Character encoding

- String: variable containing a string of characters
- In the memory, it is a list of 0s and 1s:

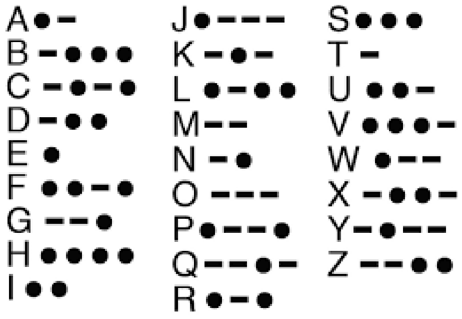
```
00001000 00100101 00101100 00101100 00101111 00000001
```

- How do we go from text to binary representation?

➡ **Character encoding**

Early history

- First binary representation of characters: Morse code

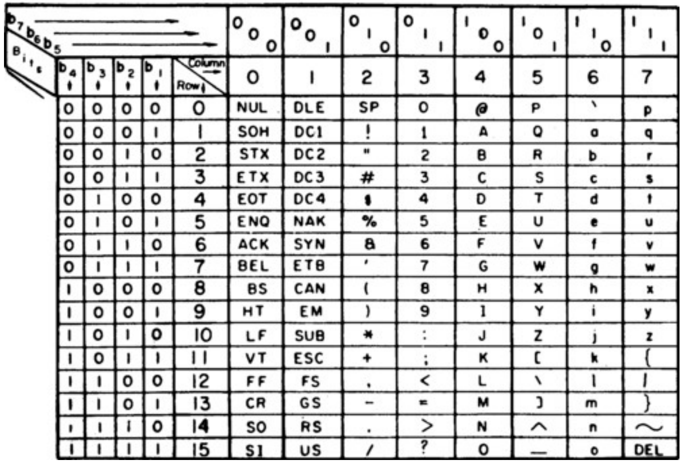


A	●▬	J	●▬▬▬	S	●●●
B	▬●●●	K	▬●▬	T	▬
C	▬●▬●	L	●▬●●	U	●●▬
D	▬●●	M	▬▬	V	●●●▬
E	●	N	▬●	W	●▬▬
F	●●▬●	O	▬▬▬	X	▬●●▬
G	▬▬●	P	●▬▬●	Y	▬●▬▬
H	●●●●	Q	▬▬●▬	Z	▬▬●●
I	●●	R	●▬●		

- Early days of computers (60s-70s):
  - Every company had its own system
  - Only standard English
  - Information transfer difficult

ASCII

- ASCII = American Standard Code for Information Interchange
- 7 bit code: $2^7 = 128$ characters
- Still only standard English
- If stored on 1 Byte = 8 bits → 128 characters undefined
- Developers started using them to define their own character sets (e.g., Latin-1)
- More than 256 characters?
- Confusion remains



USASCII code chart

Bits b7 b6 b5 →					000	001	010	011	100	101	110	111
b4	b3	b2	b1	Row ↓ / Column →	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

Unicode

- Universal Coded Character Set
- Effort to uniquely identify characters used in any language
- Right now: 143,859 characters each with an associated ID number
  - E.g.: 65 = U+0041 = A

    337 = U+0150 = ő

    128,512 = U+1F600 = 😀
- The first 128 is the same as ASCII

- This is not an encoding yet
- How are these numbers stored in the memory?

Encoding Unicode

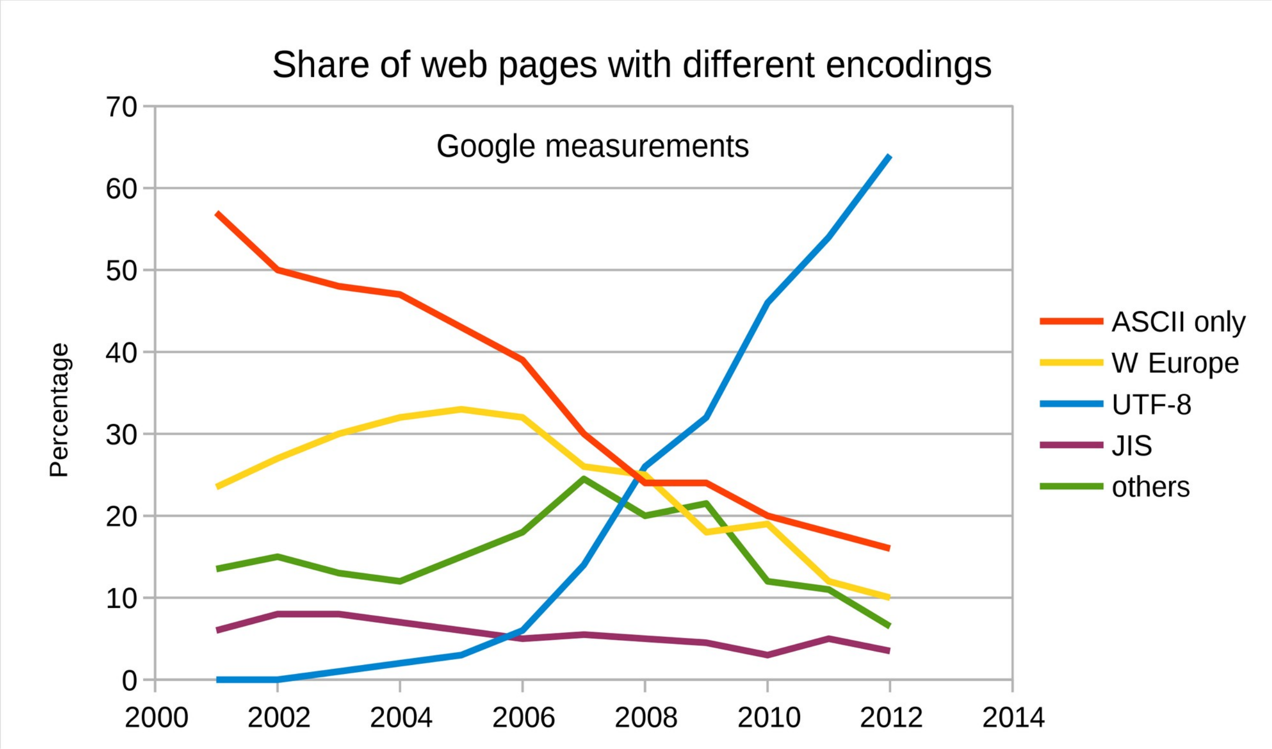
- 143,859 characters
- Possibility: store each character on 3 or more bytes

  → very wasteful, most characters are very rare
- Solution: utf-8
- Variable-width encoding

0-127:	1 byte	0xxxxxxx
128-2047	2 bytes	110xxxxx 10xxxxxx
2048-65,535	3 bytes	1110xxxx 10xxxxxx 10xxxxxx
65,536-1,114,111	4 bytes	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

UTF-8

- In 2020, 95% of www is utf-8



Source: wikipedia