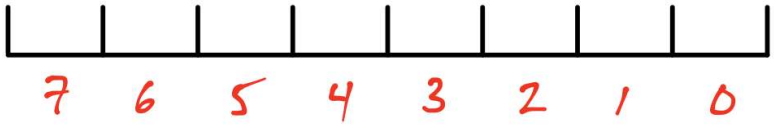
C: Bytes, strings, and character encoding

Byte:

- Basic unit of computer memory
- **1 byte** has **8 binary bits**, each 0 or 1

Schematically:

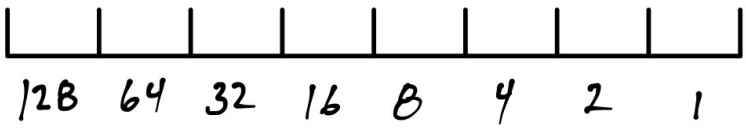


Eight spots for 0s or 1s

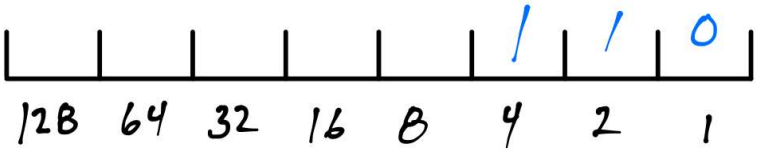
Example:

Binary: 0110 0001

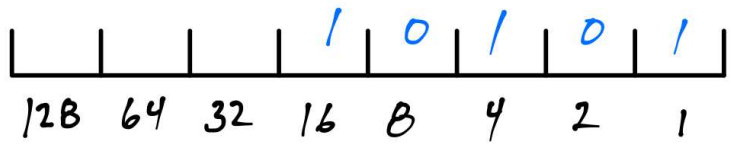
Bits have place values that are powers of 2:



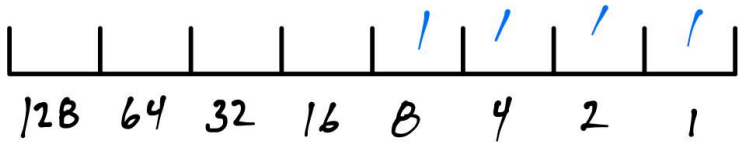
Examples:



4+2+0 = 6



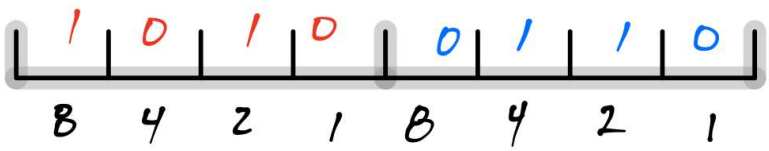
16+4+1 = 21



8+4+2+1 = 15

Representing bytes concisely: hexadecimal (or hex) notation:

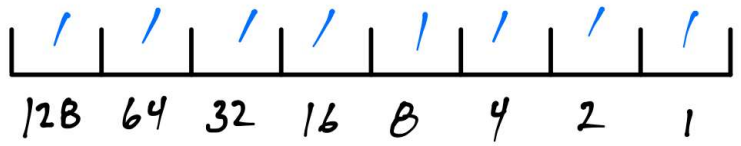
- Base 16
- Digits run 0-9 then A-F for 10-15
- Convenient for writing bytes clearly and compactly
- Widely used for web colors, special characters in URLs, etc.



Decimal: 10, 6
Hex: A6

Range of values per 8-bit byte:

- Maximum value:



Binary: 1111 1111
Hex: FF
Decimal: 255

- Range of one byte: 0 to 255

From bytes to core numeric data types:

Type	Bytes	Bits	Maximum
int	4	32	2,147,483,647
float	8	64	$\sim 2 \times 10^{308}$

Many subtypes with different bits:
int16, uint16, float32, …

From bytes to strings:

- Represent characters by **encoding** them
- **Many, many,** encoding systems have been used

ASCII (American Standard Code for Information Interchange)

- Uses 1 byte per character
- Core version uses only **7** of the bits: values 0-127
- Values 0-31: non-printing characters (tab, newline, etc.)

- Printing characters: encoded with values from 32 up:

Char	Hex	Dec		Char	Hex	Dec		Char	Hex	Dec
space	20	32		@	40	64		`	60	96
!	21	33		A	41	65		a	61	97
"	22	34		B	42	66		b	62	98
#	23	35		C	43	67		c	63	99
$	24	36		D	44	68		d	64	100
%	25	37		E	45	69		e	65	101
&	26	38		F	46	70		f	66	102
'	27	39		G	47	71		g	67	103

(	28	40		H	48	72		h	68	104
)	29	41		I	49	73		i	69	105
*	2a	42		J	4a	74		j	6a	106
+	2b	43		K	4b	75		k	6b	107
,	2c	44		L	4c	76		l	6c	108
-	2d	45		M	4d	77		m	6d	109
.	2e	46		N	4e	78		n	6e	110
/	2f	47		O	4f	79		o	6f	111
0	30	48		P	50	80		p	70	112
1	31	49		Q	51	81		q	71	113
2	32	50		R	52	82		r	72	114
3	33	51		S	53	83		s	73	115
4	34	52		T	54	84		t	74	116
5	35	53		U	55	85		u	75	117
6	36	54		V	56	86		v	76	118
7	37	55		W	57	87		w	77	119
8	38	56		X	58	88		x	78	120
9	39	57		Y	59	89		y	79	121
:	3a	58		Z	5a	90		z	7a	122
;	3b	59		[	5b	91		{	7b	123
<	3c	60		\	5c	92		|	7c	124
=	3d	61		]	5d	93		}	7d	125
>	3e	62		^	5e	94		~	7e	126
?	3f	63		_	5f	95		del	7f	127

Example:

Hi Mom!

Char:	H	i		M	o	m	!

Hex:	48	69	20	4d	6f	6d	21
Dec:	72	105	32	77	111	109	33

Aside on hex, ASCII and **URLs**:

Special characters encoded as %hex

Character	URL encoding
space	%20
(	%28
)	%29
…	…

filename on **disk**: my(ugly)file name.pdf

filename in **URL**: my%28ugly%29file%20name.pdf

Unicode:

- Uses **1+ bytes** per **character**
- Vastly more characters: > 1 million

- Widely used variant: **UTF-8** (Unicode transformation format)
- Extends ASCII: **1 ASCII byte** for ASCII characters, **2-4 bytes** for others

Example:

Good morning

Char:	G	o	o	d		m	o	r	n	i	n	g
Dec:	71	111	111	100	32	109	111	114	110	105	110	103

Buenos días

Char:	B	u	e	n	o	s		d			a	s
Dec:	66	117	101	110	111	115	32	100	195	173	97	115

í uses **two bytes**, both between 128 and 255 (above ASCII range):

195 (c3 in hex) 173 (ad in hex)