Bits and Bytes

Binary Numbers and ASCII

How is information stored at the lowest level in a computer?

Can you read this?

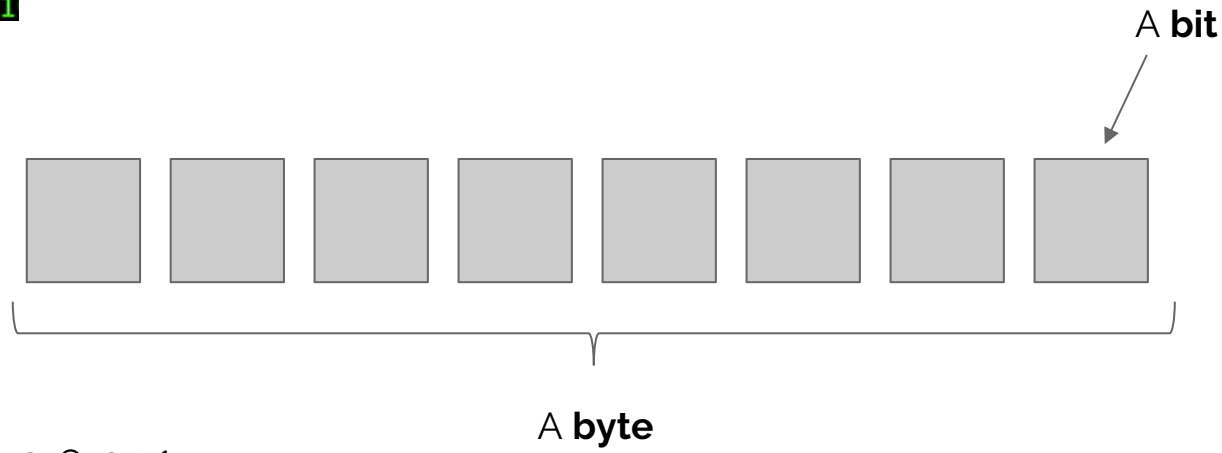
Bits and bytes

Fundamentally, digital computers are machines that convert high and low electrical signals into 0's and 1's.

- The lowest level of communication with a digital computer is 0's and 1's
- Let's learn to speak its native language!

Can you read this?

```
00111111
```



A **bit** is a 0 or 1.

A **byte** is 8 bits.

Can you read this?

```
00111111
```

Binary Place-Value

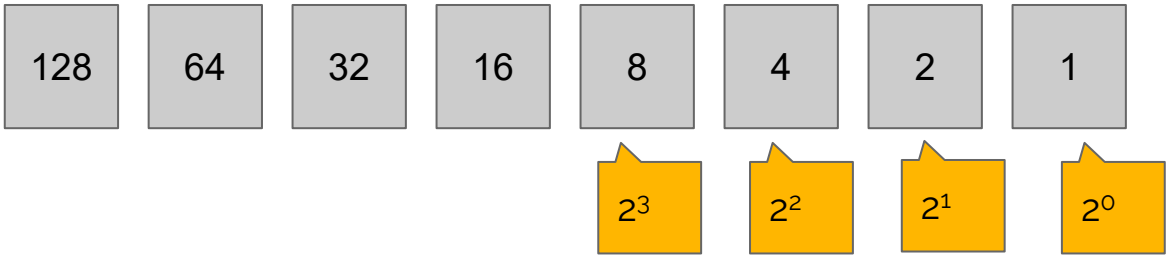
128	64	32	16	8	4	2	1

In decimal, each digit from right to left represents 1's, 10's, 100's, 1000's... etc.

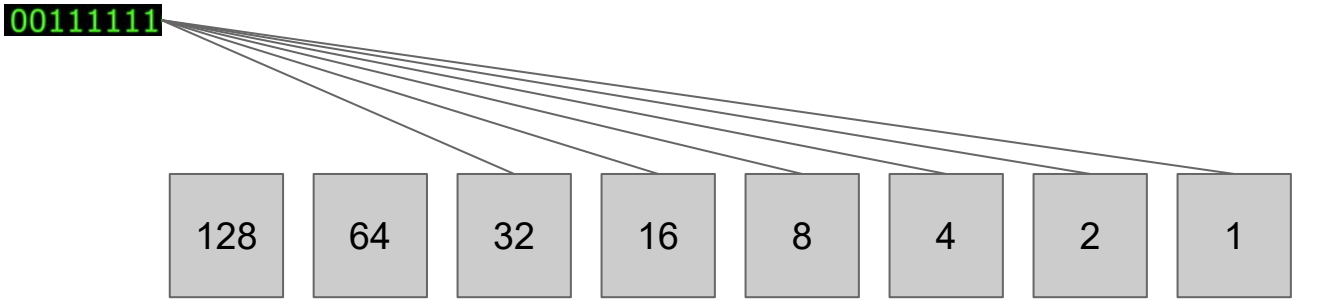
In binary, the bits represent 1, 2, 4, 8, 16...

Can you read this?

```
00111111
```



Can you read this?



32 + 16 + 8 + 4 + 2 + 1 = **63**

What is the maximum number representable with 8 bits?

Let's all translate this!

Decimal		Binary
1	→	00000001
2	→	00000010
3	→	00000011
4	→	00000100
5	→	00000101
6	→	00000110
7	→	00000111
8	→	00001000

9 _ _ _ _ _ _ _ _

15 _ _ _ _ _ _ _ _

34 _ _ _ _ _ _ _ _

85 _ _ _ _ _ _ _ _

Reminder:

128	64	32	16	8	4	2	1

Can you translate this?

Decimal		Binary
1	→	00000001
2	→	00000010
3	→	00000011
4	→	00000100
5	→	00000101
6	→	00000110
7	→	00000111
8	→	00001000

12 _ _ _ _ _ _ _ _

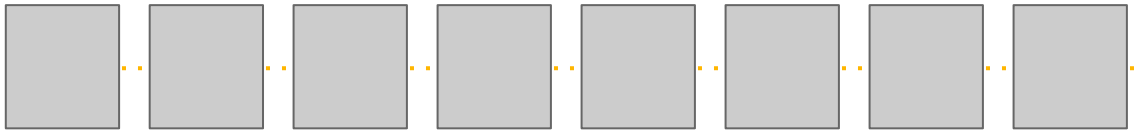
21 _ _ _ _ _ _ _ _

51 _ _ _ _ _ _ _ _

101 _ _ _ _ _ _ _ _

255 _ _ _ _ _ _ _ _

Reminder:



Ascii table

00111111 → 63

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

https://en.wikipedia.org/wiki/ASCII

From letters to ascii numbers to binary

We can encode our messages into binary just by reversing the operation.

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

A clue

01010010 01101001 01000011 01101011
00100000 01010010 01101111 01001100
01001100

Answer:

_ _ _ _ _ _ _ _ _ _ _

(Let's do the first number by hand)

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

We can convert using Python!

Let's convert from binary to decimal. How would we do that?

Assume that a binary number is represented by a string of 0's and 1's.

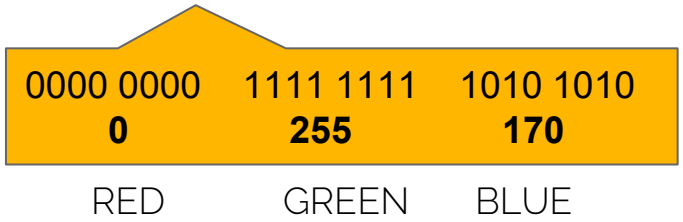
```python
# Convert binary input string to decimal

# Get binary input string (like "1011")
binary_str = input("Input a binary number: ")

# Will add up place value of each '1' bit
decimal_value = 0

# Loop through all characters
length = len(binary_str)
for i in range(length):
    # Access the next character (bit) back-to-front)
    bit_index = length - i - 1
    bit = binary_str[bit_index]

    # Add place-value of this bit
    if bit == '1':
        decimal_value += 2**i

# Print result in decimal
print(f"That equals {decimal_value}")
```

Hexadecimal

Another numbering system important in CS is hexadecimal.

You may see it in **color representations**, for example:

00FFAA



Decimal	Binary	Hexadecimal
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

More examples

All data (strings, numbers, colors, videos, music…) are expressed with bits.

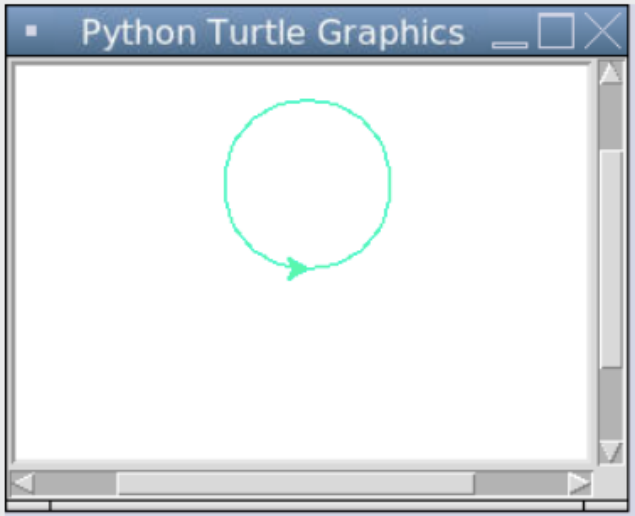
Code	example	binary	hexadecimal
ASCII	"AB"	0100000101000010	4142
ASCII	"12"	0011000100110010	3132
ASCII (note: space is coded!)	"12 AB"	0011000100110010000 00100000101000010	3132204142
Color (RGB)	Red color	111111110000000000000000	FF0000
Integer number (fixed point notation) (ex with 2 bytes, typically more)	12	0000000000001100	000C
Number with decimals (floating point notation)	123.45		

Colors in Hex

main.py ×

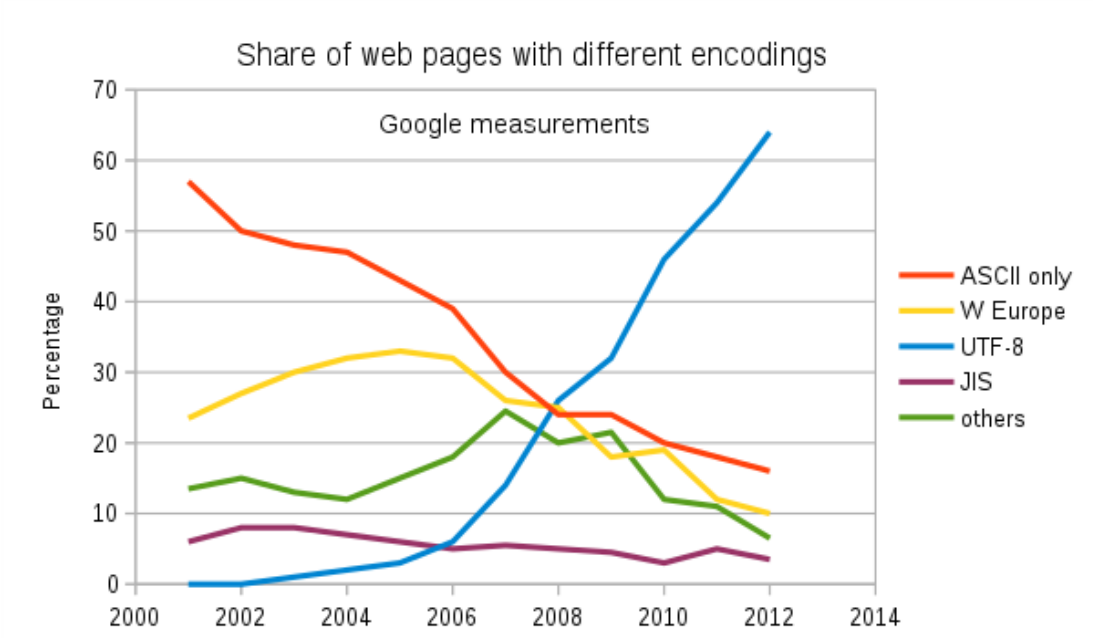
```
1   # Colors with Hex
2   # Angelica Lim
3   # Oct. 15, 2021
4
5   import turtle
6   alex = turtle.Turtle()
7
8   alex.color("#00FFAA")
9   alex.circle(40)
10
```

Green-ish



Unicode

Many languages, many characters, many encodings

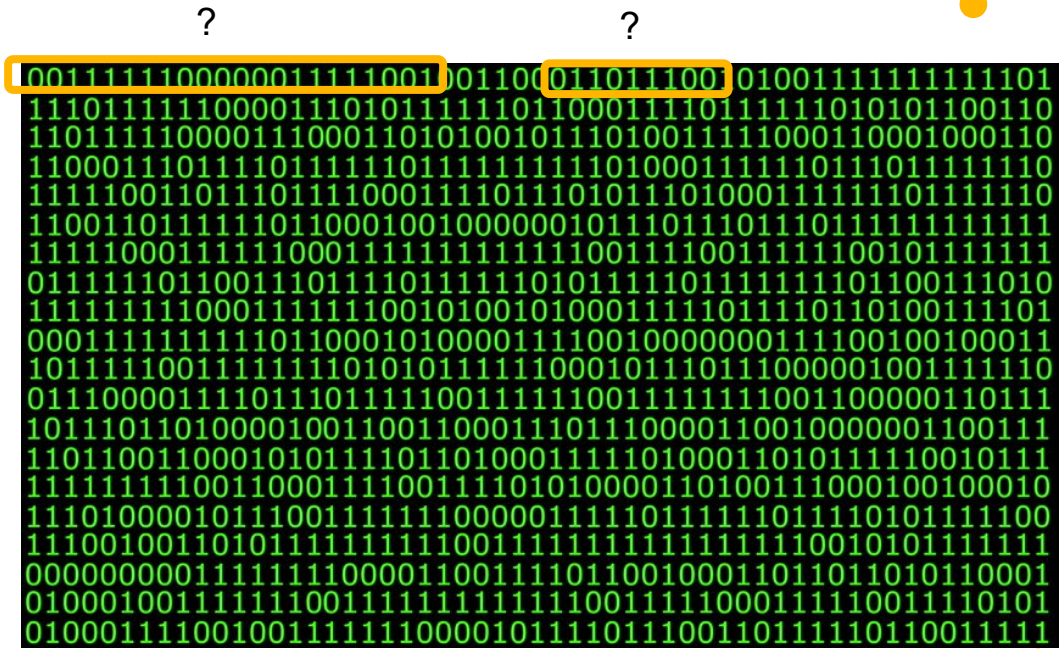


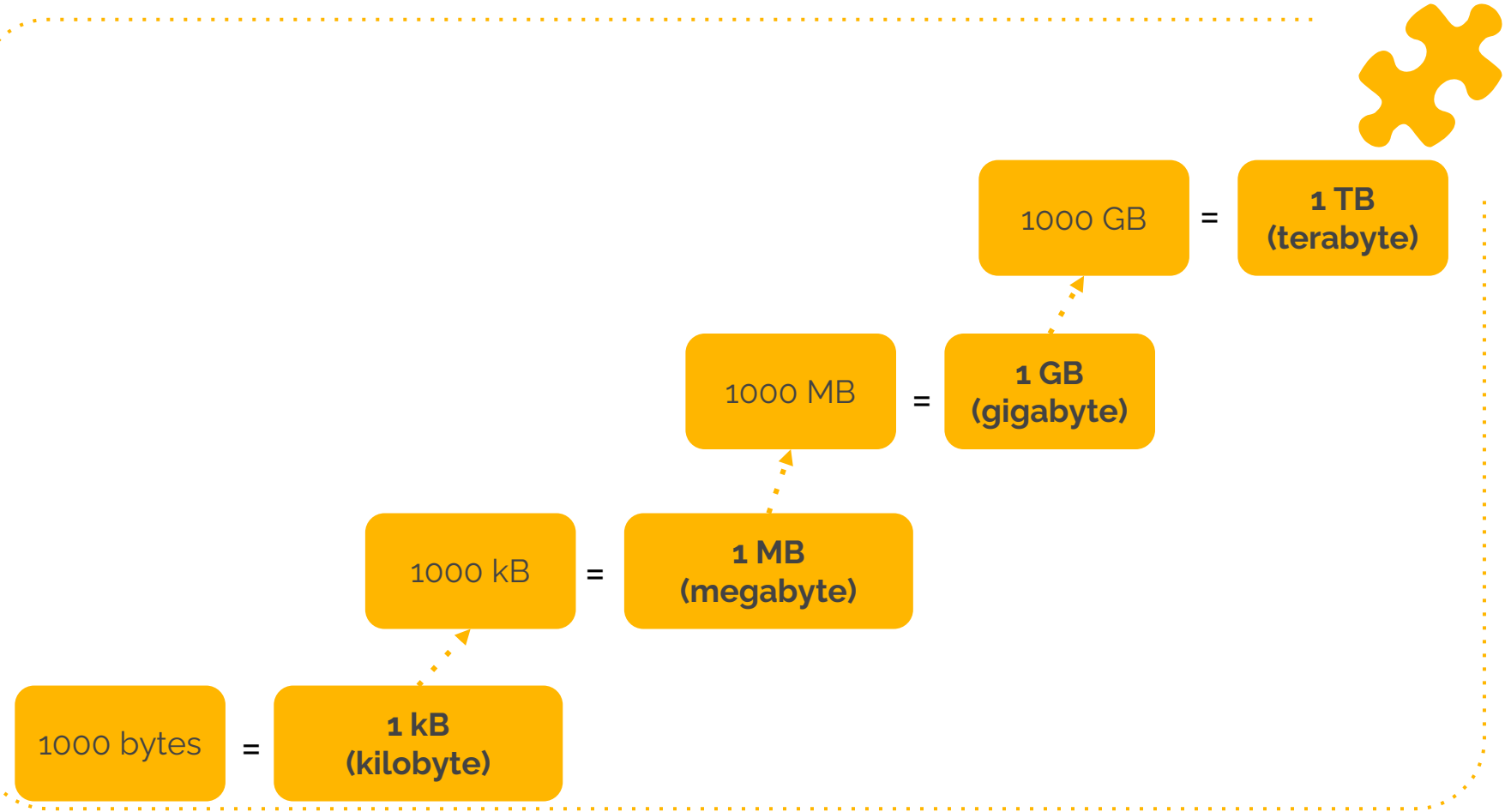
Unicode UTF-8

The maximum number of unique codes in a byte is 255 (11111111).

To type in other languages than English, we may need a lot more than 255 types of characters.

Unicode uses **up to 4 bytes** to handle more characters.




1000 bytes
=
1 kB
(kilobyte)
1000 kB
=
1 MB
(megabyte)
1000 MB
=
1 GB
(gigabyte)
1000 GB
=
1 TB
(terabyte)

iPhone X

Overview iOS Tech Specs Buy

Finish

Space Gray

Silver

Capacity[1]

64GB
256GB

These numbers look familiar!

Review

Communicating with the computer directly in binary or hexadecimal would take a lot of time and would be quite hard to read :)

Python helps translate from binary or hex into something we understand!

Grace Murray Hopper
Developer of FLOW-MATIC, the first English-like programming language in 1955, predecessor to COBOL.

Question 1

What are the numbering systems called which use:

- Only 0's and 1's?
- Digits 0-9,A-F?

a) Binary & Decimal
b) Decimal & Hexadecimal
c) Binary & Hexadecimal
d) Binary & UTF8

Question 2

What is the value of the binary number below, in decimal?

```
0101
```

a) 2

b) 5

c) 9

d) 101

Question 3

How many bits are in 2 bytes?

a) 2
b) 8
c) 16
d) 32

Question 4

How would you write the decimal number below in binary?

14

a) 0140
b) 0111
c) 1110
d) 1111