

Programowanie Aplikacji Internetowych

Podstawy

Plan na dziś

- HTML
- CSS
- DNS
- HTTP
- REST

HTML

- Co warto wiedzieć?

HTML

Skorzystajmy Najpierw z Web Developer Tools w Firefoxie



HTML

- Tim Berners-Lee
- HTML5
- [podstawowa struktura](#)

HTML

- tagi otwierające i zamykające
- atrybuty
- `
`

HTML

- `link do `
- ``
- `<p>123</p>`
- `<h2></h2>`

HTML

Listy:

- `` oraz ``
- ``

HTML

Kontenery:

- `<div>` - kontener
- `` - inline

HTML

- `<script>`
- [forms](#)

HTML - Praktyka

- [Responsive design](#),
- [Responsive typography](#),
- [media queries](#).

HTML - Praktyka

Jeśli nie mamy możliwości budowy frontendu:

- prezentacje z [revealjs](#),
- blog na wordpress lub na alternatywnej platformie,
- Warto skorzystać z gotowych komponentów, np., [bootstrap](#).

CSS

- CSS - Cascading Style Sheets
- [przykład](#)

CSS

Zacznij od korzystania z gotowych komponentów:

1. [bootstrap](#)
2. ... ([more](#))
3. [materializecss](#)
4. [ant.design](#)
5. [tailwindcss](#)

DNS

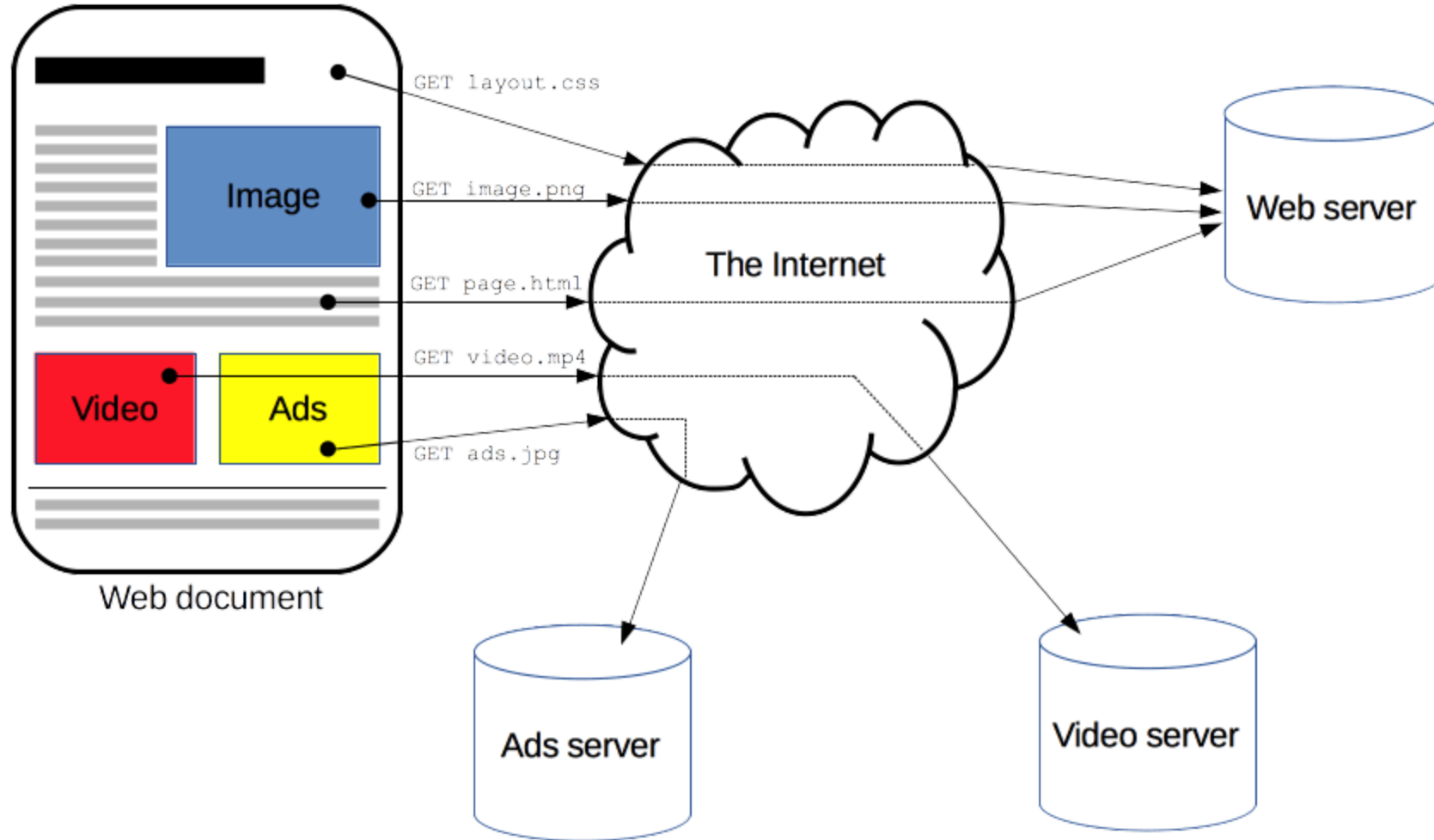
DNS - Domain Name Server

```
dig google.com
```

DNS

Sprawdźmy kto posiada domenę
`wsb.pl` na <https://www.dns.pl/whois>

HTTP

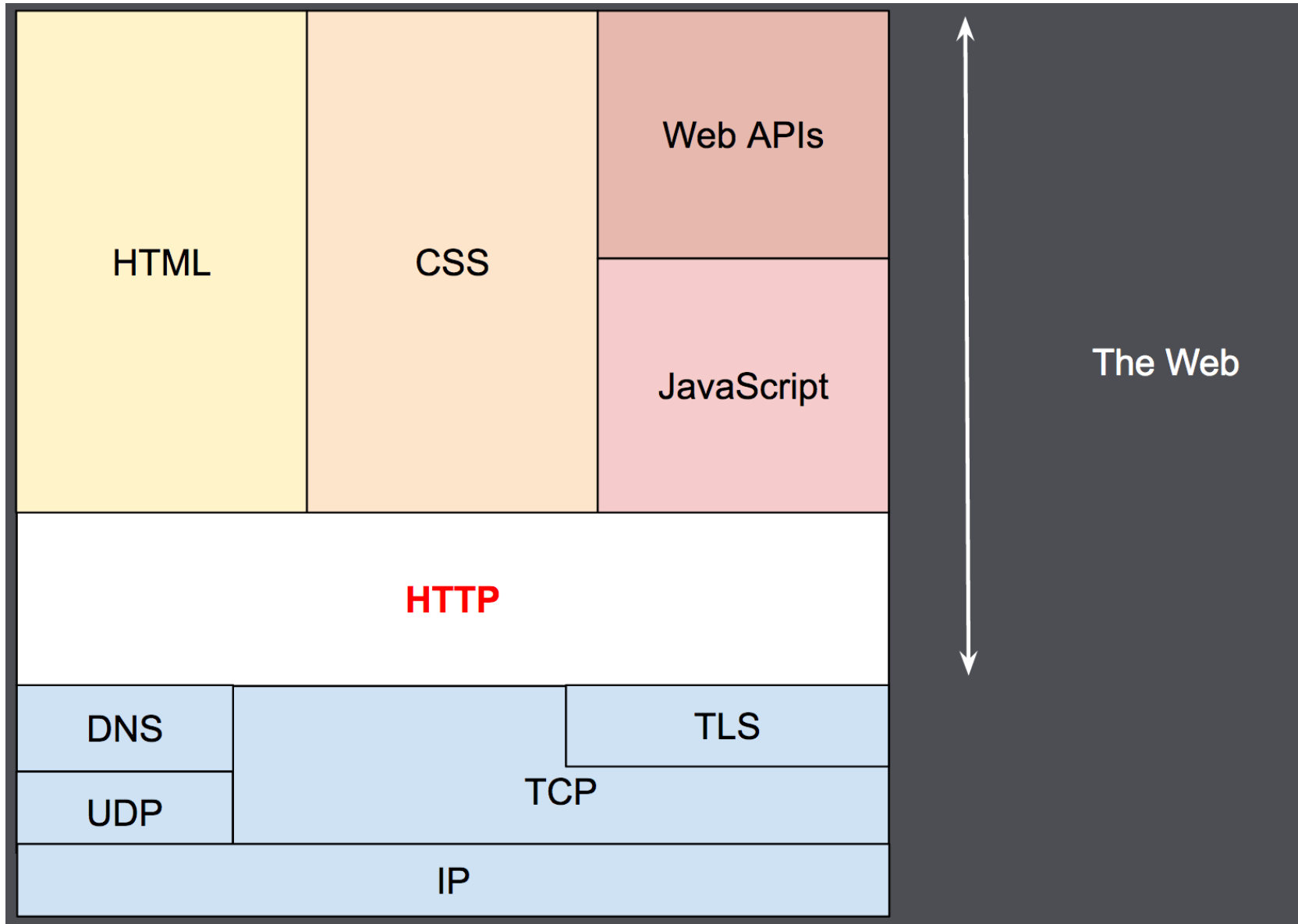


HTTP

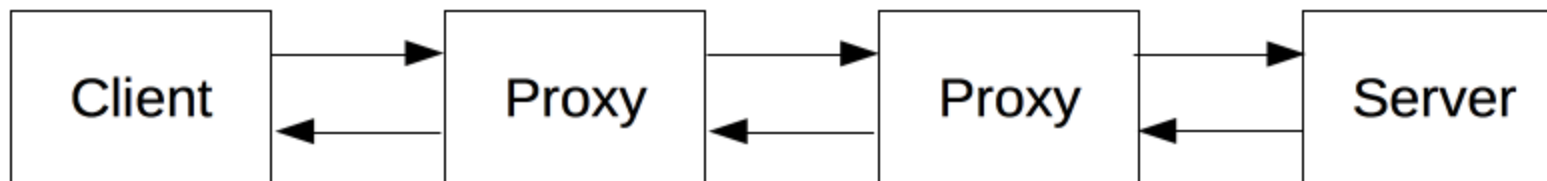
Podstawy:

- HTTP is simple
- HTTP is extensible
- HTTP is stateless (but not sessionless)
- HTTP relies on TCP (connection based)

HTTP



HTTP



HTTP

Demo:

```
curl -I www.google.com
```

```
curl -I -L google.com
```

HTTP - methods

Methods:

- GET
- POST
- DELETE

HTTP - methods

Demo:

```
http POST https://httpbin.org/post "name"="natalia"
```

HTTP - status code

Status code:

- 5xx: 500, 502
- 4xx: 404, 400, 401
- 3xx: 301, 302
- 2xx: 200, 201, 02

HTTP - status code

Demo:

```
curl -I -X GET https://httpbin.org/status/404 --fail
```

```
curl -I -X GET https://httpbin.org/status/200 --fail
```

REST

Dziękuję za uwagę