Otázky podle kategorií

RDF

1. RDF:List

- Funguje jako spojový list.
- Koukám do `first` (value), když chci druhou hodnotu, jdu do `rest` a zase `first`.
- Končí `null` (`nil`).

Příklad:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:list1 rdf:first "Hodnota1" ;
       rdf:rest [
           rdf:first "Hodnota2" ;
           rdf:rest rdf:nil
       ] .
```

2. Co je to reifikace?

- Proces přidávání metadat k RDF statementům (trojicím).

Původní trojice:

```
my:index.html my:createdBy "Jakub Klímek" .
```

Reifikace:

```
_:triple1 a rdf:Statement ;
          rdf:subject my:index.html ;
          rdf:predicate my:createdBy ;
          rdf:object "Jakub Klímek" .

_:triple1 my:createdAt "2025-01-16" .
```

3. Popište RDF*. Uveďte příklad.

- Rozšíření RDF.
- Řeší přidávání metadat k RDF statementům (trojicím).
- `<<..>>` označuje citovanou trojici (quoted triple).

Příklad:

```
my:index.html my:createdBy "Jakub Klímek" .

<< my:index.html my:createdBy "Jakub Klímek" >>
    dcterms:source "https://x.y.z"^^xsd:anyURI ;
    dcterms:created "2020-04-23"^^xsd:date .
```

4. Jak se liší prefixovaná a relativní IRI v kontextu RDF?

- **Prefixovaná IRI**:
  - Používá `@prefix` pro definici zkratek.
  - Zkracuje dlouhá IRI pomocí aliasů.
- **Relativní IRI**:
  - Používá `@base` k definici základního IRI pro relativní odkazy.

Příklad:

```
@prefix foo: <http://example.org/ns#> .
@base <http://newbase.com/> .

<#document> foo:createdBy "Jakub Klímek" .
```

- `<#document>`: Rozšíří se na `http://newbase.com/#document`.
- `foo:createdBy`: Přeloží se na `http://example.org/ns#createdBy`.

5. Co je Linked Data Vocabularies?

- **Ontologie**: Sbírka vlastností a tříd pro popis entit a vztahů mezi nimi.
- Zajišťují interoperabilitu mezi různými datovými systémy.

Příklady vokabulářů: FOAF, Dublin Core, Schema.org, SKOS.

6. Co je Open World Assumption?

- Pokud něco není v záznamu, znamená to "nevím" (není to bráno jako nepravda).
- Opačný přístup: uzavřený svět (Closed World Assumption).

7. Co je SERVICE pro SPARQL?

- Klauzule pro dotazování vzdálených SPARQL endpointů (multiendpointové dotazy).

Příklad:

```
SELECT ?name ?birthPlace
WHERE {
```

```
  ?person a dbo:Person ;
        foaf:name ?name .
  SERVICE <http://dbpedia.org/sparql> {
    ?person dbo:birthPlace ?birthPlace .
  }
}
```

8. Vysvětlete SKOS:exactMatch a použijte na příkladu.

- Použití pro linkování ekvivalentních entit mezi různými slovníky (např. "auťák" = "auto").
- **Tranzitivní**: Pokud A = B a B = C, pak A = C.

Příklad:

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ex1: <http://example.org/vocabulary1/> .
@prefix ex2: <http://example.org/vocabulary2/> .

ex1:Jazz
    a skos:Concept ;
    skos:prefLabel "Jazz"@en ;
    skos:exactMatch ex2:JazzGenre .

ex2:JazzGenre
    a skos:Concept ;
    skos:prefLabel "Jazz Music"@en .
```

BONUS

9. Prázdné uzly RDF

- Resource bez URI (lokální scope).
- Použití, pokud:
  - Neznám nebo nechci použít IRI.
  - Potřebuji přidat data k objektu lokálně.

10. Serializace RDF

- **Formáty:** N-Triples, RDF Turtle, N-Quads (podpora pojmenovaných grafů).

Wikidata

1. Vlastnosti tvrzení v Wikidatech?

- **Statement** obsahuje:
  - Property (vlastnost).

- Value (hodnotu).
- Rank (důležitost).
- Qualifiers (upřesnění).
- References (zdroje).

2. Co je QID ve Wikidatech?

- Jedinečný identifikátor pro entity na Wikidatech.
- Např. Douglas Adams: Q42.

Cypher & LPG

1. Co je set v Cypher?

- Klíčové slovo pro modifikaci vlastností uzlů nebo hran.

Příklad:

```
MATCH (n:Person {name: "John"})
SET n.age = 30
RETURN n
```

2. V čem "exceluje" LPG?

- Popisuje metadata vztahů mezi entitami.
- Obsahuje grafové algoritmy.

Příklad:

```
(:Person {name: "John", age: 30})-[:KNOWS]->(:Person {name: "Alice", age: 25})
```

XML/XPath/XSLT

1. Co je mode v XSLT?

- Atribut v šabloně pro odlišení zpracování stejných elementů různými způsoby.

Příklad:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <!-- Šablona pro "mode1" -->
  <xsl:template match="item" mode="mode1">
    <xsl:value-of select="name"/> - Mode 1
```

```
  </xsl:template>

  <!-- Šablona pro "mode2" -->
  <xsl:template match="item" mode="mode2">
    <xsl:value-of select="name"/> - Mode 2
  </xsl:template>

  <xsl:template match="/">
    <html>
      <body>
        <h2>Items in Mode 1:</h2>
        <xsl:apply-templates select="items/item" mode="mode1"/>

        <h2>Items in Mode 2:</h2>
        <xsl:apply-templates select="items/item" mode="mode2"/>
      </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

2. Uveďte příklad osy v XPath.

- `descendant-or-self`: Vrátí aktuální uzel a všechny jeho poduzly, které splňují dotaz.

Příklad:

```
/books/book/descendant-or-self::chapter
```

3. Vysvětlete XPath osu `attribute::`.

- Vrátí atributy aktuálního uzlu.

Příklad:

```
/book/@title
```

4. (a) Rozdíl mezi jednoduchým a komplexním typem v XML Schema.

- `simpleType`: Nemá subelementy ani atributy, obsahuje pouze text.
- `complexType`: Může obsahovat subelementy, atributy a text.

5. (b) Simple a complex content v XML Schema.

- `simpleContent`: Textový obsah s atributy, ale bez subelementů.
- `complexContent`: Kombinace subelementů a atributů.

6. Co je validní XML?

- XML, které odpovídá definovanému XML Schématu.
- Musí být také "well-formed" (splňovat syntaktická pravidla).

7. Popište XML DOM (Document Object Model)

- Reprezentuje XML dokument jako stromovou strukturu v paměti.
- Umožňuje náhodný přístup ke všem částem dokumentu (čtení, úpravy, mazání).
- Vhodné pro komplexní operace na celém dokumentu, ale může být náročné na paměť.

8. STAX (Streaming API for XML)

- Pull-based přístup: Dokument je parsován postupně, element po elementu.
- Efektivní pro práci s velkými dokumenty, protože nezatěžuje paměť jako DOM.
- Umožňuje řízení toku zpracování, zastaví se u požadovaného elementu.

JSON-LD

1. Co je keyword aliasing v JSON-LD?

- Definice vlastních aliasů pro klíčová slova v JSON-LD contextu.

Příklad:

```
{
  "@context": {
    "schema": "http://schema.org/",
    "name": "schema:name",
    "Person": "schema:Person",
    "PEPÍK": "@id"
  },
  "@type": "Person",
  "name": "John Doe",
  "PEPÍK": "http://example.org/person/1"
}
```

2. Uveďte 3 klíčová slova z JSON-LD.

- `@context`
- `@id`
- `@type`

3. Jakými třemi způsoby můžeme na JSON přidat JSON-LD kontext?

1. **Přímý kontext:**

```
{
  "@context": {
    "ex": "http://example.org/"
  },
  "@type": "Person",
  "ex:name": "John Doe"
}
```

2. **Externí URL kontext:**

```
{
  "@context": "http://schema.org",
  "@type": "Person",
  "name": "John Doe"
}
```

3. **Kombinovaný (scoped) kontext:**

```
{
  "@context": "http://schema.org",
  "name": "Matěj Foukal",
  "description": "Webový vývojář",
  "address": {
    "@context": {
      "postalCode": "http://schema.org/postalCode",
      "addressLocality": "http://schema.org/addressLocality"
    },
    "postalCode": "11000",
    "addressLocality": "Praha"
  }
}
```

4. Jak lze v JSON-LD zachovat pořadí hodnot v poli?

- Použitím klíčového slova `@list`.

Příklad:

```
{
  "@context": {
    "ex": "http://example.org/"
  },
  "ex:steps": {
    "@list": [
      "Step 1: Open the file",
      "Step 2: Edit the content",
```

```
        "Step 3: Save the file"
      ]
    }
  }
```

JSON Schema

1. Popište hlavní způsoby validace polí v JSON Schema.

1. **List validation:**
   - Validuje pole, kde všechny hodnoty mají stejný typ.

   Příklad:

   ```
   {
     "type": "array",
     "items": { "type": "string" }
   }
   ```

2. **Tuple validation:**
   - Validuje pole, kde každá položka má jiný typ a záleží na pořadí.

   Příklad:

   ```
   {
     "type": "array",
     "items": [
       { "type": "string" },
       { "type": "number" },
       { "type": "boolean" }
     ]
   }
   ```

2. Popište 3 validační klíčová slova v JSON Schema.

- `type`: Určuje datový typ (např. string, number).
- `required`: Specifikuje povinné vlastnosti.
- `enum`: Omezuje hodnoty na určitý seznam.

3. Popište 3 schématická klíčová slova v JSON Schema.

- `$schema`: Určuje verzi JSON Schema.
- `$id`: Identifikátor schématu.
- `$ref`: Odkazuje na jiné schéma nebo jeho část.

CSV

1. Popište CSV podle přesného RFC.

- Formát oddělený čárkami, zakončení řádků CRLF.
- Kódování UTF-8 bez BOM.
- Escapování hodnot pomocí dvojitých uvozovek (").
- Hlavička je volitelná, ale doporučená.

2. Jak pomocí URI zacílit na různé části CSV?

- Použitím fragmentového identifikátoru #.

Příklady:

```
http://example.org/data.csv#cell=5,2
http://example.org/data.csv#row=5
http://example.org/data.csv#column=3
```

3. Co je relační datový model v kontextu CSV on the Web?

- Základní jednotka: tabulka (sloupce, primární klíče).
- V JSON-LD lze nadefinovat primární a cizí klíče a vztahy mezi tabulkami.

Formáty pro grafiku a multimédia

1. Co je dithering?

- Metoda pro simulaci nedostupných barev smícháním blízkých barev.
- Používá se například v tisku nebo rastrové grafice.

2. Popište pixel/dot density. Čím je reprezentovaný?

- **PPI (Pixels Per Inch):** Metrika pro digitální obrazovky.
- **DPI (Dots Per Inch):** Metrika pro tisk.

3. Popište ztrátové a bezeztrátové kompresní metody pro rastrovou grafiku. Uveďte konkrétní formáty.

- **Bezeztrátová:**
  - Metody: Huffmanovo kódování, Run Length Encoding (RLE).
  - Formáty: PNG, BMP.
- **Ztrátová:**
  - Metody: Diskrétní kosinová transformace (DCT).
  - Formáty: JPEG.

4. Popište RGBA barevný model.

- **Red, Green, Blue, Alpha (průhlednost):** Aditivní barevný model, který určuje intenzitu světla jednotlivých složek.

5. Popište CMYK.

- **Cyan, Magenta, Yellow, Black:** Subtraktivní barevný model používaný v tisku.

6. Co je diskrétní kosinová transformace?

- Používá se v ztrátové kompresi (např. JPEG).
- Transformuje obrazová data z prostorové domény do frekvenční domény.

Zvukové formáty

1. Popište Pulse-Code Modulation v kontextu digitálního zvuku.

- **PCM:** Digitalizace analogového zvuku pomocí pravidelného vzorkování (sampling rate).
- Hodnoty amplitudy jsou kvantizovány na nejbližší digitální krok (bit depth).

Obecné technologie a koncepty

1. Co je Graph Query Language (GQL)?

- Jazyk pro dotazování nad grafovými databázemi.

Příklad:

```
MATCH (n)
RETURN n
```

2. Co je souřadnicový systém?

- Definice: Systém pro určování polohy objektů v prostoru.
- Příklad: WGS-84 (zeměpisné souřadnice).

3. Proč vznikl TeX?

- Pro profesionální sazbu dokumentů, zejména s matematickými vzorci.
- Nabízí precizní kontrolu nad formátováním textu.