National Czech Programme

eosc

JSON, JSON-LD, YAML

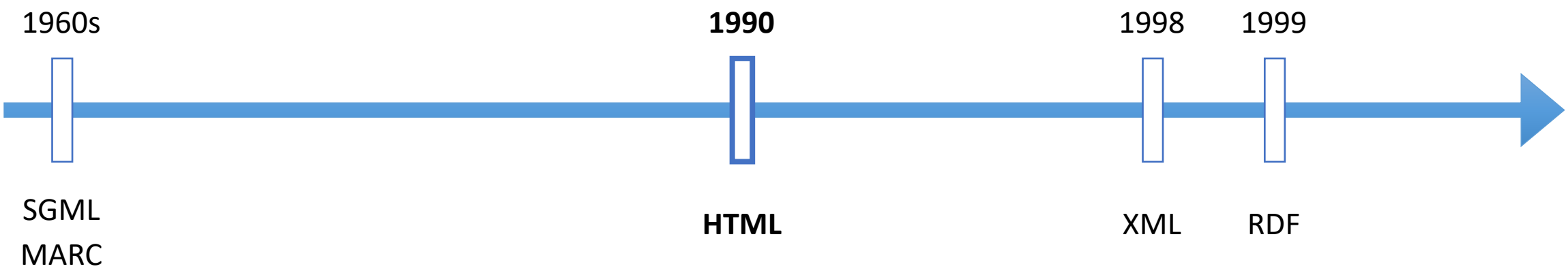
Miroslav Šimek

CESNET z.s.p.o.
NTK

Spolufinancováno Evropskou unií

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Formáty zápisu (dat)



Web - počátky

- 1989 Tim Berners-Lee, CERN
- duben 1993 WWW kód byl uvolněn jako royalty free
- konec roku 1993 - 500 serverů, 1% provozu sítě
- konec roku 1994 - 10 000 serverů, 10mil uživatelů

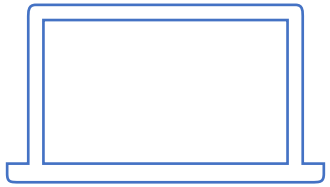
World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give univ

Everything there is online about W3 is linked directly or indirectly to this document, including an execu
Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep ,
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

Web -jak funguje

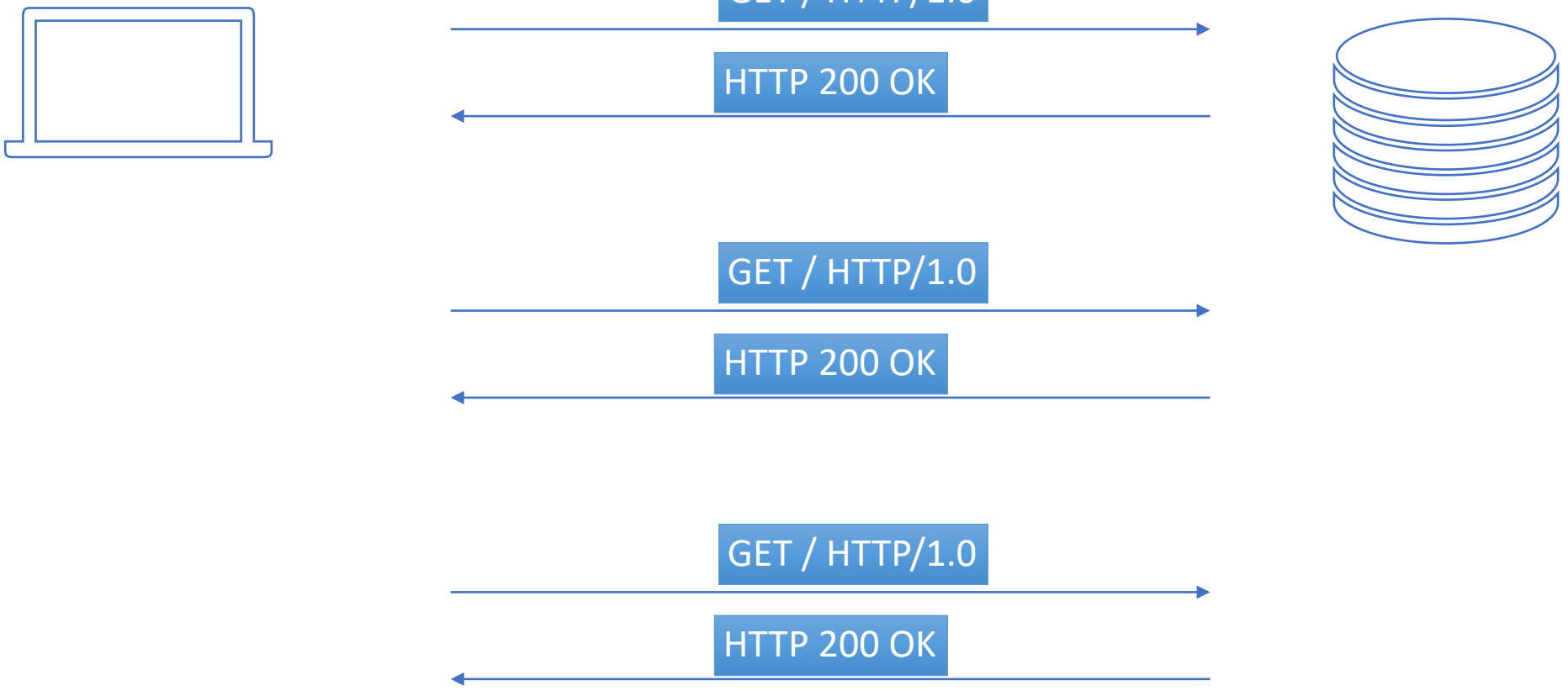


pocasi.cz

```
<html>
  <head>…</head>
  <body>
    <h1>Počasí</h1>
    <span>28</span> stupňů celsia
  </body>
</html>
```

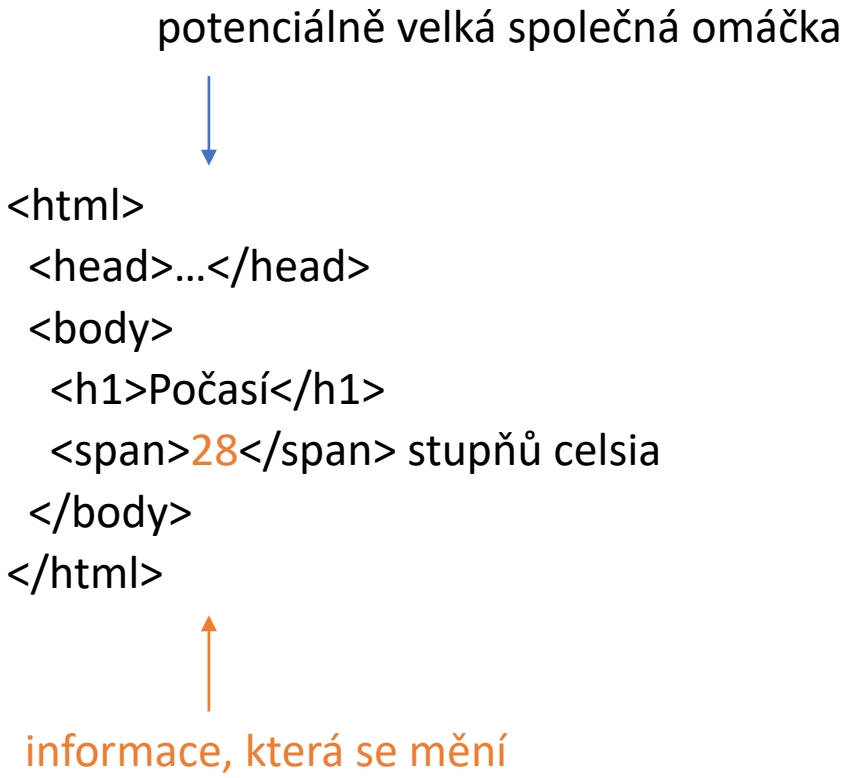
Jak udělat stránky s teplotou, která se bude aktualizovat každou minutu ?

Web -jak funguje

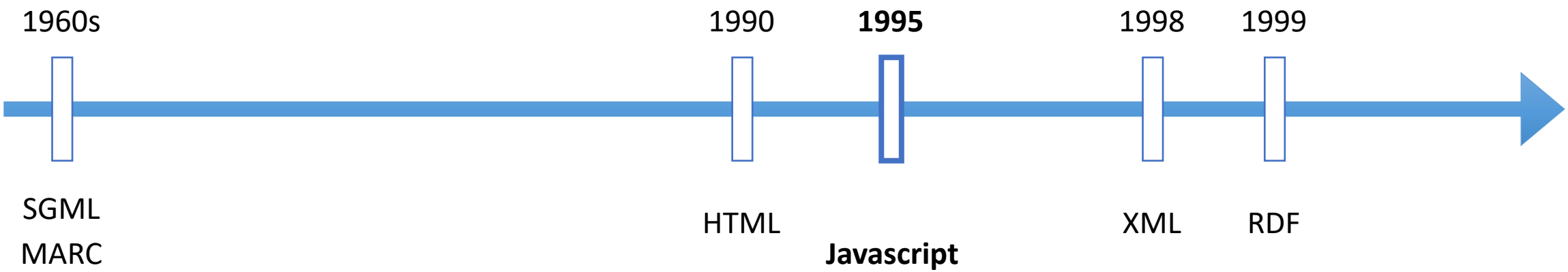


Web -jak funguje

- Funguje dobře, pokud
  - vrácené HTML je malé - to platilo na začátku webu, teď jsou stránky mnohem větší
- Vytěžuje síť (a žere data)
- Vytěžuje servery
- Je pomalé



Javascript




```
<html>
 <head>…</head>
 <body>
  <h1>Počasí</h1>
  <span id="teplota"></span> stupňů celsia
 </body>
```

```
<script>

 function nastavTeplotu(hodnota) {
  document.getElementById("teplota")
      .textContent=hodnota
 }

 nastavTeplotu(28)

</script>
```

Javascript

- Trik: skript rozdělíme na 2 části
  - část která deklaruje funkci a je v původním dokumentu
  - část která funci volá
- Pak se stačí serveru ptát periodicky pouze na druhou část, ta se v prohlížeči spustí a změní hodnotu teploty
- Technicky bylo umožněno v roce 1996

```
<html>
 <head>…</head>
 <body>
  <h1>Počasí</h1>
  <span id="teplota"></span> stupňů celsia
 </body>

<script>
 function nastavTeplotu(hodnota) {
   document.getElementById("hodnota")
       .textContent=hodnota
 }
</script>

<script>
 nastavTeplotu(28)
</script>
```

Javascript

- Takto je možno předávat nejenom čísla, ale všechny hodnoty, které podporuje Javascript:
  - čísla
  - text
  - pravda/nepravda
  - seznam v hranatých závorkách
    - [28, 28.1, 27.9]
  - komplexní objekt - více hodnot
- od 2000 - obsah de facto standardizován, pojmenován **JSON** - Javascript Object Notation
- později přibyly technické prostředky jak posílat pouze JSON, bez obálky okolo

```
<script>
  nastavTeplotu(28)
</script>

<script>
  nastavVyvojAkcii([250, 251, 252, 180])
</script>

<script>
  nastavPrijatouZpravu({
    "odesilatel": "jan.novak@gmail.com",
    "predmet": "....",
    "text": "....."
  })
</script>
```

JSON

- Dlouho de-facto standard, standardizován až 2013
- Jednoduchá struktura
  - objekt - ve složených závorkách
  - seznam - v hranatých závorkách
  - text - v uvozovkách
  - hodnoty odděleny čárkou

```
{
  "teploty": [
    {
      "cas": "12:00",
      "teplota": 28.1,
      "validovano": true
    },
    {
      "cas": "12:05",
      "teplota": 28.2,
      "validovano": false
    }
  ]
}
```

JSON - proč ne XML

- Jednodušší koncept
- Už byl přirozeně v prohlížeči, nebylo potřeba instalovat zásuvné moduly
- Protože je to část programovacího jazyka, programátoři s ním uměli přirozeně pracovat.

```
x = … nejaky json s publikaci …

prijmeni = x.autor.prijmeni
```

```
x = … nejake xml s publikaci …

prijmeni = x.getElementByTagName('autor')
        .getElementByTagName('prijmeni').textContent
```

JSON vs. XML

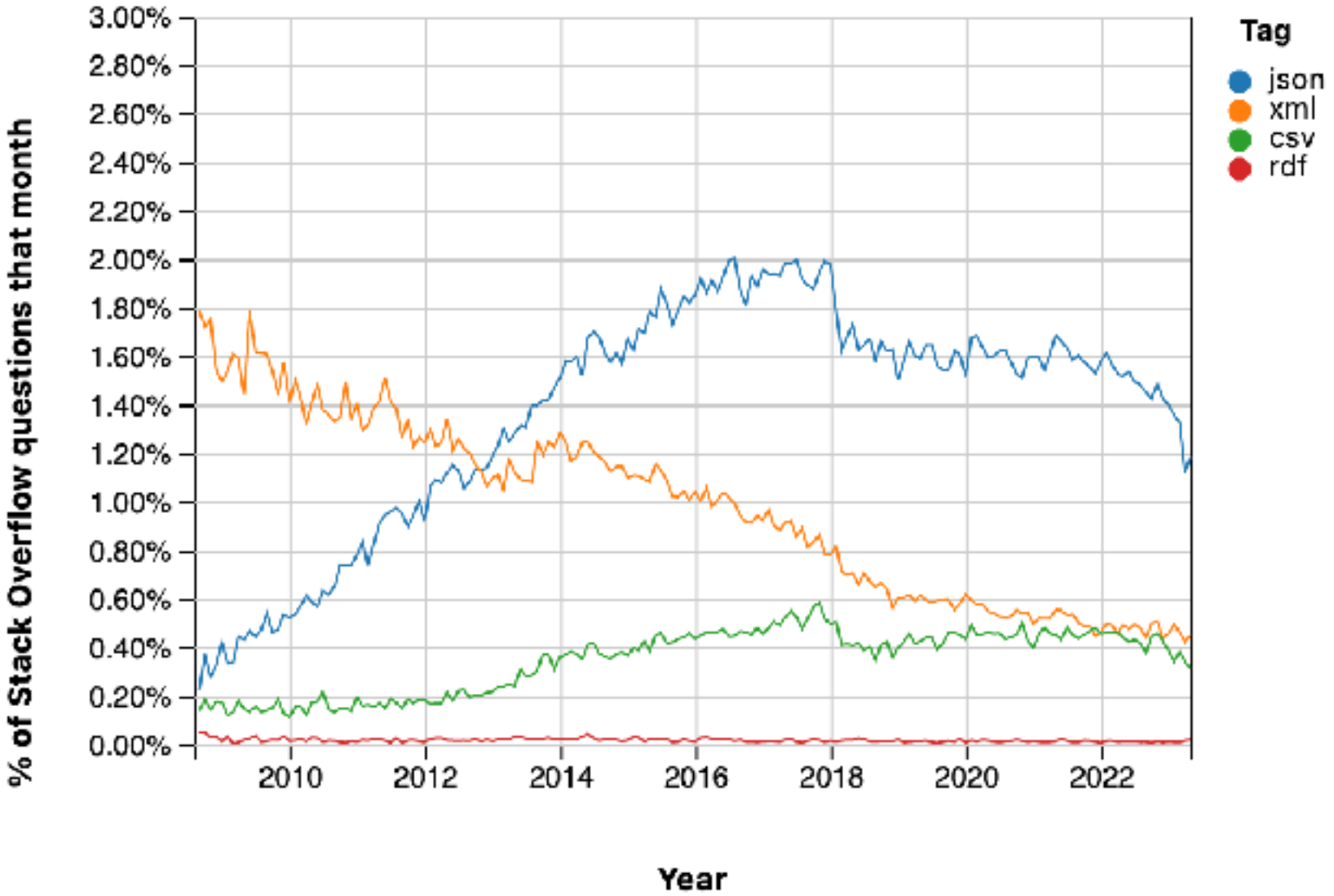
- Oba standardy popisují hierarchická data
- XML informaci dělí na atributy a dětské elementy, JSON nic takového nemá - vše je dětský element
- JSON máme vyjádřen typ, v xml je vše text
- JSON je obvykle kratší
- JSON nemá komentáře - je pro strojové zpracování, ne pro lidskou tvorbu
- JSON je obecně více podporován

```json
{
  "teploty": [
    {
      "cas": "12:00",
      "teplota": 28.1,
      "validovano": true
    },
    {
      "cas": "12:05",
      "teplota": 28.2,
      "validovano": false
    }
  ]
}
```

```xml
<teplota>
  <cas>12:00</cas>
  <validovano>true</validovano>
  <hodnota>28.0</hodnota>
</teplota>
```

```xml
<teploty>
  <teplota cas="12:00" validovano="true">
    28.1
  </teplota>
  <teplota cas="12:00" validovano="true">
    28.1
  </teplota>
  <!-- tohle je komentar -->
</teploty>
```

JSON vs. XML



https://insights.stackoverflow.com/trends?tags=json%2Cxml%2Ccsv%2Crdf

Příklad Zenodo


```
{
    "doi": "10.5281/zenodo.7808805",
    "created": "2023-04-07T14:53:24.165206+00:00",
    "updated": "2023-04-08T02:27:07.699494+00:00",
    "revision": 3,
    "id": 7808805,
    "metadata": {
      "access_right_category": "success",
      "description": "Make some beautiful corner plots
      "title": "dfm/corner.py: corner v2.2.2rc3",
      "publication_date": "2023-04-07",
      "creators": [
        {
          "affiliation": "Flatiron Institute",
          "name": "Adrian Price-Whelan"
        }
      ],
      "access_right": "open",
      "resource_type": {
        "type": "software",
        "title": "Software"
      },
      "related_identifiers": [
        {
          "scheme": "url",
          "identifier": "https://github.com/dfm/co
          "relation": "isSupplementTo"
        },
        {
          "scheme": "doi",
          "identifier": "10.5281/zenodo.591491",
          "relation": "isVersionOf"
        }
      ]
    }
}
```


https://zenodo.org/record/7808805/export/json

Národní metadatový adresář

- NMA bude založeno na repozitářové platformě Invenio
- Invenio používá JSON pro uložení (PostgreSQL) a vyhledávání (Opensearch) metadat
- Invenio používá JSON pro API přístup jako primární metadatový zápis

- Národní repozitářová platforma
  - repozitáře vytvářené CESNETem jsou založeny na platformě Invenio, takže jejich primárním metadatovým zápisem je JSON

XML Odbočka

- Máme 2 systémy, z každého přišel element "Osoba" a pokaždé jinak

- Pokud nechceme přejít na RDF, jak vyjádřit rozdíl ?

```
<osoba>
  <jmeno>Mirek</jmeno>
  <prijmeni>Šimek</prijmeni>
</osoba>
```

```
<osoba>Hurvínek</osoba>
```

Jmenné prostory

- Specifikace z roku 2002
- Kvalifikované jméno
  = jméno spojené se jmenným prostorem

{https://vscht.cz/schema}osoba

- Prefix nebo defaultní jmenný prostor
- Jmenné prostory mohou být použity jako "lightweight" semantika

Prefix

```
<vscht:osoba
  xmlns:vscht="https://vscht.cz/schema">

  <jmeno>Mirek</jmeno>
  <prijmeni>Šimek</prijmeni>
</vscht:osoba>

<osoba
  xmlns="https://pohadky.cz/schema">
  Hurvínek
</osoba>
```

Bez prefixu, defaultní jmenný prostor

Jmenné prostory — JSON

- JSON jmenné prostory nemá
- Můžeme nějak vyjádřit, že nějaká hodnota v JSONu něco znamená?

Jmenné prostory — JSON

- JSON jmenné prostory nemá
- Můžeme nějak vyjádřit, že nějaká hodnota v JSONu něco znamená?
- Ošklivé řešení

```
{
    "https://vscht.cz/schema/jmeno": "Mirek",
    "https://vscht.cz/schema/prijmeni": "Šimek"
}
```

JSON-LD intro

- Zaveďme kontext a v něm nadefinujme, jak vypadají kvalifikovaná jména. Dokument může nadále vypadat v původní podobě
- Kontext může být vložen přímo v dokumentu nebo odkázán přes URL

```
{
    "@context": {
        "autor": "https://vscht.cz/schema/autor",
        "jmeno": "https://vscht.cz/schema/jmeno",
        "prijmeni": "https://vscht.cz/schema/prijmeni"
    },
    "autor": {
        "jmeno": "Mirek",
        "prijmeni": "Šimek"
    }
}
```

```
{
    "@context": "https://vscht.cz/contexts/publication",
    "autor": {
        "jmeno": "Mirek",
        "prijmeni": "Šimek"
    }
}
```

JSON-LD intro

- Program, který neumí json-ld ignoruje "@context" a čte zbytek

```
{
    "@context": {
        "autor": "https://vscht.cz/schema/autor",
        "jmeno": "https://vscht.cz/schema/jmeno",
        "prijmeni": "https://vscht.cz/schema/prijmeni"
    },
    "autor": {
        "jmeno": "Mirek",
        "prijmeni": "Šimek"
    }
}
```

- Program, který json-ld rozumí, pak vidí plně kvalifikovaná jména

```
{
    "https://vscht.cz/schema/autor": {
        "https://vscht.cz/schema/jmeno": "Mirek",
        "https://vscht.cz/schema/prijmeni": "Šimek"
    }
}
```

JSON-LD — kontext zvládá víc

- Můžeme zkracovat hodnoty z běžných schemat

```
{
  "@context": {
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "autor": {
      "@id": "https://vscht.cz/schema/autor",
      "@type": "http://schema.org/Person"
    },
    "foaf": "http://xmlns.com/foaf/0.1/"
  },
  "autor": {
    "foaf:name": "Mirek Simek"
  }
}
```

```
{
  "https://vscht.cz/schema/autor": {
    "http://xmlns.com/foaf/0.1/name": "Mirek Simek"
  }
}
```

JSON-LD — kontext zvládá víc

- V kontextu může být u každého elementu typ - jako tady, kde je jméno explicitně označeno jako string.

```
{
  "@context": {
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "autor": "https://vscht.cz/schema/autor",
    "jmeno": {
      "@id": "https://vscht.cz/schema/jmeno",
      "@type": "xsd:string"
    },
    "prijmeni": "https://vscht.cz/schema/prijmeni"
  },
  "autor": {
    "jmeno": "Mirek",
    "prijmeni": "Šimek"
  }
}
```

- Program, který json-ld rozumí, pak ví přesně, jaký typ hodnoty má očekávat

```
{
  "https://vscht.cz/schema/autor": {
    "https://vscht.cz/schema/jmeno": {
      "@type": "http://www.w3.org/2001/XMLSchema#string",
      "@value": "Mirek"
    },
    "https://vscht.cz/schema/prijmeni": "Šimek"
  }
}
```

JSON-LD — kontext zvládá víc

- Kontext může vyjadřovat i "semantiku" - například můžeme říct, že autor je ve skutečnosti třída Person ze schema.org, jmeno je givenName a příjmení je familyName

```
{
  "@context": {
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "autor": {
      "@id": "https://vscht.cz/schema/autor",
      "@type": "http://schema.org/Person"
    },
    "jmeno": "http://schema.org/givenName",
    "prijmeni": "http://schema.org/familyName"
  },
  "autor": {
    "jmeno": "Mirek",
    "prijmeni": "Šimek"
  }
}
```

```
{
  "https://vscht.cz/schema/autor": {
    "http://schema.org/familyName": "Šimek",
    "http://schema.org/givenName": "Mirek"
  }
}
```

JSON-LD — zvládá víc

- Můžeme specifikovat i jazyky - jazyk celého json-ld dokumentu, jazyk pro konkrétní vlastnost, jazyk pro konkrétní hodnotu

```
{
  "@context": {
    "@vocab": "https://vscht.cz/",
    "@language": "en",
    "title_cs": {
        "@language": "cs"
    }
  },
  "title": "Hello world",
  "title_cs": "Ahoj světe",
  "title_fr": {
    "@value": "Salut!",
    "@language": "fr"
  }
}
```

```
{
  "https://vscht.cz/title": {
    "@language": "en",
    "@value": "Hello world"
  },
  "https://vscht.cz/title_cs": {
    "@language": "cs",
    "@value": "Ahoj světe"
  },
  "https://vscht.cz/title_fr": {
    "@language": "fr",
    "@value": "Salut!"
  }
}
```

JSON-LD — zvládá víc

- Reference

```
{
 "@context": {
  "@vocab": "http://repo.cz/context#"
 },
 "@id": "https://repo.cz/records/1234",

 "measurements": [
  {"@id": "https://repo.cz/records/1234#1", "name": "M 1"},
  {"@id": "https://repo.cz/records/1234#2", "name": "M 2"}
 ],

 "datasets": [
  {
     "name": "first dataset",
     "measurements": [
        {"@id": "https://repo.cz/records/1234#1"}
     ]
  }, …
 ]
}
```

JSON-LD shrnutí

- Uzly jsou IRI
- Uzel má @id (případně může mít automaticky generované)

```
{
 "@id": "https://repo.cz/records/1234",
 "http://repo.cz/context#datasets": {
  "http://repo.cz/context#measurements": {
   "@id": "https://repo.cz/records/1234#1"
  },
  "http://repo.cz/context#name": "first dataset"
 },
 "http://repo.cz/context#measurements": [
  {
   "@id": "https://repo.cz/records/1234#1",
   "http://repo.cz/context#name": "M 1"
  },
  {
   "@id": "https://repo.cz/records/1234#2",
   "http://repo.cz/context#name": "M 2"
  }
 ]
}
```

JSON-LD shrnutí

- Uzly jsou IRI
- Uzel má @id (případně může mít automaticky generované)
- Můžeme identifikovat jednoduché věty

"https://repo.cz/records/1234#1" — "http://repo.cz/context#name" — "M 1"

```
{
 "@id": "https://repo.cz/records/1234",
 "http://repo.cz/context#datasets": {
  "http://repo.cz/context#measurements": {
   "@id": "https://repo.cz/records/1234#1"
  },
  "http://repo.cz/context#name": "first dataset"
 },
 "http://repo.cz/context#measurements": [
  {
   "@id": "https://repo.cz/records/1234#1",
   "http://repo.cz/context#name": "M 1"
  },
  {
   "@id": "https://repo.cz/records/1234#2",
   "http://repo.cz/context#name": "M 2"
  }
 ]
}
```

JSON-LD shrnutí

=> json-ld je způsob zápisu RDF

```
{
 "@id": "https://repo.cz/records/1234",
 "http://repo.cz/context#datasets": {
  "http://repo.cz/context#measurements": {
   "@id": "https://repo.cz/records/1234#1"
  },
  "http://repo.cz/context#name": "first dataset"
 },
 "http://repo.cz/context#measurements": [
  {
   "@id": "https://repo.cz/records/1234#1",
   "http://repo.cz/context#name": "M 1"
  },
  {
   "@id": "https://repo.cz/records/1234#2",
   "http://repo.cz/context#name": "M 2"
  }
 ]
}
```

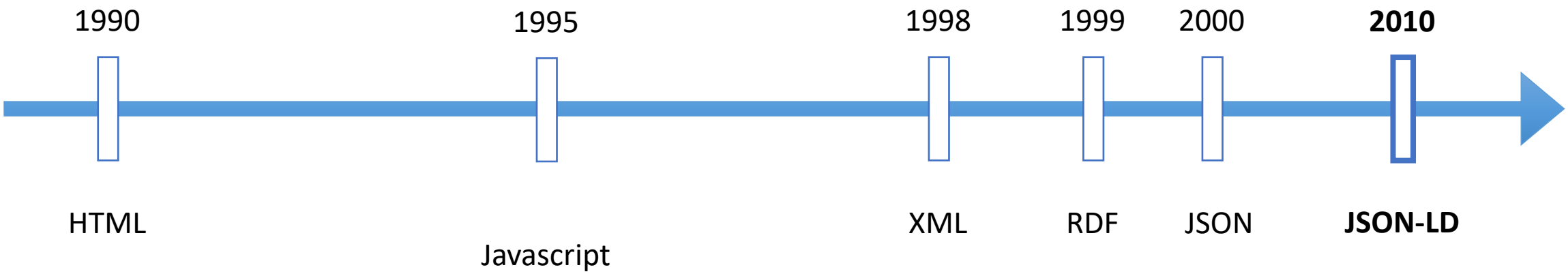
Subject	Predicate	Object
_:b0	http://repo.cz/context#measurements	https://repo.cz/records/1234#1
_:b0	http://repo.cz/context#name	first dataset
https://repo.cz/records/1234	http://repo.cz/context#datasets	_:b0
https://repo.cz/records/1234	http://repo.cz/context#measurements	https://repo.cz/records/1234#1
https://repo.cz/records/1234	http://repo.cz/context#measurements	https://repo.cz/records/1234#2
https://repo.cz/records/1234#1	http://repo.cz/context#name	Measurement 1
https://repo.cz/records/1234#2	http://repo.cz/context#name	Measurement 2

JSON-LD a RDF

Because the two are intimately compatible. Says Gregg Kellogg [1], one of the authors of JSON-LD:

> The fact that people don't know that it's RDF under the hood, that's great, you know. So what? It's not about RDF. It's about the solution. I think JSON-LD really serves that quite well, because it follows the natural progression of the way that web applications are developed. It just works.

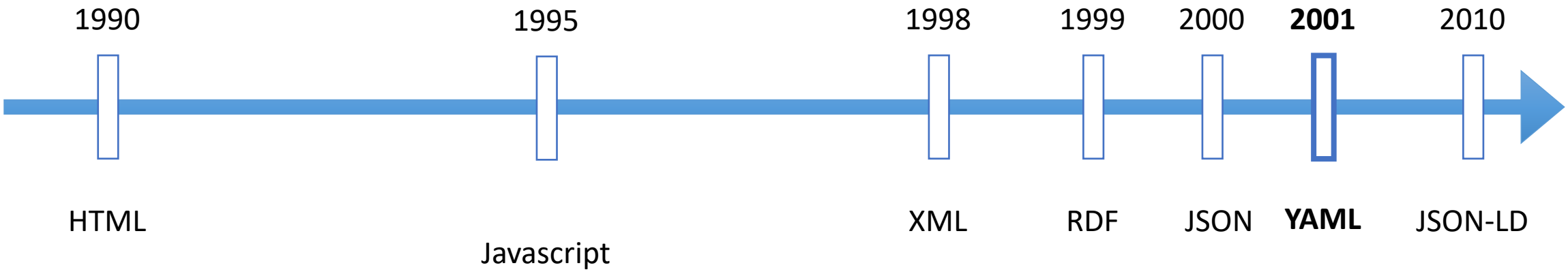
http://www.seoskeptic.com/what-is-json-ld/



JSON-LD a Národní repozitářová platforma

- Národní metadatový adresář bude exportovat JSON-LD
- Kontext bude nadefinován metadatovou skupinou
- Repozitáře vytvářené CESNETem budou exportovat JSON-LD

YAML



- Yet Another Markup Language
- YAML Ain't Markup Language
- Lidsky čitelný ale zároveň strojově zpracovatelný

YAML

- Každá hodnota na novém řádku
- Hierarchie vyjádřená zanořením
- Hodnota seznamu s pomlčkou na začátku
- Komentář je za #
- Automaticky detekuje čísla, texty, pravdivostní hodnoty (true, false)

```yaml
autor:
  jmeno: Mirek
  prijmeni: Simek
  zajmy:   # neni uplne pravda
   - cteni
   - psani
   - pocitani
```

YAML - použití

- Tam, kde chceme strojově zpracovatelný formát, který ale budou primárně editovat lidé
- Například konfigurace systémů
- YAML není koncipováno pro komunikaci mezi systémy - na to je příliš pomalé oproti JSON

YAML - národní repozitářová platforma

- V rámci repozitářů vytvářených CESNETem je YAML použit:
  - pro definici kontrolovaných slovníků (je na výběr - excel nebo YAML)
  - pro popis metadatových schemat

```yaml
properties:
  title:
    type: fulltext+keyword # Main (original) title of the object/work.
    required: true
    label.cs: Název
    label.en: Title

  additionalTitles[]:
    # Additional titles of the object/work such as subtitle, translate
    ^label.cs: Další názvy
    ^label.en: Additional titles
    ^uniqueItems: true
    properties:
      title:
        type: i18nStr
        required: true
      titleType:
        type: keyword
        required: true
        label.cs: Druh názvu
        label.en: Title type
        enum: [translatedTitle, alternativeTitle, subtitle, other]
```

Odkazy

- https://www.youtube.com/playlist?list=PLEzQf147-uEoNCeDlRrXv6ClsLDN-HtNm
- https://www.ecma-international.org/publications-and-standards/standards/ecma-404/
- https://json-ld.org/playground/
- https://json-ld.org/learn.html
- https://yaml.org/