



# Označni jezici

Interoperabilnost informacijskih sustava

Akademski godina 2025./2026.

Od XML-a i HTML-a do JSON-a, YAML-a i TOML-a – sve što trebate znati o jezicima koji pokreću razmjenu podataka

# Sadržaj predavanja

1.

## Osnove označnih jezika

Što su, zašto postoje, kako rade

2.

## Skupine i vrste označnih jezika

Od SGML-a do modernih formata

3.

## Označni jezik XML

Struktura, pravila, prostori naziva

4.

## Validacija XML dokumenata

DTD, XSD, RELAX NG, Schematron

5.

## Rukovanje XML podacima

DOM, SAX, StAX, XPath, XSLT, XQuery

6.

## Moderni formati podataka

JSON, YAML, TOML, Markdown – novi igrači

7.

## Primjene u praksi

RSS, Atom, e-Račun, ODF, OOXML i više



# 1. Osnove označnih jezika

Što su označni jezici i zašto su važni za interoperabilnost?

# Što su označni jezici?



## Označavanje (markup)

Znakovni niz koji se umeće u tekst kako bi se označio određeni dio teksta i razlikovao od ostatka.

"Marking up" – označavanje teksta dodatnim oznakama (recenzenti, urednici, lektori...)



## Označni jezik (markup language)

Sustav oznaka nad tekстом kojime se cjeline sintaktički odjeljuju od drugih.

Skup konvencija – dozvoljene oznake, njihovo značenje i pravila korištenja.



## Označni jezici i interoperabilnost

Omogućuju razmjenu podataka između sustava u točno definiranom obliku koji obje strane razumiju – na razini prijenosa podataka i na razini razumijevanja smisla (semantike) prenesenih informacija.



[XML & HTML Explained for Beginners \(Fireship\)](#)



[Isprobaj HTML uživo: codepen.io/pen](https://codepen.io/pen)

# Najpoznatiji primjer: HTML

## HTML (HyperText Markup Language)

- Temelj svakog web dokumenta
- Definira strukturu stranice
- HTML5 – aktualna verzija (2014+)
- Semantički elementi: <header>, <nav>, <article>, <section>, <footer>
- Novi API-ji: Canvas, WebGL, Web Storage, Geolocation, Web Workers...

```
<!DOCTYPE html>
<html lang="hr">
<head>
  <meta charset="UTF-8">
  <title>Moja stranica</title>
</head>
<body>
  <header>
    <h1>Pozdrav, svijete!</h1>
  </header>
  <main>
    <article>
      <p>Ovo je HTML5 dokument.</p>
    </article>
  </main>
  <footer>&copy; 2026</footer>
</body>
</html>
```



[HTML in 100 Seconds \(Fireship\)](#)



[DEMO: HTML5 primjer na CodePen](#)



## 2. Skupine i vrste označnih jezika

Pregled stotina označnih jezika različite namjene

# Skupine označnih jezika

Danas se aktivno koristi nekoliko stotina označnih jezika – vrlo važan oblik interoperabilnosti na globalnoj razini.

## Opće namjene

GML, SGML, XML, YAML,  
JSON, TOML, EBML

## Dokumenti

HTML5, XHTML, ODF,  
OOXML, Markdown, LaTeX

## Sindikacija

RSS, Atom,  
JSON Feed

## Korisničko sučelje

XAML, SwiftUI DSL,  
Jetpack Compose

## Vektorska grafika

SVG, X3D,  
glTF

## Web usluge

WSDL, SOAP, GraphQL  
Schema, OpenAPI



[Theory - Markup Languages](#)

# Označni jezici opće namjene

Služe za općeniti opis oznaka i koriste se kao osnova izgradnje drugih jezika.

<b>GML</b>	1969.	IBM – začetnik svih označnih jezika
<b>SGML</b>	1986.	ISO 8879 – metajezik, roditelj HTML-a i XML-a
<b>XML</b>	1998.	W3C – najrašireniji, podskup SGML-a
<b>JSON</b>	2001.	Douglas Crockford – lagan format za razmjenu podataka
<b>YAML</b>	2001.	"YAML Ain't Markup Language" – konfiguracija, CI/CD
<b>TOML</b>	2013.	Tom Preston-Werner – konfiguracija (Rust, Python)



**NOVO** u 2026: JSON i YAML sve više zamjenjuju XML u modernim aplikacijama, dok TOML postaje standard za konfiguraciju (Cargo.toml, pyproject.toml).



# Označni jezici dokumenata

Najrasprostranjeniji i najšire prihvaćeni označni jezici u svakodnevnom korištenju.

**HTML5 / XHTML**

Web stranice – temelj interneta

**Markdown**

README, dokumentacija, blogovi  
(GitHub, Notion)

**AsciiDoc**

Tehnička dokumentacija, knjige

**LaTeX / TeX**

Znanstveni radovi, matematika

**ODF**

Open Document Format –  
LibreOffice

**OOXML**

Office Open XML – Microsoft Office

**DocBook**

Strukturirana tehnička  
dokumentacija

**reStructuredText**

Python dokumentacija (Sphinx)

**Wikitext**

Wikipedia i slične platforme

**JSON-LD**

Linked Data za semantički web



[Markdown Crash Course \(Traversy Media\)](#)



[DEMO: Markdown Editor Online](#)

# Tri glavne vrste označnih jezika

## Prezentacijski

Oznake koje oblikuju konačni izgled dokumenta.

Najčešće sakrivene od korisnika.

Primjer: WYSIWYG uređivači, Word, HTML+CSS

## Proceduralni

Oznake koje upravljaju obradom teksta.

Pokreću akcije nad tekstualnim podacima.

Primjer: TeX, troff, PostScript

## Opisni

Označavanje u strukturnom i semantičkom smislu.

Opis značenja, ne izgleda.

Primjer: XML, HTML5 semantički elementi



Moderni trend: opisni jezici dominiraju jer odvajaju strukturu od prezentacije, što omogućuje bolju interoperabilnost i ponovnu upotrebu podataka.

# Označni jezik SGML

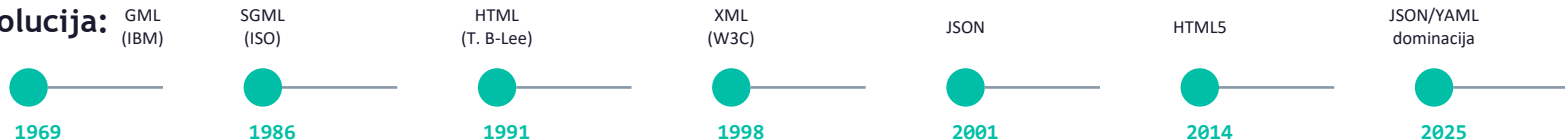
## SGML (Standard Generalized Markup Language)

- Međunarodna norma za predstavljanje tekstualnih informacija u elektroničkom obliku
- Metajezik – način formalnog opisa označnog jezika
- Neovisno o uređaju ili sustavu
- Roditelj XML-a, HTML-a i mnogih drugih

## Dijelovi SGML dokumenta:

1. Deklaracija SGML
2. Prolog (DOCTYPE, DTD)
3. Instanca (vršni element)

## Evolucija:



[SGML HTML XML What's the Difference?](#)



# 3. Označni jezik XML

Extensible Markup Language – temelj interoperabilnosti

# XML - Extensible Markup Language

## Ključne značajke XML-a

- Odvaja podatke od prezentacije
- Omogućuje dijeljenje i razmjenu podataka
- Pojednostavljuje prijenos između platformi
- Sadrži metapodatke koji opisuju sadržaj
- Omogućava semantičku interoperabilnost
- Preporuka W3C, podskup SGML-a
- Čitljiv ljudima i strojevima

```
<?xml version="1.0" encoding="UTF-8"?>
<imenik>
  <osoba oib="12345678901">
    <ime>Ana</ime>
    <prezime>Anić</prezime>
    <email>ana@example.hr</email>
    <godinaRodjenja>1999</godinaRodjenja>
  </osoba>
  <osoba oib="23456789012">
    <ime>Ivo</ime>
    <prezime>Ivić</prezime>
    <email>ivo@example.hr</email>
  </osoba>
</imenik>
```

Inačice: XML 1.0 (Fifth Ed., 2008.) – preporučena • XML 1.1 (Second Ed., 2006.) – Unicode, kontrolni znakovi



# Struktura i pravila XML dokumenta

## Prolog

```
<?xml version="1.0"
      encoding="UTF-8"
      standalone="yes"?>

<!-- Ovo je komentar -->

<?xml-stylesheet type="text/xsl"
      href="stil.xsl"?>
```

## Elementi i atributi

```
<datum format="ISO8601">
  2026-03-01
</datum>

<prazan />

<![CDATA[ Tekst s < > & ]]>
```

## Dobra oblikovanost (well-formed)

- Samo jedan korijenski element
- Elementi pravilno zatvoreni
- Pravilno gniježđeni
- Nazivi osjetljivi na velika/mala slova
- Vrijednosti atributa u navodnicima
- Samo dopušteni znakovi

## Prostori naziva (namespaces)

- Izbjegavanje konflikata naziva
- Jedinstveno imenovanje elemenata
- Ponovno korištenje postojećih schema
- Validacija s višestrukim shemama



## 4. Validacija XML dokumenata

DTD, XSD, RELAX NG, Schematron – osigurajmo ispravnost podataka

# Validacija XML dokumenata - pregled

Provjera ispravnosti XML dokumenta se može provoditi na više razina i različitim metodama.

## DTD

Document Type  
Definition

Najstariji pristup.  
Deklaracije oznaka,  
struktura, tipovi.

## XSD

XML Schema  
Definition

W3C preporuka.  
Rich tipovi podataka,  
prostori naziva.

## RELAX NG

REgular LAnguage  
for XML NG

ISO norma. Dva oblika:  
XML (.rng) i kompaktni  
(.rnc).

## Schematron

Rule-based  
validation

ISO 19757-3:2020.  
Tvrdnje i pravila,  
XPath izrazi.



Odnos: Schematron ima najveće mogućnosti, RELAX NG je "širi" od XSD, XSD moćniji od DTD. U praksi se kombiniraju – npr. XSD + Schematron.



# DTD - Document Type Definition

## DTD primjer - Imenik

```
<!DOCTYPE imenik [  
  <!ELEMENT imenik (osoba+)>  
  <!ELEMENT osoba  
    (((ime, prezime) | cijeloIme),  
    email?)>  
  <!ATTLIST osoba  
    oib CDATA #REQUIRED>  
  <!ELEMENT ime (#PCDATA)>  
  <!ELEMENT prezime (#PCDATA)>  
  <!ELEMENT cijeloIme (#PCDATA)>  
  <!ELEMENT email (#PCDATA)>  

```

## Značajke DTD-a

- Skup deklaracija oznaka
- Definira strukturu i tipove podataka
- Obaveznost elemenata
- Zastarjelo? Ne sasvim – nova verzija:
- **ISO/IEC 19757-3:2020 DSDL**
- Preporučuje se XSD ili RELAX NG

## Izraz:

Barem 1 osoba u imeniku, osoba mora imati OIB, razdvojeno ili spojeno ime/prezime, neobavezni email.



[DEMO: DTD Validator Online](#)

# XSD - XML Schema Definition

**XSD (XML Schema Definition)** – preporuka W3C-a. Definira strukturu, tipove podataka, ograničenja i prostore naziva XML dokumenata. Koristi se alternativni naziv WXS (W3C XML Schema).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=
"http://www.w3.org/2001/XMLSchema">
  <xs:element name="imenik">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="osoba"
          maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ime"
                  type="xs:string"/>
                <xs:element name="prezime"
                  type="xs:string"/>
                <xs:element name="email"
                  type="xs:string"
                  minOccurs="0"/>
              </xs:sequence>
              <xs:attribute name="oib"
                type="xs:string"
                use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

## Prednosti XSD-a

- Bogat sustav tipova podataka
- Podrška za prostore naziva
- Ograničenja (minOccurs, maxOccurs)
- Naslijeđivanje i proširenja
- Podrška u svim većim XML parserima
- Kombinira se sa Schematronom za potpunu validaciju



[DEMO: XSD Validator](#)

# RELAX NG - regularni izrazi za XML

## RELAX NG (XML oblik .rng)

```
<element name="imenik"
  xmlns="http://relaxng.org/
  ns/structure/1.0">
  <oneOrMore>
    <element name="osoba">
      <attribute name="oib">
        <text/>
      </attribute>
      <choice>
        <group>
          <element name="ime"><text/></element>
          <element name="prezime"><text/>
          </element>
        </group>
        <element name="cijeloIme"><text/>
        </element>
      </choice>
      <optional>
        <element name="email"><text/>
        </element>
      </optional>
    </element>
  </oneOrMore>
</element>
```

## Compact Syntax (.rnc)

```
element imenik {
  element osoba {
    attribute oib { text },
    ( ( element ime { text },
      element prezime { text }
    ) |
      element cijeloIme { text }
    ),
    element email { text }?
  }+
}
```

### Značajke:

- ISO/IEC 19757-2:2003
- Jednostavniji od XSD-a
- Koristi se u ODF-u, DocBook-u
- Dva oblika: XML i kompaktni

# Schematron - validacija pravilima

## Schematron primjer - Imenik

```
<schema xmlns="http://purl.oclc.org/dsdl/schematron">
  <title>Imenik</title>
  <pattern name="all">
    <rule context="/">
      <assert test="imenik">
        Korijenski element: imenik.
      </assert>
    </rule>
    <rule context="imenik">
      <assert
        test="count(*)=count(osoba)">
        Djeca: samo osoba.
      </assert>
    </rule>
    <rule context="osoba">
      <assert test="@oib">
        Osoba mora imati OIB.
      </assert>
      <assert test="count(ime)=1">
        Točno jedno ime.
      </assert>
    </rule>
  </pattern>
</schema>
```

## Ključne značajke

- Obrnuta logika od gramatika
- **Što nije zabranjeno – valjano je**
- Temelji se na XPath izrazima
- Izveden povrh XSLT
- ISO/IEC 19757-3:2020
- Kombinira se s XSD/RELAX NG

## Najmoćniji pristup validaciji!

Može izraziti pravila poput:  
"Ako je spol 'muški', dijagnoza ne smije biti 'trudan' – nemoguće u DTD/XSD!"



[DEMO: Schematron online validator](#)



## 5. Rukovanje XML podacima

DOM, SAX, StAX, XPath, XSLT, XQuery – obrada i transformacija

# DOM vs SAX vs StAX - parsiranje XML-a

## DOM

*Tree-based*

Cijeli dokument u memoriji  
XPath podrška  
Čitanje, pisanje, brisanje  
Idealno za manje dokumente

## SAX

*Event-based (push)*

Sekvencijalno, jednosmjerno  
Malo memorije  
Brže parsiranje  
Za pronalaženje elemenata

## StAX

*Streaming (pull)*

Klijent kontrolira dretvu  
Manja biblioteka  
Filtriranje sadržaja  
SOAP, WSDL podrška

Značajka	DOM	SAX	StAX
XPath	Da	Ne	Ne
Zapis XML	Da	Ne	Da
Memorija	Visoka	Niska	Niska
CRUD	Da	Ne	Ne



[SAX DOM Parsers](#)

# XPath - adresiranje dijelova XML-a

## XPath izrazi - primjeri

```
# Sve osobe
/imenik/osoba

# Ime prve osobe
/imenik/osoba[1]/ime

# Osobe rođene prije 1990.
/imenik/osoba[ godinaRodjenja<1990]

# Email svih osoba
//osoba/email
```

## Verzije XPath-a

- XPath 1.0 (1999.) – široko rasprostranjen
- XPath 2.0 (2007.) – bogatiji tipovi, ugrađen u XSLT 2.0
- XPath 3.0 (2014.) – dinamički pozivi funkcija, inline funkcije
- **XPath 3.1 (2017.) – mape i polja, JSON podrška**

Koristi se u: XSLT, XQuery, XForms, Selenium, web scraping, Java, C#, JavaScript...

XPath je za XML ono što su CSS selektori za HTML – moćan alat za navigaciju i odabir čvorova u strukturi dokumenta.



[XPath Tutorial](#)



[DEMO: XPath Tester Online](#)

# XSLT - transformacije XML dokumenata

## XSLT primjer: XML → HTML tablica

```
<?xml version="1.0,, encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
    <body>
      <h1>Imenik</h1>
      <table border="1">
        <tr>
          <th>Ime</th>
          <th>Prezime</th>
          <th>Email</th>
        </tr>
        <xsl:for-each
          select="imenik/osoba">
          <tr>
            <td>
              <xsl:value-of select="ime"/>
            </td>
            <td>
              <xsl:value-of
                select="prezime"/>
            </td>
            <td>
              <xsl:value-of
                select="email"/>
            </td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

## XSLT verzije

- XSLT 1.0 (1999.) – široko prihvaćena
- XSLT 2.0 (2007.) – grupiranja, funkcije
- **XSLT 3.0 (2017.) – streaming, JSON podrška, mape**

## Procesori:

Saxon, Xalan, xsltproc, MSXML, Sablotron...

Mogu se izvršavati na klijentu (preglednik) ili poslužitelju.



[DEMO: XSLT Online Transformer](#)



[XSLT Beginner Tutorial with Demo](#)



# XQuery - upiti nad XML podacima

## XQuery primjeri

```
(: Pronađi osobe mlađe od 1990. :)  
doc("podaci.xml")/imenik  
  /osoba[ godinaRodjenja<1990]  
  
(: FLWOR izraz :)  
for $a in doc("podaci.xml")  
  /imenik/osoba  
where $a/godinaRodjenja < 1990  
order by $a/prezime  
return $a/ime
```

## XQuery → HTML

```
<table>  
  <tr><th>IME</th>  
    <th>PREZIME</th></tr>  
  {  
    for $a in  
      doc("podaci.xml")/imenik/osoba  
    return  
      <tr>  
        <td>{data($a/ime)}</td>  
        <td>{data($a/prezime)}</td>  
      </tr>  
  }  
</table>
```

## XQuery je za XML ono što je SQL za baze podataka

- XQuery 1.0 (2007.) – preporuka W3C, razvijano paralelno s XSLT 2.0
- XQuery 3.0 (2014.) – grupiranje, try/catch, switch, inline funkcije
- **XQuery 3.1 (2017.) – podrška za JSON i XML, mape i polja**
- Izvedbe: Saxon, MarkLogic, BaseX, eXist-db, RaptorXML (60+ implementacija)





## 6. Moderni formati podataka

JSON, YAML, TOML, Markdown, GraphQL – novi igrači u svijetu interoperabilnosti

# JSON - JavaScript Object Notation

## JSON primjer

```
{
  "imenik": {
    "osobe": [
      {
        "oib": "12345678901",
        "ime": "Ana",
        "prezime": "Anić",
        "email": "ana@example.hr",
        "godine": 27,
        "aktivan": true
      },
      {
        "oib": "23456789012",
        "ime": "Ivo",
        "prezime": "Ivić",
        "email": "ivo@example.hr",
        "godine": 35,
        "aktivan": false
      }
    ]
  }
}
```

## Zašto JSON dominira?

- Lagan i čitljiv format
- Nativna podrška u JavaScriptu
- Standard za REST API-je
- Podržava: objekte, polja, stringove, brojeve, boolean, null
- **JSON Schema za validaciju**
- JSON-LD za semantički web
- RFC 8259 (2017.)

## JSON vs XML

JSON je 30-50% kraći od ekvivalentnog XML-a.  
Nema prostora naziva, komentara ni atributa.  
Nema validacijske sheme kao XSD  
(ali ima JSON Schema!).



[Learn JSON in 10 Minutes](#)



[DEMO: JSON Schema Validator](#)



[DEMO: JSON Editor Online](#)

# YAML - YAML Ain't Markup Language

## YAML primjer

```
# Konfiguracija aplikacije
app:
  name: "Moja aplikacija"
  version: "2.0"
  debug: true

database:
  host: localhost
  port: 5432
  credentials:
    username: admin
    password: tajno123

serveri:
  - name: web-01
    role: frontend
    port: 8080
  - name: api-01
    role: backend
    port: 3000

deploy:
  environment: production
  replicas: 3
```

## Gdje se koristi YAML?

- Docker Compose konfiguracija
- Kubernetes manifesti
- GitHub Actions / GitLab CI/CD
- Ansible playbook-ovi
- Spring Boot (application.yml)
- OpenAPI / Swagger specifikacije



## Pazi na uvlačenje!

YAML koristi uvlačenje (indentaciju) za strukturu – razmaci, NIKAD tabovi! Česta greška kod početnika.



[DEMO: YAML Validator](#)



[YAML Tutorial \(TechWorld with Nana\)](#)

# TOML - Tom's Obvious, Minimal Language

## TOML primjer

```
# Cargo.toml (Rust projekt)
[package]
name = "moja-aplikacija"
version = "0.1.0"
edition = "2021"
authors = ["Ana Anić <ana@hr>"]

[dependencies]
serde = { version = "1.0",
  features = ["derive"] }
tokio = { version = "1",
  features = ["full"] }

[profile.release]
opt-level = 3

# pyproject.toml (Python)
[project]
name = "moj-paket"
version = "1.0.0"
requires-python = ">=3.10"

[tool.pytest.ini_options]
testpaths = ["tests"]
```

## Zašto TOML?

- Minimalan i čitljiv
- **EksPLICITNI tipovi (datum, vrijeme!)**
- Nedvosmislen za parsere
- Standard za Rust (Cargo.toml)
- Standard za Python (pyproject.toml)
- Hugo, Go projekti

## TOML 1.0 (2021.) - stabilna specifikacija

Kreirao Tom Preston-Werner (suosnivač GitHuba). Cilj: spoj čitljivosti YAML-a i strogosti JSON-a, bez zamke indentacije.



[DEMO: TOML Validator](#)



[Data Formats: XML, JSON, and YAML](#)

# Usporedba: XML vs JSON vs YAML vs TOML

Značajka	XML	JSON	YAML	TOML
Čitljivost	Srednja	Dobra	Odlična	Odlična
Opširnost	Visoka	Srednja	Niska	Niska
Komentari	Da	Ne*	Da	Da
Tipovi podat.	Tekst	6 tipova	Automatski	Eksplcitni
Validacija	DTD/XSD/RNG	JSON Schema	Ograničena	Ograničena
Namespaces	Da	Ne	Ne	Ne
Primarna namjena	Dokumenti, razmjena	API-ji, web	Konfiguracija, DevOps	Konfiguracija, paketi
Godina	1998.	2001.	2001.	2013.

\* JSON5 i JSONC podržavaju komentare



2025. trend: JSON za API-je, YAML za DevOps/Kubernetes, TOML za konfiguraciju (Rust, Python), XML za enterprise sustave i dokumente.

# Markdown - lagan označni jezik za dokumentaciju

## Markdown primjer

```
# Naslov 1
## Naslov 2

Tekst s bold i italic.

- Lista stavki
- Druga stavka
  - Pod-stavka

```python
def pozdrav(ime):
    return f"Bok, {ime}!"
```

[Link](https://example.com)
![Slika](slika.png)

Stupac 1	Stupac 2
Podatak	Podatak
```

## Gdje se koristi?

- GitHub README.md
- Notion, Obsidian, Joplin
- Dokumentacija (MkDocs, Docusaurus)
- Blogovi (Hugo, Jekyll, Gatsby)
- Discord, Slack, Reddit poruke
- Jupyter Notebook ćelije
- **CommonMark standard (2019.)**

Varijante: GitHub Flavored Markdown (GFM), MultiMarkdown, Markdown Extra, R Markdown



[DEMO: Markdown Live Preview](#)



[Markdown Crash Course \(Traversy Media\)](#)

# SVG i GraphQL - specijalizirani označni jezici

## SVG - Scalable Vector Graphics

```
<svg width="200" height="200"
  xmlns="http://www.w3.org/
    2000/svg">
  <circle cx="100" cy="100" r="80"
    fill="#00BFA6"
    stroke="#0D1B2A"
    stroke-width="3"/>
  <text x="100" y="110"
    text-anchor="middle"
    fill="white"
    font-size="24">
    SVG!
  </text>
</svg>
```

- XML-baziran format za 2D vektorsku grafiku
- Animacije, filteri, interaktivnost
- **D3.js** – vizualizacija podataka putem SVG-a

## GraphQL - jezik upita za API-je

```
# Schema
type Osoba {
  oib: String!
  ime: String!
  prezime: String!
  email: String
}

type Query {
  osobe: [Osoba!]!
  osoba(oib: String!): Osoba
}

# Upit
query {
  osobe {
    ime
    prezime
    email
  }
}
```

- Razvio Facebook (2015.), open-source
- Klijent traži točno ono što treba
- **Alternativa REST API-jima**



[DEMO: GraphQL Playground](#)



[GraphQL in 100 Seconds \(Fireship\)](#)





## 7. Primjene označnih jezika u praksi

RSS, Atom, e-Račun, ODF, OOXML, OpenAPI i druge primjene

# RSS i Atom - sindikacija sadržaja

## RSS (Really Simple Syndication)

```
<?xml version="1.0"
  encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Moj Blog</title>
    <link>https://blog.hr</link>
    <description>Novosti</description>
    <item>
      <title>Novi post</title>
      <link>https://blog.hr/1</link>
      <pubDate>
        Mon, 01 Mar 2026
      </pubDate>
    </item>
  </channel>
</rss>
```

## JSON Feed - moderna alternativa

```
{
  "version":
    "https://jsonfeed.org/v1.1",
  "title": "Moj Blog",
  "home_page_url":
    "https://blog.hr",
  "items": [
    {
      "id": "1",
      "title": "Novi post",
      "url": "https://blog.hr/1",
      "date_published":
        "2026-03-01T18:00:00Z"
    }
  ]
}
```

**Atom** (RFC 4287, 2005.) – naprednija alternativa RSS-u, XML-bazirana. **JSON Feed** (2017.) – moderna alternativa u JSON formatu, jednostavnija za web developere. Sve tri tehnologije su još uvijek aktivno korištene – RSS u podcast platformama, Atom u GitHub feedovima, JSON Feed u modernim blogovima.



# Poslovne primjene označnih jezika

## e-Račun

Elektronički račun u RH (CII/UBL XML format)

*Obavezno za B2G u EU od 2024.*

## UBL

Universal Business Language – ISO/IEC 19845

*Narudžbe, računi, otpremnice...*

## ebXML

Electronic Business XML – UN/CEFACT + OASIS

*Infrastruktura e-poslovanja*

## ODF

Open Document Format – ISO/IEC 26300

*LibreOffice, Google Docs*

## OOXML

Office Open XML – ISO/IEC 29500

*Microsoft Office formati*

## OpenAPI

Specifikacija za REST API-je (bivši Swagger)

*YAML/JSON opis API endpointova*



[OpenAPI/Swagger in 10 min \(APIs You Won't Hate\)](#)



[DEMO: Swagger Editor](#)

# Preglednici i DOM u 2026. godini

## DOM - Document Object Model

- Konvencija za interakciju s objektima označnih jezika (HTML, XML...)
- **DOM Level 4 (Living Standard)** – aktualna specifikacija
- JavaScript koristi DOM kao osnovni način komuniciranja s HTML-om
- Web Components, Shadow DOM – moderni pristup

```
// Moderni DOM API (2026.)  
document.querySelector('.card')?.classList.toggle('active');
```

## Preglednici - layout engines

Engine	Preglednici
Blink	Chrome, Edge, Opera, Brave, Samsung
WebKit	Safari, iOS svi preglednici
Gecko	Firefox, Thunderbird

*Blink (Chromium) dominira s ~70% tržišnog udjela u 2026.*












Moderni pristup: Virtual DOM (React), Incremental DOM (Angular), Signals (Solid.js, Angular 17+, Svelte 5) – novi paradigmi za učinkovit rad s DOM-om.









Izvor: [gs.statcounter.com/browser-market-share](https://gs.statcounter.com/browser-market-share)

# Praktični resursi i alati za učenje

## Online demo alati

-  [CodePen – HTML/CSS/JS playground](#)
-  [JSFiddle – Web eksperimenti](#)
-  [XML Validator Online](#)
-  [JSON Editor Online](#)
-  [YAML Lint – provjera YAML-a](#)
-  [XSLT Online Transformer](#)
-  [XPath Tester](#)
-  [GraphQL Playground](#)
-  [Swagger Editor – OpenAPI](#)
-  [Markdown Live Preview](#)

## YouTube kanali i videa

-  [HTML in 100 Seconds](#)
-  [JSON in 100 Seconds](#)
-  [YAML Tutorial \(Nana\)](#)
-  [GraphQL in 100 Seconds](#)
-  [XML Tutorial \(Beginner\)](#)
-  [SVG in 100 Seconds](#)
-  [Markdown Crash Course](#)
-  [REST API vs GraphQL](#)
-  [What is RSS? \(Explained\)](#)
-  [XSLT Tutorial](#)

# Sažetak - ključne poruke

- 1 Označni jezici su temelj interoperabilnosti – omogućuju komunikaciju između sustava na sintaksoj i semantičkoj razini.
- 2 XML ostaje relevantan u enterprise sustavima, ali JSON, YAML i TOML dominiraju u modernim aplikacijama.
- 3 Validacija je ključna – od jednostavnog DTD-a do moćnog Schematrona, svaki pristup ima svoju svrhu.
- 4 Moderni alati (XPath 3.1, XQuery 3.1, XSLT 3.0) podržavaju i JSON, čime se premošćuje jaz između XML i JSON ekosustava.
- 5 Markdown, GraphQL, OpenAPI, JSON Feed – novi formati koji definiraju kako komuniciramo i razmjenjujemo podatke u 2026. godini.

Domaća zadaća: Ponovite gradivo HTML, XML i JSON. Isprobajte demo alate s prethodnog slajda. Usporedite isti podatak zapisan u XML, JSON, YAML i TOML formatu.

# Pitanja?

Hvala na pažnji!