## **Formát JSON vs. XML**

### **JSON – obecný popis, historie a důvod vzniku**

JSON (JavaScript Object Notation) je **lehký formát pro výměnu dat**, který je snadno čitelný pro lidi i stroje. **Vznikl jako alternativa k XML** pro jednodušší a rychlejší přenos dat mezi webovými aplikacemi a servery.

**Historie:**

* JSON se poprvé objevil kolem roku **2001** jako součást JavaScriptu.
* **Douglas Crockford** jej formalizoval jako standard.
* Díky své jednoduchosti a efektivitě se stal hlavním formátem v API a webových službách.

**Důvod vzniku JSON:**

* XML bylo těžkopádné a příliš složité pro jednoduché datové struktury.
* JSON je **jednodušší, kratší, rychlejší** na zpracování.
* **Lépe se používá v JavaScriptu**, protože má stejnou syntaxi jako objekty v JavaScriptu.

### **Porovnání JSON a XML na konkrétním příkladu**

Pokud si otevřeš [**https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45**](https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45), uvidíš data v JSON:

{  
 "results": {  
 "sunrise": "7:45:08 AM",  
 "sunset": "6:34:54 PM"  
 },  
 "status": "OK"  
}

Stejná data v XML by vypadala takto:

<response>  
 <results>  
 <sunrise>7:45:08 AM</sunrise>  
 <sunset>6:34:54 PM</sunset>  
 </results>  
 <status>OK</status>  
</response>

### **Proč je JSON vhodnější než XML?**

* **Je kratší** – méně znaků = rychlejší přenos.
* **Snadněji se čte a zapisuje** – jednoduchá struktura.
* **Lépe se zpracovává v JavaScriptu** – přímo kompatibilní.
* **Menší režie** – nepotřebuje značky <tag>, stačí klíče a hodnoty.

## **Linux přístupová práva**

V Linuxu má každý soubor a adresář **přístupová práva** a vlastníka.

### **Základní pojmy:**

* **Owner (vlastník)** – uživatel, který soubor vytvořil.
* **Group (skupina)** – skupina uživatelů, kteří mohou mít různá práva k souboru.
* **Others (ostatní)** – všichni ostatní uživatelé systému.

### **Jak jsou práva zobrazena ve výpisu souboru?**

Příkaz ls -l zobrazí:

-rwxr-xr-- 1 user group 1234 Jan 01 12:34 soubor.txt

Rozbor:

* **První znak (- nebo d)**:
  + - = soubor
  + d = adresář
* **Další devět znaků určují práva:**
  + rwx (práva vlastníka)
  + r-x (práva skupiny)
  + r-- (práva ostatních)
* **Vysvětlení:**
  + r (read) – čtení souboru
  + w (write) – zápis do souboru
  + x (execute) – spuštění souboru

### **Příklad nastavení práv**

Chceme vytvořit skript, který můžeme spouštět pouze my:

1. **Vytvoříme soubor:** echo "ls -al" > skript.sh
2. **Nastavíme mu práva:** chmod 700 skript.sh  
   1. **700 znamená:**
      1. **Vlastník může číst, zapisovat, spouštět (rwx).**
      2. **Nikdo jiný nemá žádná práva (---).**

### **Význam příkazů chmod**

* **chmod 600 soubor.txt** → **jen vlastník může číst a zapisovat, ostatní nic**.
* **chmod u-w soubor.txt** → **odebere vlastníkovi právo zapisovat**.
* **chmod ug=rwx soubor.txt** → **vlastník i skupina mají všechna práva, ostatní nic**.

## **Python - operátory**

### **Co jsou to operátory?**

Operátory jsou **speciální symboly**, které umožňují provádět operace nad hodnotami.

### **Rozdělení operátorů**

1. **Aritmetické operátory** (pro výpočty):
   1. + (sčítání)
   2. - (odčítání)
   3. \* (násobení)
   4. / (dělení)
   5. // (celé dělení)
   6. % (zbytek po dělení)
   7. \*\* (mocnina)
2. **Porovnávací operátory** (vrací True nebo False):
   1. == (rovná se)
   2. != (nerovná se)
   3. > (větší než)
   4. < (menší než)
   5. >= (větší nebo rovno)
   6. <= (menší nebo rovno)
3. **Logické operátory** (pro podmínky):
   1. and (a zároveň)
   2. or (nebo)
   3. not (negace)
4. **Přiřazovací operátory** (pro ukládání hodnot):
   1. = (přiřazení)
   2. += (přičtení hodnoty)
   3. -= (odečtení hodnoty)
   4. \*= (násobení)
   5. /= (dělení)
5. **Bitové operátory** (pracují s binárními čísly):
   1. & (bitový AND)
   2. | (bitový OR)
   3. ^ (bitový XOR)
   4. ~ (bitová negace)
   5. << (bitový posun doleva)
   6. >> (bitový posun doprava)
6. **Členské operátory** (ověřují přítomnost v sekvenci):
   1. in (je v seznamu)
   2. not in (není v seznamu)
7. **Identitní operátory** (porovnávají identitu objektů):
   1. is (odkazuje na stejný objekt)
   2. is not (neodkazuje na stejný objekt)

## **Shrnutí – co se naučit (tučné části)**

* **JSON** – co to je, proč vznikl, výhody oproti XML
* **Příklad JSON a XML – porovnání výstupu API**
* **Linux přístupová práva – owner, group, others, rwx**
* **Výpis práv v ls -l, příklad chmod 700**
* **Co dělají příkazy chmod 600, chmod u-w, chmod ug=rwx**
* **Python operátory – rozdělení do skupin a příklady**

S tímhle materiálem bys měl být schopný mluvit o tématu 15 minut.