

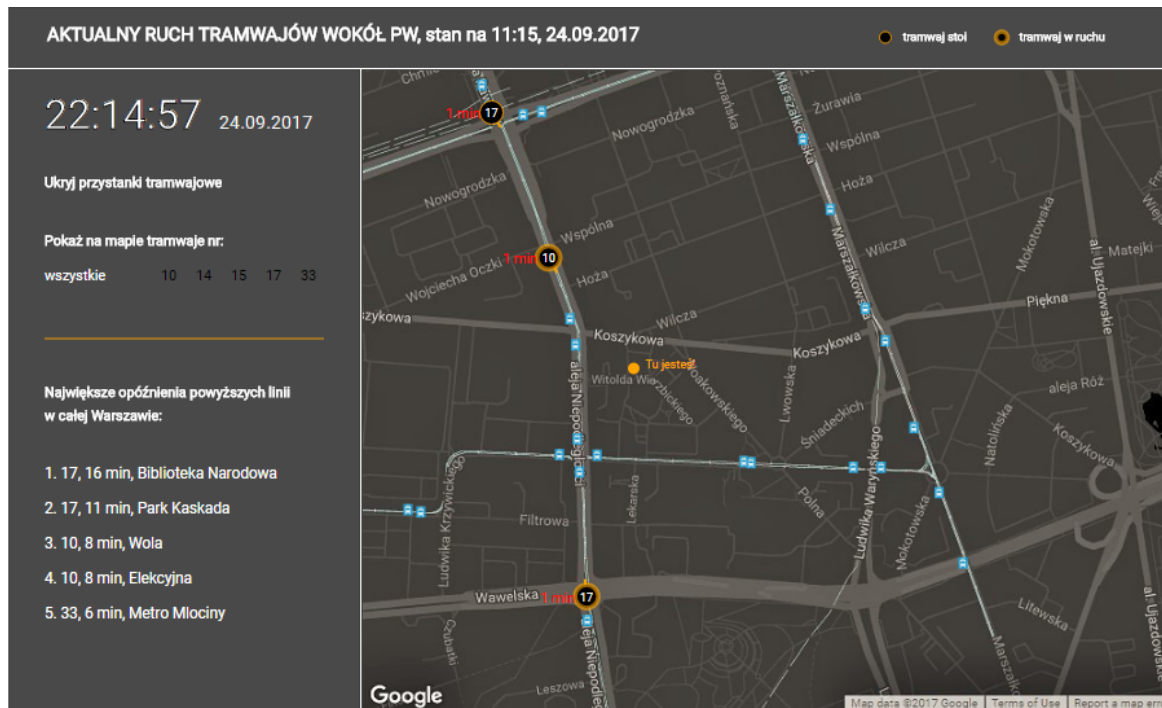
Warsztaty z biblioteki D3.js

WhyR? 2017



Plan warsztatów

1. Krótki wstęp o D3 (5 – 10 minut)
2. Use case z danymi o tramwajach (20 – 25 min)

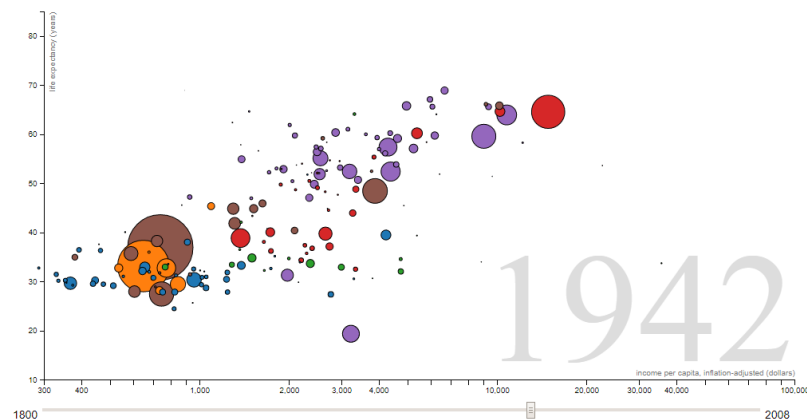


D3 - definicja

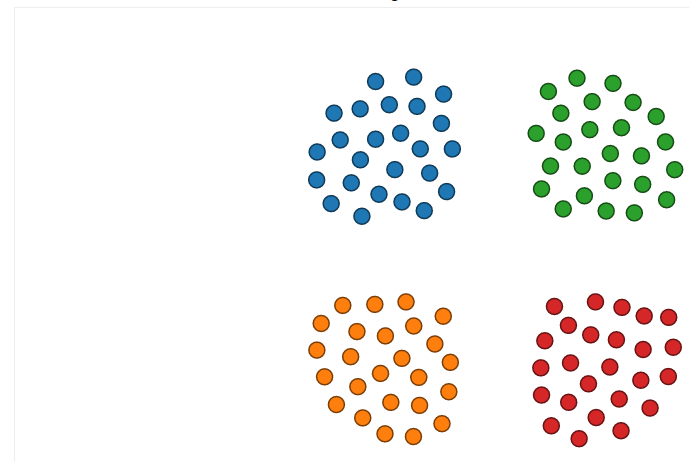
D3 to biblioteka JavaScriptowa służąca do generowania spersonalizowanych, interaktywnych wizualizacji, działających w przeglądarkach internetowych.

A Re-Recreation of Gapminder's Wealth & Health of Nations

Use your mouse to click and drag countries (represented as circles) to explore +200 years of life expectancy and income indicators.



Multi-Foci Force Layout



D3 - nazwa

D3 = **Data-Driven Documents**

	A	B	C	D	E
1	€10,015	69.98%	€1,029	0.013	Thu Jun 18 2015
2	€9,929	37.43%	€952	0.512	Fri Jun 19 2015
3	€10,054	52.03%	€1,023	1.796	Sat Jun 20 2015
4	€10,011	26.50%	€983	0.058	Sun Jun 21 2015
5	€9,923	57.00%	€1,035	-0.672	Mon Jun 22 2015
6	€10,062	37.78%	€1,005	0.124	Tue Jun 23 2015
7	€9,781	15.10%	€1,013	0.299	Wed Jun 24 2015
8	€10,032	6.09%	€947	0.070	Thu Jun 25 2015
9	€10,145	85.90%	€990	-0.626	Fri Jun 26 2015
10	€9,959	30.00%	€992	-0.171	Sat Jun 27 2015

dane

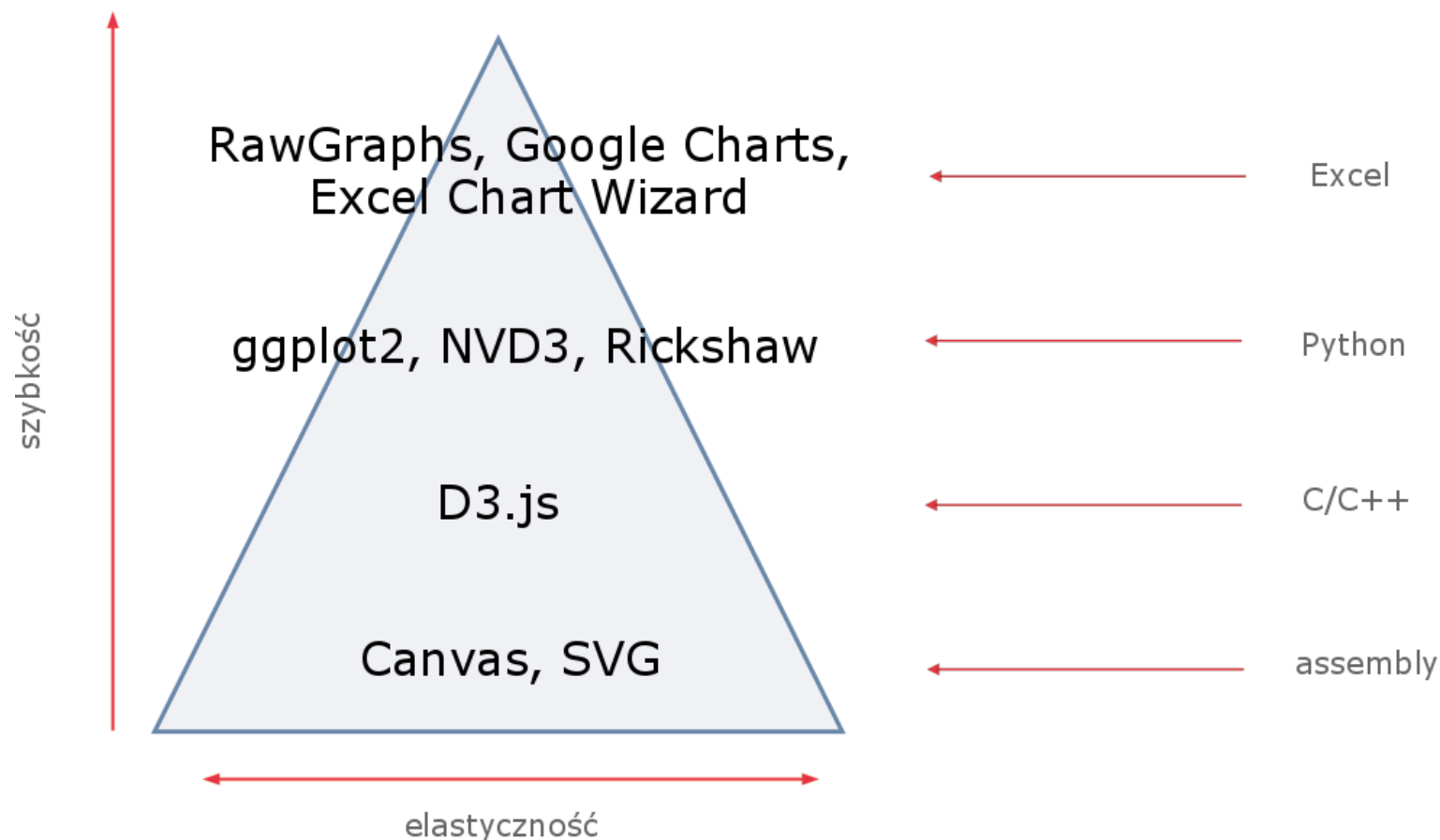


wiązane przez D3

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5     <meta http-equiv="Content-Type" content="text/html;"
6     <title>Bardzo ważna strona</title>
7
8 </head>
9
10 <body>
11
12 <p>Paragraf</p>
13
14 <h1>Nagłówek wielki</h1>
15
16 <h2>Nagłówek mniej wielki</h2>
17
18 <span>Lorem ipsum dolor sit amet enim. Etiam ullamcorpei
19 pellentesque dui, non felis. Maecenas malesuada elit
20 </span>
21
22 <a href="http://d3js.org/">The D3 website</a>
23
```

z dokumentem HTML

D3 – prezentacja, nie eksploracja



Używane technologie

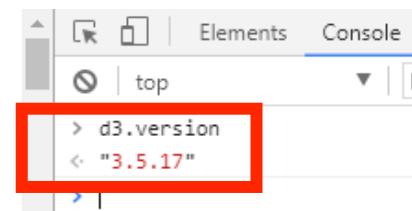


D3 – podstawowe informacje

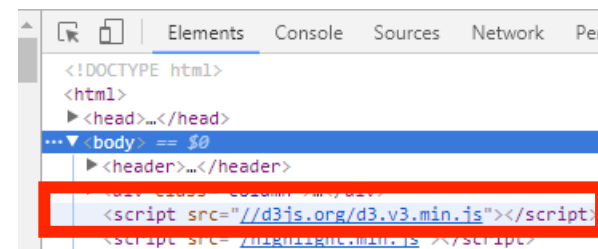
- darmowa biblioteka JavaScriptowa
- autor: Mike Bostock
- <https://d3js.org/> - strona główna
- <https://github.com/d3/d3/wiki/Gallery> - przykłady
- [API reference D3](#) - dokumentacja

Wersje D3.js: v3 vs v4

- nowa wersja 4 od poprzedniego roku
- różnice m.in.:
 - podział biblioteki na moduły
 - nazewnictwo funkcji (`d3.scale.linear()` -> `d3.scaleLinear`)
 - działanie selections (kluczowych elementów D3)
- => **nie wystarczy podmienić wersji w pliku**
- gdzie znaleźć informacje o używanej wersji:
 - `d3.version` w konsoli JavaScript w przeglądarce
 - dołączany skrypt w nagłówku htmla
- wszystkie zmiany są omówione zbiorczo na [githubie](https://github.com)



A screenshot of a web browser's developer console. The 'Console' tab is selected. A command `> d3.version` has been entered, and the output is `< "3.5.17"`. The entire console output area is highlighted with a red rectangular box.



A screenshot of a web browser's developer source code view. The 'Elements' tab is selected. The HTML document structure is shown, including `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>`. In the `<head>` section, a `<script>` tag is highlighted with a red rectangular box. The text of the script tag is `src="//d3js.org/d3.v3.min.js"></script>`.

Materiały online

D3 przykłady: bl.ocks.org

tutoriale D3: [tutorial Scott Murray](#), <http://d3indepth.com/>

SVG: [dokumentacja MDN o SVG](#), [dokumentacja W3 o SVG](#)

CSS: [tutorial MDN o CSS](#)

JavaScript: [tutoriale MDN o JavaScript](#)

Inne:

[cheatsheet z D3 Jerome Cukier](#)

[wstęp "narzędziowy" do D3](#)

[otwieranie konsoli JS w różnych przeglądarkach](#)

Idea D3 = dane + elementy htmla

- przykład - czym dysponujemy?
 - 3 obserwacje z naszego zbioru:

```
var new_data_R_1=  
[  
{"brigade":"2","line":"10","time":"2017-09-25T11:00:43Z","lon":20.977,"lat":52.236,"r":  
{"brigade":"16","line":"33","time":"2017-09-22T17:43:08Z","lon":20.9297,"lat":52.2904,  
{"brigade":"2","line":"33","time":"2017-09-25T11:00:21Z","lon":20.9281,"lat":52.2803,'
```

- 3 kółka:

```
<svg id="mapPanel">  
  <circle class="tramGroup" r="9px" cx = "10" cy="12"> </circle>  
  <circle class="tramGroup" r="9px" cx = "15" cy="12"> </circle>  
  <circle class="tramGroup" r="9px" cx = "20" cy="12"> </circle>  
</svg>
```

Faza UPDATE

- tyle samo obserwacji co elementów

elementy HTML-a/SVG

dane

`<circle></circle>`



obserwacja nr 1

`<circle></circle>`



obserwacja nr 2

`<circle></circle>`



obserwacja nr 3

update

Faza UPDATE

```
// step JOIN
var tramCircles = d3.select("#mapPanel").selectAll("circle.tramGroup")
    .data(data, function(d) { return d.brigade + d.line; })

// step: UPDATE
tramCircles
    .attr("cx", function(d) { return scaleLon(d.lon); })
    .attr("cy", function(d) { return scaleLat(d.lat); })
    .attr('stroke', function(d) {
        if(d.status == "STOPPED") { return "orange"; }
        else { return "rgba(255,165,0,0.6)"; }
    })
    .attr("stroke-width", function(d) {
        if(d.status == "STOPPED") { return "1px"; }
        else { return "4px"; }
    })
```

Faza ENTER

- więcej obserwacji niż elementów

elementy HTML-a/SVG

`<circle></circle>`

`<circle></circle>`

`<circle></circle>`

dane

faza update

obserwacja nr 1

obserwacja nr 2

obserwacja nr 3

obserwacja nr 4

obserwacja nr 5

faza enter

Faza ENTER

```
// step: ENTER
tramCircles.enter().append("circle")
.attr("class", function(d) { return "tram_" + d.line + " tramGroup"; })
.attr("r", "10px")
.attr("cx", function(d) { return scaleLon(d.lon); })
.attr("cy", function(d) { return scaleLat(d.lat); })
.attr('stroke', function(d) {
    if(d.status == "STOPPED") { return "orange"; }
    else { return "rgba(255,165,0,0.6)"; }
})
.attr("stroke-width", function(d) {
    if(d.status == "STOPPED") { return "1px"; }
    else { return "4px"; }
})
```

Faza EXIT

- mniej obserwacji niż elementów

elementy HTML-a/SVG

`<circle></circle>`

`<circle></circle>`

`<circle></circle>`

faza exit

dane

faza update

obserwacja nr 1

obserwacja nr 2

Faza EXIT

```
// step: EXIT  
tramCircles.exit().remove();
```