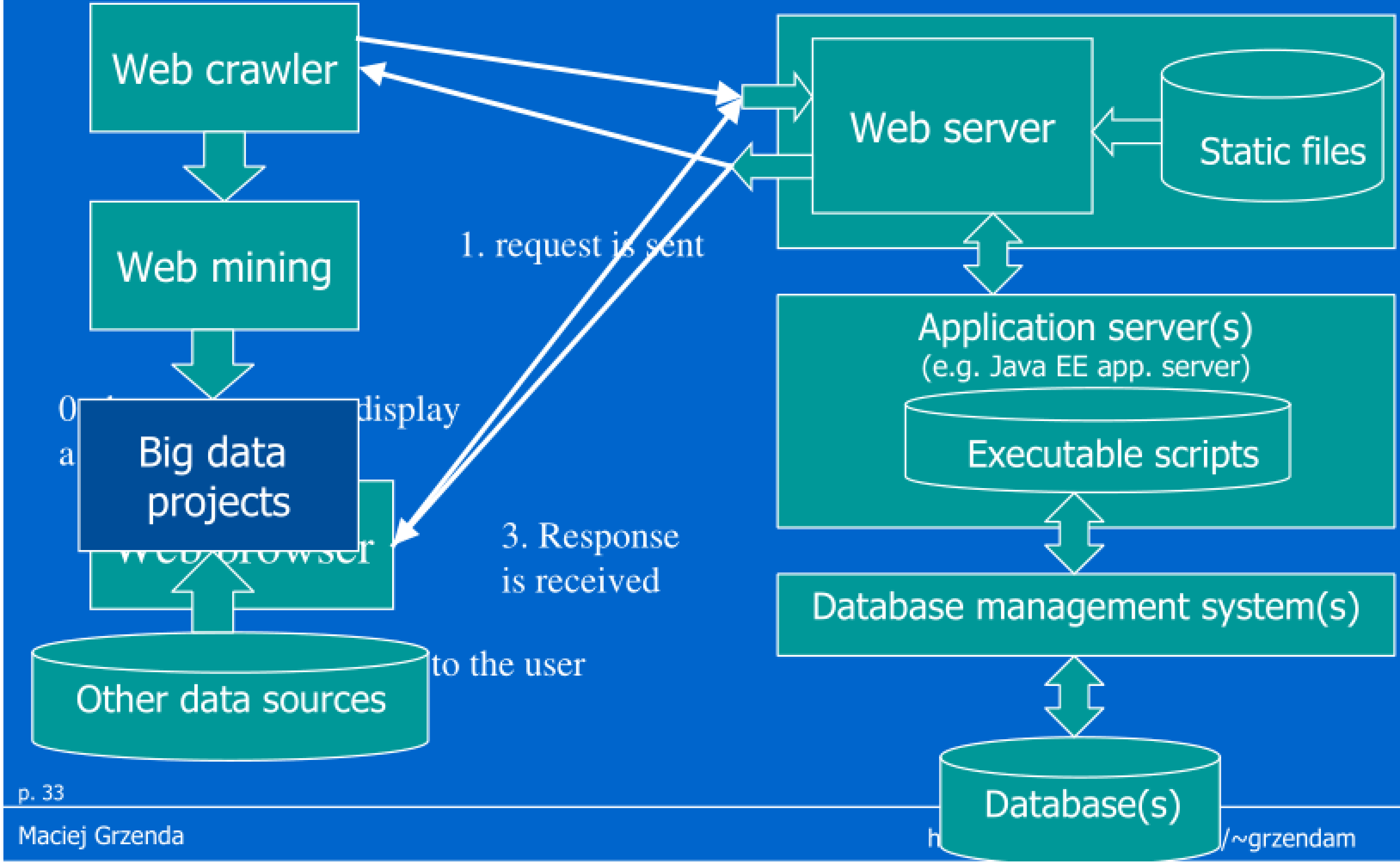
Database management system(s)

Database(s)

HUMAN CAPITAL
NATIONAL COHESION STRATEGY

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

Dynamic documents: a wider perspective+2



Web crawler

Web mining

Big data
projects

Web browser

Other data sources

0

display

a

to the user

1. request is sent

3. Response
is received

Web server

Static files

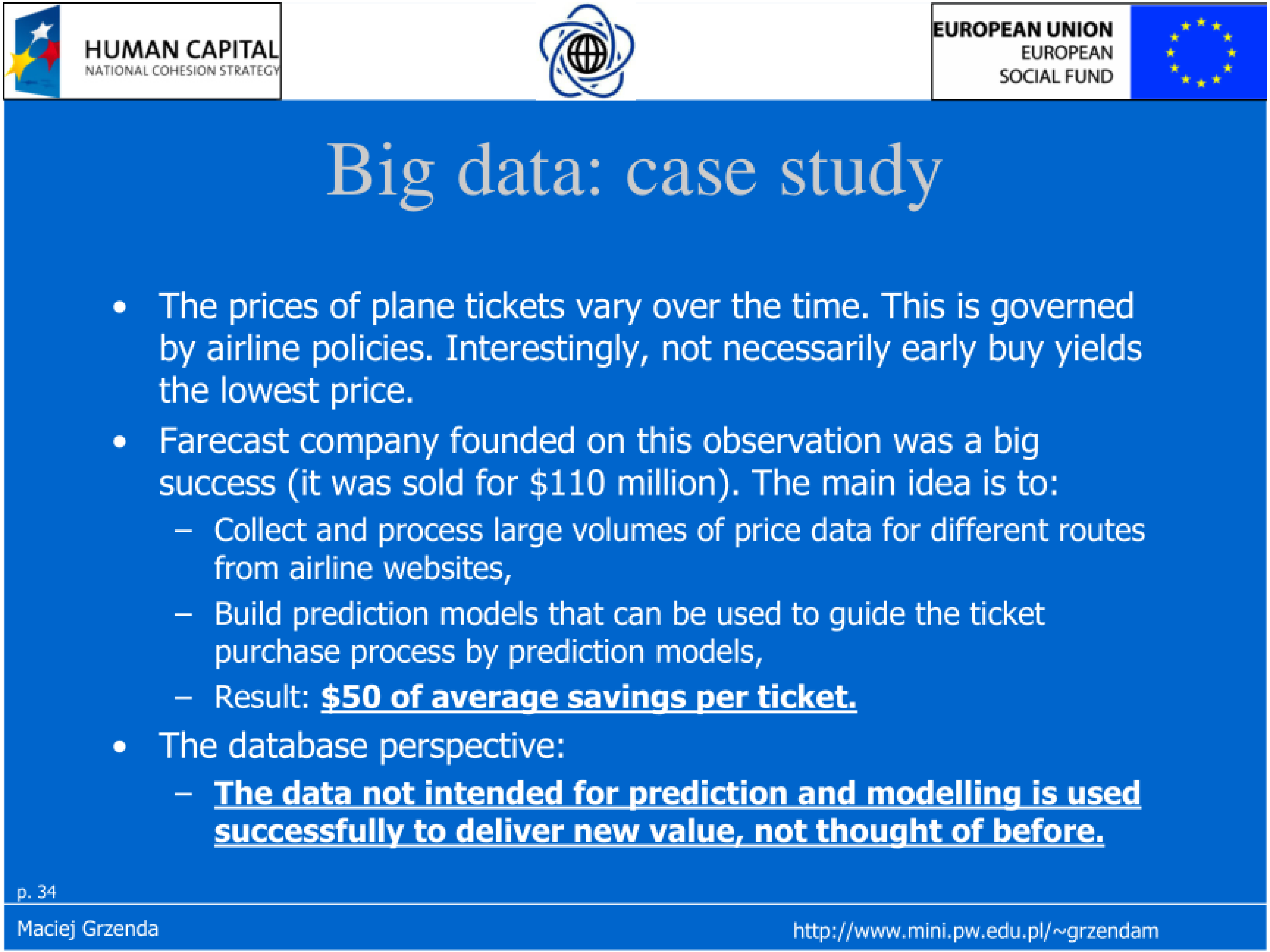
Application server(s)
(e.g. Java EE app. server)

Executable scripts

Database management system(s)

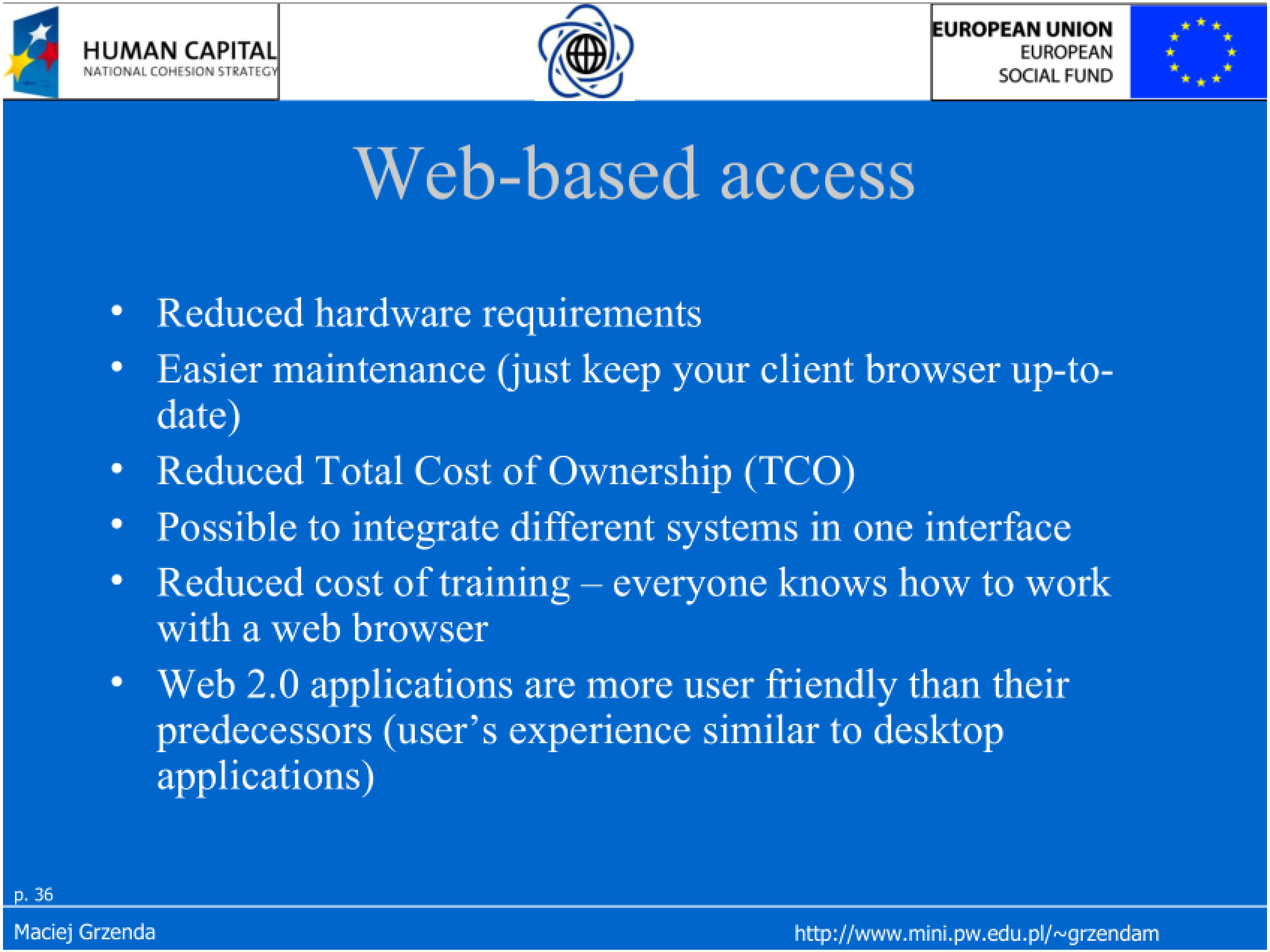
Database(s)

p. 33



Big data: case study

- The prices of plane tickets vary over the time. This is governed by airline policies. Interestingly, not necessarily early buy yields the lowest price.
- Farecast company founded on this observation was a big success (it was sold for $110 million). The main idea is to:
  - Collect and process large volumes of price data for different routes from airline websites,
  - Build prediction models that can be used to guide the ticket purchase process by prediction models,
  - Result: **$50 of average savings per ticket.**
- The database perspective:
  - **The data not intended for prediction and modelling is used successfully to deliver new value, not thought of before.**



HUMAN CAPITAL
NATIONAL COHESION STRATEGY

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

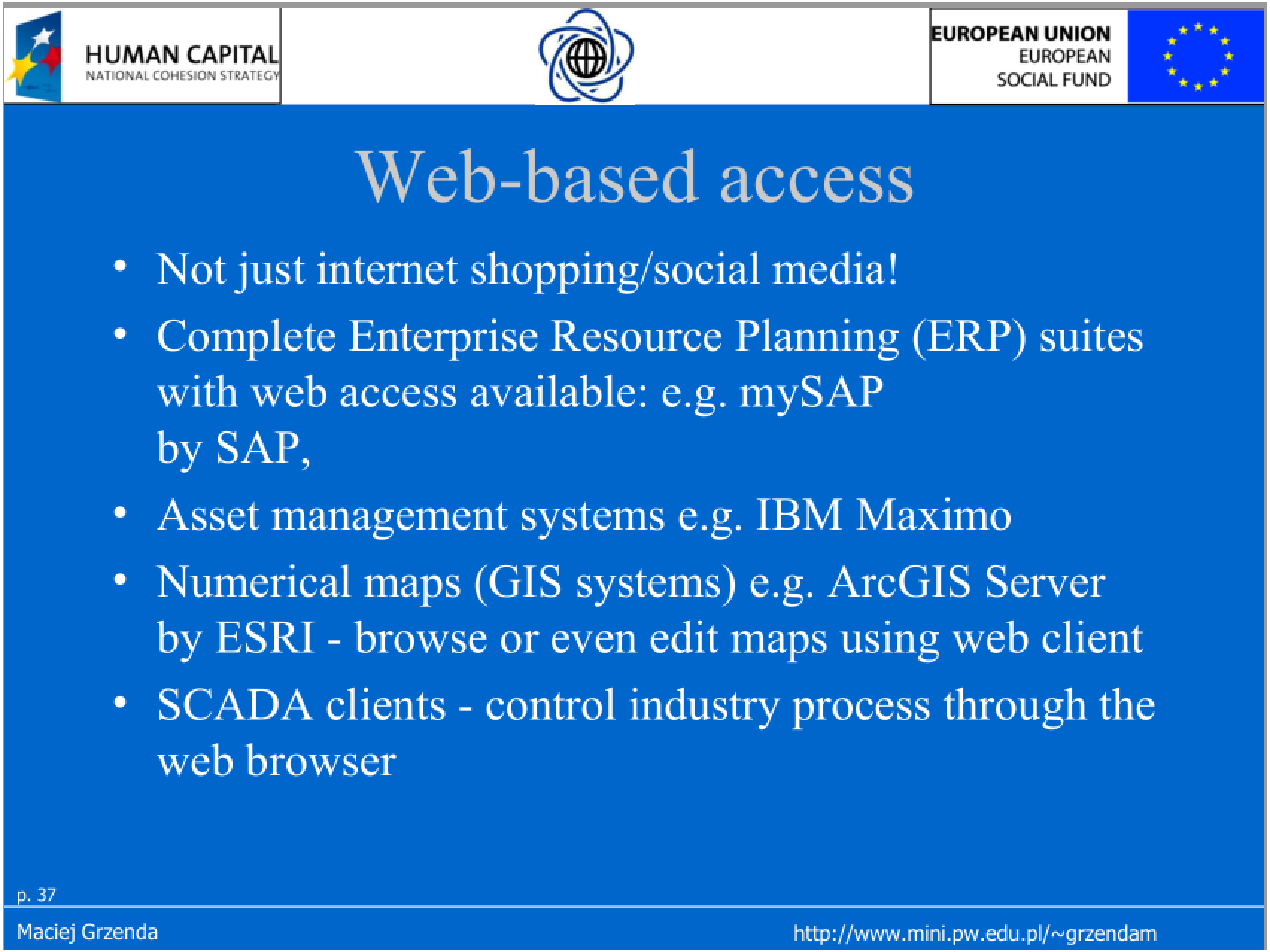
Web-based access

- Reduced hardware requirements
- Easier maintenance (just keep your client browser up-to-date)
- Reduced Total Cost of Ownership (TCO)
- Possible to integrate different systems in one interface
- Reduced cost of training – everyone knows how to work with a web browser
- Web 2.0 applications are more user friendly than their predecessors (user’s experience similar to desktop applications)

p. 36

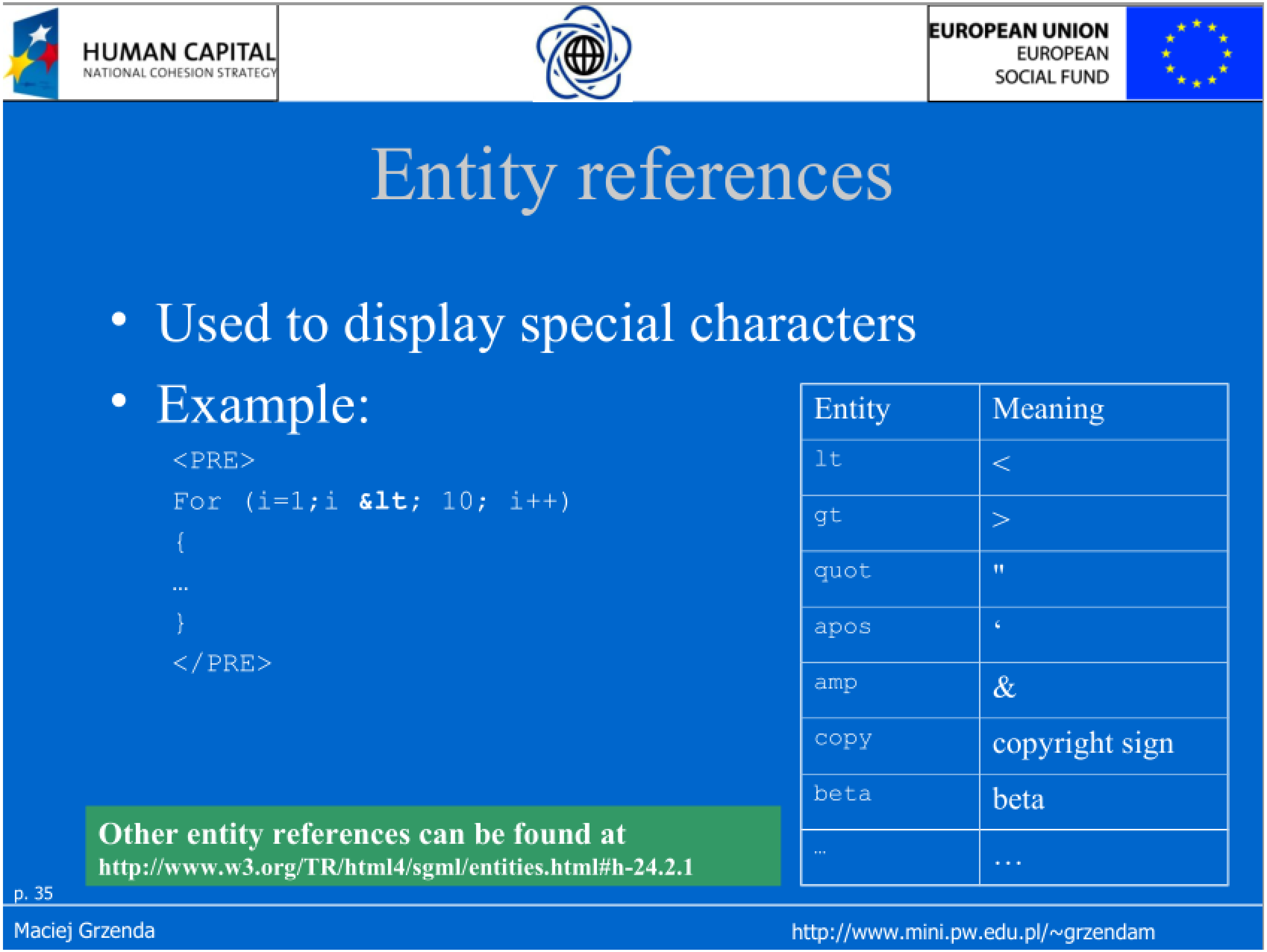
Maciej Grzenda

http://www.mini.pw.edu.pl/~grzendam



Web-based access

- Not just internet shopping/social media!
- Complete Enterprise Resource Planning (ERP) suites with web access available: e.g. mySAP by SAP,
- Asset management systems e.g. IBM Maximo
- Numerical maps (GIS systems) e.g. ArcGIS Server by ESRI - browse or even edit maps using web client
- SCADA clients - control industry process through the web browser



Entity references

- Used to display special characters
- Example:

```
<PRE>
For (i=1;i &lt; 10; i++)
{
…
}
</PRE>
```

Entity	Meaning
lt	<
gt	>
quot	"
apos	‘
amp	&
copy	copyright sign
beta	beta
…	…

Other entity references can be found at
http://www.w3.org/TR/html4/sgml/entities.html#h-24.2.1

Entities

https://www.w3schools.com/html/html_entities.asp

HTML Character Entities

semicolon

	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	double quotation mark	&quot;	&#34;
'	single quotation mark (apostrophe)	&apos;	&#39;

Combining Diacritical Marks

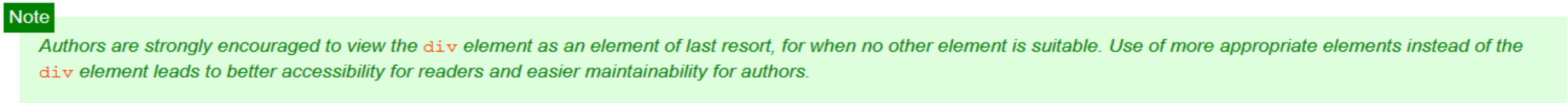
`	O	O&#768;	Ò
´	O	O&#769;	Ó
ˆ	O	O&#770;	Ô
˜	O	O&#771;	Õ

Block and inline

- Every HTML element has a default display property value, depending on what type of element it is. There are two display values: block and inline.
- In HTML, the default display property value is taken from the HTML specifications or from the browser/user default style sheet. The default value in XML is inline, including SVG elements.

https://www.w3schools.com/cssref/pr_class_display.asp

- DIV
  - A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).
  - has no required attributes, but style, class and id are common
  - often used as a container for other HTML elements
  - starts on a new line and takes up the full width available
- SPAN
  - An inline element does not start on a new line and it only takes up as much width as necessary.
  - An inline element cannot contain a block-level element does not start on a new line and it only takes up as much width as necessary
  - used to mark up a part of a text, or a part of a document
  - The span element doesn't mean anything on its own, but can be useful when used together with the global attributes, e.g. class, lang, or dir

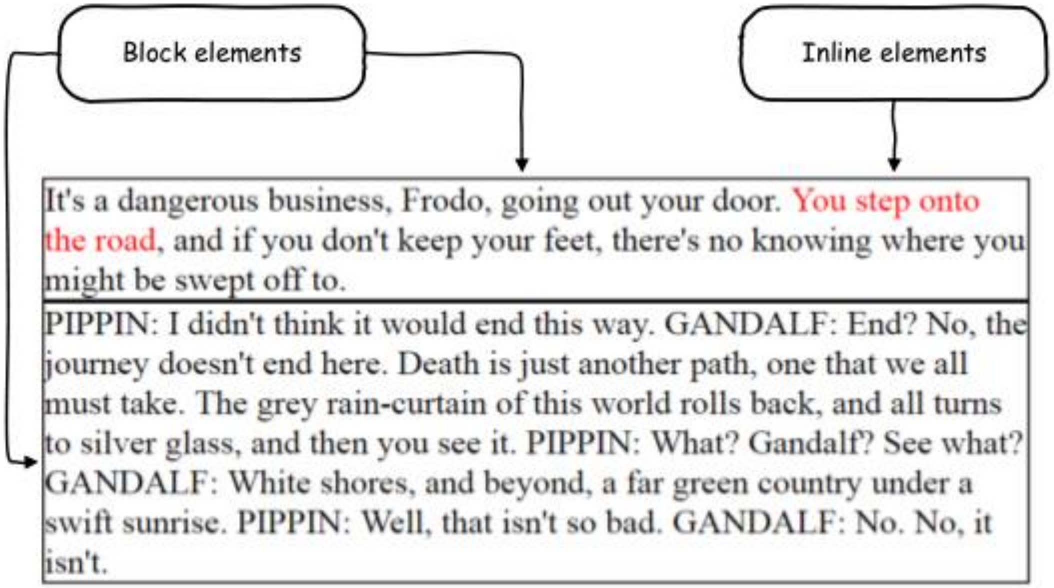


Note

Authors are strongly encouraged to view the `div` element as an element of last resort, for when no other element is suitable. Use of more appropriate elements instead of the `div` element leads to better accessibility for readers and easier maintainability for authors.

https://html.spec.whatwg.org/multipage/grouping-content.html#the-div-element

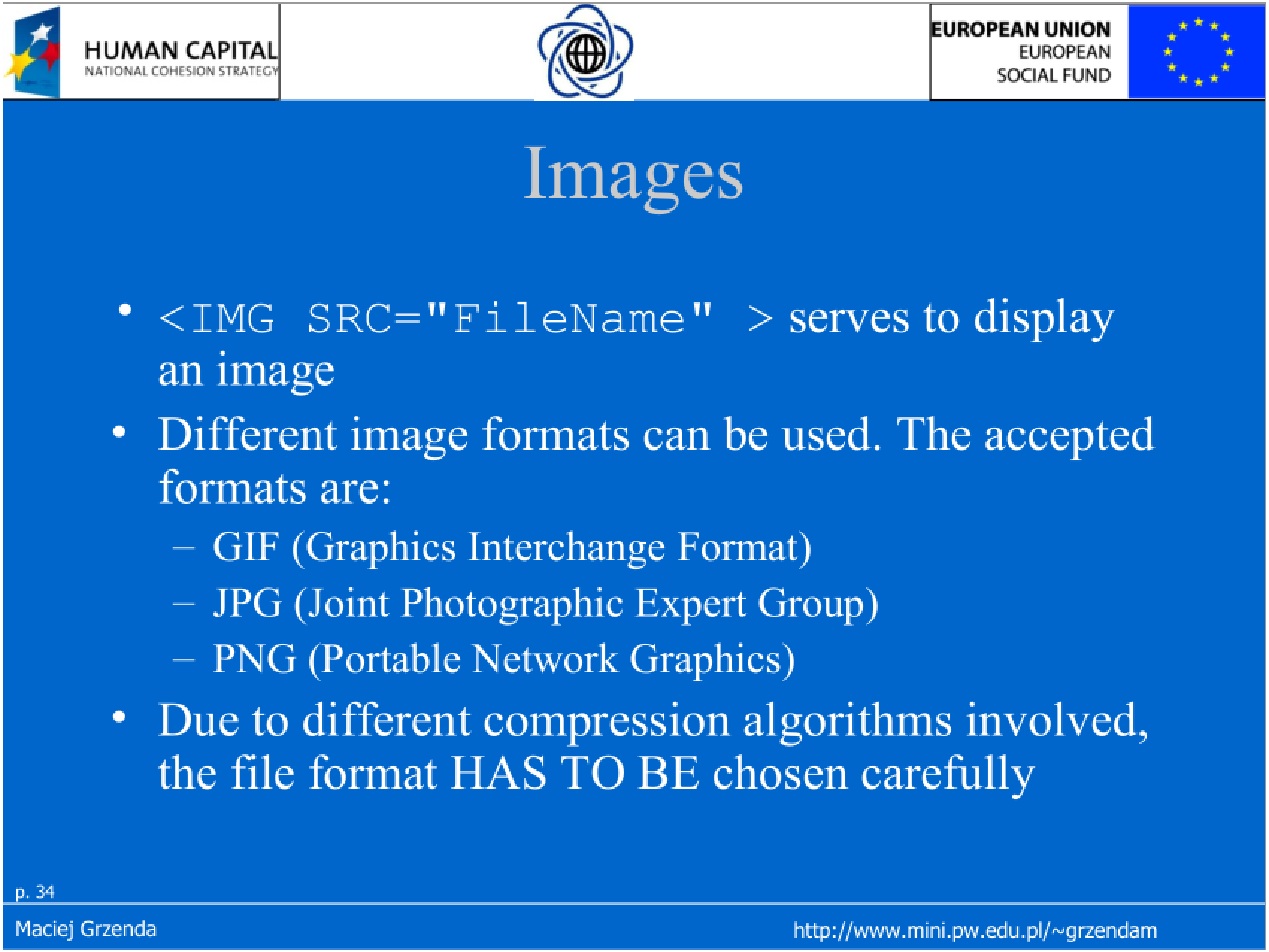
Block vs Inline



It's a dangerous business, Frodo, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

PIPPIN: I didn't think it would end this way. GANDALF: End? No, the journey doesn't end here. Death is just another path, one that we all must take. The grey rain-curtain of this world rolls back, and all turns to silver glass, and then you see it. PIPPIN: What? Gandalf? See what? GANDALF: White shores, and beyond, a far green country under a swift sunrise. PIPPIN: Well, that isn't so bad. GANDALF: No. No, it isn't.

SHOW EXAMPLE



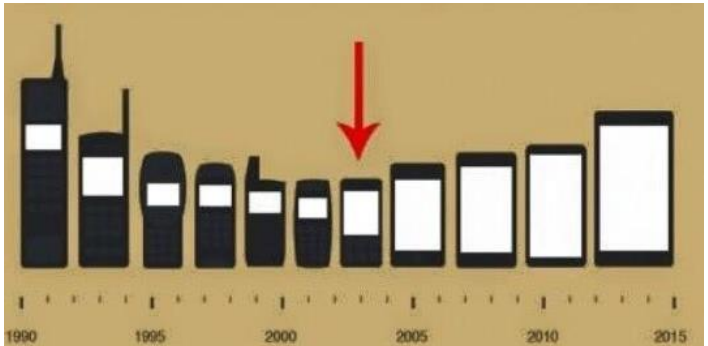
Images

- `<IMG SRC="FileName" >` serves to display an image
- Different image formats can be used. The accepted formats are:
  - GIF (Graphics Interchange Format)
  - JPG (Joint Photographic Expert Group)
  - PNG (Portable Network Graphics)
- Due to different compression algorithms involved, the file format HAS TO BE chosen carefully

Multimedia

- GIF - Graphicts Interchange Format
- PNG - GIF free alternative,some problems
- JPEG
- WebP - Google (Asymmetric Numeral Systems)
- SVG - Vector Graphic (XML Application)

- Only MP4, WebM, and Ogg video are supported by the HTML standard
- Only M4A, MP3, WAV, and Ogg audio are supported by the HTML standard



Design

sRGB = standard RGB colorspace *IEC 61966-2-1:1999 Standard*

https://www.w3.org/TR/css-color-3/#SRGB

Desing:
-10-bit per primary color depth display
-Color calibration
- Adobe color gamut

→

Consumer:
- 8 bit
- color depends on device
-- srgb

- Color perception

Print
72 points / inch -> 1/6 inch is 12pt

Web
96 pixels / inch -> 1/6 inch is 16px
-Page scaling
- High DPI os display

<a> element (or *anchor* element)

- creates a hyperlink to
  - web pages, files, email addresses, locations in the same page (fragment), or anything else a URL can address
  - browser settings define action

```
<a href="https://example.com">Website</a>
<a href="mailto:m.bluth@example.com">Email</a>
<a href="tel:+123456789">Phone</a>
```

- target

_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window (default)
_parent	The response is displayed in the parent frame, If no parent = _self.
_top	The response is displayed in the full body of the window
iframe_name	The response is displayed in a named iframe

<a> element cont.

Fragment

```
<A id="section1" />
...
<A HREF="#section1"> Section 1 </A>
```

http://example.com/index.html#section1

Location

```
<a href="//example.com"> Scheme-relative URL </a>
-http:
<a href="/en-US/docs/Web/HTML"> Origin-relative URL </a>
-http://example.com
<a href="./p"> Directory-relative URL</a>
<a href="http://example.com/path/to/file.zip"> Absolute URL</a>
```

Image as link

```
<a href="default.asp">
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

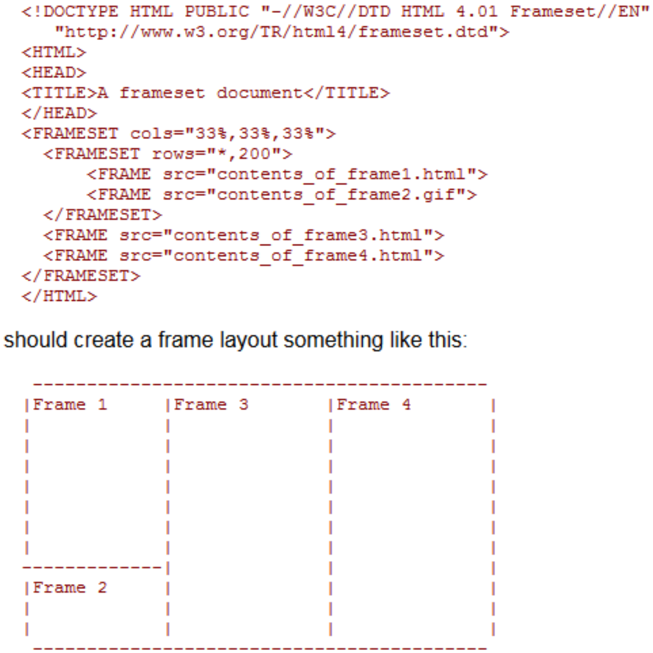
Button as link

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

iframe

- <iframe> tag specifies an *inline frame*
- src attribute – define URL to embeed
- can be sandboxed (isolated)
- is able to "float"
  within content in a page
- <a target="iframe_name">

Obsolete HTML4
- <frameset> ~ <body>
- <frame>



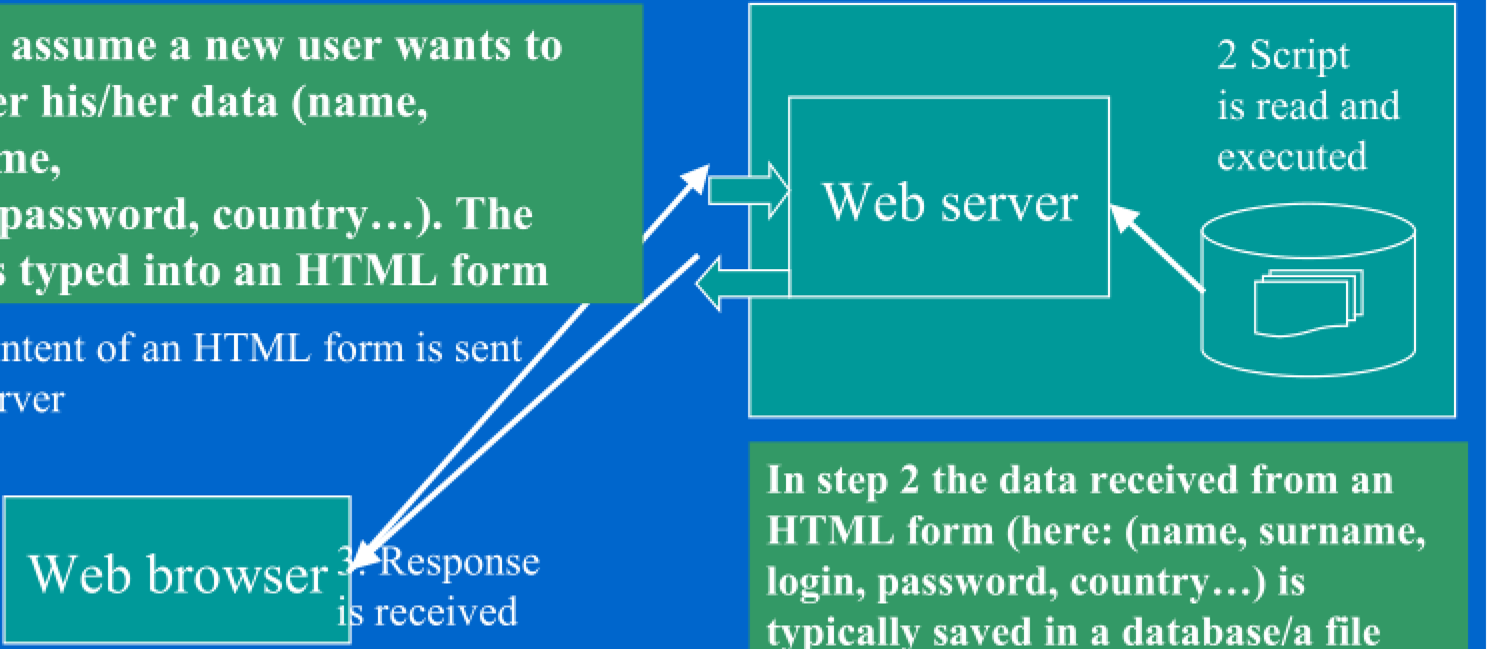
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
   "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>A frameset document</TITLE>
</HEAD>
<FRAMESET cols="33%,33%,33%">
  <FRAMESET rows="*,200">
      <FRAME src="contents_of_frame1.html">
      <FRAME src="contents_of_frame2.gif">
  </FRAMESET>
  <FRAME src="contents_of_frame3.html">
  <FRAME src="contents_of_frame4.html">
</FRAMESET>
</HTML>
```

should create a frame layout something like this:

```
 ------------------------------------------
|Frame 1     |Frame 3       |Frame 4       |
|            |              |              |
|            |              |              |
|            |              |              |
|            |              |              |
|            |              |              |
|            |              |              |
-------------|              |              |
|Frame 2     |              |              |
|            |              |              |
|            |              |              |
 ------------------------------------------
```

HTML forms – an example

**Let us assume a new user wants to register his/her data (name, surname,
login, password, country…). The data is typed into an HTML form**

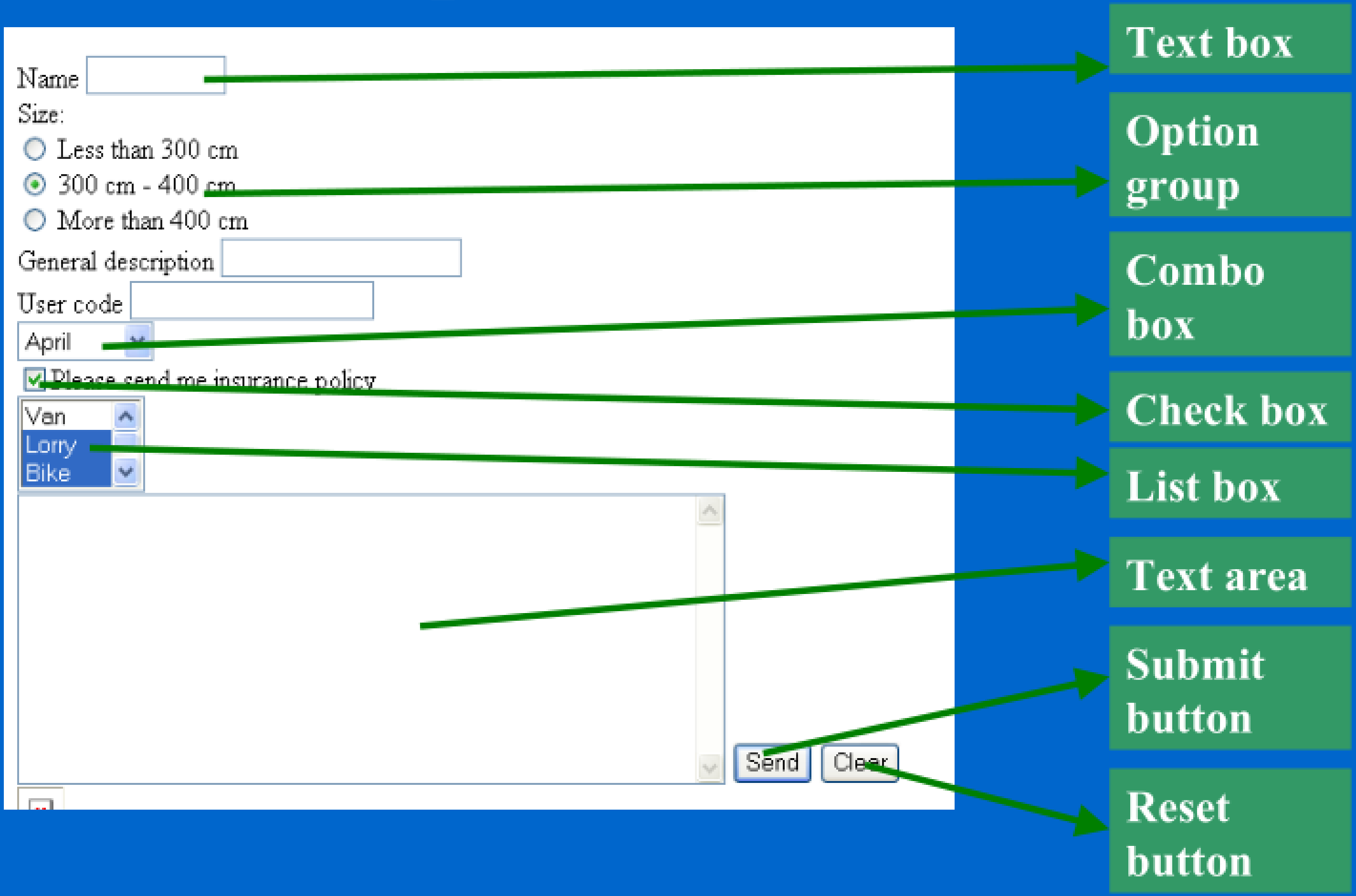


1. the content of an HTML form is sent to the server

4. Response is displayed to the user

In step 2 the data received from an HTML form (here: (name, surname, login, password, country…) is typically saved in a database/a file and/or used to answer the query e.g. display a list of books on a certain topic

An example: „Rent a car" form



Name

Size:

○ Less than 300 cm

● 300 cm - 400 cm

○ More than 400 cm

General description

User code

April

☑ Please send me insurance policy

Van
Lorry
Bike

Text box

Option group

Combo box

Check box

List box

Text area

Submit button

Reset button

HTML Forms – FORM tag

- Encloses a group of input items
- Define where and how send the data of the form
- Form can not be embedded in another form
- Template code:

```
<BODY>
  …
<FORM …>
<!-- input items -->
  …
</FORM>
…
</BODY>
```

<!-- and --> are used to start and finish block of comments in HTML

<form> element

- The HTML <form> element can contain one or more of the following form elements:

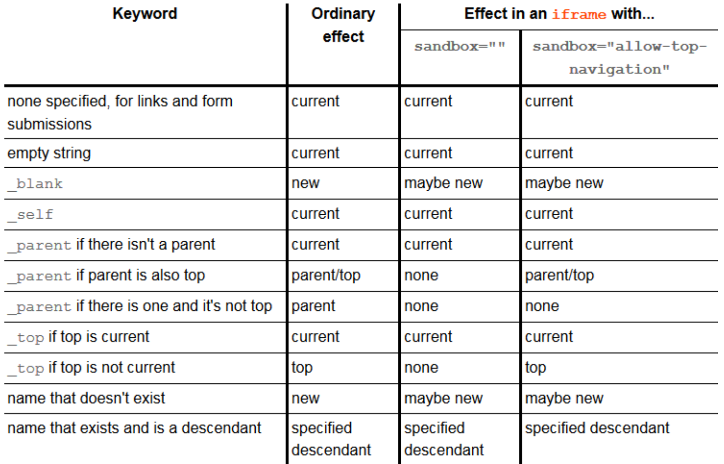
<input>	most used form element
<label>	defines a label for several form elements.
<select>	defines a drop-down list z <option>
<textarea>	multi-line input field
<button>	<button type="button" onclick="alert('Hello World!')">Click Me!</button>
<fieldset>	group related data in a form
<legend>	element defines a caption for the <fieldset> element.
<datalist>	list of pre-defined options/suggestions for an <input> element.
<output>	result of a calculation
<option value=“”>	defines an option in a select list <select>, <optgroup>, or <datalist>
<optgroup>	groups of related options are easier to handle for a user.

Notice that each input **field must have a <u>name</u> attribute to be submitted**.
If the name attribute is omitted, the value of the input field will not be sent at all.

<form> attributes

Attribute	Description
• accept-charset	Specifies the character encodings used for form submission (sending data)
~~• accept~~	removed in HTML5, use the accept attribute on <input type=file> elements.
• **action**	Specifies where to send the form-data when a form is submitted
• autocomplete	Specifies whether a form should have autocomplete on or off
• enctype	Specifies how the form-data should be encoded when submitting it to the server (only for method="post") - application/x-www-form-urlencoded (default) - multipart/form-data (input file) - text/plain: Introduced by HTML5 for debugging purposes
• **method**	Specifies the HTTP method to use when sending form-data
• name	Specifies the name of the form
• novalidate	Specifies that the form should not be validated when submitted
• rel	Specifies the relationship between a linked resource and the current document
• target	Specifies where to display the response that is received after submitting the form

The <input type="submit"> defines a button for submitting the form data to a form-handler.
The form-handler is typically a file on the server with a script for processing input data.
The form-handler is specified in the form's action attribute.



Keyword	Ordinary effect	Effect in an iframe with...	
		sandbox=""	sandbox="allow-top-navigation"
none specified, for links and form submissions	current	current	current
empty string	current	current	current
_blank	new	maybe new	maybe new
_self	current	current	current
_parent if there isn't a parent	current	current	current
_parent if parent is also top	parent/top	none	parent/top
_parent if there is one and it's not top	parent	none	none
_top if top is current	current	current	current
_top if top is not current	top	none	top
name that doesn't exist	new	maybe new	maybe new
name that exists and is a descendant	specified descendant	specified descendant	specified descendant

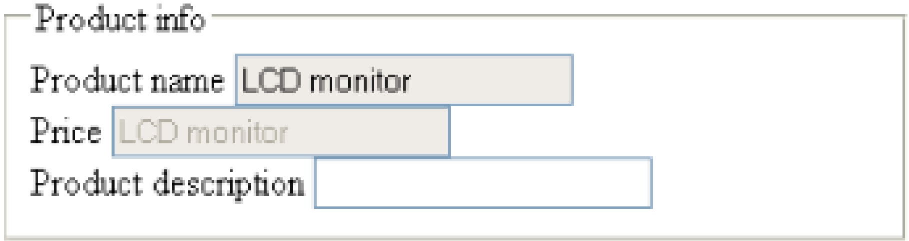
https://html.spec.whatwg.org/multipage/browsers.html#valid-browsing-context-name-or-keyword

Forms – element grouping

Forms – element grouping

- LABEL, FIELDSET and LEGEND tags are used to group and describe form items
- An example:

```
<form …>
…
<fieldset>
      <legend accesskey="M">Product info</legend>
      <label for="pname">Product name</label>
      <input type="text" value="LCD monitor" readonly="readonly" tabindex="2” id="pname"
      name="prodname"><br />
      <label for="price">Price</label>
      <input type="text" value="LCD monitor" disabled="disabled" id="price„ name="prodprice“> <br />
      <label for="info">Product description</label>
      <input  type="text" tabindex="1" id="info" name="prodinfo">
</fieldset>
…
</form>
```



Product info

Product name LCD monitor

Price LCD monitor

Product description

Global attributes

-readonly / readonly=“readonly”

-disabled / disabled=“disabled”

-accesskey=“K” / tabindex=“2”

Form items – text and combo boxes

- Text boxes

```
<INPUT TYPE= "TEXT" [VALUE=DefaultValue] [MAXLENGTH=MaxNoOfChars] [SIZE=ScreenSize] />
```

– Example:

```
<INPUT TYPE= "TEXT" NAME= "Surname" MAXLENGTH="50" />
```

- Pop-up lists (combo boxes)

– Pop-up lists can be defined using the SELECT tag. Every SELECT tag contains a list of OPTION tags,for instance:

```
<SELECT NAME="month">
      <OPTION>January
      <OPTION>February
      <OPTION>March
      <OPTION selected>April
      ....
</SELECT>
```

– In addition, SELECTED attribute defines default value for the list.

Text area and check box

- Text area:

– Used to define multiline input text box

– Example:

```
<TEXTAREA NAME="Description" ROWS="10" COLS="50">
Write here problem description
</TEXTAREA>
```

- Check box:

– Can be set on or off

– Example:

```
<INPUT TYPE="CHECKBOX" id="Policy” name=‘’Policy” CHECKED VALUE= "YES" />
<label for=„Policy"> Send me Policy</label>
```

Label without id – implictly known; id – semantic association eg. For screen readers

```
<label> <INPUT TYPE="CHECKBOX" Name="Policy" CHECKED VALUE= "YES" /> Label <label>
```

Form items – list boxes

List boxes

– Defined as combo boxes, with the size attribute defining the number of displayed items

– Example:

Size = number of visible elements in box

<SELECT NAME="Preferences" SIZE="4">

…

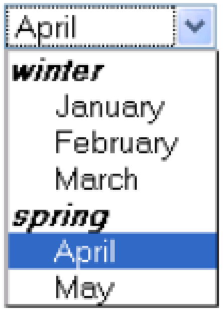
– The SIZE attribute does not define the number of options, which can be greater than the one defined by the SIZE.

– MULTIPLE attribute can be used to allow multiple items to be selected at a time.

Combo boxes and list boxes – element grouping

- Options can be grouped. Moreover a value to be sent to the server in case an option is selected can be specified.
- An example:

```
<SELECT NAME="month">
    <optgroup label="winter">
        <OPTION VALUE="1">January</option>
        <OPTION VALUE="2">February</option>
        <OPTION VALUE="3">March</option>
    </optgroup>
    <optgroup label="spring">
        <OPTION VALUE="4" SELECTED>April</option>
        <OPTION VALUE="5"> May</option>
    </optgroup>
</SELECT>
```



Option groups

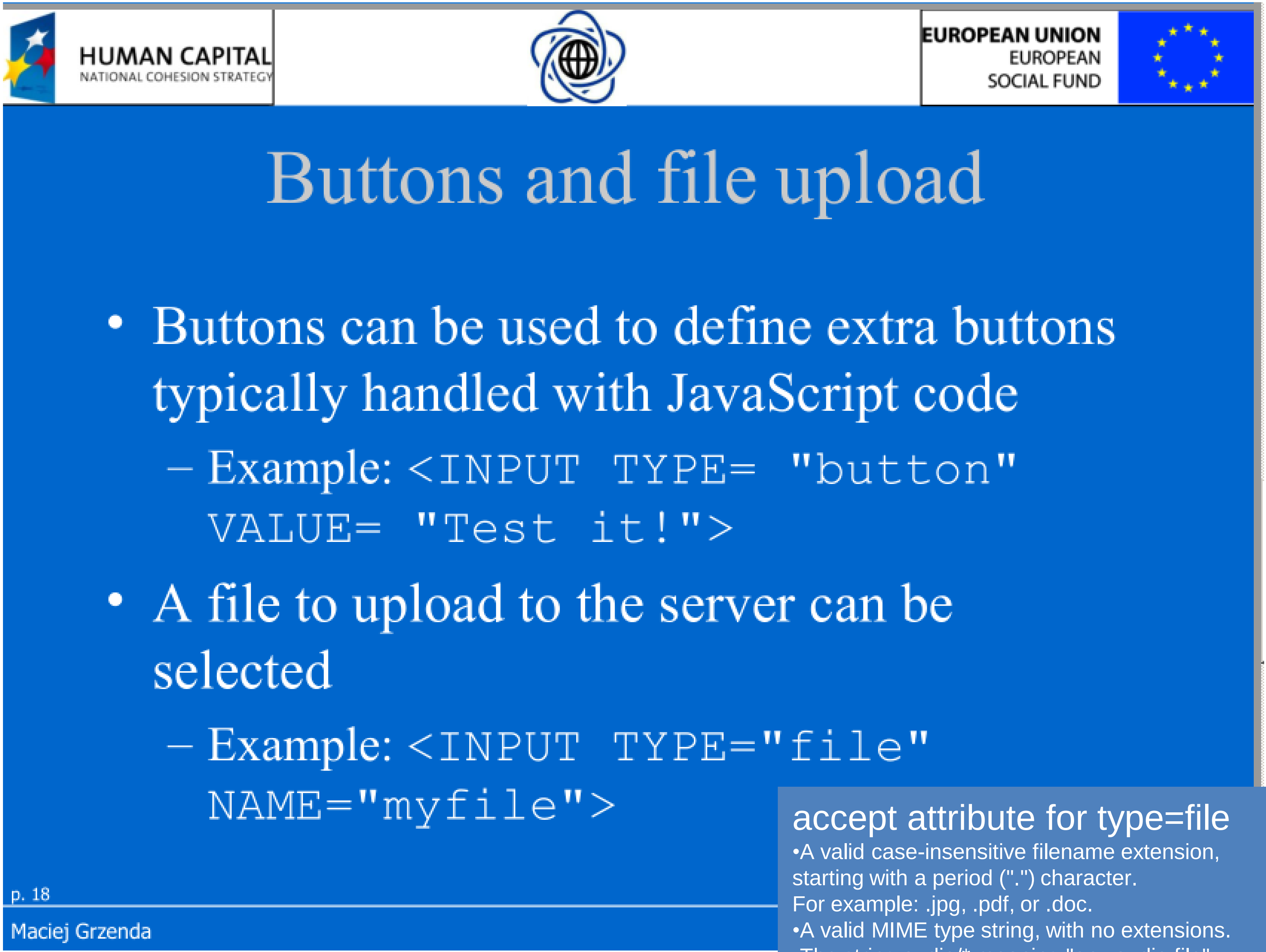
- Option groups are defined through the use of **RADIO** field. In order to define a group of options only one of which can be selected at a moment, the NAME attribute has to be set to the same value for every single group item.
- Consequently, several different option groups can exist in one form, these have to use different names however.

- Example:

```
<BR> <INPUT TYPE= "RADIO" NAME= "Type" VALUE= "1" > Less than 300 cm
<BR> <INPUT TYPE= "RADIO" NAME= "Type" VALUE= "2” CHECKED> 300 cm - 400 cm
<BR> <INPUT TYPE= "RADIO" NAME= "Type" VALUE= "3" > More than 400 cm
```

- In addition, the CHECKED attribute declares the initially chosen option.

```
<form action="/action_page.php">
      <label>Yes            <input type="radio"  name="answer" value="yes“> </label>
      <label>No             <input type="radio"  name="answer" value="no"> </label>
      <label>Whatever <input type="radio“  name="answer" value="whatever"> </label>
      <input type="submit">
   </form>
```



HUMAN CAPITAL
NATIONAL COHESION STRATEGY

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

Buttons and file upload

- Buttons can be used to define extra buttons typically handled with JavaScript code
  - Example: `<INPUT TYPE= "button" VALUE= "Test it!">`
- A file to upload to the server can be selected
  - Example: `<INPUT TYPE="file" NAME="myfile">`

accept attribute for type=file

•A valid case-insensitive filename extension, starting with a period (".") character.
For example: .jpg, .pdf, or .doc.
•A valid MIME type string, with no extensions.
•The string audio/* meaning "any audio file".
•The string video/* meaning "any video file".
•The string image/* meaning "any image file"

Server configuration may be needed to upload file

Other items

- Hidden fields
  - Do not have any visual representation
  - Allow to send extra data to the server
  - Example:
    - <INPUT TYPE= "HIDDEN" VALUE= "Reg_Form" NAME= "extra" />
- Password fields:
  - No real security of the data in pure HTTP!
  - Example:
    - <INPUT TYPE= "PASSWORD" NAME= "password" />
- Submit and reset fields
  - Serve to send the data to the server and reset the items to their default values (not necessary empty)
  - Example:

```
<INPUT TYPE="Submit" VALUE="Send">
   <!--- instead of default Submit use Send --!>
<INPUT TYPE="Reset" VALUE="Clear">
   <!--- instead of default Reset use Clear --!>
```

HTML forms – ENCTYPE attribute

- Not only www servers define content-type of documents sent, but also HTML forms.
- ENCTYPE attribute of a form defines MIME type used to encode form content
- Because of encoding applied by the browser, the server-side script must parse the received content to retrieve the values of input items.

application/x-www-form-urlencoded

- Default content type for HTML forms. The following rules are applied by the browser whilst submitting form content:
  - Control names and values are escaped. Space characters are replaced by `+', and then reserved characters are escaped.
  - Non-alphanumeric characters are replaced by `%HH', a percent sign and two hexadecimal digits representing the ASCII code of the character.
  - Line breaks are represented as "CR LF" pairs (i.e., `%0D%0A').
  - The control names/values are listed in the order they appear in the document. The name is separated from the value by `=' and name/value pairs are separated from each other by `&'.
- An example of this encoding:

```
http://127.0.0.1/cgi?Name=&Type=3&Code=&Extra=R
eg_Form&month=April&Description=Please+specify+
%0D%0Ayour+requirements%0D%0A
```

multipart/form-data

- The content type "application/x-www-form-urlencoded" is inefficient for sending large quantities of binary data or text containing non-ASCII characters.
- The content type "multipart/form-data" should be used for submitting forms that contain files, non-ASCII data, and binary data.
- The content "multipart/form-data" follows the rules of all multipart MIME data streams
- Complicated parser is required to retrieve information received in this encoding

Sample form with multipart/form-data

```
<FORM action="http://server.com/cgi/handle"
       enctype="multipart/form-data"
       method="post">
   <P>
   What is your name? <INPUT type="text"
  name="submit-name"><BR>

  What files are you sending? <INPUT type="file"
  name="files"><BR>

  <INPUT type="submit" value="Send"> <INPUT
  type="reset">
</FORM>
```

<INPUT TYPE=”file” ..> can be used to upload a file from local file system to the server

Multipart/form-data - sample content

```
Content-Type: multipart/form-data; boundary=AaB03x
   --AaB03x
   Content-Disposition: form-data; name="submit-name"
   John
   --AaB03x
   Content-Disposition: form-data; name="files"
   Content-Type: multipart/mixed; boundary=BbC04y
   --BbC04y
   Content-Disposition: file; filename="file1.txt"
   Content-Type: text/plain
   ... contents of file1.txt ...
   --BbC04y
   Content-Disposition: file; filename="file2.gif"
   Content-Type: image/gif
   Content-Transfer-Encoding: binary
   ...contents of file2.gif...
   --BbC04y--
   --AaB03x--
```

Due to significant complication of data stream in multipart/form-data, third-party libraries are used to retrieve actual data sent in it.

Forms – content-types

ENCTYPE	Discussion
application/x-www-form-urlencoded	•Default – to be used in majority of cases. •Both GET and POST methods can be used. •Lower processing overhead.
multipart/form-data	Used for uploading files with POST method only.

HTML forms – METHOD attribute

METHOD	Discussion
GET	• Default – used with application/x-www-form-urlencoded • Form content is sent together with URL. Thus, limitations concerning the number of bytes sent take place • Allows to use script address together with HTML form field settings in hypertext links. Example: <A HREF=" http://127.0.0.1/cgi?Name=&Type=3&Code=&Extra=Reg_Form&month=April&Description=Please+specify...">...</A>
POST	• Used with both ENCTYPE settings. • No limits regarding number of bytes sent • Data not sent with URL, thus not displayed in address field.

Commandline

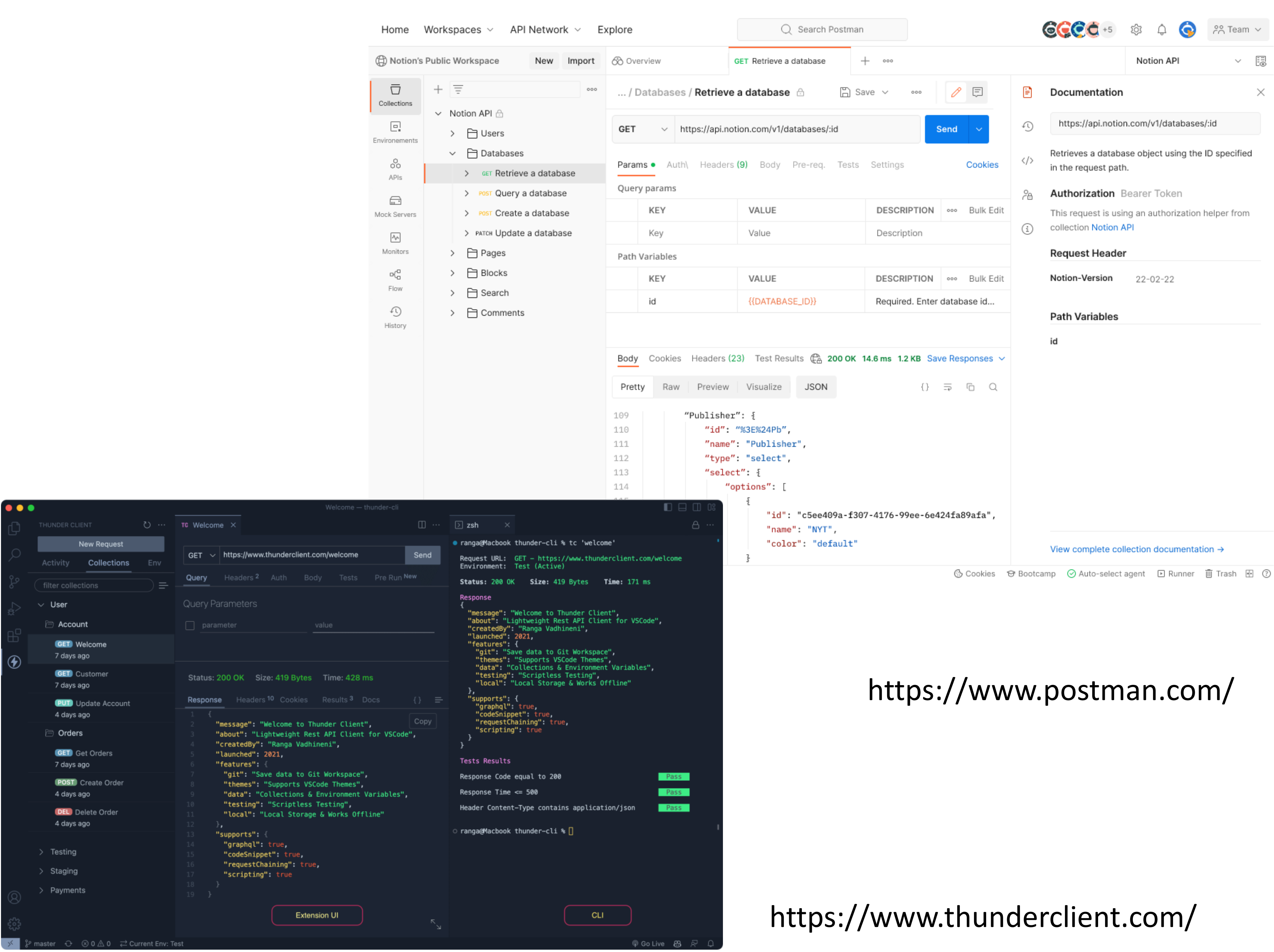
-
  ```
  curl -X GET 'http://localhost/?fname=3&ble=pppp‘
  ```

-
  ```
  curl --verbose --data 'fname=9;ppp=2' --request "POST" http://localhost/test.php
  ```

-
  ```
  curl -X " PUT " URL \
  --header 'Content-Type: multipart/form-data; boundary=---------BOUNDARY' \
   --data-binary @file
  ```

-
  ```
  curl -b AAA=1234 -v http://localhost          //send cookie
  ```

https://curl.haxx.se/windows/
https://github.com/httpie/httpie



GET https://api.notion.com/v1/databases/:id

Query params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Path Variables

KEY	VALUE	DESCRIPTION
id	{{DATABASE_ID}}	Required. Enter database id...

200 OK 14.6 ms 1.2 KB

```
"Publisher": {
    "id": "%3E%24Pb",
    "name": "Publisher",
    "type": "select",
    "select": {
        "options": [
            {
                "id": "c5ee409a-f307-4176-99ee-6e424fa89afa",
                "name": "NYT",
                "color": "default"
            }
```

Documentation

https://api.notion.com/v1/databases/:id

Retrieves a database object using the ID specified in the request path.

Authorization Bearer Token

This request is using an authorization helper from collection Notion API

Request Header

Notion-Version 22-02-22

Path Variables

id

View complete collection documentation →

GET https://www.thunderclient.com/welcome

Status: 200 OK Size: 419 Bytes Time: 428 ms

```
{
  "message": "Welcome to Thunder Client",
  "about": "Lightweight Rest API Client for VSCode",
  "createdBy": "Ranga Vadhineni",
  "launched": 2021,
  "features": {
    "git": "Save data to Git Workspace",
    "themes": "Supports VSCode Themes",
    "data": "Collections & Environment Variables",
    "testing": "Scriptless Testing",
    "local": "Local Storage & Works Offline"
  },
  "supports": {
    "graphql": true,
    "codeSnippet": true,
    "requestChaining": true,
    "scripting": true
  }
}
```

```
ranga@Macbook thunder-cli % tc 'welcome'

Request URL:  GET - https://www.thunderclient.com/welcome
Environment:  Test (Active)

Status: 200 OK    Size: 419 Bytes    Time: 171 ms

Response
{
  "message": "Welcome to Thunder Client",
  "about": "Lightweight Rest API Client for VSCode",
  "createdBy": "Ranga Vadhineni",
  "launched": 2021,
  "features": {
    "git": "Save data to Git Workspace",
    "themes": "Supports VSCode Themes",
    "data": "Collections & Environment Variables",
    "testing": "Scriptless Testing",
    "local": "Local Storage & Works Offline"
  },
  "supports": {
    "graphql": true,
    "codeSnippet": true,
    "requestChaining": true,
    "scripting": true
  }
}

Tests Results

Response Code equal to 200                        Pass

Response Time <= 500                              Pass

Header Content-Type contains application/json     Pass

ranga@Macbook thunder-cli %
```

https://www.postman.com/

https://www.thunderclient.com/

data-* attributes

- Used to store custom data private to the page or application
  - The attribute name should not contain any uppercase letters, and must be at least one character long after the prefix "data-"
  - The attribute value can be any string
  - engaging user experience (without any Ajax calls or server-side database queries).
  - Custom attributes prefixed with "data-" will be completely ignored by the user agent.
  - Accessed : tag.dataset property

data-* attributes

- HTML : <span id="hi" data-test="a value 1" data-test-two="sth important">hi</span>
- JS console :
  const spanhi = document.querySelector("#hi")
  console.log(spanhi.dataset)
- DOMStringMap { test → "a value 1", testTwo → "sth important" }
  test: "a value 1"
  testTwo: "sth important"

The -attribute-name to camelCase

Form Validation best practices

- Always have server side validation, since malicious actors can bypass client side validation.
- Provide detailed error messages in-context with the field that produced the error.
- Provide an example of what the data should look like in case of an error message, such as - "Email did not match format - test@example.com"
- Avoid using single error pages that involve redirection. This is bad user experience and forces the user to go back to a previous page to fix the form and lose context.
- Always mark required fields.

https://css-tricks.com/almanac/selectors/r/required/

Other <input> HTML5

<input type="button"> <input type="button" onclick="alert('Hello World!')" value="Click Me!">

<input type="checkbox">

<input type="color">

<input type="date"> <input type="date" id="datemax" name="datemax" max="1979-12-31">

<input type="datetime-local">

<input type="email"> Depending on browser support, the e-mail address can be automatically validated when submitted.

<input type="file">

<input type="hidden"> hidden but passed , ie. A currently processed record

<input type="image"> defines an image as a submit button.

<input type="month">

<input type="number"> <input type="number" id="quantity" name="quantity" min="1" max="5">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset"> reset button that will reset all form values to their default values:

<input type="search">

<input type="submit">

<input type="tel"> <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">

<input type="text"> default

<input type="time"> Depending on browser support, a time picker can show up in the input field.

<input type="url">

<input type="week"> only Chrome

Input Attributes

value attribute specifies an initial value for an input field

readonly attribute specifies that an input field is read-only

disabled attribute specifies that an input field should be disabled.

size attribute specifies the visible width, in characters, of an input field The default value for size is 20. : text, search, tel, url, email, password.

maxlength attribute specifies the maximum number of characters allowed in an input field. if you want to alert the user, you must write JavaScript code

placeholder attribute specifies short a hint that describes the expected value of an input field

height and width attributes specify the height and width of an <input type="image"> element.

autocomplete attribute specifies whether a form or an input field should have autocomplete on or off. text, search, url, tel, email, password, datepickers, range, and color.

pattern Specifies a regular expression to check the input value against

readonly Specifies that an input field is read only (cannot be changed)

required Specifies that an input field is required (must be filled out) field must be filled out before submitting the form.

step attribute specifies the legal number intervals for an input field. number, date

min and max attributes specify the minimum and maximum values for an input field.: number, range, date, datetime-local, month, time and week

<input type="date" id="datemin" name="datemin" min="2000-01-02"> <input type="number" id="quantity" name="quantity" min="1" max="5">

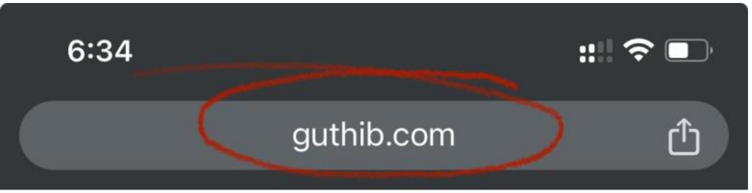
- Power effective
- Eco effective

- Opera Mini introduced a cloud (proxy) browser because of mobile bandwidth restrictions in 2005.

  15 years later, @MightyApp is taking that idea to the desktop because of the heavy CPU and memory usage of today's web apps. https://mightyapp.com
  Maximiliano Firtman @firt

Newbie Programmers
AI & ML
ALGORITHMS
DATA STRUCTURES
OOP
HELLO WORLD

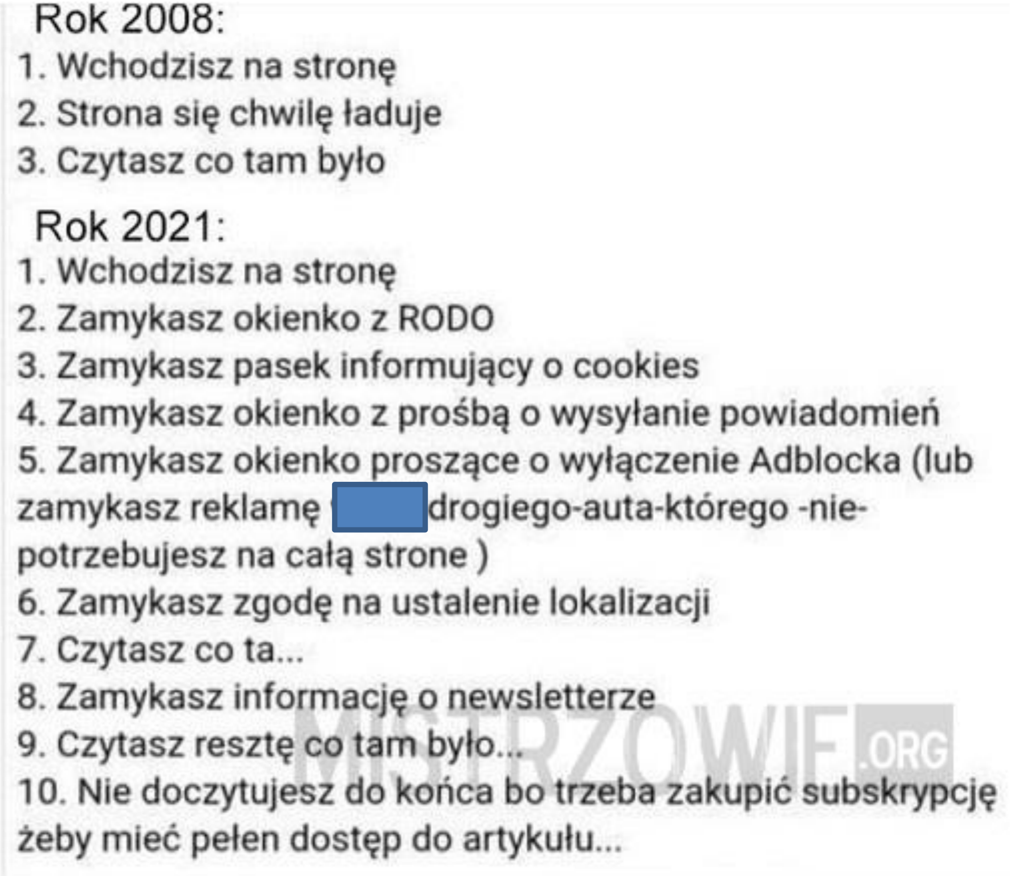
New Web Devs
React
Git & GitHub
JavaScript
HTML/CSS
imgflip.com
@kvncnls



You spelled it wrong.

Gdy przychodzi wojna i dostajesz powołanie na front

a nigdy nie pisałeś w JS'ie

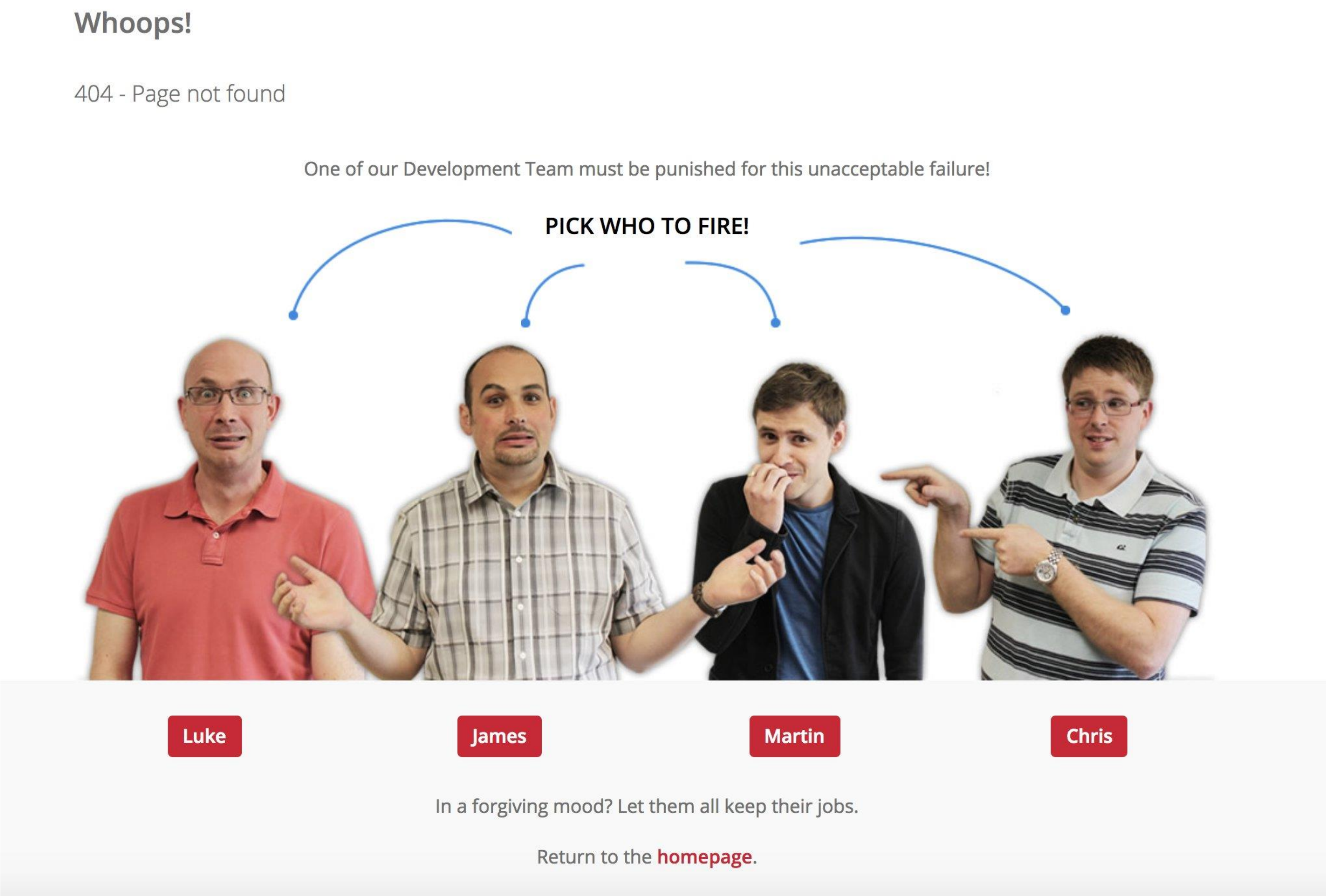


Rok 2008:

1. Wchodzisz na stronę
2. Strona się chwilę ładuje
3. Czytasz co tam było

Rok 2021:

1. Wchodzisz na stronę
2. Zamykasz okienko z RODO
3. Zamykasz pasek informujący o cookies
4. Zamykasz okienko z prośbą o wysyłanie powiadomień
5. Zamykasz okienko proszące o wyłączenie Adblocka (lub zamykasz reklamę [redacted] drogiego-auta-którego -nie-potrzebujesz na całą strone)
6. Zamykasz zgodę na ustalenie lokalizacji
7. Czytasz co ta...
8. Zamykasz informację o newsletterze
9. Czytasz resztę co tam było...
10. Nie doczytujesz do końca bo trzeba zakupić subskrypcję żeby mieć pełen dostęp do artykułu...



Whoops!

404 - Page not found

One of our Development Team must be punished for this unacceptable failure!

PICK WHO TO FIRE!

In a forgiving mood? Let them all keep their jobs.

Return to the homepage.

- François Chollet @fchollet
- In the first few years, the web meant freedom. But the rise of ubiquitous social media made it more of a prison.
- 5:51 AM · May 29, 2024

- thoughts on AI @thoughts_on_AI
- In the beginning, every web user was an educated person. Now everyone uses the web. And with attention on the web being a democracy, what we have is mob rule electing charlatans to lead us.