

Digital humanities

Životní cyklus digitálních dat

Jindřich Marek

Životní cyklus digitálních dat

- v humanitních vědách
1. Nízkoúrovňová reprezentace digitálních dat
 2. Formáty textových dat
 3. Životní cyklus dat

Co jsou data v kontextu DH?

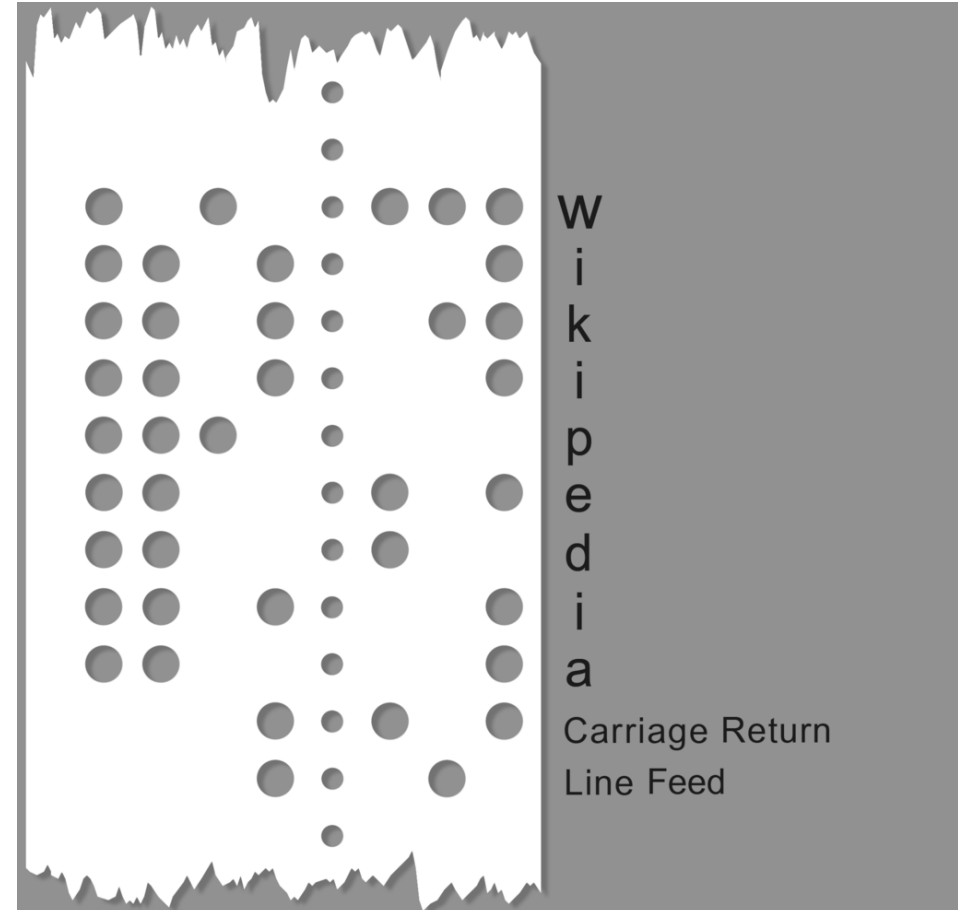
- čísla
 - celá čísla
 - desetinná čísla
- texty
- logická data
 - ano/ne
- datum a čas
- tabulární data
- multimédia
 - obrazy (bitové mapy, vektory)
 - video
 - audio
- prostorová data
 - mapy
- síťová data
- ...

1. Nízkoúrovňová reprezentace digit. dat

- počítače pracují s bity a byty
- jedničky a nuly

Text: kódování znaků

- reprezentace znaků v digitálním prostředí
 - jedničky a nuly
 - ASCII: W = 1010111
 - v desítkové soustavě 87
 - v šestnáctkové soustavě 57









Ein unentbehrlicher Reise-
begleiter ist die kleine

Erika

Aktiengesellschaft vorm.
Seidel & Naumann, Dresden

ASCII (1963)

- *American Standard Code for Information Interchange*
- 7 bitů na 1 byte
 - $W = 1010111$
- 128 znaků (pozice 0 až 127)
 - z toho 95 tisknutelných znaků
 - také kontrolní znaky (telekomunikace)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Changed or added in 1963 version
 Changed in both 1963 version and 1965 draft

ISO/IEC 8859 (1987)

- 8 bitů na 1 byte
 - W = 01010111
- 15 částí (tabulek) pro různé jazykové oblasti
- až 256 znaků (pozice 0 až 255)
- původně ECMA-94, 4 tabulky (1985)
- později podobné Windows code pages

ISO/IEC 8859-2 (Latin-2)																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x																
1x																
2x	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8x																
9x																
Ax	NBSP	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î
Bx	°	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
Cx	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß		
Dx	Đ	Ñ	Ń	Ó	Ô	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	
Ex	í	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	đ	ñ	ń	ó	ô	õ	ö	÷	ř	ů	ű	ű	ý	ÿ		

Unicode

- různé počty bitů na byte
- prvních 256 znaků odpovídá ISO/IEC 8859-1
 - W = 01010111
- verze 1.0 (1992)
 - všeob. přijetí mnohem později
- verze 17.0 (2025)
 - 159 801 znaků
 - 172 písem

- UTF-8
 - 1-4x 8 bitů pro jeden znak

Code point ↔ UTF-8 conversion

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0xxxxxxx			
U+0080	U+07FF	110xxxxx	10xxxxxx		
U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
U+10000	^[b] U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- kompatibilní s ISO/IEC 8859-1
- UTF-16
- (Čína:) GB 18030

(◐◑) SYM名L

Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00D0	00D1	00D2	00D3	00D4	00D5	00D6	00D7	00D8	00D9	00DA	00DB	00DC	00DD	00DE	00DF
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00E0	00E1	00E2	00E3	00E4	00E5	00E6	00E7	00E8	00E9	00EA	00EB	00EC	00ED	00EE	00EF
đ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
00F0	00F1	00F2	00F3	00F4	00F5	00F6	00F7	00F8	00F9	00FA	00FB	00FC	00FD	00FE	00FF
Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	010A	010B	010C	010D	010E	010F
Đ	đ	Ē	ē	Ĕ	ĕ	Ė	ė	Ę	ę	Ě	ě	Ĝ	ĝ	Ğ	ğ
0110	0111	0112	0113	0114	0115	0116	0117	0118	0119	011A	011B	011C	011D	011E	011F
Ġ	ġ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Ĩ	ĩ	Ī	ī	Ĭ	ĭ	Į	į
0120	0121	0122	0123	0124	0125	0126	0127	0128	0129	012A	012B	012C	012D	012E	012F
İ	ı	IJ	ij	Ĵ	ĵ	ķ	ķ	κ	Í	í	Ľ	Ľ	Ł	ł	Ł
0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	013A	013B	013C	013D	013E	013F
Į	į	Ń	ń	Ņ	ņ	Ň	ň	ʼ	Ŋ	ŋ	Ō	ō	Ŏ	ŏ	
0140	0141	0142	0143	0144	0145	0146	0147	0148	0149	014A	014B	014C	014D	014E	014F
Ŏ	ŏ	Œ	œ	Ŕ	ŕ	Ŗ	ŗ	Ř	ř	Ś	ś	Ŝ	ŝ	Ş	ş
0150	0151	0152	0153	0154	0155	0156	0157	0158	0159	015A	015B	015C	015D	015E	015F
Š	š	Ţ	ţ	Ť	ť	Ŧ	ŧ	Ũ	ũ	Ū	ū	Ŭ	ŭ	Ů	ů
0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	016A	016B	016C	016D	016E	016F
Ů	ů	Ų	ų	Ŵ	ŵ	Ŷ	ŷ	Ÿ	Ž	ž	Ž	ž	Ž	ž	ƀ
0170	0171	0172	0173	0174	0175	0176	0177	0178	0179	017A	017B	017C	017D	017E	017F

↑ Back to top

🔍 Search Unicode Blocks

0: Basic Multilingual Plane

Basic Latin 0000–007F

Latin-1 Supplement 0080–00FF

Latin Extended-A 0100–017F



Latin Extended-B 0180–024F

IPA Extensions 0250–02AF

Spacing Modifier Letters 02B0–02FF

Combining Diacritical Marks 0300–036F

Greek and Coptic 0370–03FF

Cyrillic 0400–04FF

Cyrillic Supplement 0500–052F

Armenian 0530–058F

Hebrew 0590–05FF

Arabic 0600–06FF

Syriac 0700–074F

Arabic Supplement 0750–077F

Thaana 0780–07BF

NKo 07C0–07FF

Samaritan 0800–083F

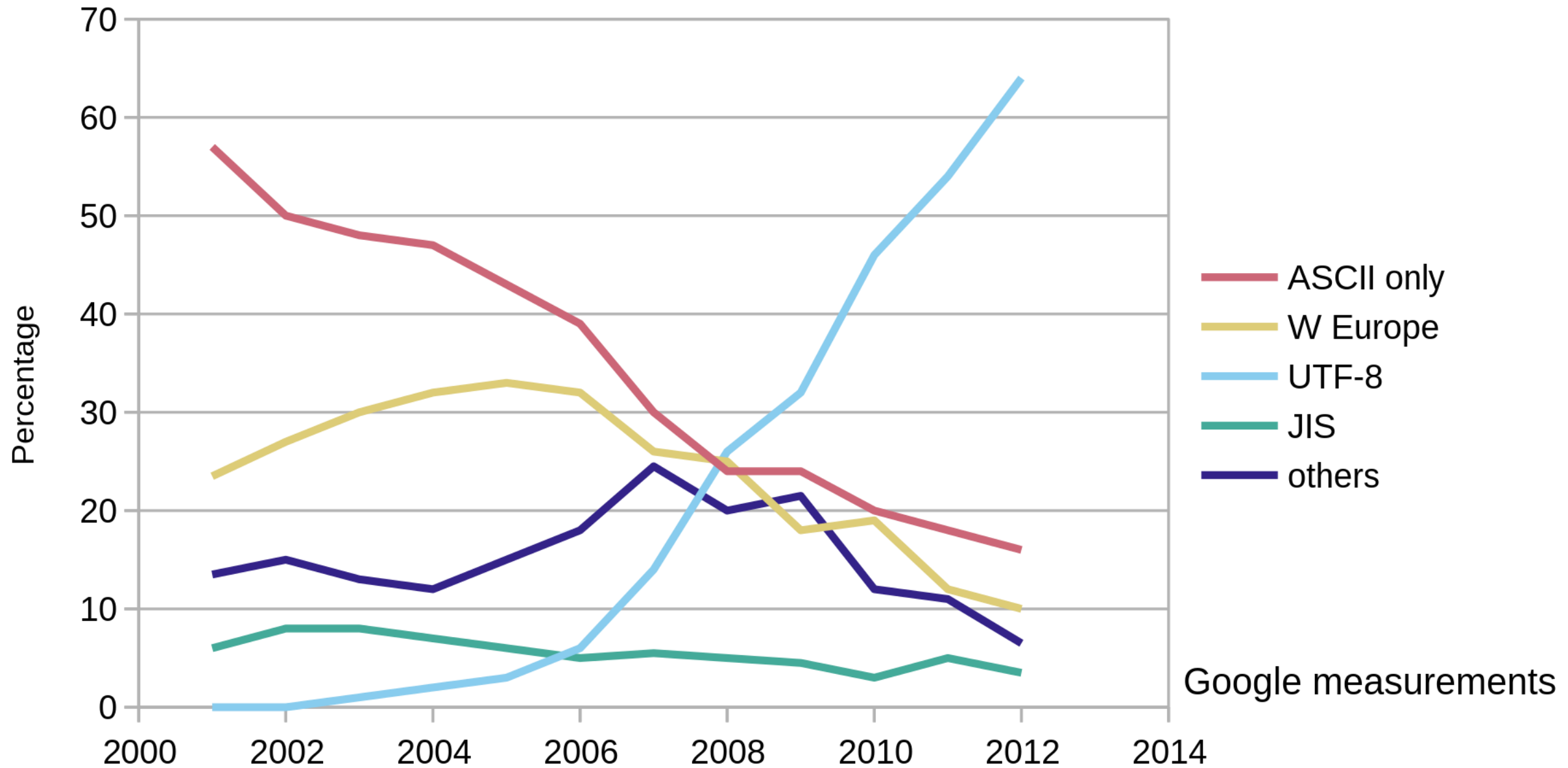
Mandaic 0840–085F

Syriac Supplement 0860–086F

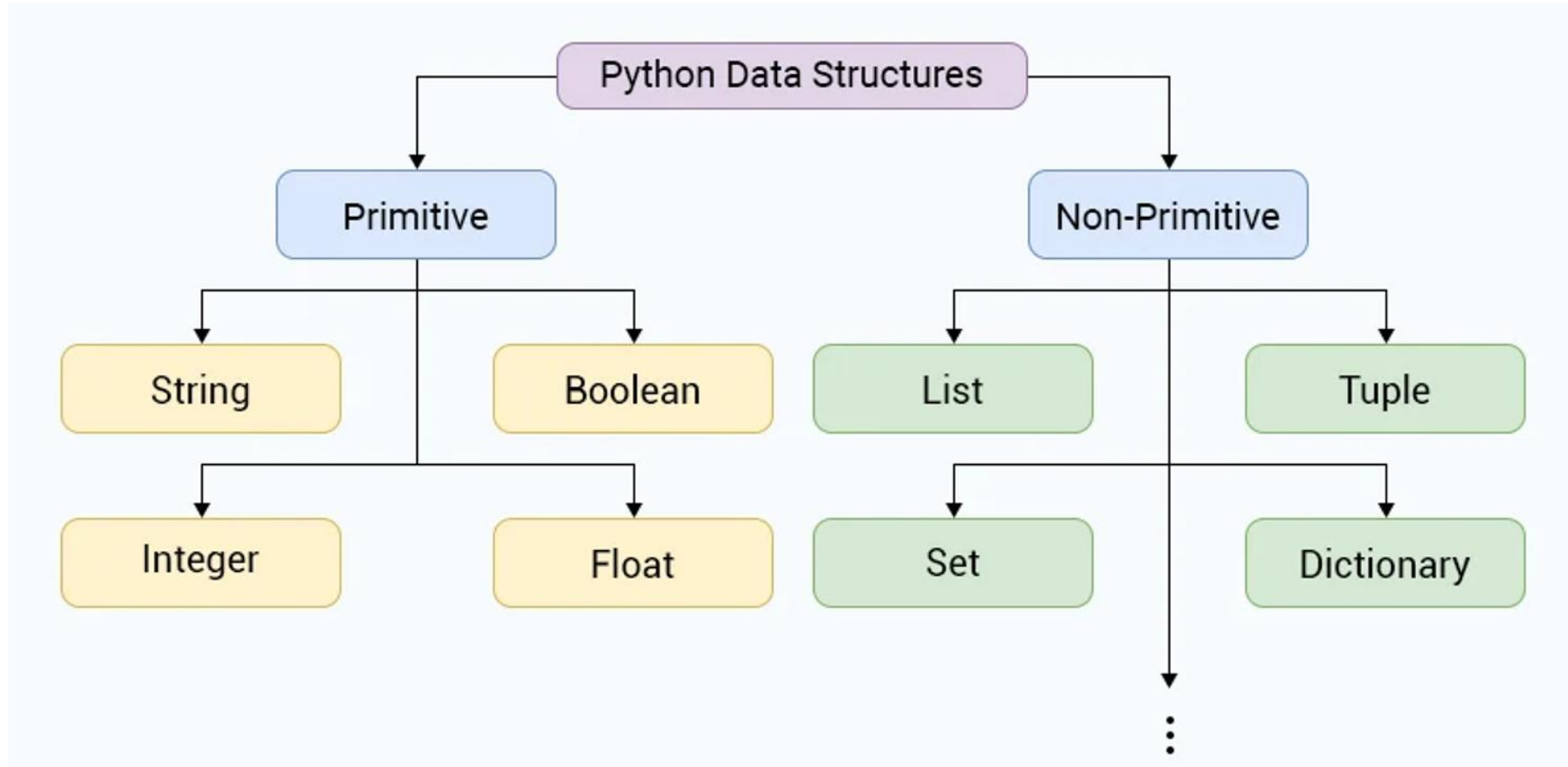
Arabic Extended-B 0870–089F

EN

Share of web pages with different encodings



Datové struktury v Pythonu



Python Primitive Data Structures

These store simple data values.

DESCRIPTION		EXAMPLES
String	Collection of characters surrounded by single or double quotation marks	'Alice', "Bob"
Boolean	Logical values	True, False
Integer	Whole number of unlimited length	0, -273
Float	Floating-point decimal	1.618, 3.1415926

Python Built-In Non-Primitive Data Structures

These data structures, which store values and collections of values, are inherent to Python.

DESCRIPTION	ORDERED	ALLOW DUPLICATES	MUTABLE	SYNTAX	EXAMPLES
List	✓	✓	✓	[]	• [1, 2.3, True] • ['John', 'Doe']
Tuple	✓	✓	×	()	• ('age', 22) • (7.89, False)
Set	×	×	×	{ }	• {6, 4.5} • {'nice', True}
Dictionary	✓ > 3.7	×	✓	{key: value}	• {"FB": "Facebook", "WA": "WhatsApp", "IG": "Instagram"} • {'name': 'Bob', 'id': 255}
Map, storing key-value pairs.	×	×			

Význam pro DH

- „médium je poselství“
- digitální materialita

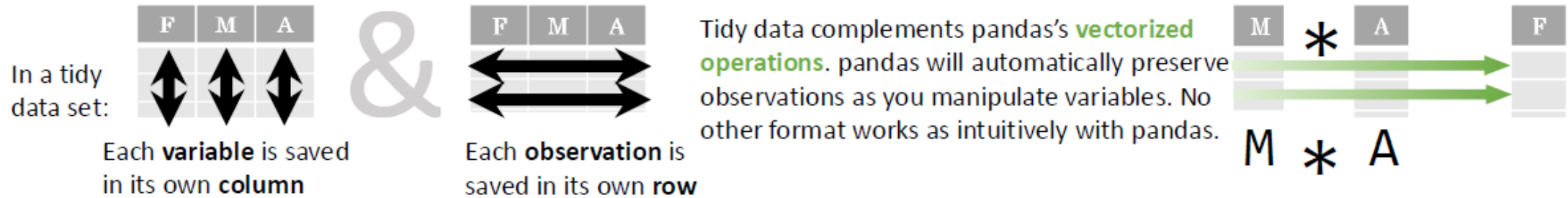
2. Formáty textových dat

kategorie	formáty	popis
tabulární data	CSV, TSV, ...	prostý text s oddělovači (čárka, tabulátor)
značkovací jazyky	XML, HTML, LaTeX, ...	strukturovaný text (se značkováním)
data s atributy	JSON, ...	hierarchická data (API, web)
formáty pro publikaci nebo výměnu	DOCX, PDF, ...	balíček XML dat (DOCX), popis strany (PDF), ...

2.1 Tabulární data

- reprezentace dat: čistá data (*tidy data*)
 - kanonická podoba tabulárních dat, cíl pro případné úpravy
- kvalita dat
 - přesnost, úplnost, konzistence, vhodnost, platnost, jedinečnost
- operace s daty: vylepšování kvality dat
 - „gramatika pro operace s daty“ (balíček dplyr pro R)
- *formální teoretické modely (relační algebra) pomíjíme*

Represent data: *tidy data*



[další info pro Python](#)

Tabulárními data: *data wrangling*

aspekt	funkce (pandas)	formální stránka
datový model	pd.DataFrame	relace + schéma
vývoj schématu	df.assign, df.drop, df.rename	sloupce: přidání, odebrání, přejm.
	df[[...]]	sloupce: výběr
filtry po řádcích	df.query, df[cond]	výběr
řazení hodnot	df.sort_values	uspořádání
výběr sloupců	df[['col1','col2']]	projekce
spojení více tabulek	df.merge, df.join	přirozené spojení
agregace/seskupování	df.groupby.agg	rozšířená relační algebra
přeskupení (dlouhý, krátký formát)	df.pivot, df.melt	transpozice matice + seskupování
úprava datových typů	df.replace, df.astype	transformace hodnot (bodově)
deduplikace	df.drop_duplicates	deduplikace (teorie množin)
práce s chybějícími hodnotami	df.fillna, df.dropna	rozšíření částečné funkce

široký (L) a dlouhý (R) formát tabulky

df3

	first	last	height	weight
0	John	Doe	5.5	130
1	Mary	Bo	6.0	150



df3.melt(id_vars=['first', 'last'])

	first	last	variable	value
0	John	Doe	height	5.5
1	Mary	Bo	height	6.0
2	John	Doe	weight	130
3	Mary	Bo	weight	150

2.2 a 2.3 Značkovací jazyky, data s atributy

- zejm. explicitní členění textu pomocí metadat
 - XML
 - HTML
 - JSON

The following JSON and XML examples both define an employees object, with an array of 3 employees:

JSON Example

```
{ "employees": [
  { "firstName": "John", "lastName": "Doe" },
  { "firstName": "Anna", "lastName": "Smith" },
  { "firstName": "Peter", "lastName": "Jones" }
]}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON is Like XML Because

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages
- Both JSON and XML can be fetched with an XMLHttpRequest

JSON is Unlike XML Because

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

3. Životní cyklus dat

1. Sběr
2. Vytváření
3. Obohacování
4. Analýza
5. Interpretace
6. Uložení
7. Komunikace

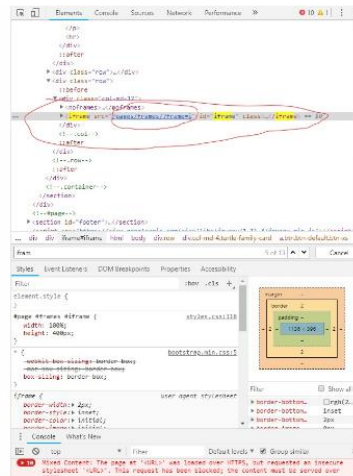
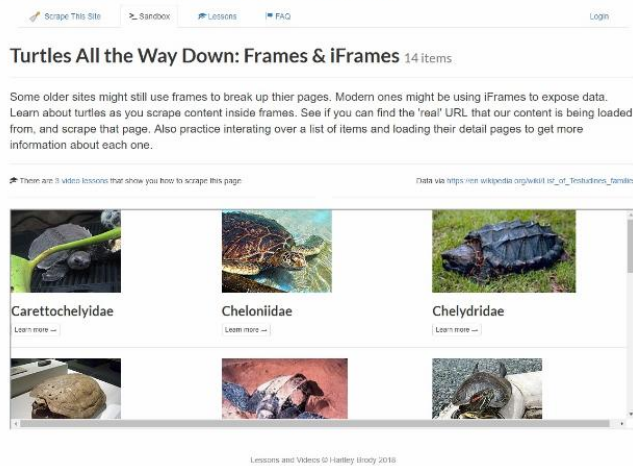
3.1. Sběr dat

- API
 - např. Europeana REST API
- web scraping
 - pozor na podmínky služeb a přetěžování serverů
- export z databází
 - nejčastější formáty: CSV, JSON, XML

Příklad: webscraping

2 Find the iframe

To extract turtles' name, we need to find the link to the iframe. Let's use Chrome Developer Tool to find the link to iframe: We can use ctrl+F and search for keyword "iframe" and here we find the link.



```
from requests_html import HTMLSession
from urllib.parse import urljoin
```

```
my_url = "https://scrapethissite.com/pages/frames/"
```

```
# extrahovat zdroj jako je ukázáno na obrázku výše
session = HTMLSession()
r = session.get(my_url)
iframe_src = r.html.find("#iframe", first=True).attrs['src']
```

```
# získat url k tomuto iframe
iframe_url = urljoin("https://scrapethissite.com", iframe_src)
```

```
# extrahovat jména želv:
r2 = session.get(iframe_url)
turtle_names = r2.html.find(".family-name")
turtle_names_text = [name.text for name in turtle_names]
```

```
print(turtle_names_text)
```

3.2 Vytváření dat

- čištění
- transformace
 - regulární výrazy (VS Code)
 - práce se sloupci (R, Python)
- úprava dat pomocí LLM

Příklad: čištění dat

- cíl: přehledná datová struktura
 - „tidy data“
- prostředky
 - rozdělení/spojení buněk
 - řešení chybějících hodnot
 - sjednocení údajů
 - oprava chyb

Tidy data



A data set is **tidy** iff:

1. Each **variable** is in its own **column**
2. Each **case** is in its own **row**
3. Each **value** is in its own **cell**

Příklad: čištění dat v OpenRefine

- dokumentace: <https://openrefine.org/docs>

Main features



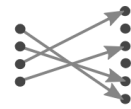
Faceting

Drill through large datasets using facets and apply operations on filtered views of your dataset.



Clustering

Fix inconsistencies by merging similar values thanks to powerful heuristics.



Reconciliation

Match your dataset to external databases via reconciliation services.



Infinite undo/redo

Rewind to any previous state of your dataset and replay your operation history on a new version of it.



Privacy

Your data is cleaned on your machine, not in some dubious data laundering cloud.



Wikibase

Contribute to Wikidata, the free knowledge base anyone can edit, and other Wikibase instances.

Příklad: úprava dat pomocí LLM

- prompt

Encode the following table with popular baby names of 2022 in JSON:

Rank	Male name		Female name
1	Liam	Olivia	
2	Noah	Emma	
3	Oliver	Charlotte	
4	James	Amelia	
5	Elijah	Sophia	
6	William	Isabella	
7	Henry	Ava	
8	Lucas	Mia	
9	Benjamin		Evelyn
10	Theodore		Luna

- odpověď

```
{
  "names_2022": {
    "male_names": [
      {"rank": 1, "name": "Liam"},
      {"rank": 2, "name": "Noah"},
      {"rank": 3, "name": "Oliver"},
      {"rank": 4, "name": "James"},
      {"rank": 5, "name": "Elijah"},
      {"rank": 6, "name": "William"},
      {"rank": 7, "name": "Henry"},
      {"rank": 8, "name": "Lucas"},
      {"rank": 9, "name": "Benjamin"},
      {"rank": 10, "name": "Theodore"}
    ],
    "female_names": [
      {"rank": 1, "name": "Olivia"},
      {"rank": 2, "name": "Emma"},
      {"rank": 3, "name": "Charlotte"},
      {"rank": 4, "name": "Amelia"},
      {"rank": 5, "name": "Sophia"},
      {"rank": 6, "name": "Isabella"},
      {"rank": 7, "name": "Ava"},
      {"rank": 8, "name": "Mia"},
      {"rank": 9, "name": "Evelyn"},
      {"rank": 10, "name": "Luna"}
    ]
  }
}
```

3.3 Obohacování dat

- reconciling (OpenRefine)
- geolokace
- NER (named entity recognition)
- textové korpusy: anotace
 - tokenizace, lemmatizace, rozpoznání slovních druhů

3.4 Analýza a interpretace dat

- výpočty (tabulky s čísly)
- vizualizace dat

Příklad: explorativní analýza

- základní statistika
 - počty, distribuce
- vizualizace
 - časová řada, geografická mapa
- příp. identifikace anomálií

3.5. Interpretace dat

- výzkumná otázka: co data říkají o historickém/kulturním jevu?
 - vždy vzhledem ke kontextu

3.6. Uložení dat

- GitHub
- Zenodo

Featured communities



Biodiversity Literature Repository

A community to share publications related to bio-systematics.

[Browse](#)

Recent uploads

March 13, 2024 (v1)

Presentation

Open

NOAA NESDIS Operational Sea Surface Temperatures (SSTs)

Baker-Yeboah, Sheekela; Casey, Kenneth; Harris, Andy

This presentation was given by Sheekela Baker-Yeboah on March 7th, 2024 for a GHR SST Talk. Find the event details and the recording of the GHR SST Talk on the GHR SST Website here.

Uploaded on March 13, 2024

Part of [Group for High Resolution Sea Surface Temperature \(GHR SST\)](#)

0 0

March 13, 2024 (2024-03-13)

Dataset

Open

SARS-CoV-2 Sequenzdaten aus Deutschland

Robert Koch-Institut

Ein zentraler Bestandteil einer erfolgreichen Erregersurveillance ist das Verständnis der Verbreitung eines Erregers sowie seiner pathogenen Eigenschaften. Hierbei stellt das

Why use Zenodo?

- **Safe** — your research is stored safely for the future in CERN's Data Centre for as long as CERN exists.
- **Trusted** — built and operated by CERN and OpenAIRE to ensure that everyone can join in Open Science.
- **Citeable** — every upload is assigned a Digital Object Identifier (DOI), to make them citable and trackable.
- **No waiting time** — Uploads are made available online as soon as you hit publish, and your DOI is registered within seconds.
- **Open or closed** — Share e.g. anonymized clinical trial data with only medical professionals via our restricted access mode.
- **Versioning** — Easily update your dataset with our versioning feature.
- **GitHub integration** — Easily preserve your GitHub repository in Zenodo.
- **Usage statistics** — All uploads display standards compliant usage statistics

3.7. Komunikace dat

- patří sem i zpřístupnění v jiném formátu

Příklad: export dat z Jupyter Lab

1. vestavěné možnosti

- Python Script
- HTML
- PDF (vyžaduje instalaci LaTeXu)

2. nbconvert (z příkazové řádky)

- HTML
- LaTeX
- PDF
- Reveal JS
- Markdown (md)
- ReStructured Text (rst)
- executable script

z něho do dalších formátů,
např. DOCX (MS Word)

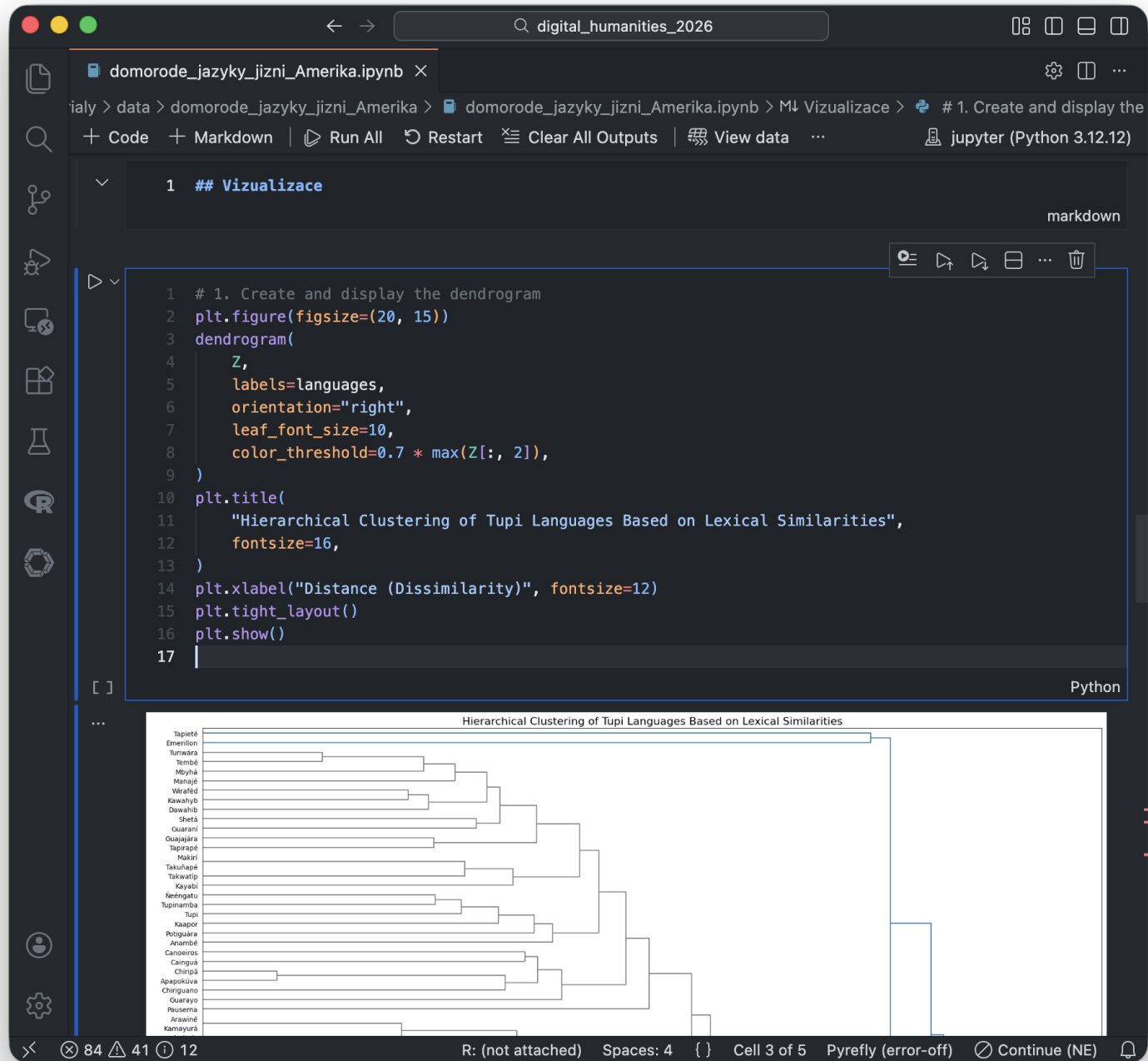
Konverze přes MD do DOCX (MS Word)

z příkazové řádky:

```
# pip install nbconvert
```

```
jupyter nbconvert --to markdown notebook.ipynb
```

```
pandoc notebook.md -o output.docx
```



← → digital_humanities_2026

domorode_jazyky_jizni_Amerika.md ×

materialy > data > domorode_jazyky_jizni_Amerika > domorode_jazyky_jizni_Amerika.md > abc ## Vizualizace

156
157 ## Vizualizace
158
159
160 ```python
161 # 1. Create and display the dendrogram
162 plt.figure(figsize=(20, 15))
163 dendrogram(
164 Z,
165 labels=languages,
166 orientation="right",
167 leaf_font_size=10,
168 color_threshold=0.7 * max(Z[:, 2]),
169)
170 plt.title(
171 "Hierarchical Clustering of Tupi Languages Based on Lexical Similarities",
172 fontsize=16,
173)
174 plt.xlabel("Distance (Dissimilarity)", fontsize=12)
175 plt.tight_layout()
176 plt.show()
177
178 ```
179
180 |
181
182 ![png](domorode_jazyky_jizni_Amerika_files/domorode_jazyky_jizni_Amerika_2_0.png)
183
184
185
186
187 ```python
188 # 2. Create and display the heatmap
189 plt.figure(figsize=(20, 18))
190 sns.heatmap(
191 similarity_matrix,
192 xticklabels=languages,
193 yticklabels=languages,
194 cmap="viridis",
195 annot=False,

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

✕ 0 ⚠ 11 ⓘ 12 R: (not attached) Ln 180, Col 1 Spaces: 4 UTF-8 LF { } Markdown ⌂ Continue (NE) 🔔

Vizualizace

1. Create and display the dendrogram

```
plt.figure(figsize=(20, 15))
dendrogram(
    Z,
    labels=languages,
    orientation="right",
    leaf_font_size=10,
    color_threshold=0.7 * max(Z[:, 2]),
)
plt.title(
    "Hierarchical Clustering of Tupi Languages Based on Lexical Similarities",
    fontsize=16,
)
plt.xlabel("Distance (Dissimilarity)", fontsize=12)
plt.tight_layout()
plt.show()
```

