Lecture 4: Bits, bytes, binary numbers, and the representation of information continued

- **computers represent, process, store, copy, and transmit everything as numbers**
  – hence "digital computer"

- **the numbers can represent anything**
  – not just numbers that you might do arithmetic on

- **the meaning depends on context**
  – as well as what the numbers ultimately represent
  – e.g., numbers coming to your computer or phone from your wi-fi connection could be email, movies, music, documents, apps, Zoom meeting, ...

Some things are intrinsically discrete / digital

- **another kind of conversion**
  - letters are converted into numbers when you type on a keyboard
  - the letters are stored (a Word document), retrieved (File/Open...), processed (paper is revised), transmitted (submitted by email), printed on paper
- **letters and other symbols are inherently discrete**
- **encoding them as numbers is just assigning a numeric value to each one, without any intrinsic meaning**
- **what letters and other symbols are included?**
- **how many digits/letter?**
  - determined by how many symbols there are
  - how do we disambiguate if symbols have different lengths?
- **how do we decide whose encoding to use?**
- **the representation is arbitrary**
- **but everyone has to agree on it**
  - if they want to work together

ASCII: American Standard Code for Information Interchange

- an arbitrary but agreed-upon representation for USA
- widely used everywhere

32	space	33	!	34	"	35	#	36	$	37	%	38	&	39	'
40	(	41	)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[	92	\	93	]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	del

```
00100000 space   00100001 !   00100010 "   00100011 #   ...
```

Hexadecimal notation

- **binary numbers are bulky**
- **hexadecimal notation is a shorthand**
- **it combines 4 bits into a single digit, written in base 16**
  - a more compact representation of the same information
- **hex uses the symbols A B C D E F for the digits 10 .. 15**

0 1 2 3 4 5 6 7 8 9 A B C D E F

0	0000	1	0001	2	0010	3	0011
4	0100	5	0101	6	0110	7	0111
8	1000	9	1001	A	1010	B	1011
C	1100	D	1101	E	1110	F	1111

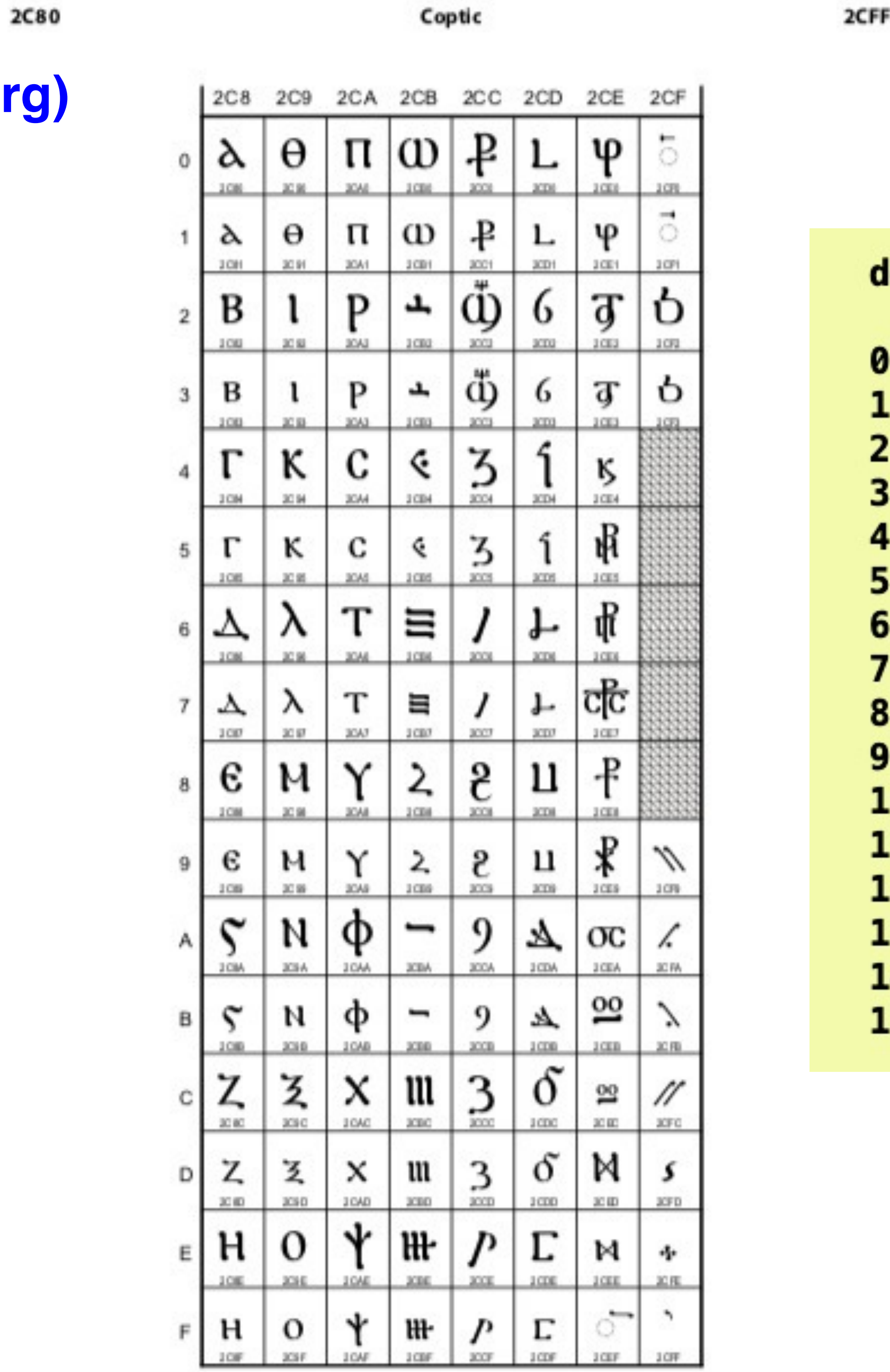
Decimal, binary, hex

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

ASCII, using hexadecimal numbers

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

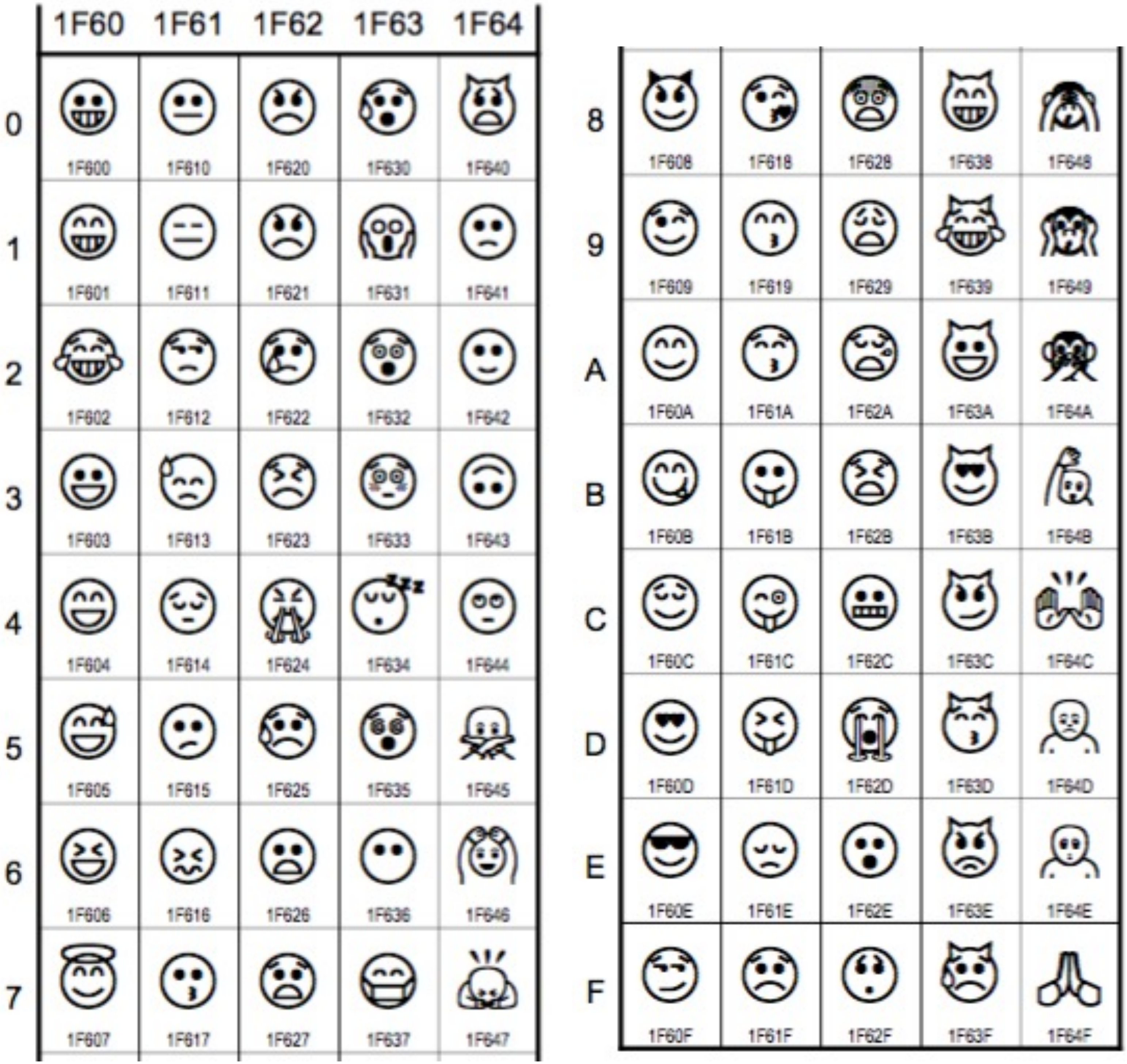
Coptic (unicode.org)



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

Emoji

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



Color

- **TV & computer screens use Red-Green-Blue (RGB) model**

- **each color is a combination of red, green, blue components**
  – R+G = yellow, R+B = magenta, B+G = cyan, R+G+B = white

- **for computers, color of a pixel is usually specified by three numbers giving amount of each color, on a scale of 0 to 255**

- **this is often expressed in hexadecimal so the three components can be specified separately (in effect, as bit patterns)**
  – 000000 is black, FFFFFF is white

- **printers, etc., use cyan-magenta-yellow[-black] (CMY[K])**

Approximations using 2^n

2^24 = 2^4 * 2^20

= 16 * 1,000,000 (16,777,216)

2^32 = 2^2 * 2^30

= 4 * 1,000,000,000 (4,294,967,296)

2^64 = 2^4 * 2^60

= 16 * 1,000,000,000,000,000,000

(18,446,744,073,709,551,616)

A <u>very</u> important idea

- **number of items and number of digits are tightly related:**
  - one determines the other
  - maximum number of different items = base $^{\text{number of digits}}$
  - e.g., 9-digit SSN: 10^9 = 1 billion possible numbers

  - e.g., to represent up to 100 “characters”: 2 digits is enough
  - but for 1000 characters, we need 3 digits

  - the same for bits: 9 bits can represent up to 2^9 = 512 items

- **interpretation depends on context**
  - without knowing that, we can only guess what numbers mean

Things to remember

- **digital devices represent everything as numbers**
  - discrete values, not continuous or infinitely precise
- **all modern digital devices use binary numbers (base 2) internally**
  - instead of decimal (base 10)
- **it's all bits at the bottom**
  - a bit is a "binary digit", that is, a number that is either 0 or 1
  - computers ultimately represent and process <u>everything</u> as bits
- **groups of bits represent larger things**
  - numbers, letters, words, names, pictures, sounds, instructions, ...
  - the interpretation of a group of bits depends on their context
  - the representation is arbitrary; standards (often) define what it is
- **the number of digits used in the representation determines how many different things can be represented**
  - number of values = base $^{\textbf{number of digits}}$
  - e.g., 10^{2} , 2^{10}