JAK FUNGUJE

UNICODE

Ondřej Caletka | OpenAlt 2026 | listopad 2026

VŠECHNO JE ČÍSLO

POČÍTAČE PRACUJÍ JEN S ČÍSLY

- všechno zpracovatelné počítačem musí být **převoditelné na číslo**
- základní jednotka - **bity 0/1**
- paralelní zpracování **8/16/32/64 bitů** (byte, word, dword, qword)
- paměť je nejčastěji **organizovaná po bajtech**/oktetech

LIDÉ POUŽÍVAJÍ ŠESTNÁCTKOVOU A OSMIČKOVOU SOUSTAVU

- ušetří psaní dlouhých čísel
- přitom stále pracujeme s **celistvým počtem bitů**

bin	hex/oct/dec	bin	hex	dec
0000	0	1000	8	8
0001	1	1001	9	9
0010	2	1010	A	10
0011	3	1011	B	11
0100	4	1100	C	12
0101	5	1101	D	13
0110	6	1110	E	14
0111	7	1111	F	15

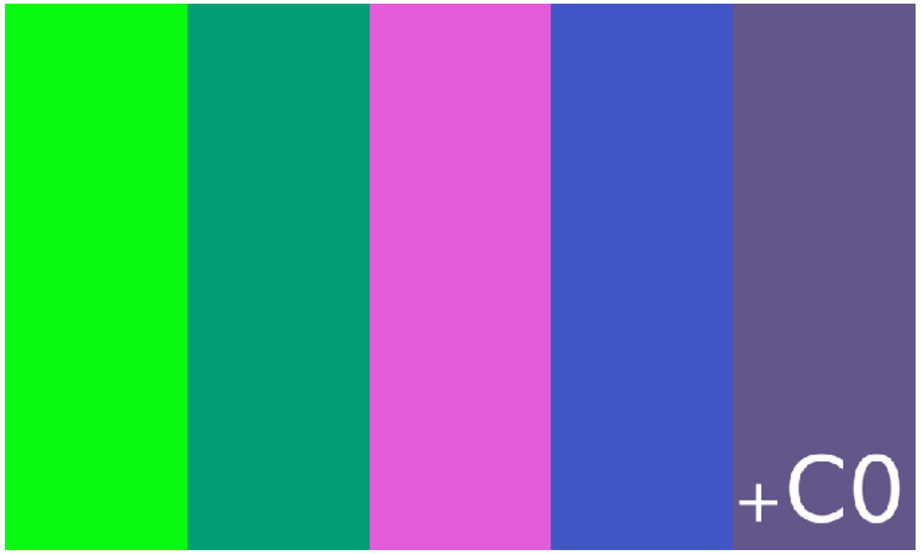
ZÁLEŽÍ NA REPREZENTACI

- skupina čísel, nebo jedno číslo?
- kladné/záporné/desetinné?
- **pořadí oktetů**:
  - **Big Endian** - nejprve nejvýznamnější (MSB)
  - **Little Endian** - nejprve nejméně významný (LSB)

1111 1111 0000 0111	BINARY
F F 0 7	HEX
255 7	8-BIT DEC
65287	16-BIT DEC BE
2047	16-BIT DEC LE
-249	16-BIT DEC BE TWO'S COMPLEMENT

KONCEPT NELEGÁLNÍCH ČÍSEL

- jakákoli počítačová informace je **číslo**
- některé počítačové informace je zákázáno šířit
- je tedy zakázáno **šířit určitá čísla**
- například 128bitový šifrovací klíč AACS pro HD DVD a Blu-ray



FREE SPEECH FLAG

Wikipedia: Illegal number

NĚKTERÁ ČÍSLA NEJSOU NELEGÁLNÍ, JEN MAJÍ BÝT TAJNÁ



EMUDHRA PRIVATE KEY FLAG

ASCII JE VŠUDE

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

- sedmibitový kód
- řídicí znaky (např. konec řádku)
- číslice, velká a malá písmena
- pouze americká angličtina
- **základ spousty dalších kódování**

$b_7 b_6 b_5$ → / Bits b_4 b_3 b_2 b_1 ↓ / Column → / Row ↓	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	—	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

ASCII SE VYPLATÍ ZNÁT (ČÁSTEČNĚ) NAZPAMĚŤ

- řídicí znaky 0x00 - 0x1f, 0x7f
- mezera 0x20
- číslice 0x30 - 0x39
- velká písmena 0x41 - 0x5A
- malá písmena 0x61 - 0x7A

	0	1	2	3	4	5	6	7
0			SPC	0	@	P		p
1			!	1	A	Q	a	q
2				2	B	R	b	r
3			#	3	C	S	c	s
4				4	D	T	d	t
5				5	E	U	e	u
6				6	F	V	f	v
7				7	G	W	g	w
8				8	H	X	h	x
9				9	I	Y	i	y
A	LF		*		J	Z	j	z
B					K		k	
C					L		l	
D	CR				M		m	
E					N		n	
F					O		o	DEL

ASCII VYSVĚTLUJE KLÁVESOVÉ ZKRATKY

- Shift přeřazuje o **0x20**
- Ctrl přeřazuje o **0x40**
  - **^L** smaže obrazovku
  - **^S** umrtví terminál, **^Q** ho oživí
  - **^D** ukončí vstup
  - **^A** resp. **^E** přesune kurzor na začátek resp. konec řádku
  - **^M** = Enter, **^I** = Tab, **^[** = ESC

	0	1	2	3	4	5	6	7
0	NUL		SPC	0	@	P		p
1	SOH	XOn	!	1	A	Q	a	q
2				2	B	R	b	r
3		XOff	#	3	C	S	c	s
4	EOT			4	D	T	d	t
5	ENQ			5	E	U	e	u
6				6	F	V	f	v
7	BEL			7	G	W	g	w
8	BS			8	H	X	h	x
9	TAB			9	I	Y	i	y
A	LF		*		J	Z	j	z
B		ESC			K	[	k	{
C	FF				L	\	l	|
D	CR				M	[	m	}
E					N		n	~
F					O		o	DEL

ANSWER TO THE ULTIMATE QUESTION OF LIFE, THE UNIVERSE, AND EVERYTHING

ALSO THE RESPONSE OF CRYPT(3) FUNCTION IN MUSL C LIBRARY FOR INVALID INPUT



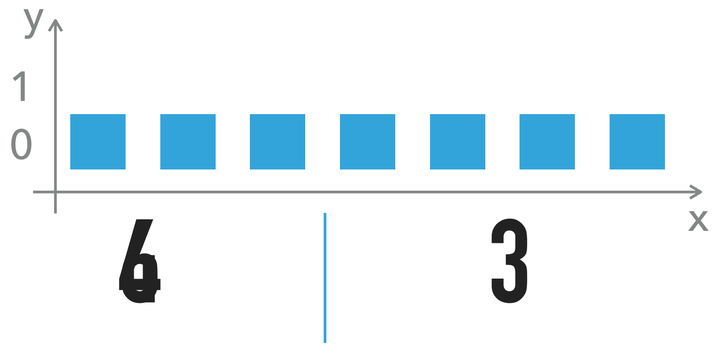
0x 2A

2*16 + 10 = 42

0b 010 1010

ZAČALI JSME PÍSMENEM C. NAJEDNOU SE CESNET USMÁL

cesnet



Příběh loga CESNET 2017

TELETEXT ČT – STRANA 899

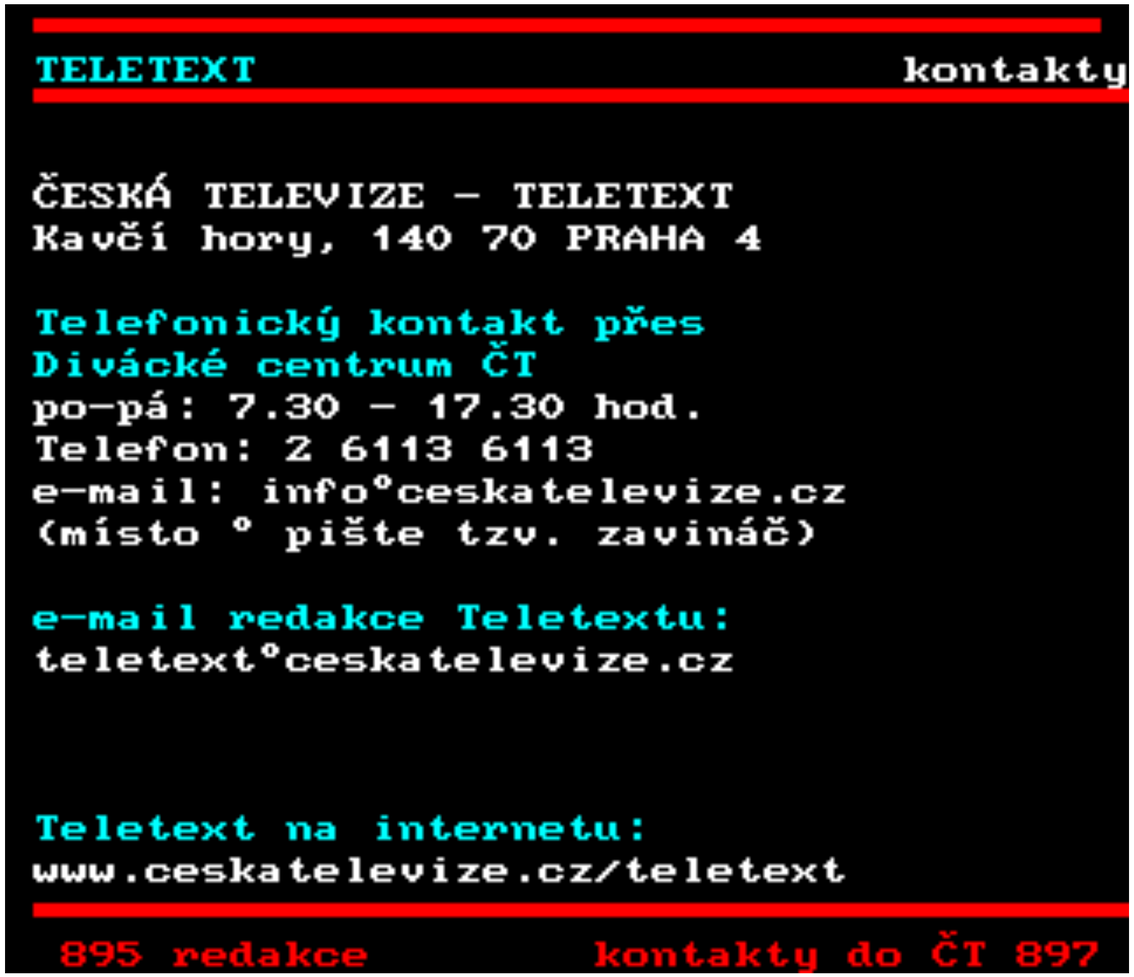


- 40 znaků na 24 řádcích
- přenášeno v **nevyužitých televizních řádcích** jako kombinace šedých teček

* = 010 1010

U = 101 0101

KDYŽ ASCII NESTAČÍ



- národní znaky nahrazují některé původní znaky z ASCII
- česká verze nemá znak @

ROZŠÍŘENÍ ASCII

PŮVODNÍ OSMIBITOVÁ KÓDOVÁNÍ

- přidávají dalších 128 znaků
  - diakritiku
  - semigrafiku
- mnoho **vzájemně nekompatibilních kódování**
- diakritika často na úkor semigrafiky



CP437
EGA/VGA
DEFAULT



CP850
DOS
LATIN 1

KÓD BRATŘÍ KAMENICKÝCH

- osmibitové kódování pro **české a slovenské prostředí**
- zachovává všechny semigrafické znaky z CP437
- zachovává **určitou čitelnost** i na počítačích bez možnosti změny fontu

P─íli¿ ælufouçkÿ kûñ úpêl âábelské ódy.

PPtsïLI¢ ÆLUåOUÇK¥ KªÑ ùPëL àÅBELSKÉ òDY.

STANDARDNÍ A PROPRIETÁRNÍ ROZŠÍŘENÍ

- jeden znak = 8bitů
  - CP852
  - Windows-1250
  - ISO-8859-2
- „Latin-2“
- variabilní délka
  - ISO-6937
- „**non spacing diacritical mark**“ - podobně jako na psacím stroji
- používá se například v metadatech digitální televize DVB

PROBLÉMY S OSMIBITOVÝMI ROZŠÍŘENÍMI

- nedostatek místa pro všechny možné kombinace znaků
- nemožnost kombinovat **různé jazyky v jednom textu**
- často chybějící **metadata** o tom, které kódování se používá



PŘICHÁZÍ UNICODE

ZÁKLADNÍ MYŠLENKA

- **osm bitů je málo**
- **šestnáctibitová** čísla lze zpracovávat **téměř stejně efektivně**
- díky dostatečné kapacitě lze pro každý znak definovat **unikátní kódový bod**
  - například U+002A = '*' = "\N{**ASTERISK**}"

PYTHON 3 SYNTAX



NEJVĚTŠÍ OMYLY HISTORIE

- „**640 kB paměti** přeci musí stačit každému.“
- „**Čtyři miliardy IP adres** jsou nevyčerpatelně mnoho.“
- „Všechny jazyky světa používají **míň než 65 tisíc různých znaků**“
  - aktualizace Unicode 2.0 (1996) rozšiřuje kódový prostor až do **U+10FFFF**
  - přidává dalších 16 rovin k ***Basic Multilingual Plane*** (BMP)
  - v BMP (U+0000 – U+FFFF) jsou stále **všechny znaky všech živých jazyků**

KROMĚ EMOJI 😎

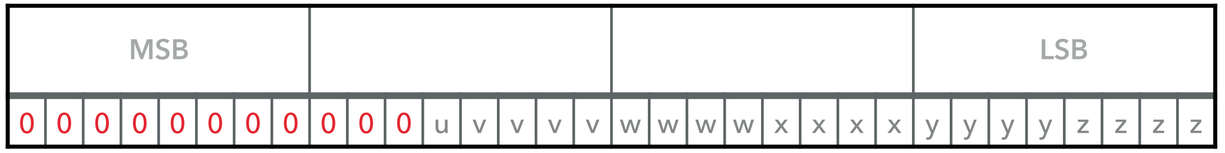
KÓDOVÁNÍ UNICODE

- způsob uložení sekvence **kódových bodů** do souboru
- dříve **Unicode Transformation Format**, nyní **Unicode Encoding Form**/Scheme
- dělení podle délky slova (**Code Unit**):
  - UTF-32
  - UTF-16
  - **UTF-8**

KÓDOVÁNÍ UTF-32

- jeden kódový bod = jedno 32bitové číslo **0x0000 0000 - 0x0010 FFFF**
- jedenáct nejvýznamnějších bitů **vždy nulových**
- **extrémně prostorově neefektivní**
- nutnost řešit pořadí oktetů
  - UTF-32**BE** – big endian
  - UTF-32**LE** – little endian

U+UVWXYZ V KÓDOVÁNÍ UTF-32



KÓDOVÁNÍ UTF-8

- kódování do osmibitových slov s **proměnnou délkou**
- kódové body U+0000 - U+007F jsou reprezentovány **jedním oktetem**
- **plná zpětná kompatibilita s ASCII** (v 8bitových slovech)
- vyšší kódové body se kódují do **dvou- až čtyř-oktetových sekvencí**

U+00YZ UTF-8

Byte 1							
0	y	y	y	z	z	z	z

U+0XYZ V UTF-8

Byte 1								Byte 2							
1	1	0	x	x	x	y	y	1	0	y	y	z	z	z	z

U+WXYZ V KÓDOVÁNÍ UTF-8

Byte 1								Byte 2								Byte 3							
1	1	1	0	w	w	w	w	1	0	x	x	x	x	y	y	1	0	y	y	z	z	z	z

U+UVWXYZ V KÓDOVÁNÍ UTF-8

Byte 1								Byte 2								Byte 3								Byte 4							
1	1	1	1	0	u	v	v	1	0	v	v	w	w	w	w	1	0	x	x	x	x	y	y	1	0	y	y	z	z	z	z

PROČ JE UTF-8 NEJLEPŠÍ?

- podporuje **nulou ukončené řetězce**
- neřeší **little/big endian**
- automatická **synchronizace** a **zotavení** z neúplných znaků
- znaky ASCII jsou **kódovány efektivně** *(například značky HTML)*

Hello OndÅ™ej,

UTF-8 INTERPRETOVANÉ JAKO WINDOWS-1252



8BITOVÉ KÓDOVÁNÍ INTERPRETOVANÉ JAKO UTF-8

KÓDOVÁNÍ UTF-16 – IT'S COMPLICATED

- původně kódování s pevnou délkou, také známé jako **UCS-2**
- dokáže zakódovat **pouze kódové body z BMP**
- spousta aplikací **stále předpokládá pevnou délku**
- kódování hodnot **mimo BMP** se provádí pomocí **dvojice** kódových bodů **uvnitř BMP**

SURROGATE CODE POINTS

- oblast kódových bodů BMP od **U+D800** do **U+DFFF**
- slouží pro reprezentaci kódových bodů **mimo BMP** v UTF-16
- dvě skupiny pro horní/první a dolní/druhou část
- jsou **zakázané mimo UTF-16**



U+01F995

−0x10000

0x0F995

00001111100110010101

00001111100110010101

D 8 D D

1101100000111110 1101110110010101

high surrogate *low surrogate*

Chaotic Enby, Wikipedia

UNICODE V PRAXI

BYTE ORDER MARK (BOM)

- kódový bod **U+FEFF**, "\N{**ZERO WIDTH NO-BREAK SPACE**}"
- kódový bod **U+FFFE není platným znakem**
- vkládán na začátek textů v **UTF-16** a **UTF-32** k rozpoznání pořadí oktetů
- někdy používán i v **UTF-8** *(jako EF BB BF)* k rozlišení UTF-8 od starých kódování
- **rozbíjí shellové skripty**

```
00000000  ef bb bf 23 21 2f 62 69  6e 2f 62 61 73 68 0a     |...#!/bin/bash.|
```

NEJEDNOZNAČNOST

```
>>> o1
'Ondřej'
>>> o2
'Ondřej'
>>> o1 == o2
False
>>> len(o1)
6
>>> len(o2)
7
```

- kódové body nejsou vždy **znaky**
- jako znaky obvykle vnímáme *grafémy*
- stejný *grafém* lze reprezentovat **různou kombinací kódových bodů**
- nejde o *homoglyfy*: jde o totožný *glyf* reprezentující tentýž *grafém*

NEJMENŠÍ JEDNOTKA PSANÉHO JAZYKA

STEJNÉ GLYFY S RŮZNÝM VÝZNAMEM

KONKRÉTNÍ REPREZENTACE GRAFÉMU URČITÝM FONTEM

KOMPONOVANÉ A DEKOMPONOVANÉ ZNAKY

```
>>> import unicodedata
>>> def explain(s):
...     """Print the sequence of code points in the string."""
...     for c in s:
...         print(f"{c}\tU+{ord(c):04x}\t{unicodedata.name(c)}")
...
>>> explain(o1)
O U+004f LATIN CAPITAL LETTER O
n U+006e LATIN SMALL LETTER N
d U+0064 LATIN SMALL LETTER D
ř U+0159 LATIN SMALL LETTER R WITH CARON
e U+0065 LATIN SMALL LETTER E
j U+006a LATIN SMALL LETTER J
```

```
>>> explain(o2)
O U+004f LATIN CAPITAL LETTER O
n U+006e LATIN SMALL LETTER N
d U+0064 LATIN SMALL LETTER D
r U+0072 LATIN SMALL LETTER R
ˇ U+030c COMBINING CARON
e U+0065 LATIN SMALL LETTER E
j U+006a LATIN SMALL LETTER J
```

EXTRÉMNÍ KOMBINACE (ZALGO)






```
>>> ms = "Michal Špaček"
>>> len(ms)
93
```

S	U+0053	LATIN CAPITAL LETTER S
◌̵	U+0335	COMBINING SHORT STROKE OVERLAY
◌̒	U+0312	COMBINING TURNED COMMA ABOVE
◌̒	U+0312	COMBINING TURNED COMMA ABOVE
◌̓	U+0343	COMBINING GREEK KORONIS
◌̐	U+0310	COMBINING CANDRABINDU
◌͜	U+035c	COMBINING DOUBLE BREVE BELOW
◌͕	U+0355	COMBINING RIGHT ARROWHEAD BELOW
◌͙	U+0359	COMBINING ASTERISK BELOW
◌͙	U+0339	COMBINING RIGHT HALF RING BELOW
◌̭	U+032d	COMBINING CIRCUMFLEX ACCENT BELOW
◌̣	U+0323	COMBINING DOT BELOW
◌̘	U+0318	COMBINING LEFT TACK BELOW
◌̩	U+0329	COMBINING VERTICAL LINE BELOW
◌̞	U+031e	COMBINING DOWN TACK BELOW
◌ͅ	U+0345	COMBINING GREEK YPOGEGRAMMENI
◌̙	U+0319	COMBINING RIGHT TACK BELOW
◌̡	U+0321	COMBINING PALATALIZED HOOK BELOW
◌̗	U+0317	COMBINING ACUTE ACCENT BELOW

p	U+0070	LATIN SMALL LETTER P
◌̶	U+0336	COMBINING LONG STROKE OVERLAY
◌̫	U+035b	COMBINING ZIGZAG ABOVE
◌͈	U+0348	COMBINING DOUBLE VERTICAL LINE BELOW
◌ͫ	U+032b	COMBINING INVERTED DOUBLE ARCH BELOW
◌̧	U+0327	COMBINING CEDILLA
◌̝	U+031d	COMBINING UP TACK BELOW
◌̧	U+0327	COMBINING CEDILLA
◌̧	U+0327	COMBINING CEDILLA
◌̥	U+0325	COMBINING RING BELOW
◌̲	U+0332	COMBINING LOW LINE
◌͜	U+035c	COMBINING DOUBLE BREVE BELOW

a	U+0061	LATIN SMALL LETTER A
◌̵	U+0335	COMBINING SHORT STROKE OVERLAY
◌̛	U+031b	COMBINING HORN
◌̂	U+0302	COMBINING CIRCUMFLEX ACCENT
◌̈	U+0308	COMBINING DIAERESIS
◌͕	U+0355	COMBINING RIGHT ARROWHEAD BELOW
◌̜	U+031c	COMBINING LEFT HALF RING BELOW
◌͕	U+0355	COMBINING RIGHT ARROWHEAD BELOW
◌͕	U+0355	COMBINING RIGHT ARROWHEAD BELOW
◌͍	U+034d	COMBINING LEFT RIGHT ARROW BELOW
◌̼	U+033c	COMBINING SEAGULL BELOW
◌̻	U+033b	COMBINING SQUARE BELOW
◌͔	U+0354	COMBINING LEFT ARROWHEAD BELOW
◌̻	U+033b	COMBINING SQUARE BELOW
◌̥	U+0325	COMBINING RING BELOW
◌̻	U+033b	COMBINING SQUARE BELOW

NORMALIZACE V UNICODE

- Každý znak má definovanou **nejvýš jednu dekompozici**
  - **kanonická** ≡ ekvivalence
  - **kompatibilní** ≈ podobný význam, jiná forma
    - například: MICRO SIGN ≈ GREEK SMALL LETTER MU
- **normalizační forma D** = všechny znaky kanonicky dekomponované
- **normalizační forma C** = po dekompozici jsou znaky kanonicky složené
- **normalizační forma KD/KC** = používá taktéž **kompatibilní dekompozice**

PRAKTICKÉ VYUŽITÍ NORMALIZACE – VYTVÁŘENÍ SLUGŮ

```
>>> o1
'Ondřej'
>>> # takhle raději ne
>>> "".join(c for c in o1.lower() if c <= 'z')
'ondej'
>>>
>>> # lepší
>>> nfkd = unicodedata.normalize('NFKD', o1)
>>> "".join(c for c in nfkd.lower() if c.isalnum())
'ondrej'
```

EMOJI ≠ EMOTIKONY

- emotikony **kódované jako znaky Unicode**
- mimo BMP, nejčastěji uvnitř **Supplementary Multilingual Plane** (SMP), **U+1FXXX**
- standardizované modifikátory, kombinace (limit 32 kódových bodů)
  - barva kůže
  - typ vlasů
  - pohlaví
  - směr pohybu

KOMPLEXNÍ EMOJI V PŘÍKLADECH



❤ U+2764 HEAVY BLACK HEART
U+fe0f VARIATION SELECTOR-16
U+200d ZERO WIDTH JOINER
🔥 U+1f525 FIRE



🍋 U+1f34b LEMON
U+200d ZERO WIDTH JOINER
🟩 U+1f7e9 LARGE GREEN SQUARE



🫱 U+1faf1 RIGHTWARDS HAND
🏼 U+1f3fb EMOJI MODIFIER FITZPATRICK TYPE-1-2
U+200d ZERO WIDTH JOINER
🫲 U+1faf2 LEFTWARDS HAND
🏿 U+1f3ff EMOJI MODIFIER FITZPATRICK TYPE-6



👨 U+1f468 MAN
🏼 U+1f3fb EMOJI MODIFIER FITZPATRICK TYPE-1-2
U+200d ZERO WIDTH JOINER
🦲 U+1f9b2 EMOJI COMPONENT BALD

KOMPLEXNÍ EMOJI V PŘÍKLADECH



✏ U+270f PENCIL
U+fe0f VARIATION SELECTOR-16



👱 U+1f9d4 BEARDED PERSON
U+200d ZERO WIDTH JOINER
♀ U+2640 FEMALE SIGN
U+fe0f VARIATION SELECTOR-16



👩 U+1f469 WOMAN
🏽 U+1f3fc EMOJI MODIFIER FITZPATRICK TYPE-3
U+200d ZERO WIDTH JOINER
🚒 U+1f692 FIRE ENGINE



👩 U+1f469 WOMAN
🏽 U+1f3fc EMOJI MODIFIER FITZPATRICK TYPE-3
U+200d ZERO WIDTH JOINER
🦯 U+1f9af PROBING CANE
U+200d ZERO WIDTH JOINER
➡ U+27a1 BLACK RIGHTWARDS ARROW
U+fe0f VARIATION SELECTOR-16

EMOJI VLAJKY ÚZEMNÍCH CELKŮ

- používají 26 písmen skupiny **Regional indicator symbol** kódující ISO 3166-1 alpha-2
- **implementace rozhoduje**, kterým kódům přiřadí vlajky a které zobrazí jako **dvojici barevných písmen**

🇨🇿

🇨 U+1f1e8 REGIONAL INDICATOR SYMBOL LETTER C
🇿 U+1f1ff REGIONAL INDICATOR SYMBOL LETTER Z

🇬🇧

🇬 U+1f1ec REGIONAL INDICATOR SYMBOL LETTER G
🇧 U+1f1e7 REGIONAL INDICATOR SYMBOL LETTER B

🇽🇰

🇽 U+1f1fd REGIONAL INDICATOR SYMBOL LETTER X
🇰 U+1f1f0 REGIONAL INDICATOR SYMBOL LETTER K

OSTATNÍ EMOJI VLAJKY

- kombinace bílé nebo černé vlajky a jiného symbolu
- černá vlajka a **tag**, kódující zemi a region pěti malými písmeny

NEVIDITELNÁ METAINFORMACE PŮVODNĚ URČENÁ K OZNAČENÍ JAZYKA

🏳️‍🌈

```
🏳  U+1f3f3  WAVING WHITE FLAG
    U+fe0f VARIATION SELECTOR-16
    U+200d ZERO WIDTH JOINER
🌈  U+1f308  RAINBOW
```

🏴‍

```
🏴  U+1f3f4  WAVING BLACK FLAG
    U+e0067  TAG LATIN SMALL LETTER G
    U+e0062  TAG LATIN SMALL LETTER B
    U+e0065  TAG LATIN SMALL LETTER E
    U+e006e  TAG LATIN SMALL LETTER N
    U+e0067  TAG LATIN SMALL LETTER G
    U+e007f  CANCEL TAG
```

UNICODE UMÍ VÍC

UNICODE COMMON LOCALE DATA REPOSITORY

- místo odkud **všichni čerpají lokalizační data** jako
  - abecední pořadí
  - názvy měsíců (ledna, února,…)
  - styl prezentace čísel (desetinná čárka, oddělovač tisíců)

Z „leden“ je „ledna“ – glibc mění názvy měsíců

UNICODE NENÍ JEDNODUCHÉ, PROTOŽE SVĚT NENÍ JEDNODUCHÝ

- ale aspoň je **jednotné**
- a **stabilní** – znaky se nikdy **nevyřazují** a **nerecyklují**
- nesnažte se **počítat znaky** – nemá to cenu
- porovnávejte řetězce jen po **normalizaci**
- **používejte UTF-8** a počítejte s tím, že jeden kódový bod může mít **až 4 bajty**

DÍKY ZA POZORNOST

ONDŘEJ CALETKA
HTTPS://ONDŘEJ.CALETKA.CZ
ONDŘEJ@CALETKA.CZ

Prezentace je už nyní k dispozici ke stažení