**Zpracování a parsování textých dat, regulární výrazy, kódování a stringy**

**Parsování textu**

ČESKY – **SYNTATICKÁ ANALÝZA TEXTU**

Analýza textového řetězce pro zisk námi potřebných dat

Data se převádí do strukturovaného formátu pro snadnější zpracování (jedna z možností)

Nejčastější metoda parsování textu jsou REGEXY

Příklady dat, které se často **parsují**:

* Soubory s konfigurací
* Logy
* Webové stránky
* Uživatelský vstup

**Co je to REGEX?**

**Sekvence znaků, která definuje pattern pro vyhledávání nebo manipulování v textovým řetězci**

Dokáže výrazně zjednodušit zpracování textových dat

Je velice populární a využívá se prakticky všude

Je podporovatelný mnohými programovacími jazyky

* **Ne všechny mají stejnou syntaxi na REGEX**
* Základy jsou ale v podstatě stejné

|  |  |  |  |
| --- | --- | --- | --- |
| **Znak** | **Co se hledá** | Příklad | Shodu najde v: |
| ^ | Shoda na začátku | ^pt | pt, ptkkdfjfk, ptrkm, ptyfdfdf, atd. |
| $ | Shoda na konci | pt$ | Jsapt,ptlsdkpt, ukrajinapt, atd |
| . | Shoda jakéhokoliv charakteru | k.c | kfc, kasasasac, k:csasasasa |
| […] | Shoda čehokoliv v závorkách | [koc] | „k“ a „o“ a „c“ |
| {x} | Najde přesně x-krát shodu | S{2} | SS,SSkassjad,laslSSsasl |
| + | Shoda znaku před + vícekrát | a+c | ac,aac,aaaaaaaac |
| ? | Shoda před ? Nulakrát nebo jednou | ab?c | ac,abc |

**Použití REGEXU v PYTHON:**

A screenshot of a computer

Description automatically generated with medium confidence



**REGEX FLAGS (Vlajky)**

* Ke klasickým regexům si můžeme také pomoct přidáním VLAJKY
* Vlajky jsou parametry, který dodají nějakou vlastnost vyhledávání
* I – ignorace „case sensitive“
* M – shoda ve více řádcích
* Atd…
* Jako příklad jsme k minulému vyhledávání pouze přidali vlajku pro rozlišení velkých a malých písmen a výsledné vyhledávání se nám změnilo





**REGEX Cheat sheet**

(je dobré se v něm orientovat, na maturitě dostaneme nejspíše identický, pokud se vylosuje tato otázka)

A picture containing text, screenshot, menu, website

Description automatically generated

**Speciální znaky**

Speciální znaky v programovacích jazycích jsou využívány pro formátování, výpis omezených znaků (v stringu), ověření vstupu, encoding a decoding.

Python speciální znaky:

A screenshot of a computer

Description automatically generated with low confidence A screenshot of a computer

Description automatically generated with medium confidence

**String**

**Co je to string?**

String je datový typ reprezentující textový řetězec – ukládá znaky.

Je obvykle prezentován jako text v uzavřených uvozovkách (jednoduchých nebo dvojtých)

Má většinou svoje vlastní operace:

* Spojování
* Porovnávání
* Vyhledávání

**STRING v PYTHON:**

* Je označován jak uvozovkami jedoduchými tak i složenými
* „MULTILINE STRING“ za použití trojtých uvozovek jde string přenést na více řádků

A picture containing text, font, screenshot

Description automatically generated

* Má BUILD-IN metody
* len()
* lower()
* upper()
* strip() – odstraní mezery na začátku a na konci stringu
* split() – rozdělí string na základě delimetru
* Může se jim iterovat jako s listem
* Každý znak má svůj index na kterým je pevně daný
* Pomocí indexu se můžeme dotázat na třeba 5 prvek
* Můžeme řetězcem iterovat pomocí FOR
* Operování s metodou IN
* Lze se dotázat, zda ve stringu existuje daný string

A picture containing text, font, screenshot, line

Description automatically generated

* Toto vypíše TRUE

**Kódování**

**Co to je?**

**Proces konvertování stringu do sekvence bytes**

Důvod:

* Přesné a efektivní definování pro počítač (počítač rozumí pouze binárně)
* Uložení a přenos dat přes síť

Reprezentace stringu záleží na typu kódování

Standartní kódování – 1 byte (8 bit)

Pro další nestandartní znaky se využívaly více bajtové

To přivádí k dnešnímu stavu kódování

Nejpopulárnější způsoby kódování

* ASCII
* UNICODE
* UTF-8
* UTF-16

Příklad ASCII tabulky:

A screenshot of a computer

Description automatically generated with low confidence

UTF-8

Každému znaku (více než milion znaků) **je přiřazeno unikátní číslo**

**První BYTE v sekvenci** uvádí, **kolik BYTES celkově reprezentuje daný znak**

* Ostatní BYTES reprezentují daný znak
* Čím vyšší číslo – tím víc potřeba ke kódování bajtů

Byl původně vytvořen pro spolupoužití s ASCII

* Pokud WEB obsahuje UTF-8 znaky, ale systém podporuje pouze ASCII, tak se znaky v pořádku zobrazí

A black and white sign with white text

Description automatically generated with low confidence

A table with numbers and symbols

Description automatically generated with low confidence

UTF-16

Může reprezentovat absolutně všechny UNICODE znaky

Je víc kompaktní

Není podporovaný s ASCII