Technologie Webowe

Warstwa prezentacyjna

Tomek Sułkowski

Plan na dziś

- Ogólny zarys technologii webowych
- Warstwy front-endu
- Narzędzia i techniki

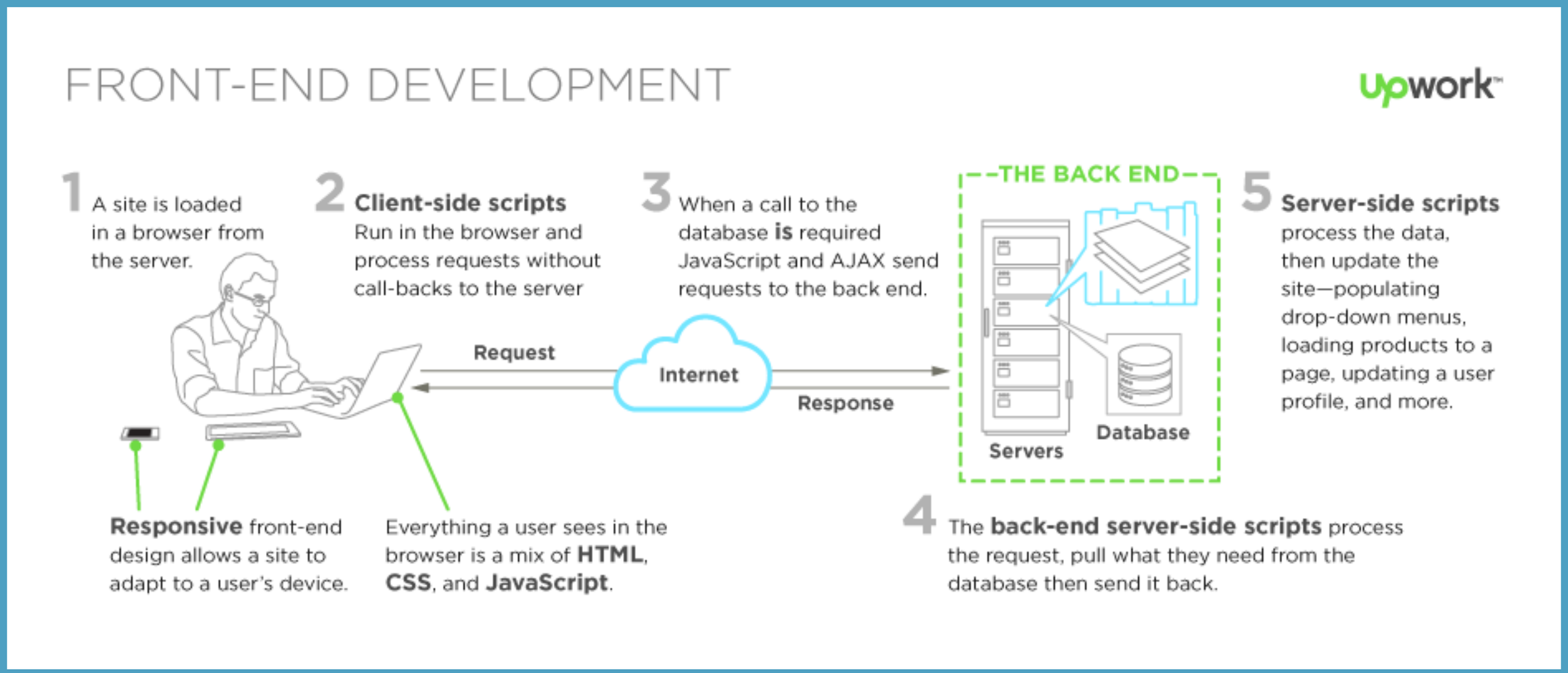
poznajmy się

- Tomek Sułkowski
- Front-end Engineer @ StackBlitz / US
- Lead Trainer @ Sages / Warszawa

- 13 lat pracy z WWW
- 7 lat 100% z front-endem

WWW: ogólnie

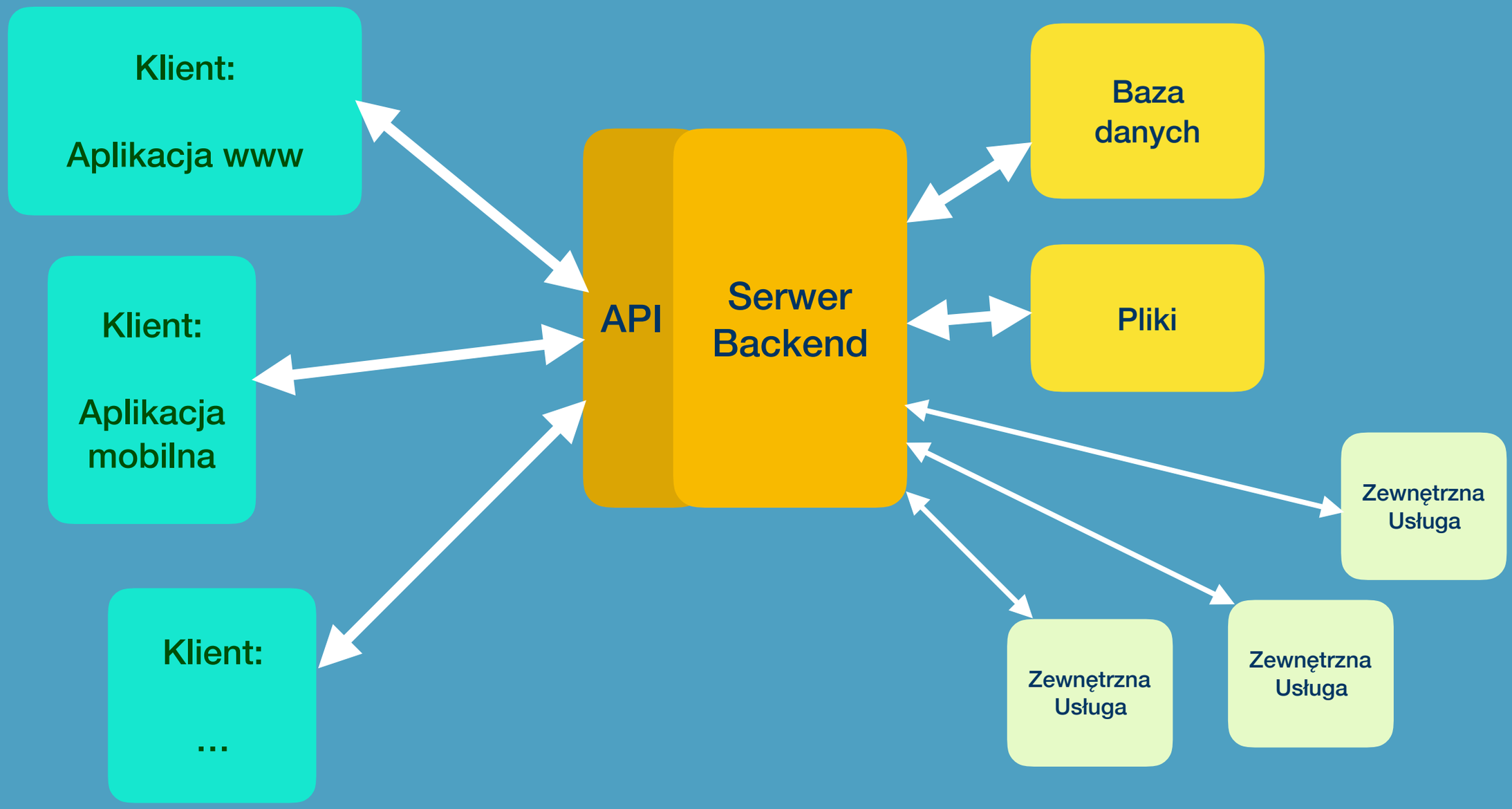
Co jest czym w internecie?



Frontend vs Backend

WWW: ogólnie

Typowa architektura



Backend?

- Typowe języki:
  Java, PHP, .Net, Ruby, Python

- Również bazy danych, DevOps

- Niezależny od front-endu
  Front-end może nie wiedzieć o technologii back-endu

- 2 typowe sposoby komunikacji:
  - “REST API” - frontend pyta->backend odpowiada
  - Socket connection - połączenie w czasie rzeczywistym

Frontend?

- Typowe języki:
  JavaScript, CSS, HTML, TypeScript, Sass

- Niezależny od back-endu
  Back-end może nie wiedzieć o technologii front-endu

- 2 typowe sposoby komunikacji:
  - “REST API” - frontend pyta->backend odpowiada
  - Socket connection - połączenie w czasie rzeczywistym

Frontend developer?

jak nas zwą

- ~~Webmaster~~
- Front-end Developer
- Front-end Engineer
- CSS/HTML Developer
- Front-End Web Designer
- UI Developer
- Front-end Accessibility Expert
- QA

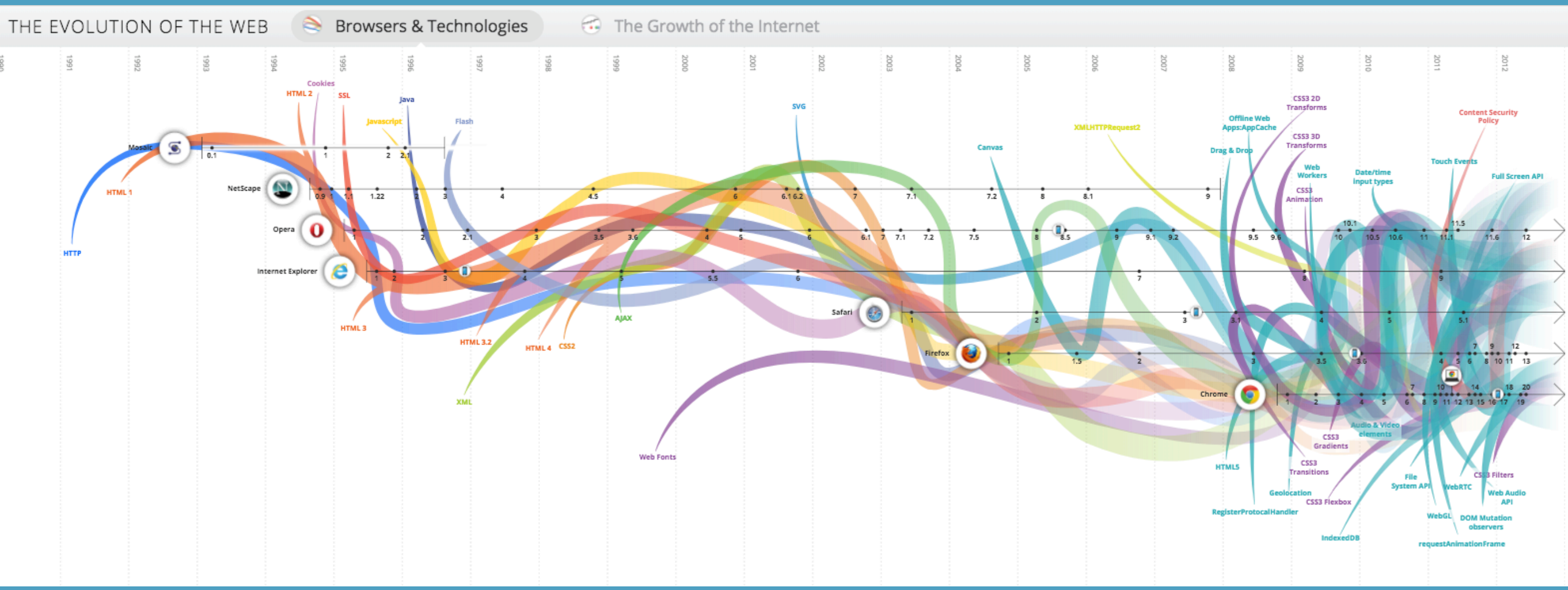
Uwaga!

JAVA *is to* JAVASCRIPT *as* HAM *is to* HAMSTER

Przeglądarka

pierwszy front

“nie wszyscy na raz”



http://www.evolutionoftheweb.com

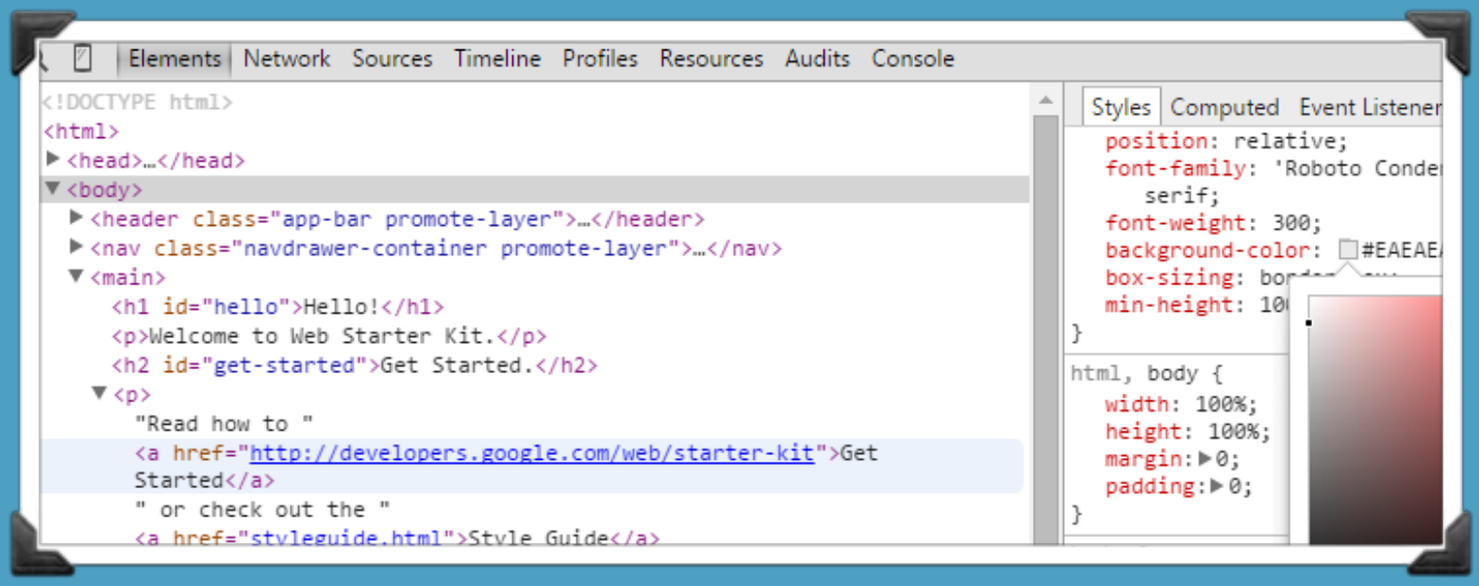
Przeglądarka

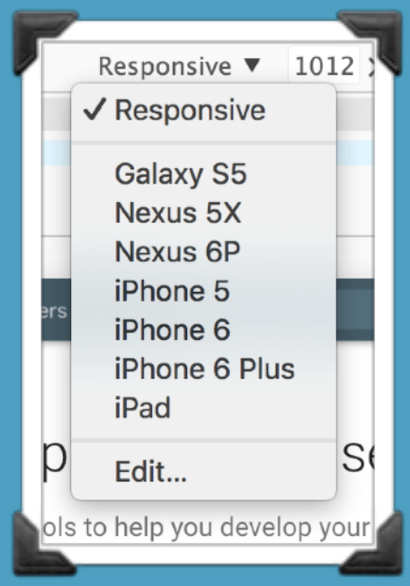
pierwszy front

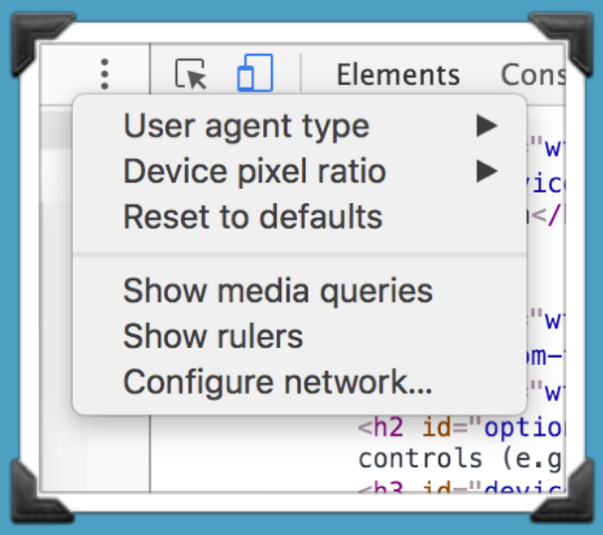
- Ale nie tylko 
  - Mniej popularne (?) silniki
    - Czytniki
    - “Headless browsers”

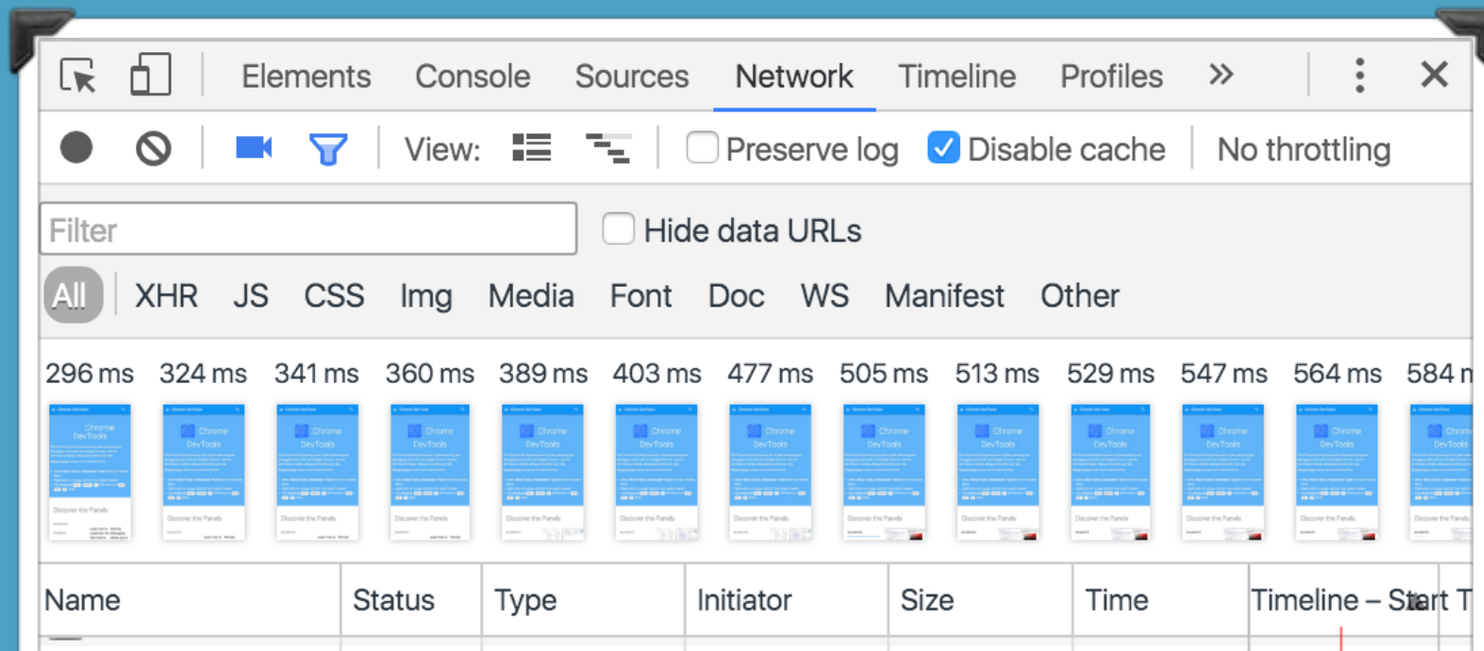
- Statystyki: http://gs.statcounter.com
  - Ważniejsze statystyki: **własne!**

Przeglądarka
potężne narzędzie developerskie









Warstwy Front-endu

Warstwy Front-endu

HTML

- HyperText Markup Language
- Struktura
- Znaczenie / Semantyka
- HTML5

Warstwy Front-endu

CSS

- Cascading Style Sheets
- Wygląd
- Statyczne i dynamiczne (animacje)
- CSS3 → 4

Warstwy Front-endu

JavaScript

- Zachowanie

Warstwy Front-endu

inne kluczowe akronimy

- DOM
  Document Object Model

- Web APIs / Browser APIs/ HTML5 APIs

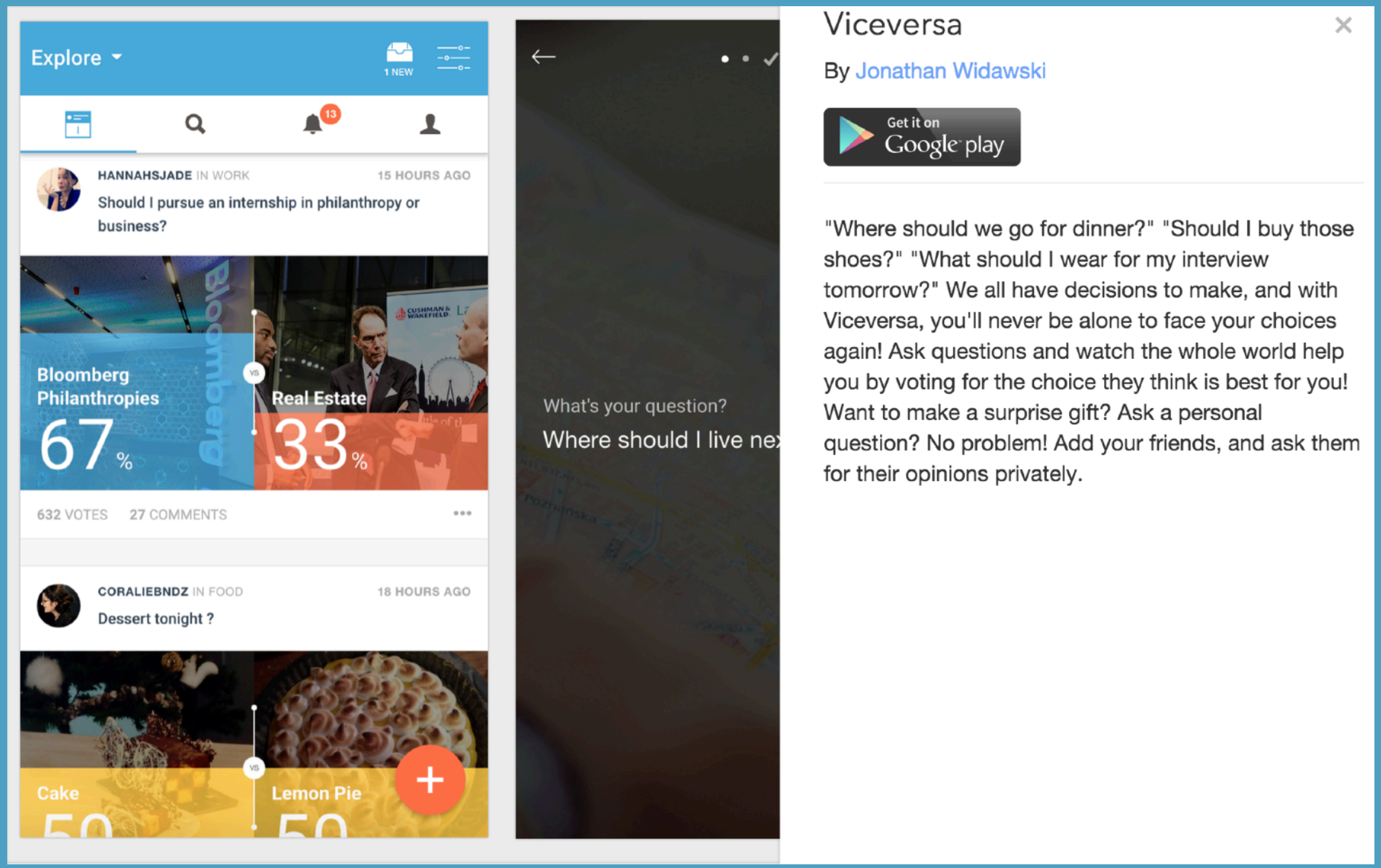
- JSON
  JavaScript Object Notation

- WCAG
  Web Content Accessibility Guidelines

- WAI-ARIA
  Web Accessibility Initiative: Accessible Rich Internet Applications

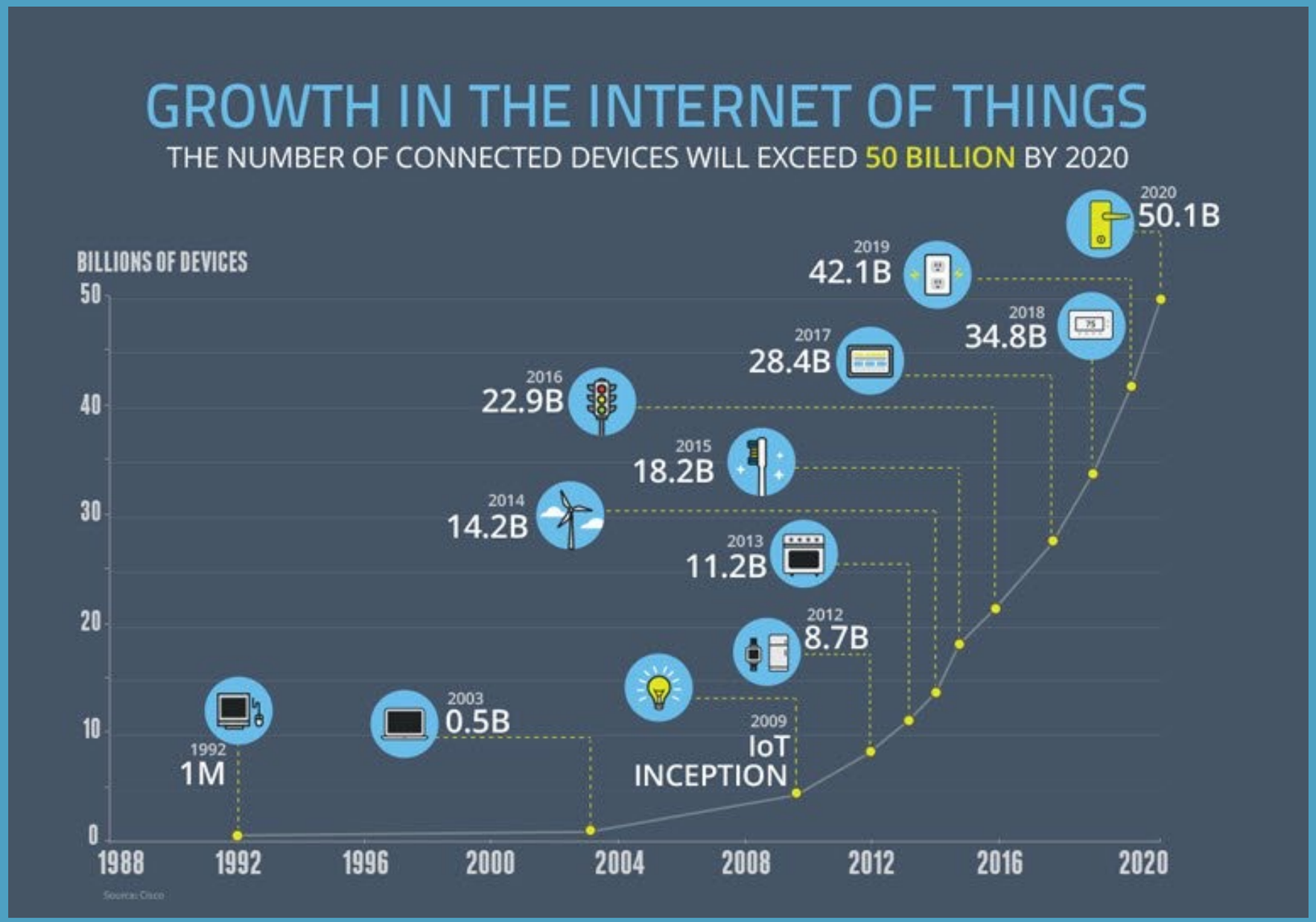
Czy tylko front-end?

Desktop & Mobile



Czy tylko front-end?

Internet of Things



HTML

Podstawa

HTML

Podstawowy dokument

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello world!</title>
    <meta name="description"
          content="My first basic HTML document">
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

HTML

Elementy języka

```
<!-- komentarz -->

<p>       <!-- znacznik otwierający -->
  If Tetris has taught me anything
  it's that errors pile up and accomplishments disappear.
</p>      <!-- znacznik zamykający -->

<br>      <!-- znacznik samozamykający się -->
<br/>

<!-- atrybuty -->
<img src="https://placehold.it/50x50" alt="Description" />
```

HTML

2 rodzaje elementów

Blokowe:

-
- <p>
- ,
- <h1>, ... , <h6>
- <blockquote>
- <form>
- <article>, <section>, ...

Inline:

-
- <a>
- , <i>, , <b>
- <q>
- <abbr>
- <img>
- <input>, ...

HTML

Tabele

```
<table>
    <thead>                 <!-- nagłówek -->
        <tr>                <!-- wiersz -->
            <th>ID</th>     <!-- komórka nagłówka -->
            <th>Name</th>
        </tr>
    </thead>
    <tbody>                 <!-- część główna -->
        <tr>                <!-- nagłówek -->
            <td>1</td>      <!-- komórka -->
            <td>John</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Anne</td>
        </tr>
    </tbody>
    <tfoot>                 <!-- stopka -->
        <tr>
            <th colspan="2">
                2 people
            </th>
        </tr>
    </tfoot>
</table>
```

HTML

Semantyka

```
<div id="main">
  <div class="article">
    <div class="header">
      <h1>Blog post</h1>
      <p>Published: <span>21st Feb, 2015</span></p>
    </div>
    <p>…</p>
  </div>
</div>
```

HTML
Accessibility (a11y)

```
<!-- bad -->
<h1><img alt="Logo" src="logo.png"></h1>

<!-- good -->
<h1><img alt="My Company, Inc." src="logo.png"></h1>
```

- Linki używane do linkowania a przyciski do klikania
- Nie polegamy jedynie na kolorze jako nośniku informacji
- Jasno określamy etykiety w polach formularzy
- … http://romeo.elsevier.com/accessibility_checklist

HTML5 czyli...

- nowa specyfikacja znaczników
  - nowe elementy
  - nowe atrybuty do już istniejących elementów
- nowe specyfikacje interakcji z przeglądarką
  - canvas
  - drag & drop
  - offline storage
- technologie powiązane
  - file API
  - indexedDB
  - geolokacja
  - ...

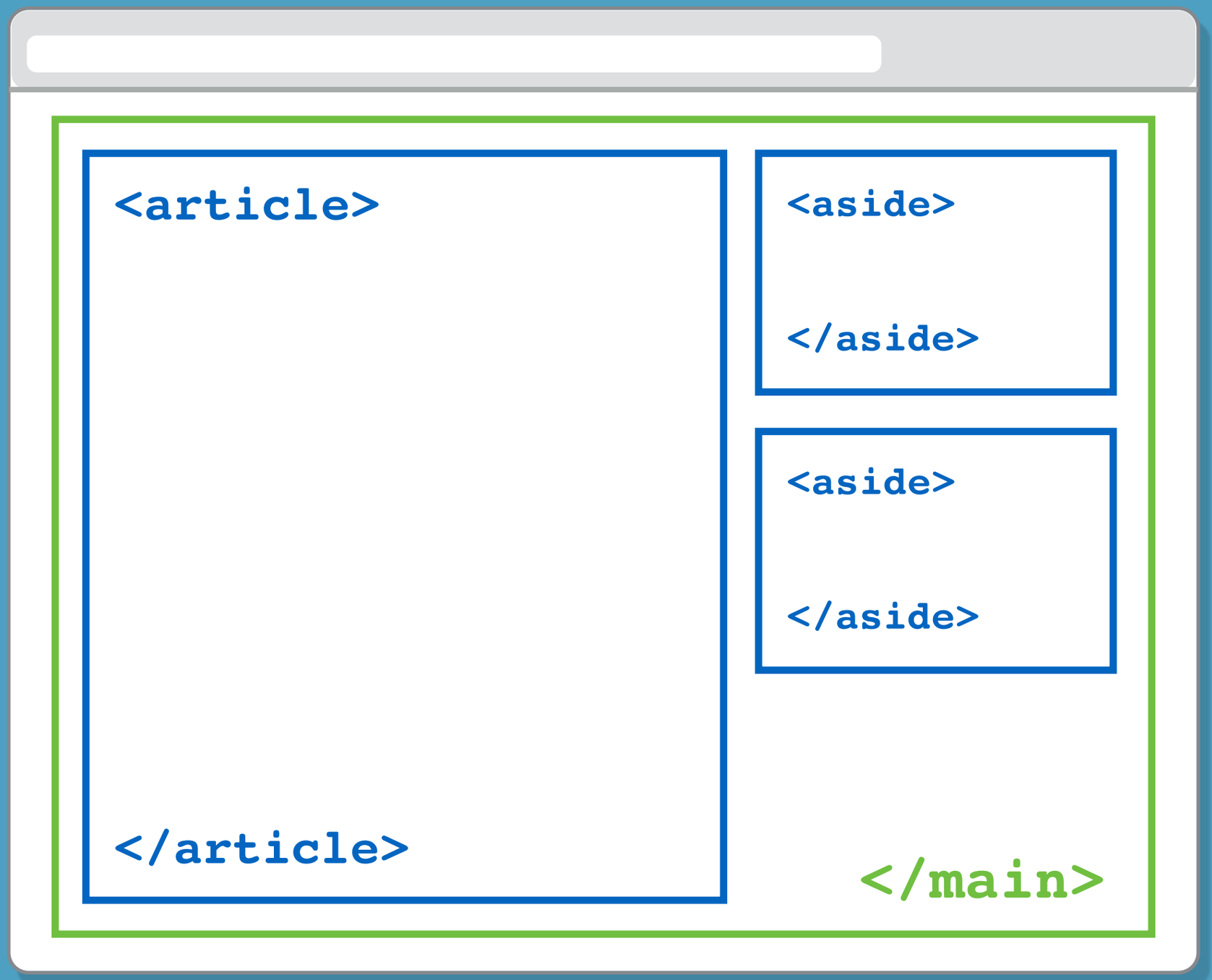
HTML5 czyli...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="content-type"
          content="text/html;charset=utf-8" />
```

```
<!DOCTYPE html>
<html>
  <head>
  <meta charset="utf-8">
```

HTML

<article>, <aside>, <main>



<video> i <audio>

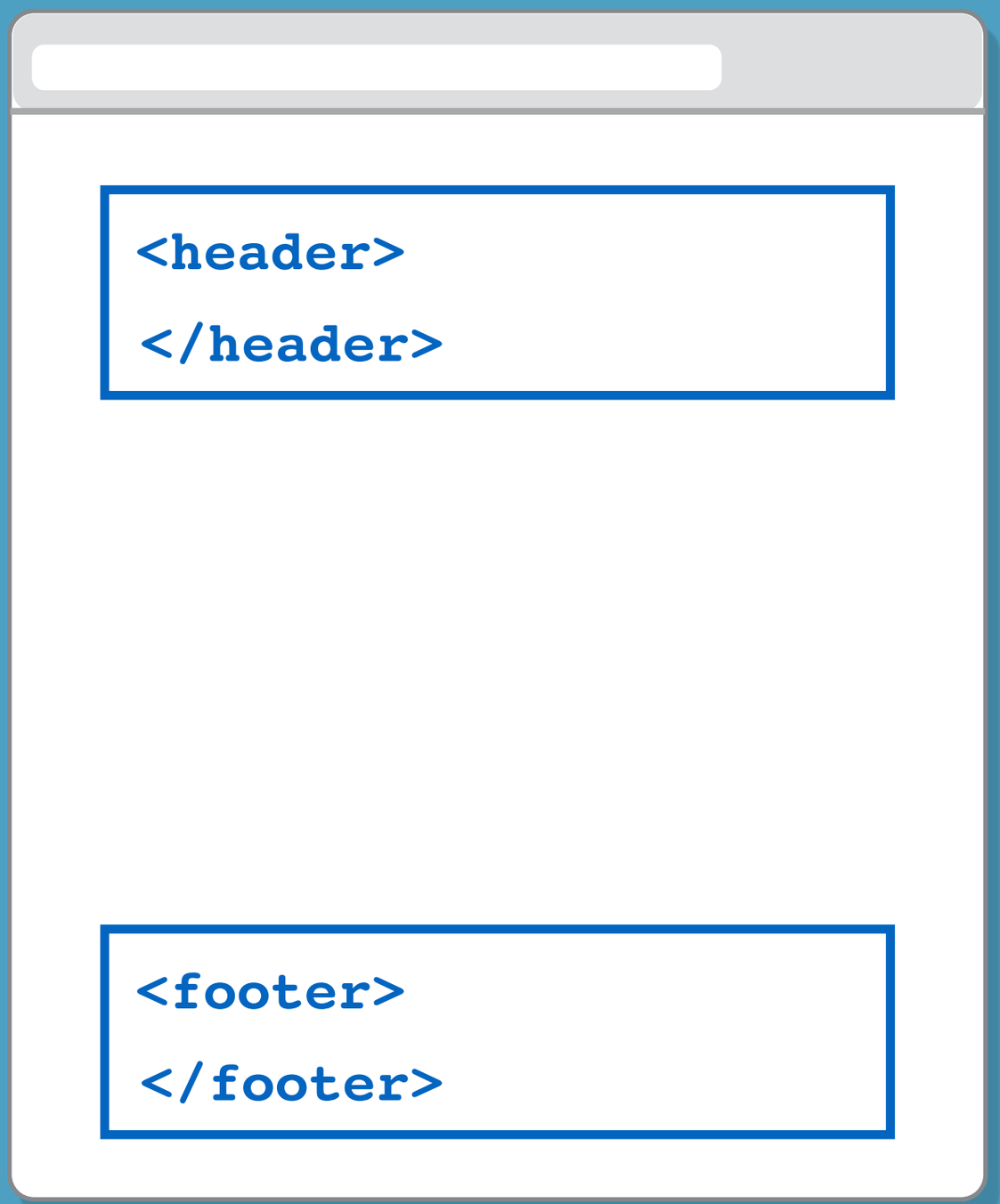
```
<video>
    <source src="video.mp4"
      type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
    </source>
    <source src="video.3gp"
      type='video/3gpp; codecs="mp4v.20.8, samr"'>
    </source>
    <source src="video.ogv"
      type='video/ogg; codecs="theora, vorbis"'>
    </source>
    <track kind="subtitles" label="English Subtitles"
        src="en.vtt" srclang="en">
</video>
```

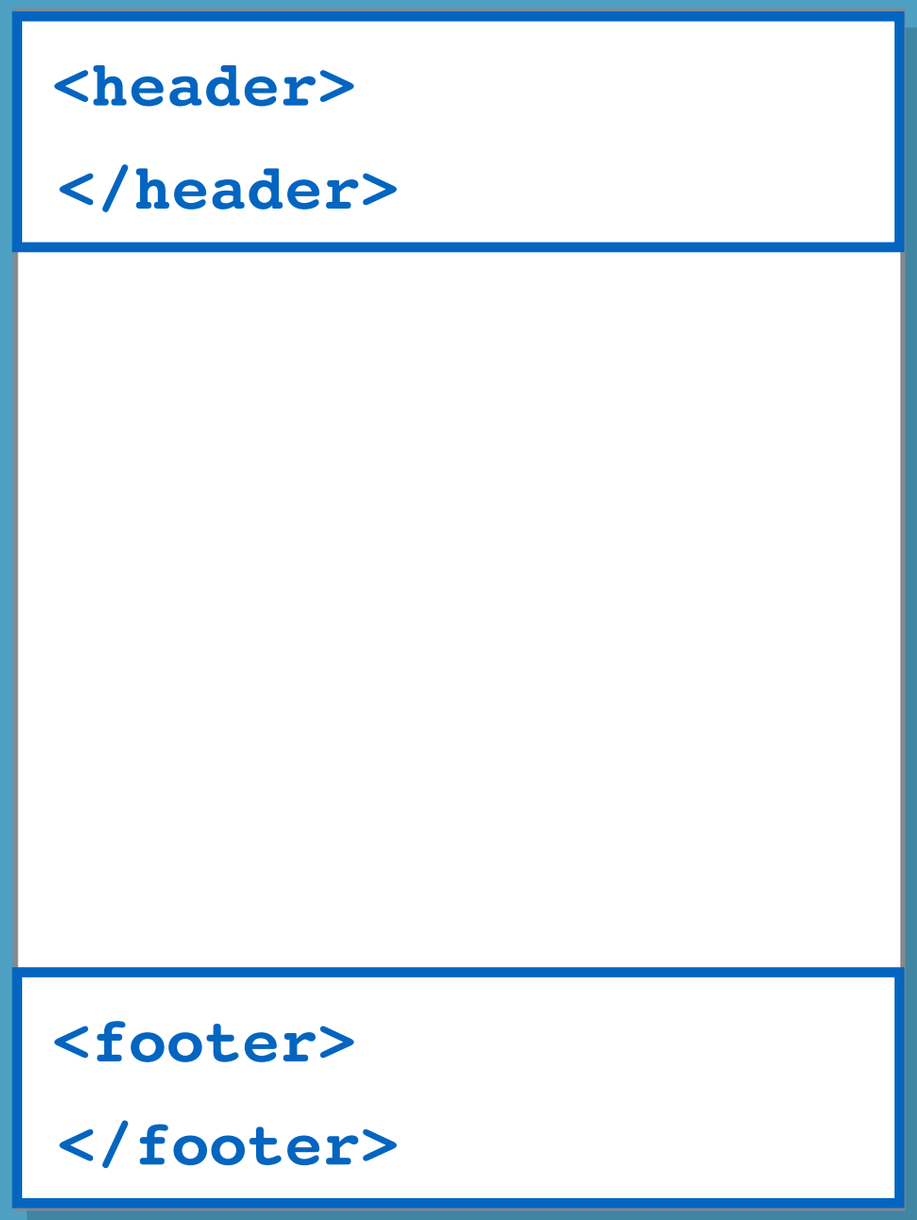
<figure>

A cheeky macaque, Lower Kintaganban River, Borneo. Original by Richard Clark

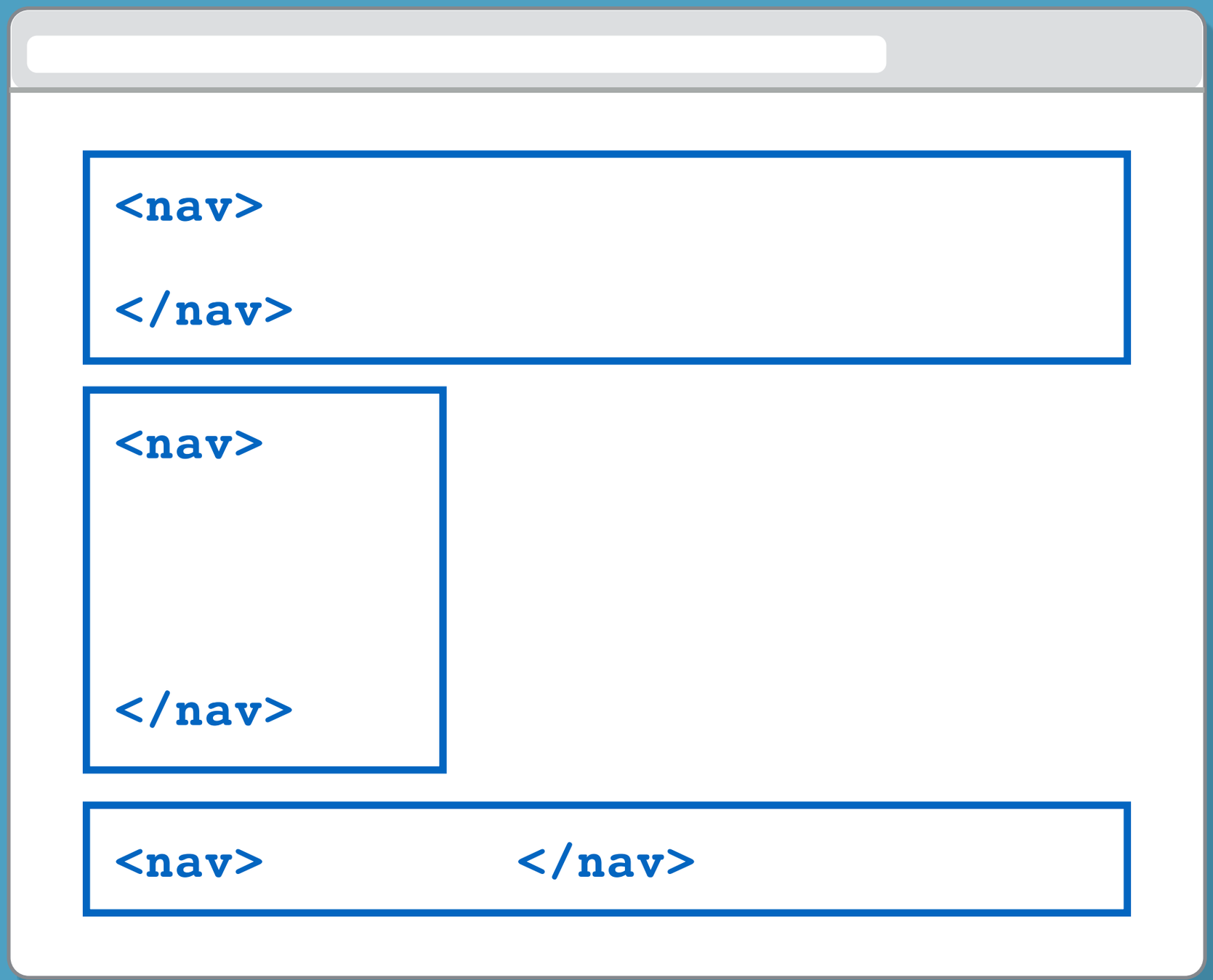
```
<figure>
  <img src="/macaque.jpg" alt="Macaque in the
            trees">
  <figcaption>A cheeky macaque, Lower
  Kintaganban River, Borneo. Original by <a
  href="http://www.flickr.com/photos/rclark/">
  Richard Clark</a></figcaption>
</figure>
```

<header> i <footer>





<nav>



HTML5: przydatne zasoby

- HTML developer guide
  https://developer.mozilla.org/en-US/docs/Web/Guide/HTML

- HTML Doctor
  http://html5doctor.com/

- Dive Into HTML5
  http://diveintohtml5.info/

- HTML5 validator
  http://html5.validator.nu/

Sprawdzanie możliwości przeglądarek

- Can I Use ... ?
  http://caniuse.com
- 
  http://modernizr.com/

Storage API

Zalety względem cookies:

- duże limity (>=5MB)
- nie przesyłane każdorazowo na serwer
- nie trzeba wyświetlać info o ciasteczkach ;)

Wady:

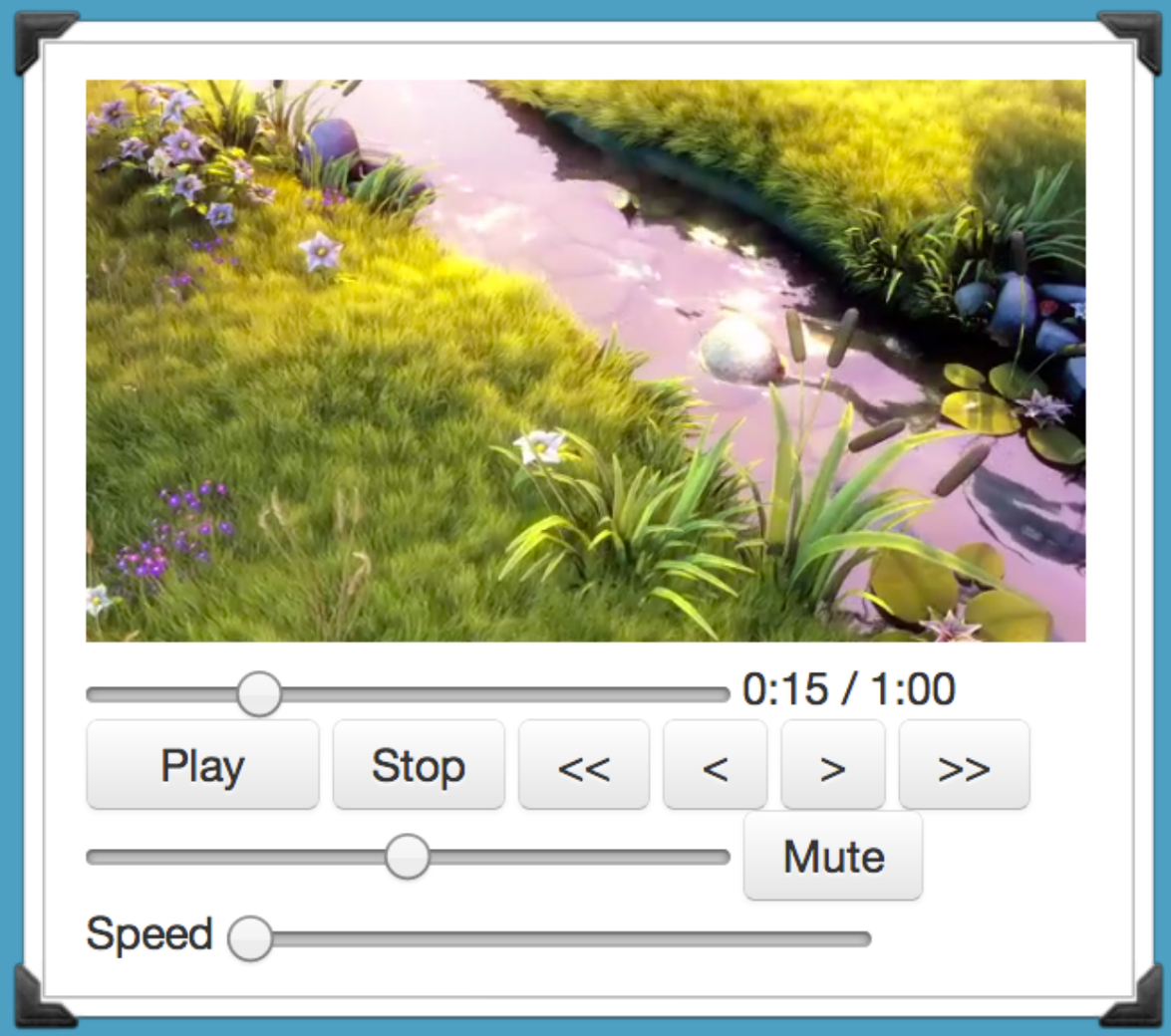
- brak mechanizmu dookreślania czasu ważności

Dwa rodzaje:

- window.**sessionStorage**
- window.**localStorage**

Media API

- **events:**
  - paused
  - ended
- **methods:**
  - play()
  - pause()
  - load()
  - currentTime =
  - playbackRate =
  - volume =
  - muted =



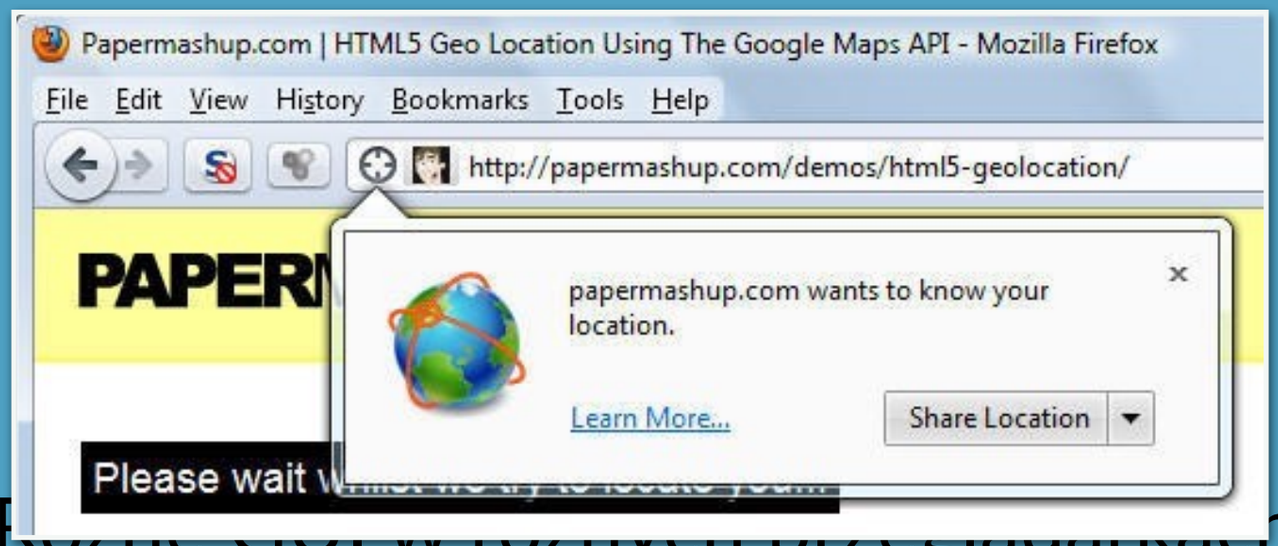
Geolocation API

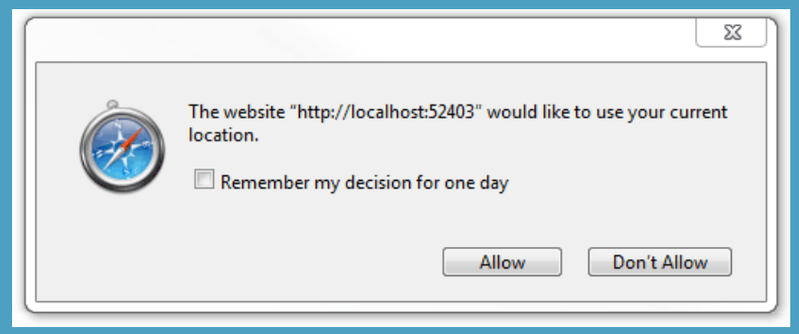
- Możliwość pobrania informacji o położeniu geograficznym urządzenia
- Za pomocą IP, GPS, WiFi lub danych sieci kom.
- Po udzieleniu zgody przez użytkownika!
- Informacje pobierane są **asynchronicznie**
- Jednorazowo lub stale

opcje:

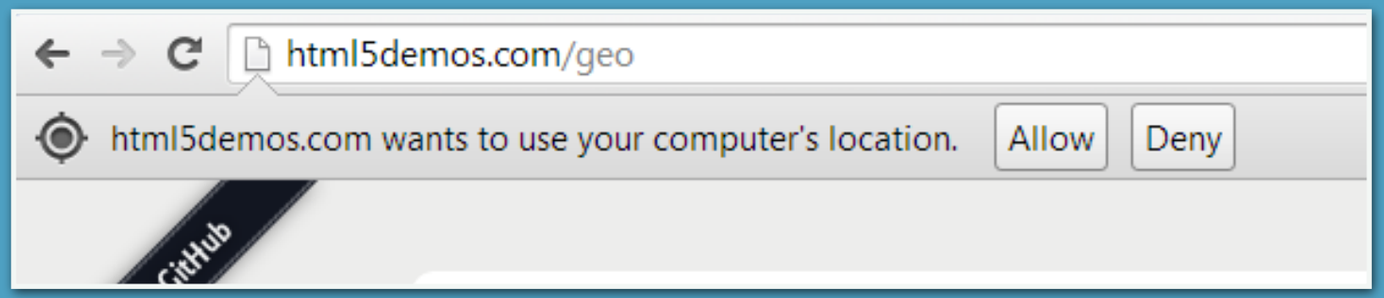
```
{
      enableHighAccuracy: true,
      maximumAge        : 32000,
      timeout           : 8000
}
```

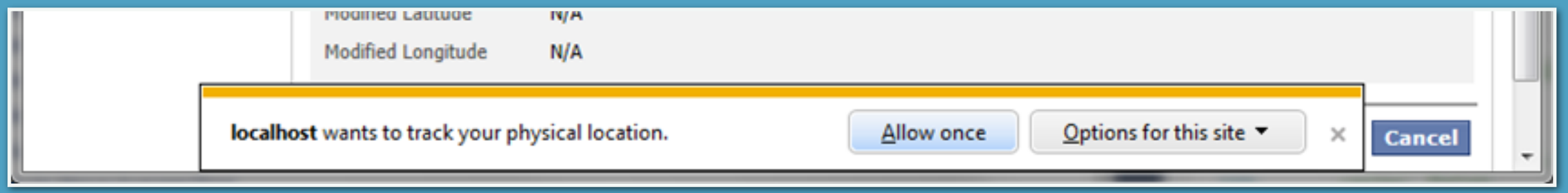
Geolocation API





Różne GUI w różnych przeglądarkach...





HTML

Formularze

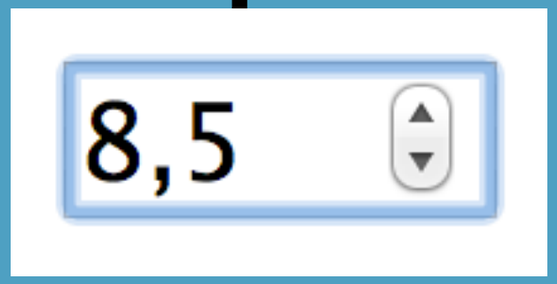
„Stare” typy elementu <input>

```
<input type="text">
<input type="radio">
<input type="checkbox">
<input type="file">
<input type="hidden">
<input type="password">
<input type="submit">
<input type="button">
```

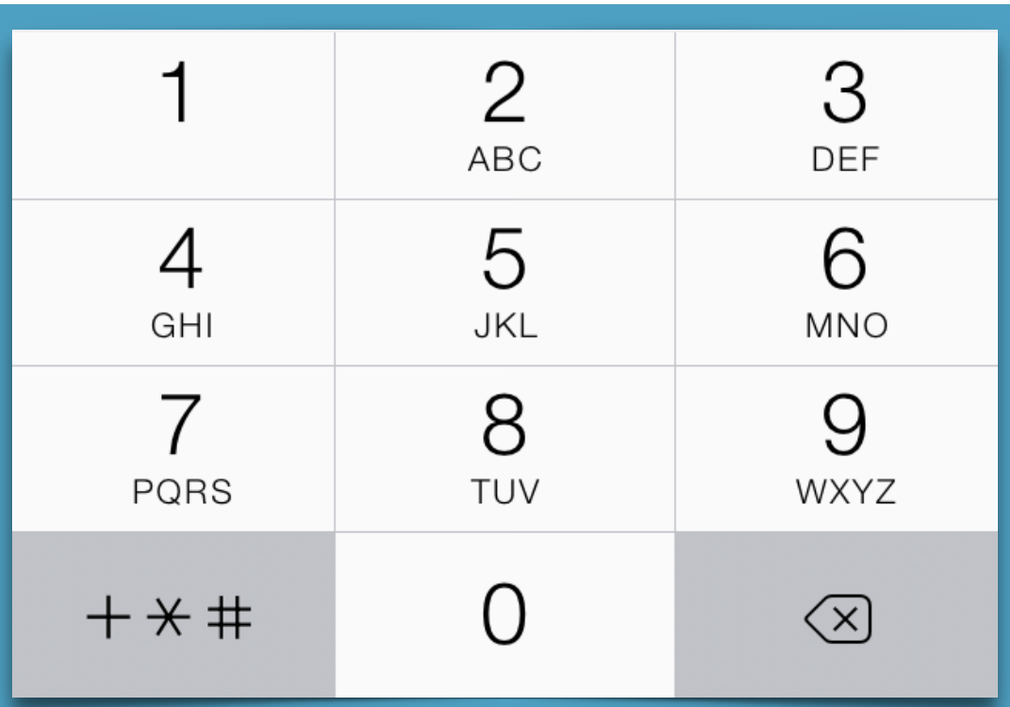
Nowe typy elementu <input>

```
<input type="color">
<input type="datalist">
<input type="date">
<input type="datetime">
<input type="datetime-local">
<input type="email">
<input type="month">
<input type="number">
<input type="range">
<input type="search">
<input type="tel">
<input type="time">
<input type="url">
<input type="week">
```

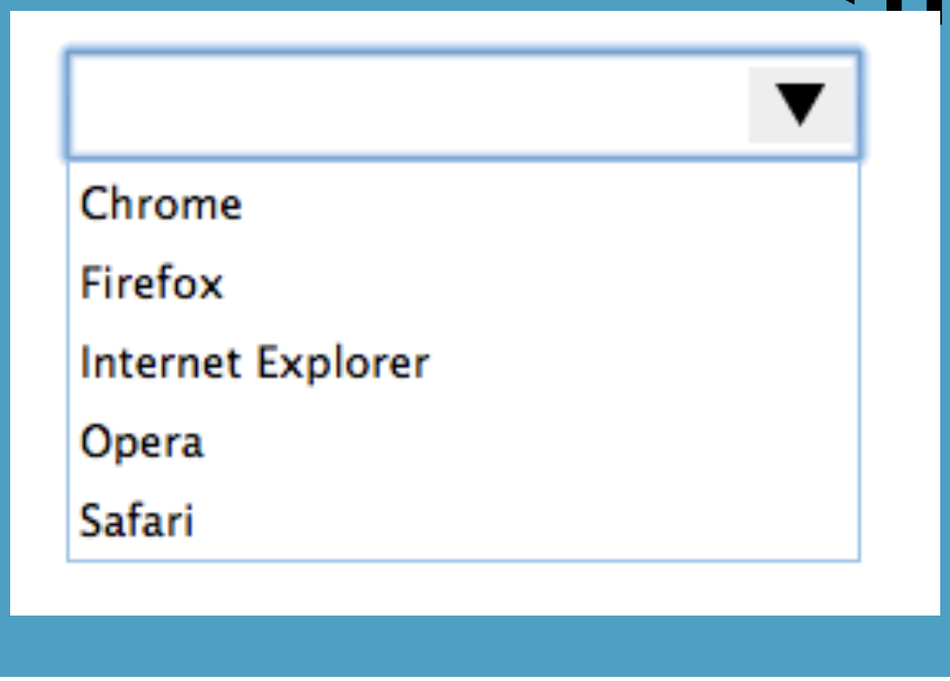
Przykładowe typy elementu <input>

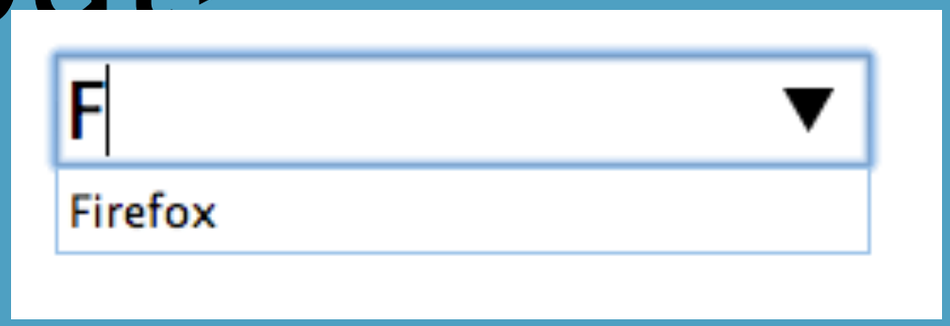



```
<input type="number" min="5" max="18" step="0.5"
value="8.5" name="shoe-size" >
```



Przykładowe typy elementu <input>






```
<input list="browsers" />
<datalist id="browsers">
    <option value="Chrome">
    <option value="Firefox">
    <option value="Internet Explorer">
    <option value="Opera">
    <option value="Safari">
</datalist>
```

Inne właściwości elementów formularzy

```
1 <form novalidate>
2       <input type="text" name="code"
3         placeholder="0-ABC"
4         autofocus
5         autocomplete="off"
6         pattern="[0-9]-[A-Z]{3}">
7 </form>
```

CSS

Podstawy

CSS

Składnia

```
/* Komentarz */

selektor { wlasnosc: wartosc;}
```

Selektory podstawowe

```
<ul id="menu">
    <li>Witamy!</li>
    <li><a href="#">Start</a></li>
    <li id="about" class="active"><a href="#">O nas</a>
        <ul>
            <li class="important active"><a href="#">O firmie</a></li>
            <li><a href="#">Zarząd</a></li>
            <li><a href="#">Partnerzy</a></li>
        </ul>
    </li>
    <li class="important"><a href="#">Kontakt</a></li>
</ul>
```

```
li {text-transform: uppercase}
.important {font-weight: bold}
#about {font-style: italic}
```

Selektory relacji

```html
<ul id="menu">
    <li>Witamy!</li>
    <li><a href="#">Start</a></li>
    <li id="about" class="active"><a href="#">O nas</a>
        <ul>
            <li class="important active"><a href="#">O firmie</a></li>
            <li><a href="#">Zarząd</a></li>
            <li><a href="#">Partnerzy</a></li>
        </ul>
    </li>
    <li class="important"><a href="#">Kontakt</a></li>
</ul>
```

```css
ul li {}
#menu > li {}
.important + li {}
.active ~ li {}
```

Selektory atrybutów

```
1 a[href] {}
2 a[href="http://google.com"] {}
3 a[data-tags~="frameworks"] {}
4 a[lang|="en"] {}
5 a[class^="btn-"] {}
6 a[class$="xs"] {}
7 a[href*="network.static"] {}
```

CSS Specificity

Który selektor wygra?

a

1 x element selector

Sith power: 0,0,1

p a

2 x element selectors

Sith power: 0,0,2

.foo

1 x class selector *

Sith power: 0,1,0

a.foo

1 x element selector
1 x class selector

Sith power: 0,1,1

p a.foo

2 x element selectors
1 x class selector

Sith power: 0,1,2

.foo .bar

2 x class selectors

Sith power: 0,2,0

p.foo a.bar

2 x element selectors
2 x class selectors

Sith power: 0,2,2

#foo

1 x id selector

Sith power: 1,0,0

https://stuffandnonsense.co.uk/archives/css_specificity_wars.html

CSS Specificity

Który selektor wygra? (cd)

a#foo
1 x element selector
1 x id selector

Sith power: 1,0,1

.foo a#bar
1 x element selector
1 x class selector
1 x id selector

Sith power: 1,1,1

.foo .foo #foo
2 x class selectors
1 x id selector

Sith power: 1,2,0

style
1 x style attribute

Sith power: 1,0,0,0

*** Same specificity**
class selector =
attribute attribute =
pseudo-classes

!important

https://stuffandnonsense.co.uk/archives/css_specificity_wars.html

Pseudo-elementy

```
::after {}
.ok::before {
    content: "✔ " attr(data-reason);
    display: inline-block;
    position: absolute;
    left: -20px;
    width: 20px;
    text-align: center;
}
::first-letter {}
::first-line {}
::selection {}
```

::before i ::after

- działają dla **niepustych elementów**
- wymagają atrybutu **content**

Pseudo-klasy

:active, :focus, :hover
:checked
:disabled, :enabled
:empty
:first-child
:last-child, :nth-child(), :nth-last-child()
:first-of-type, :last-of-type, :nth-last-of-type(odd), :nth-of-type(2n+1)
:invalid, :valid
:lang()
:left, :right
:not(***simple selector***)
:only-child, :only-of-type
:optional, :required
:target

Kolory

```
{
    color: white;
    color: #fff;
    color: #FFFFFF;

    color: rgb(255, 255, 255);
    color: rgba(255, 255, 255, 1);
    color: rgba(100%, 100%, 100%, 1);
    color: hsl(0, 100%, 100%);
    color: hsla(0, 100%, 100%, 1);
    color: transparent;
    text-shadow: 0 0 currentColor;
}
```

transform

```
1 {
2       transform:
3           rotate(45deg)
4           translate(200px, 0)
5           skew(0, 0)
6           scale(1);
7       transform-origin: right bottom;
8       transform-origin: 200px 100%;
9 }
```

transition

```
{
    transition-property: all;
    transition-duration: 2000ms;
    transition-timing-function: ease | ease-in |
      linear | steps(4, end);
    transition-delay: 2s;
    transition: transform 1s ease-in-out 2s;
}
```

co podlega transition?

animacje

```
@keyframes animacja {
    from {
        left: 0;
    }
    to {
        left: 100%;
    }
}
```

VS

```
@keyframes animacja {
    0% {
        left: 0;
    }
    100% {
        left: 100%;
    }
}
```

```
.dynamic {
    animation-name: animacja;
    animation-duration: 2s;
    animation-timing-function: ease-in-out;
    animation-iteration-count: 4; // lub infinite
    animation-delay: 1s;
    animation-direction: alternate; // normal | reverse
}
```

Responsive Web Design

RWD: definicje

Fixed - Szerokość strony jest stała bez względu na rozmiar okna przeglądarki czy rodzaj urządzenia.

Fluid - Szerokości elementów na stronie określone są za pomocą wartości procentowych. Kolumny utrzymują rozmiary proporcjonalne do siebie i okna przeglądarki i płynnie zmieniają szerokość.

Adaptive - Strona posiada media queries, które definiują określone szerokości elementów dla poszczególnych rozmiarów urządzeń.

Responsive - strona jest zbudowana na płynnej siatce i używa media queries do określania zmian w layoucie w zależności od szerokości okna przeglądarki.

RWD: proces

Mobile first

RWD: proces

Definiowanie breakpoints („wartości granicznych”)

Standardowe wartości to:

- 320px i 480px („iPhone”)
- 768px i 1024px („iPad”)
- powyżej 1024px („desktop”)

Bardziej uniwersalna metoda:

1. Projektujemy najmniejszy ekran
2. Rozszerzamy okno, póki design wygląda dobrze
3. W momencie, w którym przestaje - ustawiamy breakpoint

RWD: meta viewport

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

RWD: media queries - zapis

Media Queries (2.1) używamy od lat:

```
1 <link rel="stylesheet" href="css/styles.css"
2               media="screen" type="text/css">
3 <link rel="stylesheet" href="css/print.css"
4               media="print" type=„text/css">
```

```
1 @media screen {
2       p {color: blue;}
3 }
4
5 @media print {
6       p {color: red;}
7 }
```

RWD: media queries - zapis

Którego zapisu używać?

```
@media (max-width:632px) {}

@media screen and (max-width:632px) {}

@media only screen and (max-width:632px) {}
```

RWD: media queries - dostępne warunki

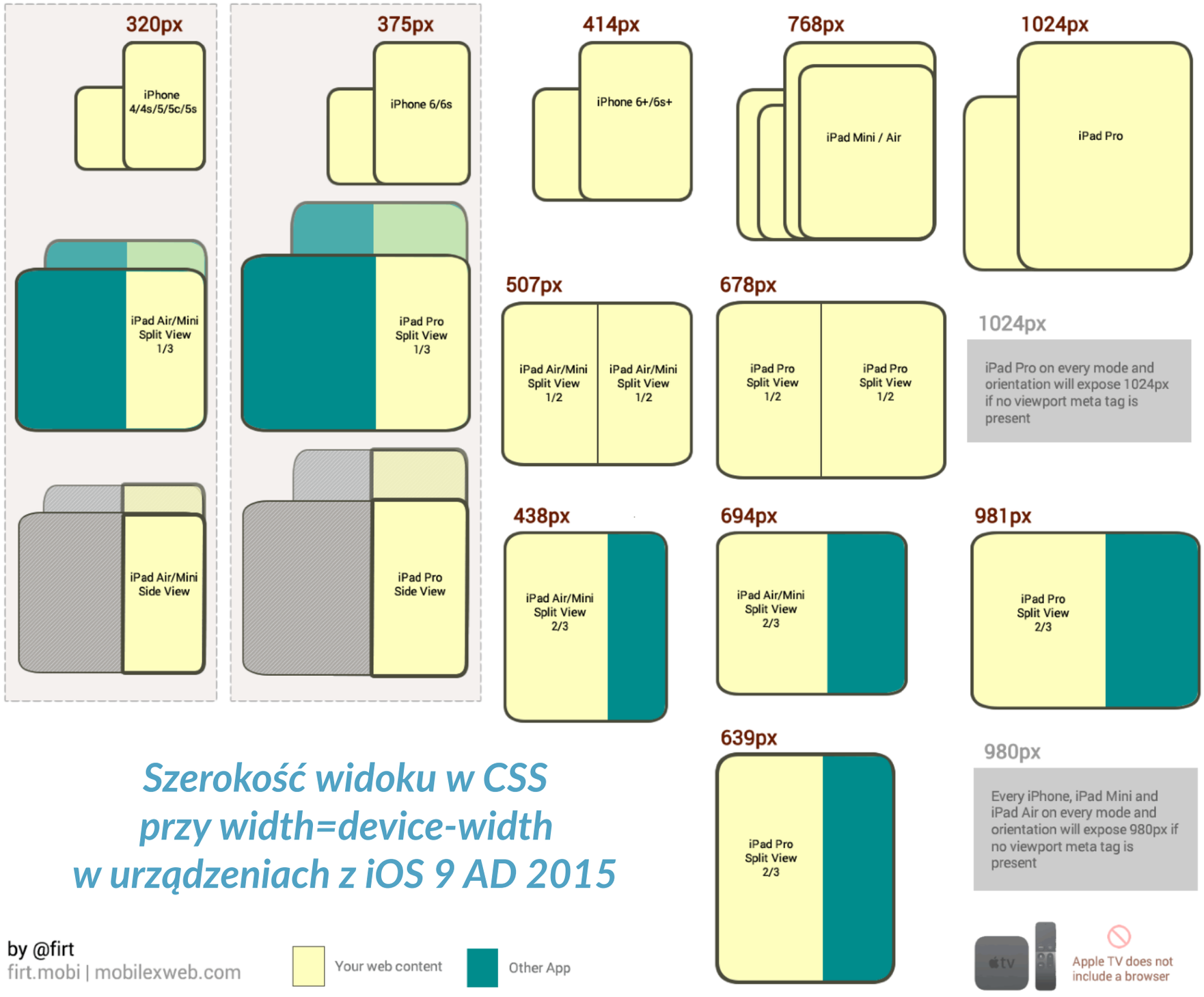
CSS2.1

1. all
2. aural
3. handheld
4. braille
5. embossed
6. **print**
7. projection
8. **screen**
9. tty
10. tv

CSS3

1. **width**
2. **height**
3. **device-width**
4. **device-height**
5. **orientation**
6. **aspect-ratio**
7. **device-aspect-ratio**
8. color
9. color-index
10. monochrome
11. resolution
12. scan
13. grid

RWD: media queries - width vs device-width



RWD: img

podstawa:

```
img {max-width:100%;}
```

lepiej:

```
img {max-width:100%;
     height:auto;}
```

RWD: img

RWD: img

city-lg.png
[337kB]

city-md.png
[188kB]

city-sm.png
[90kB]

RWD: img srcset

srcset

Lista adresów URL wskazujących na pliki,
które mogą być wykorzystane przez przeglądarkę.

```
<img
     src="city-lg.png"
     srcset="city-sm.png, city-md.png, city-lg.png"
>
```

RWD: img srcset

w (Width Descriptor)

Opcjonalny opis wskazujący na szerokość każdej z grafik w *srcset*.

```
<img
     src="city-lg.png"
     srcset="city-sm.png 400w,
             city-md.png 800w,
             city-lg.png 1200w"
>
```

RWD: img srcset

x (Pixel Density Descriptor)

Opcjonalny opis wskazujący na docelową gęstość pikseli dla każdej z grafik w *srcset*.

```
<img
     src="city-lg.png"
     srcset="city-sm.png,
             city-md.png 2x,
             city-lg.png 3x"
>
```

RWD: picture

<picture>

Określa warunki, dla których mają zostać załadowane odpowiednie grafiki.

```
<picture>
    <source srcset="city-lg.png" media="(min-width: 800px)">
    <source srcset="city-md.png" media="(min-width: 600px)">
    <img src="city-sm.png" alt="A city landscape">
</picture>
```

RWD: can I use „picture”?

RESPONSIVE IMAGES COMMUNITY GROUP

https://responsiveimages.org/

Picturefill

A RESPONSIVE IMAGE POLYFILL

https://scottjehl.github.io/picturefill/

Typografia: dlaczego to ważne?

„Web design is 95% typography”

Naprawdę?

„Typography is the foundation of web design”

Tak lepiej.

Typografia: gdzie?

Popularne zastosowania tekstu:

- nagłówki
- długie paragrafy
- cytaty, zajawki mające przykuć uwagę
- nawigacja

Typografia: jak?

Na co będziemy zwracać uwagę:

- **konsekwentna** reprezentacja
  w różnych kontekstach

- czytelna długość linii
  na wszystkich rozmiarach ekranów

- waga strony

Typografia: skąd?

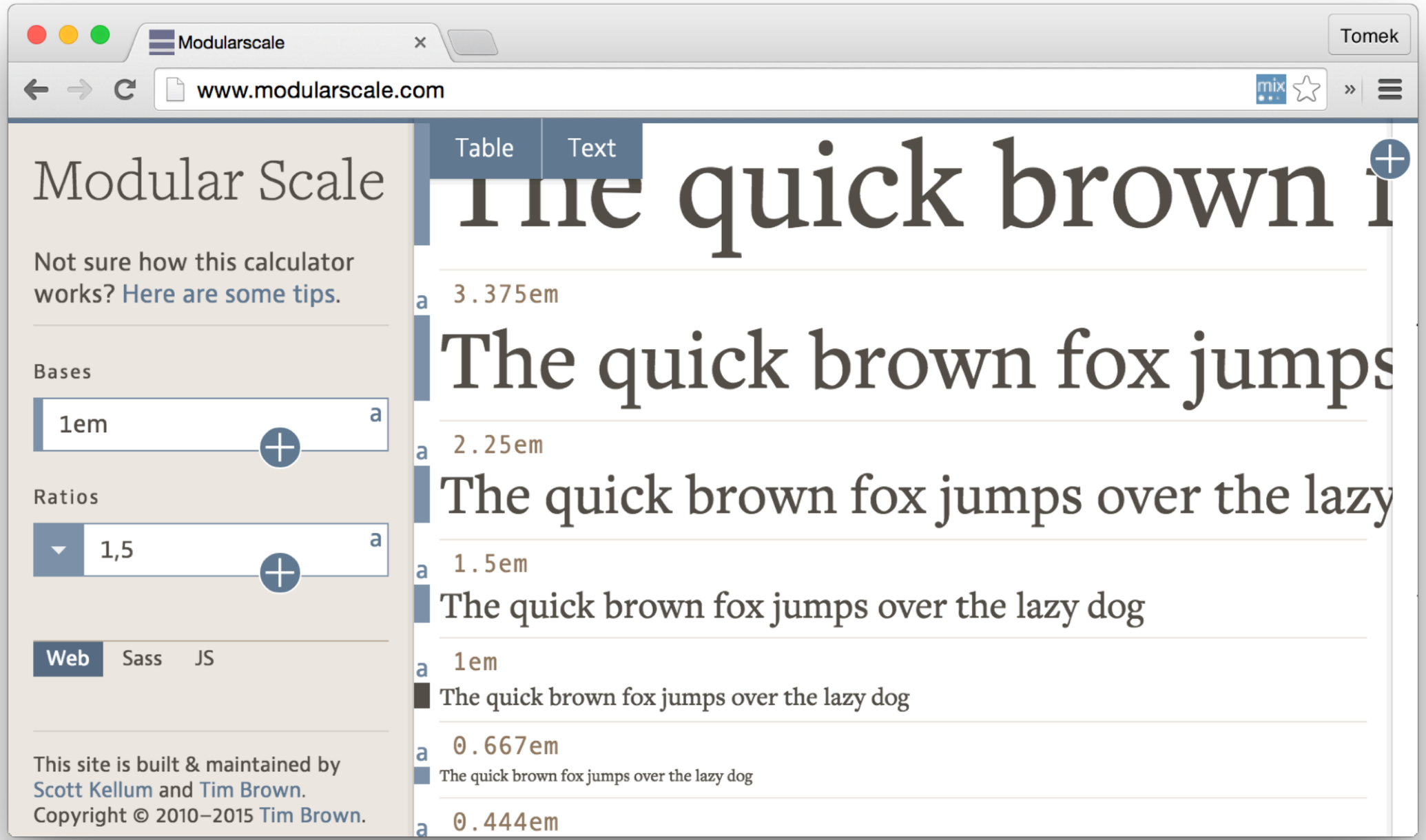
Źródło fontów:

Self-hosted: FONT SQUIRREL

Fonts providers:

Google Fonts Adobe Typekit FONT DECK

Typografia: konsekwencja = skala



Budowa skali typograficznej

Typografia: jednostki

	px	em	rem
😊	wartości absolutne	wartości relatywne	wartości relatywne
	łatwe do przeniesienia z designu	liczone względem kontekstu	liczone względem globalnego kontekstu (html)
	„najbardziej precyzyjne”	łatwe do globalnej modyfikacji	łatwe do globalnej modyfikacji
☹️	problemy z dostępnością w IE	problemy ze zmianą kontekstu	wsparcie przez IE >8 (pełne przez IE >10)
	mało elastyczne	mniej „przenośne”	

Typografia: jednostki

vw / vh / vmin / vmax

liczone względem wymiarów okna

„bardzo responsywne”

wsparcie przez IE >8
(oprócz *vmax*, z jakiegoś powodu)

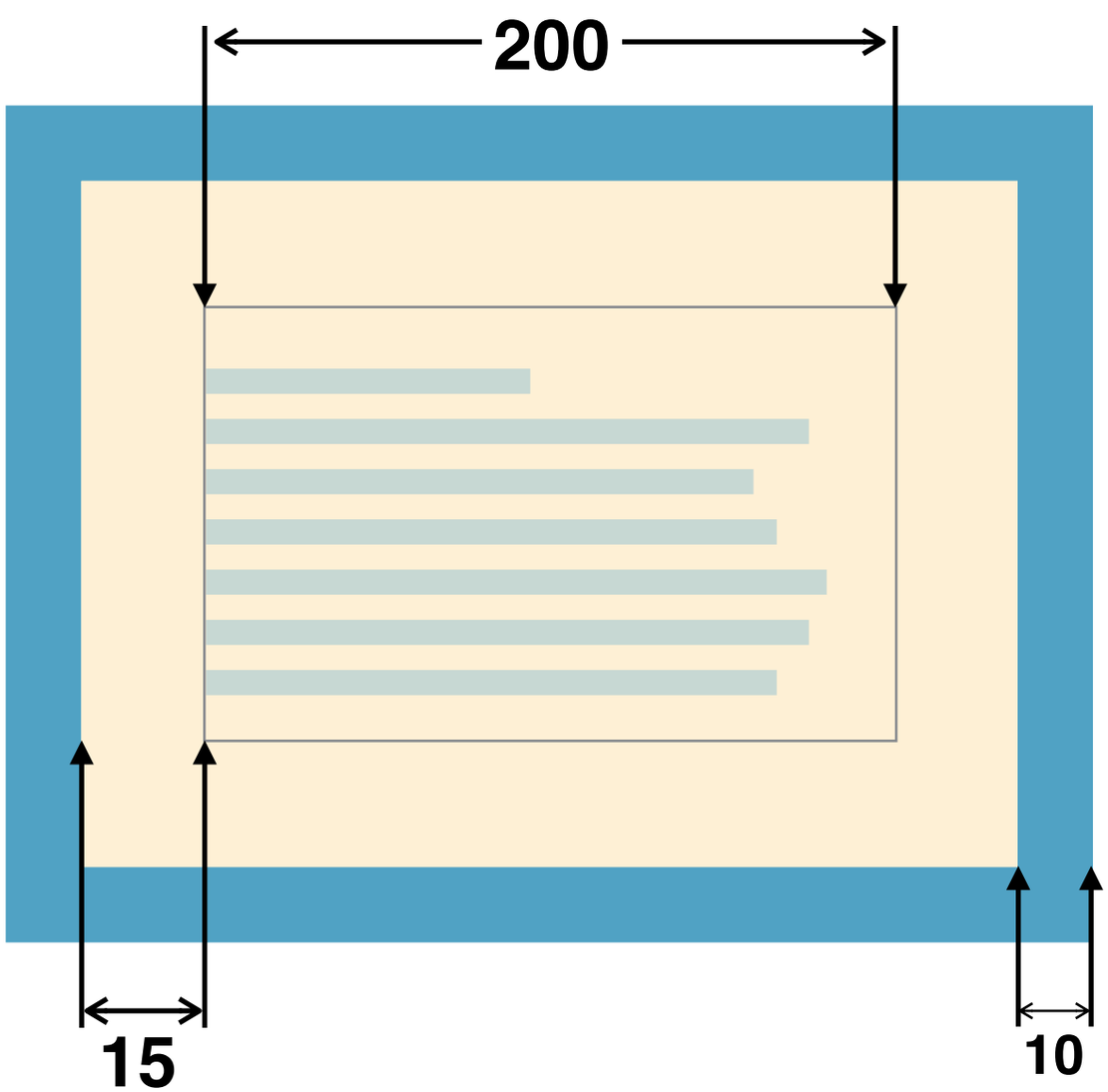
zwykle wymagają dookreślania
dla mocno różniących się rozmiarów ekranów

Typografia: parametry wpływające na czytelność

typografia	css
stopień pisma	size
leading	line-height
measure	width

- kombinacja tych trzech parametrów wpływa na wygodę czytania
- optymalna szerokość linii to 45-75 znaków
- *text-align: left* również pomaga

CSS: box model



„Ile miejsca zajmuje ten blok?”

```
.box {
    width: 200px;
    margin: 15px;
    border-width: 15px;
}
```

CSS: flexbox

Flexible Box Layout to nowy tryb definiowania układu elementów w CSS 3, który pozwala na uzyskanie zaawansowanych layoutów aplikacji internetowych.

Dostępne właściwości:

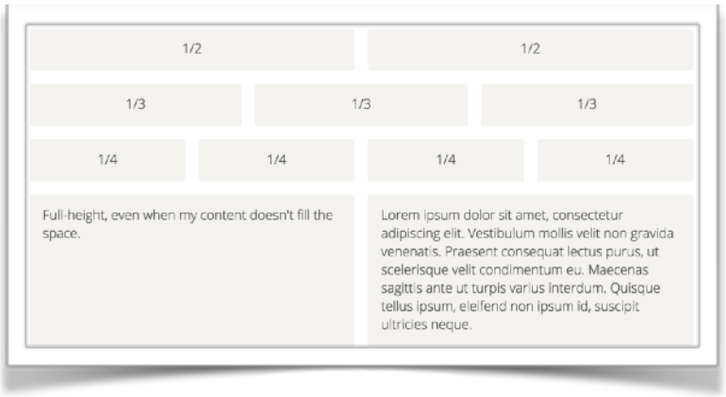
CONTAINER

```
flex-direction, flex-wrap, flex-flow,
justify-content, align-items, align-content
```

CHILD ITEM

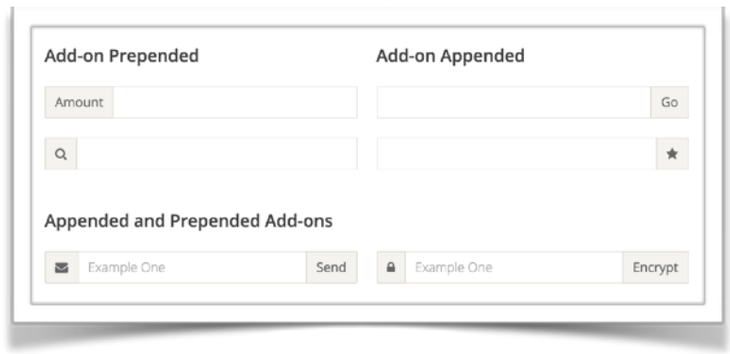
```
order,
align-self,
flex-grow, flex-shrink, flex-basis
```

CSS: flexbox - rozwiązania

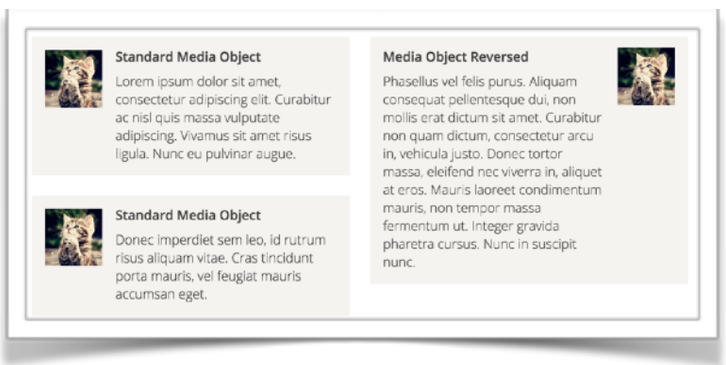


Grid system

„Holy Grail Layout"



Input addons



Media element

Sticky footer

Centering!

Wzorce RWD: nawigacja

Popularne wzorce responsywnej nawigacji:

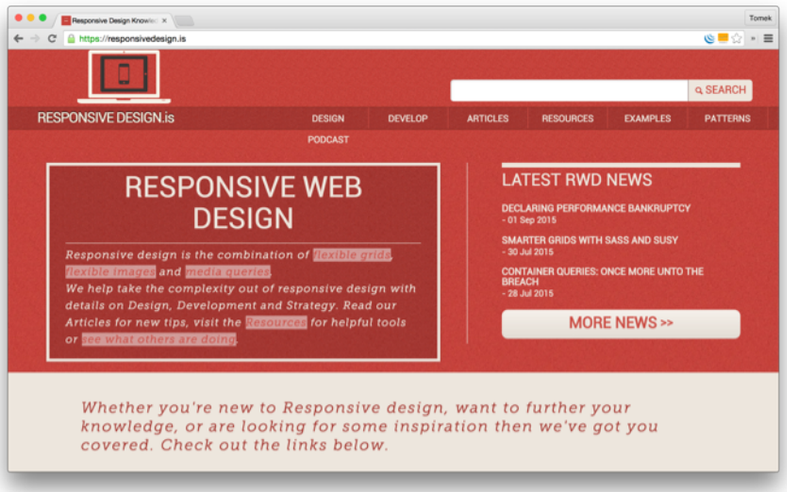
- zwężane menu („bez zmian”)
- menu w stopce
- menu <select>
- dropdown toggle
- off-canvas
- pull-down

Wzorce RWD: nawigacja

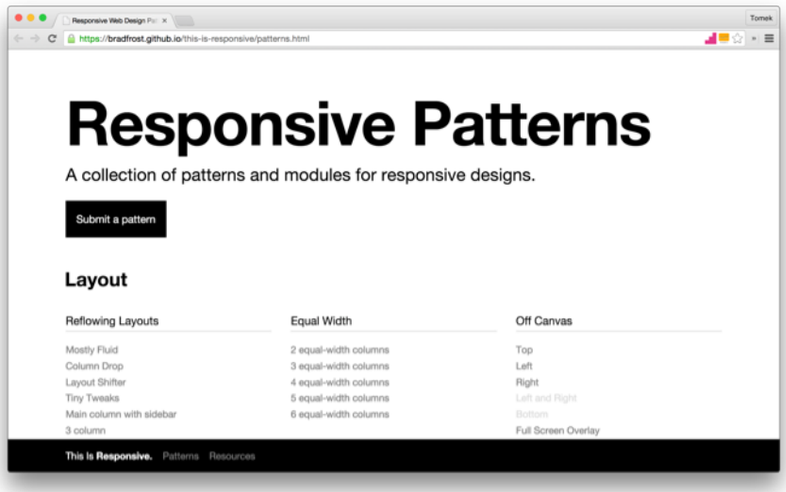
Rozwiązania responsywnej zaawansowanej nawigacji:

- multi-toggle („zagnieżdżony akordeon”)
- „prawy do lewego”
- bez podmenu
- „priority+”

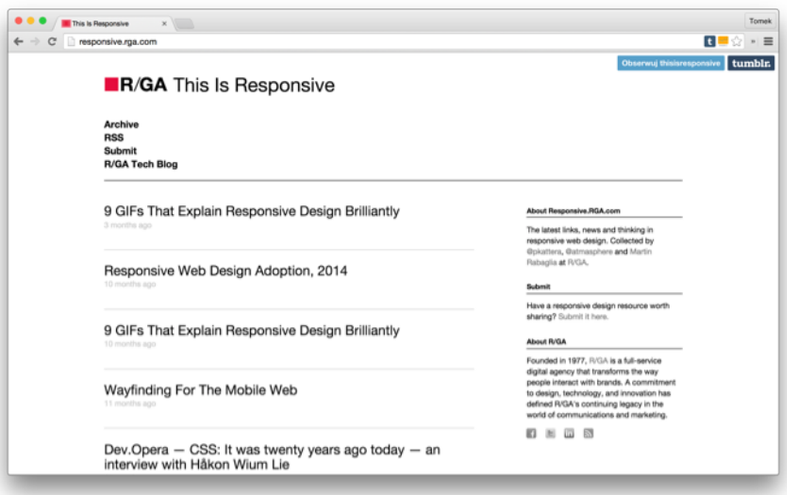
RWD: przydatne zasoby



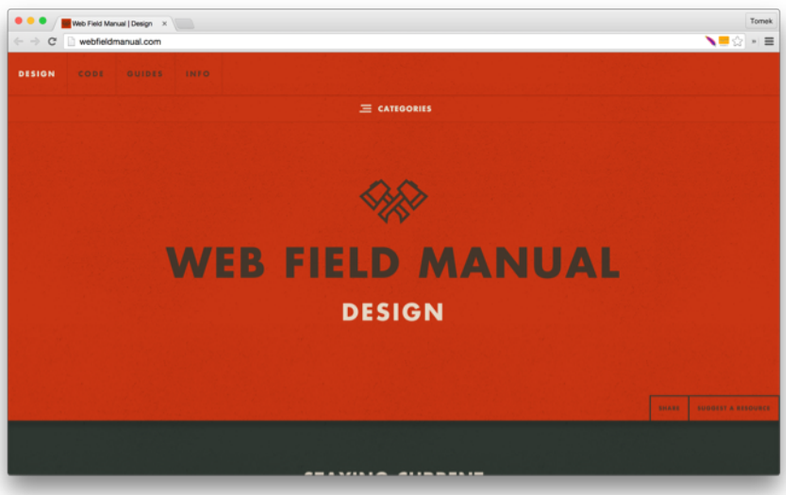
https://responsivedesign.is/



https://bradfrost.github.io/this-is-responsive/patterns.html



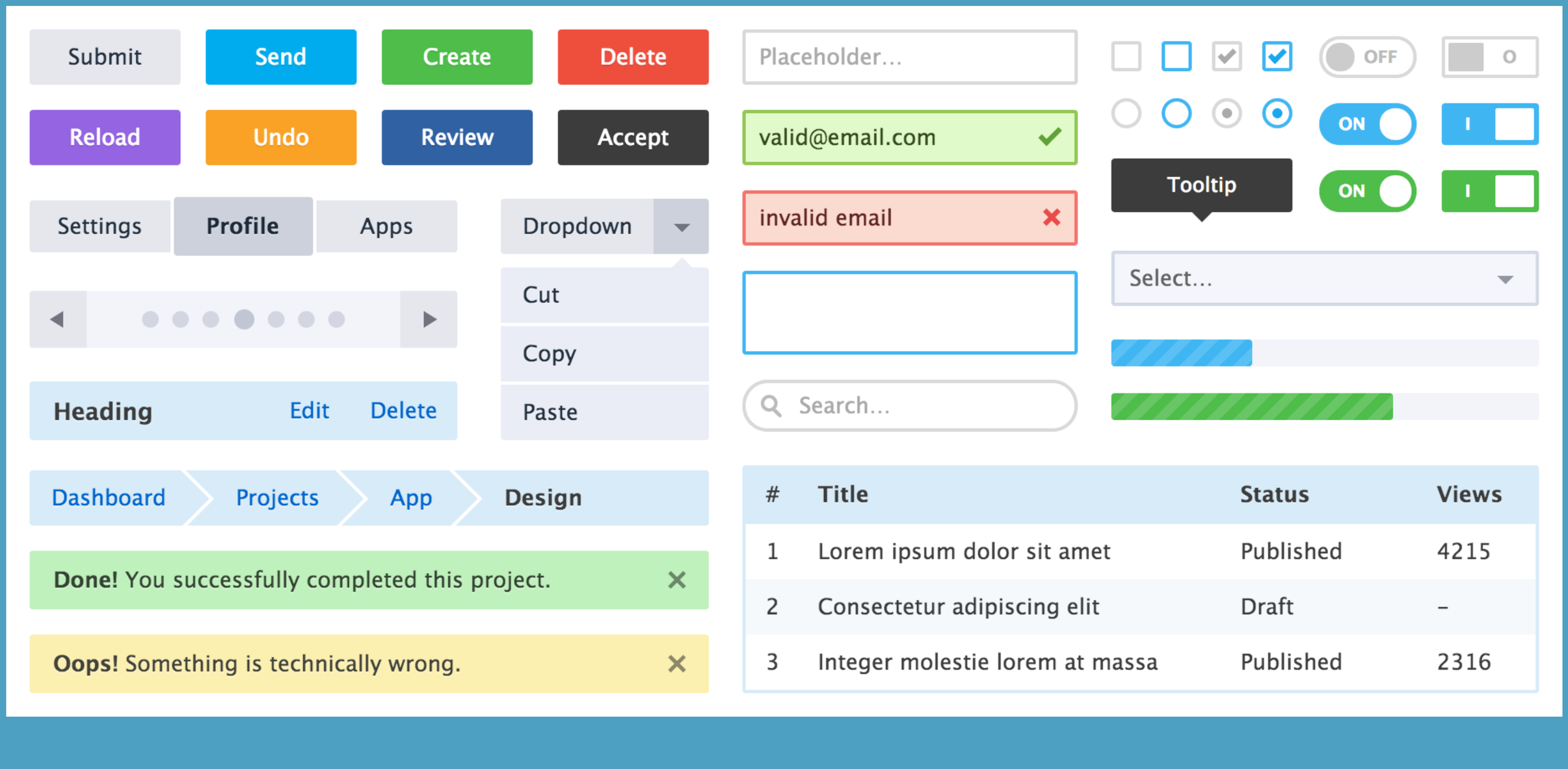
http://responsive.rga.com/



http://webfieldmanual.com/

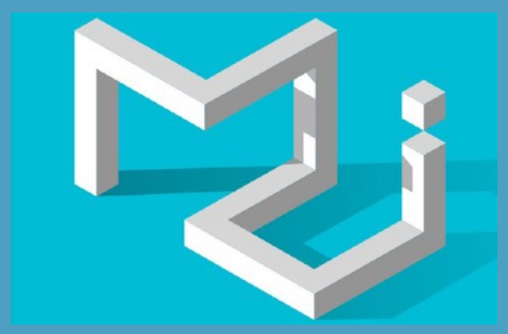
CSS Frameworks

typowe problemy – rozwiązane



CSS Frameworks

typowe problemy – rozwiązane



JavaScript

JavaScript frameworks

nie wymyślajmy koła od nowa

- Rozwiązanie podstawowych, typowych problemów
- Zapewnienie przewidywalnej struktury*
- I konwencji*
- Abstrakcja nad trudniejszymi problemami
- Własne ekosystemy dodatków

JavaScript frameworks

Aktualnie najpopularniejsze:

React *Angular* *Vue.js*

Linki

przykłady i inspiracje:

- https://stuffandnonsense.co.uk/
- https://boagworld.com/
- https://clearleft.com/casestudies
- https://www.awwwards.com/

html, css:

- https://marksheet.io/
- https://bradfrost.github.io/this-is-responsive/patterns.html
- https://philipwalton.github.io/solved-by-flexbox/

narzędzia:

- https://codesandbox.io
  https://stackblitz.com/
- https://codepen.io/
- https://webflow.com/
- https://www.invisionapp.com/
- https://www.figma.com/
- https://www.sketchapp.com/
- https://zeplin.io/

🤓

- https://twitter.com/sulco
- https://medium.com/@tomsu