unicodedata

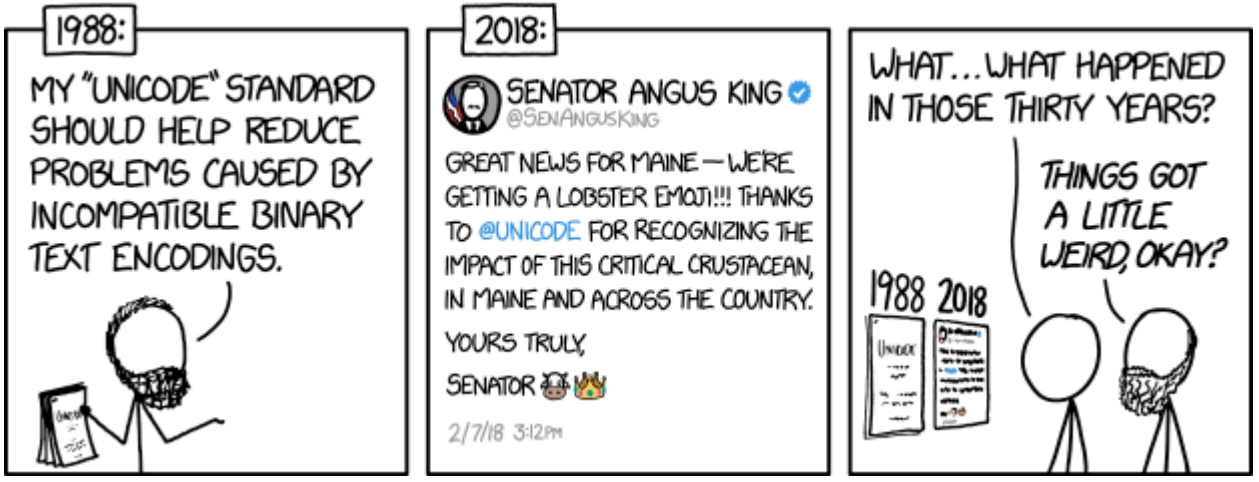

Image from XKCD (CC-BY 2.5)

unicodedata

or: How I Learned to Stop Worrying and Love the Unicode Standard

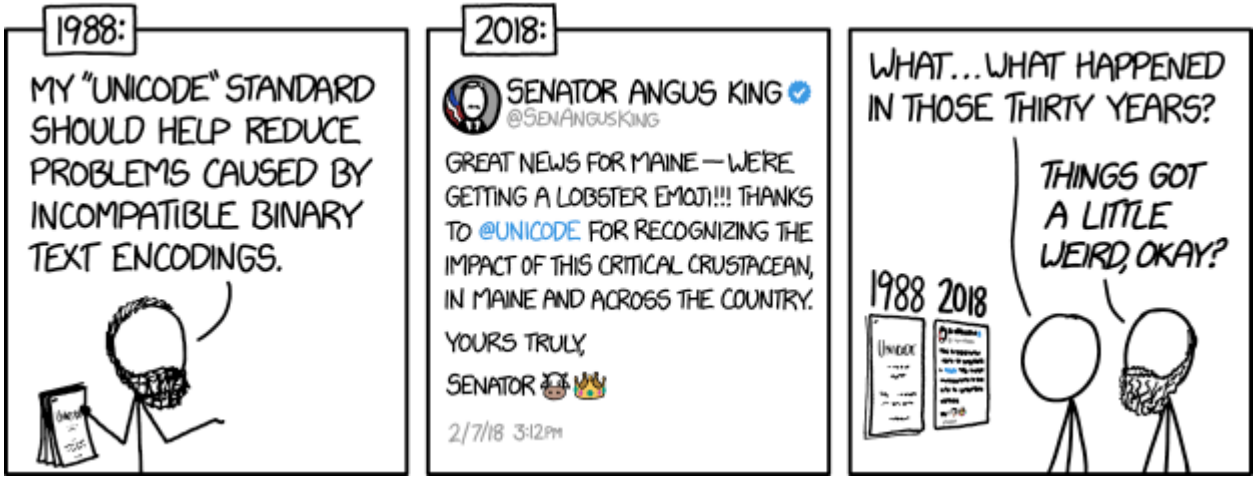

Image from XKCD (CC-BY 2.5)

First: what even is text?!

- A: there ain’t no such thing
- Computers are built around numerical data (integers or bytes)
  - All the way down to the hardware!

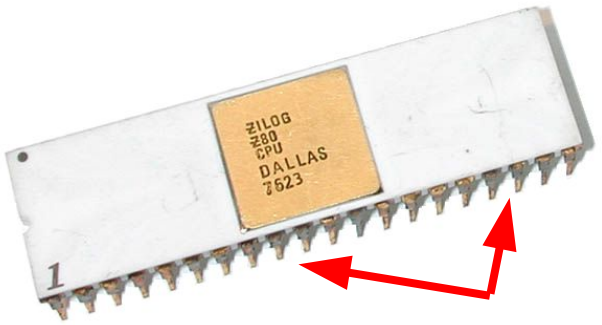

Image from Wikimedia Commons (CC-BY 2.5)

First: what even is text?!

- Once upon a time, computers had entire chips dedicated to generating text characters!

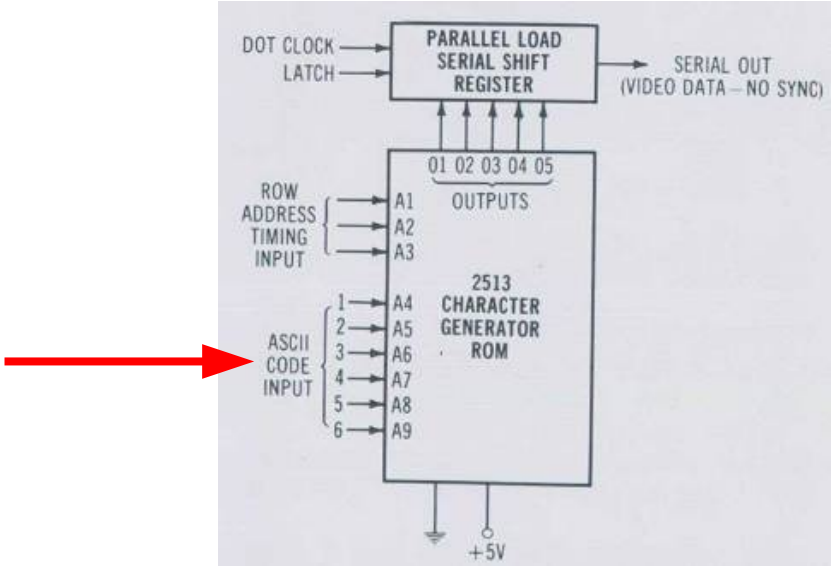

Image from AtariArchives.org (fair use)

First: what even is text?!

- So, we'll define a **coding** that associates characters with numbers (i.e. "codes")
  - e.g. ASCII
  - The IANA maintains an incomplete* list of these

*258 listed, but there are more!

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Image from Wikimedia Commons (public domain)

English: not the only language in the world!

- Wouldn’t it be great if there were a **universal coding** for text?

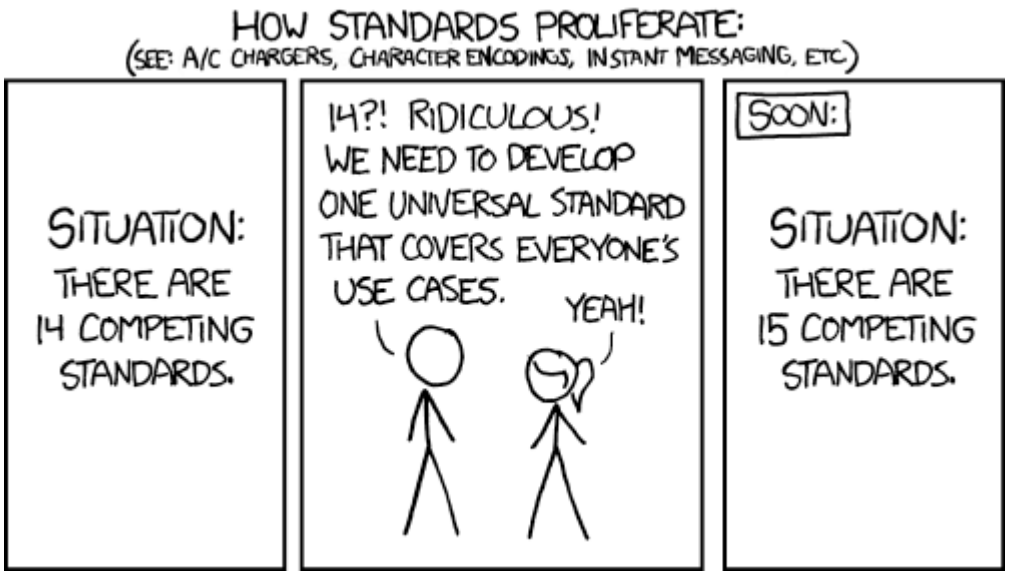

Image from XKCD (CC-BY 2.5)

What is Unicode?

- In 1987, Joe Becker and others proposed a *“wide-body ASCII”* using more bits per code in order to represent many more characters
- This idea eventually became the **Unicode Standard**
  - 1,114,112 “code points”
  - 149,186 of which are “assigned”

Unicode vs. Unicode Consortium?

- Aside: when people say “Unicode”, they *usually* are referring to the **Unicode Standard…**
- …but the name can also refer to the **Unicode Consortium** and its other work

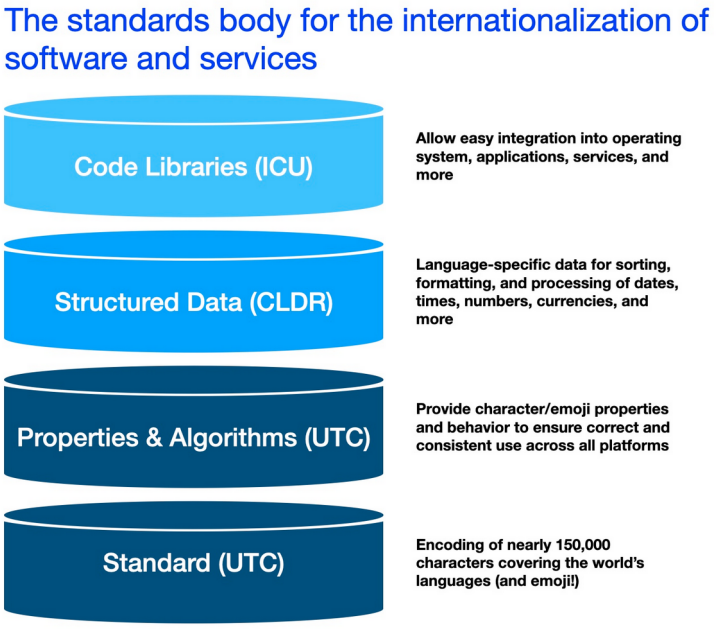

Image from unicode.org (fair use)

What is Unicode?

- The Unicode Standard
  - Lots of space for growth
  - Encodings (numbers → bytes transformations) try to be compatible with other codings when possible
  - e.g. UTF-8 is fully compatible with ASCII

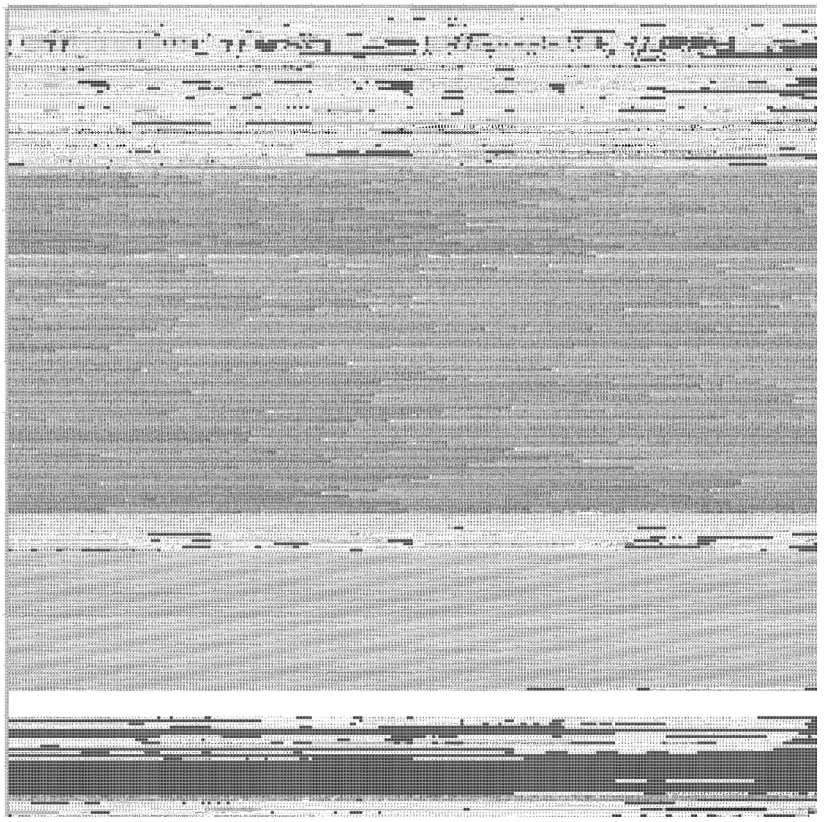

Image from Wikimedia Commons (public domain)

What is Unicode?

- The Unicode Standard
  - Has internal units of organization (planes, blocks, etc.)
  - Previous slide is one **plane** out of 16!

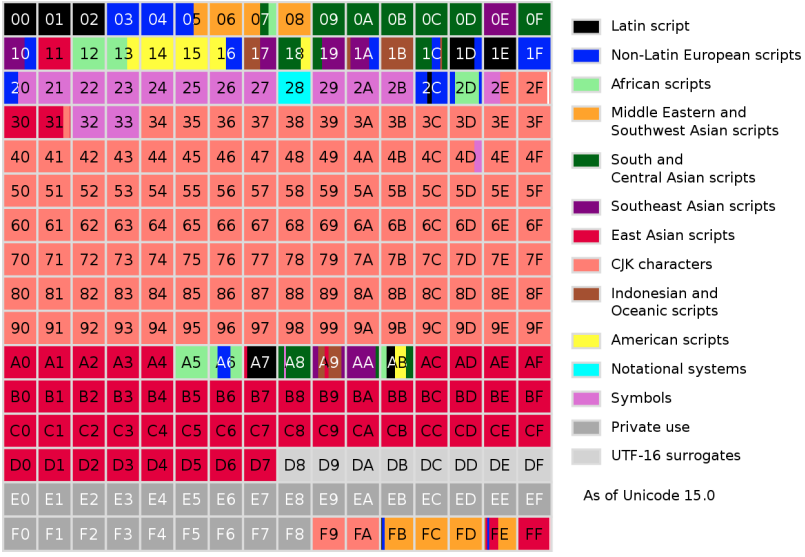

Image from Wikimedia Commons (public domain)

What is Unicode?

- The **Unicode Character Database (UCD)**
  - Extra data associated with the characters defined by the Unicode Standard
  - This data has (some) information about the *meaning* of the characters
- …way simpler than you might think
  - Basically a bunch of text files!

What is Unicode?

- The UCD gives us information about each character:
  - Name
  - Category
  - Properties
    - e.g. numerical information
  - Information about how this character interacts with other characters (e.g. is it a “combining” character?)
  - …and more!

What is unicodedata?

- A module wrapped around Unicode’s data files
  - Note: only updates when Python does!
- Most of the functions are named for their UCD counterparts
  - e.g. unicodedata.category() tells us about the category of the given codepoint

What is unicodedata?

- lookup()
- name()
- decimal(), digit(), numeric()
  - These aren’t all the same concept!
- category()
- normalize()

unicodedata: examples

- Get the name of a character
- Lookup a character by name
- Find all of the numerical characters in Unicode

unicodedata: normalization

- “ça va” vs. “ça va?”
- How do we compare strings that “look the same” to a human being, but which are distinct?
  - There’s a Unicode guideline for that!
  - Several **normalization forms** defined
- Equivalent identifiers in Python
  - パイソン and ﾊﾟｲｿﾝ are the same Python name
  - This is actually a Unicode rule!