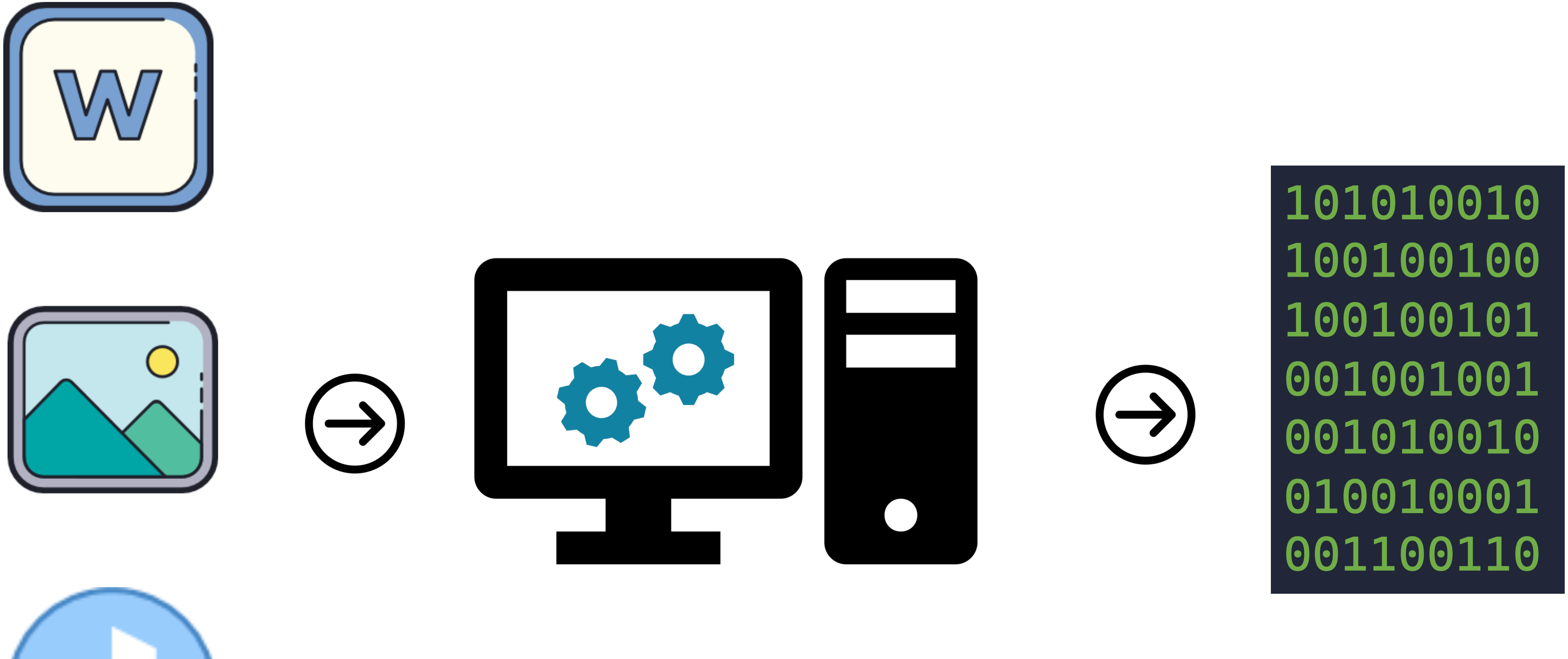
Topic 2

Data Representation

What language do computers think in?



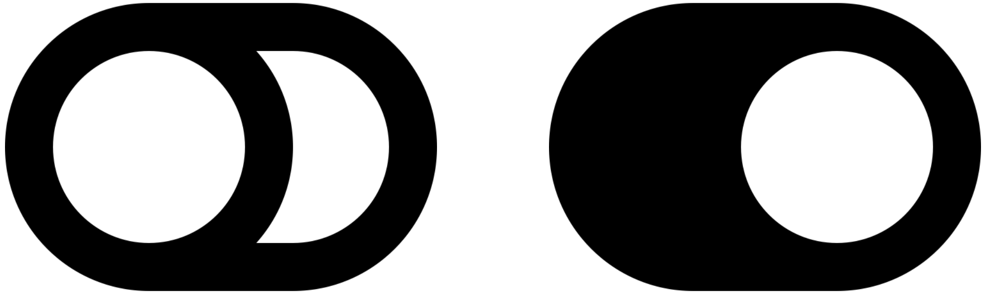
To interpret information, computers use **binary language** based in bits.

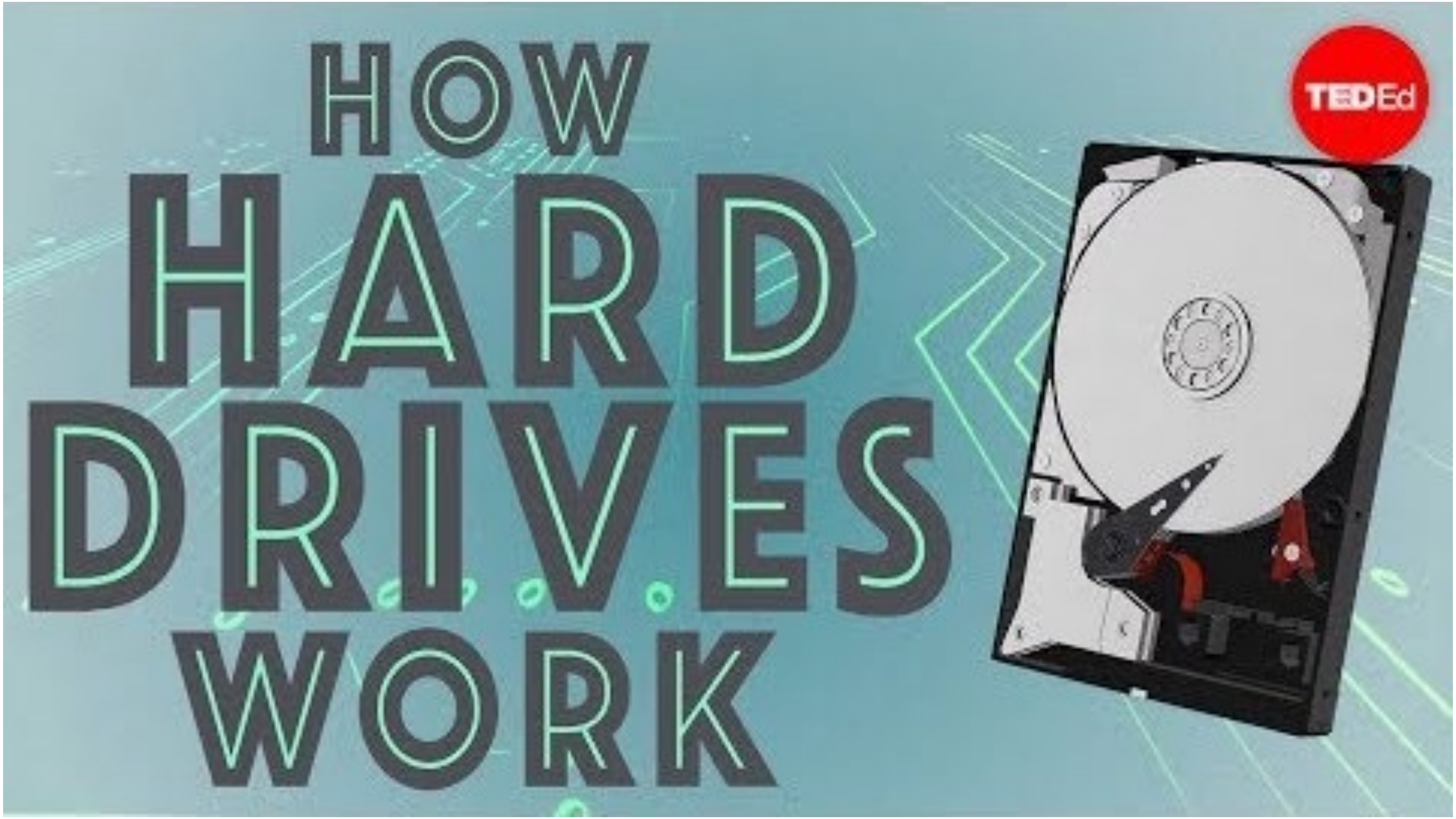
Bits

Binary digIT

A bit is the most basic unit of information in the digital world. A bit can be represented using two values: **0** or **1**.

Other representations such as **TRUE** or **FALSE**, or **on** and **off**.




HOW
HARD
DRIVES
WORK
TED Ed

Why do we use bits?

1

Transistors have two states: on / off

2

Data storage is more reliable when using 2 posible states

3

Processing of data in binary is quick and

Assume we have the following readings from a hard disk drive.

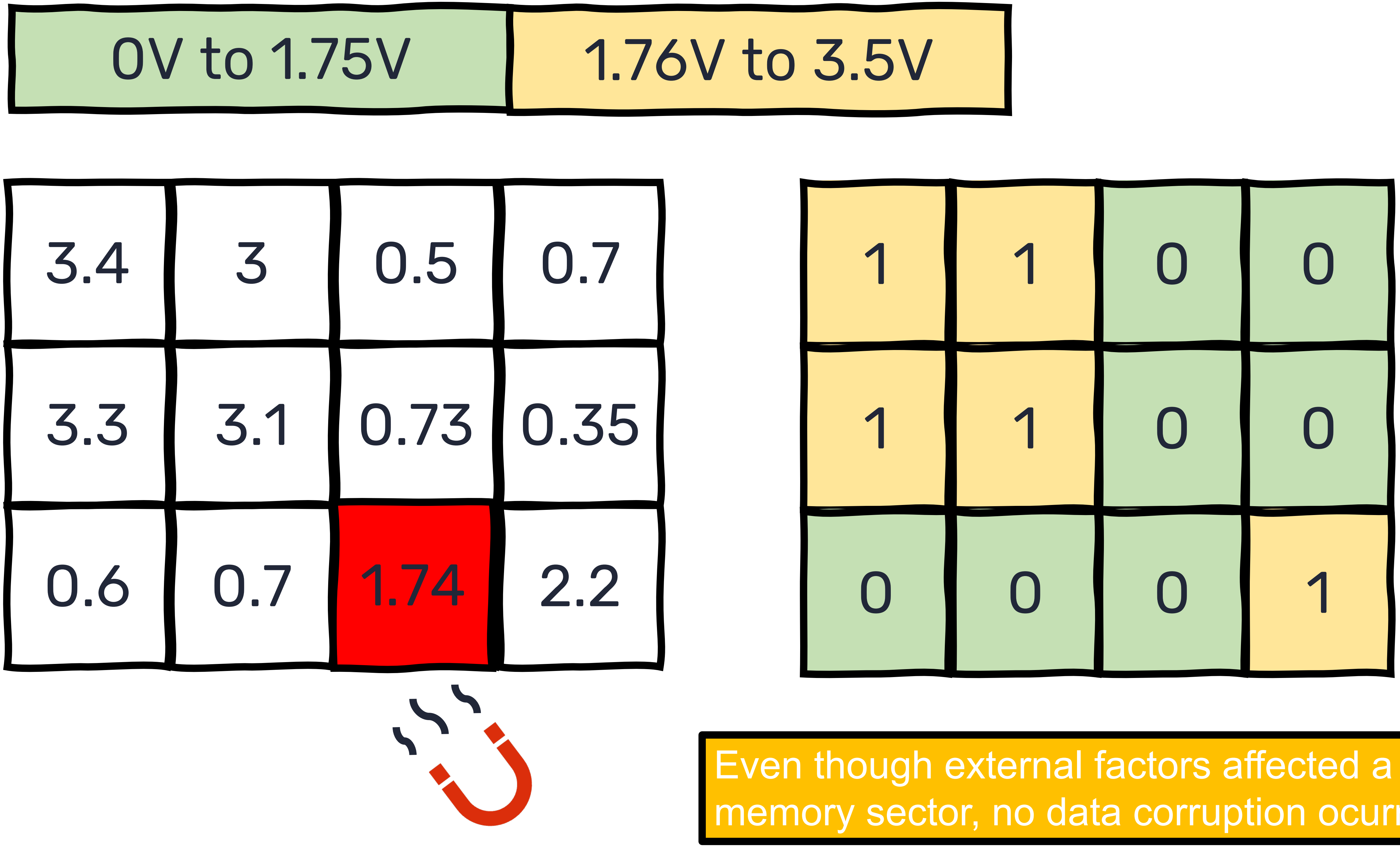
0 = 0V to 1.75V
1 = 1.76V to 3.5V

3.4	3	0.5	0.7
3.3	3.1	0.73	0.35
0.6	0.7	0.8	2.2

🟩 0V to 1.75V	🟨 1.76V to 3.5V

3.4	3	0.5	0.7
3.3	3.1	0.73	0.35
0.6	0.7	0.8	2.2

🟨 1	🟨 1	🟩 0	🟩 0
🟨 1	🟨 1	🟩 0	🟩 0
🟩 0	🟩 0	🟩 0	🟨 1



🟩 0V to 1.75V	🟨 1.76V to 3.5V

3.4	3	0.5	0.7
3.3	3.1	0.73	0.35
0.6	0.7	🟥 1.74	2.2

1	1	0	0
1	1	0	0
0	0	0	1

Even though external factors affected a memory sector, no data corruption ocurred

Now we assume we have 4 ranges:

0 = 0V to 0.75V
1 = 0.76V to 1.50V
2 = 1.51 to 2.25V
3 = 2.26V to 3.0V

1.76	2	1	0
2.5	2.2	1.2	0.9
0.8	1.6	1.74	2.2

0V to 0.75V	0.76V to 1.5V	1.51V to 2.25V	2.26V to 3.0V

1.76	2	1	0.75
2.5	1.51	1.2	0.9
0.8	1.6	1.74	2.2

2	2	1	0
3	2	1	1
1	2	2	2

0V to 0.75V	0.76V to 1.5V	1.51V to 2.25V	2.26V to 3.0V

1.76	2	1	0.75
2.5	1.51	1.2	0.9
0.8	1.6	2.26	2.2

2	2	1	0
3	2	1	1
1	2	~~2~~	2

A small interference on the data array impacts the contents of the registers. The more ranges there are, the higher risk of data corruption.

Bits y Bytes

Byte

A single bit

1

A group of 8 bits is known as a **byte**.

0	1	0	0	0	0	1	0

Internet Speed

How is data bandwidth measured by your ISP?
(Internet Service Provider)

Hasta 50 Megas
Antes ~~20 Megas~~
NETFLIX
Claro video
LLAMADAS ilimitadas
Incluye 1 línea TELMEX
Claro drive
ALMACENAMIENTO EN LA NUBE
100 GB

50 Megabits per second

↓ ↓ ↓ ↓

Divide by 8

↓ ↓ ↓ ↓

6.25 Megabytes per second

Metric

IEC
Binary

JEDEC
Binary

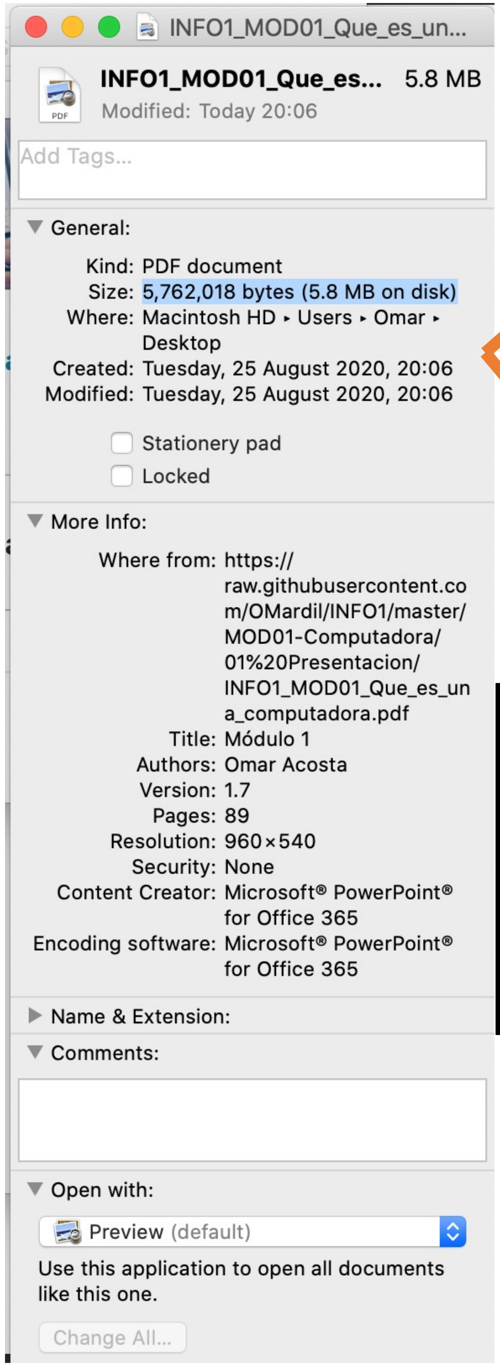
The difference in this systems is if they measure data using powers of 10, or powers of 2

Metric System

Valor (bytes)	Decimal	
1	B	byte
1000	kB	kilobyte
1000^2	MB	megabyte
1000^3	GB	gigabyte
1000^4	TB	terabyte
1000^5	PB	petabyte

Binary System

Valor (bytes)	IEC		JEDEC	
$2^0 = 1$	B	byte	B	byte
$2^{10} = 1024$	KiB	kibibyte	KB	kilobyte
2^{20}	MiB	mebibyte	MB	megabyte
2^{30}	GiB	gibibyte	GB	gigabyte
2^{40}	TiB	tebibyte	TB	terabyte
2^{50}	PiB	pebibyte	PB	petabyte



INFO1_MOD01_Que_es_un...

INFO1_MOD01_Que_es... 5.8 MB
Modified: Today 20:06

Add Tags...

▼ General:

Kind: PDF document
Size: 5,762,018 bytes (5.8 MB on disk)
Where: Macintosh HD ▸ Users ▸ Omar ▸ Desktop
Created: Tuesday, 25 August 2020, 20:06
Modified: Tuesday, 25 August 2020, 20:06

□ Stationery pad
□ Locked

▼ More Info:

Where from: https://raw.githubusercontent.com/OMardil/INFO1/master/MOD01-Computadora/01%20Presentacion/INFO1_MOD01_Que_es_una_computadora.pdf
Title: Módulo 1
Authors: Omar Acosta
Version: 1.7
Pages: 89
Resolution: 960×540
Security: None
Content Creator: Microsoft® PowerPoint® for Office 365
Encoding software: Microsoft® PowerPoint® for Office 365

▶ Name & Extension:

▼ Comments:

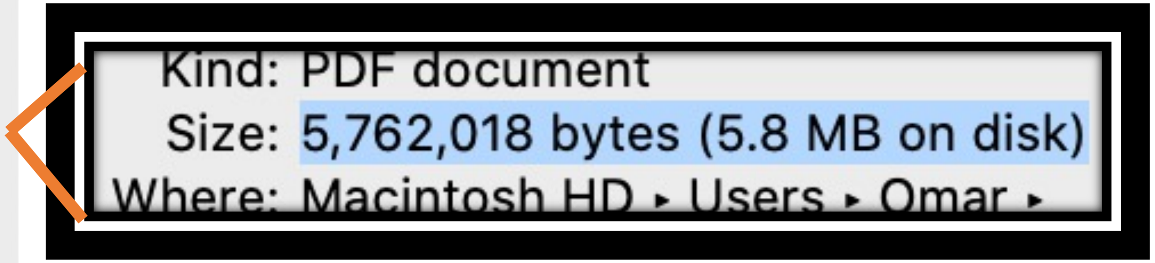
▼ Open with:

Preview (default)

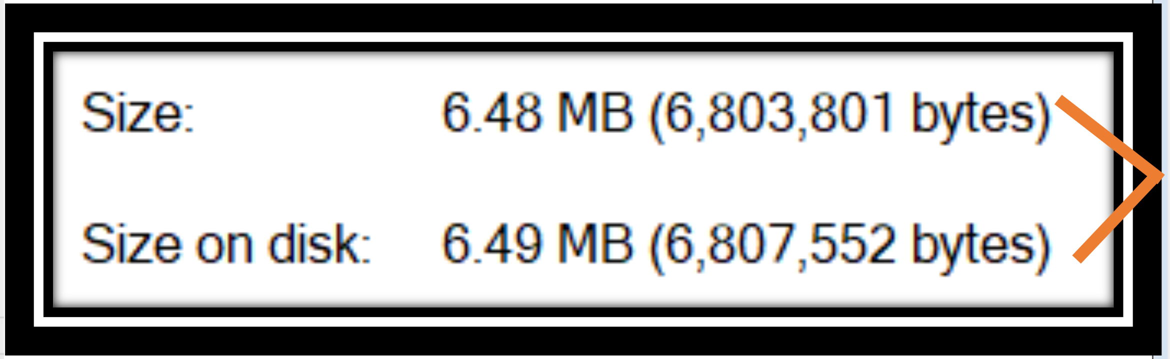
Use this application to open all documents like this one.

Change All...

Mac OS uses the metric system (powers of 10) to measure file sizes

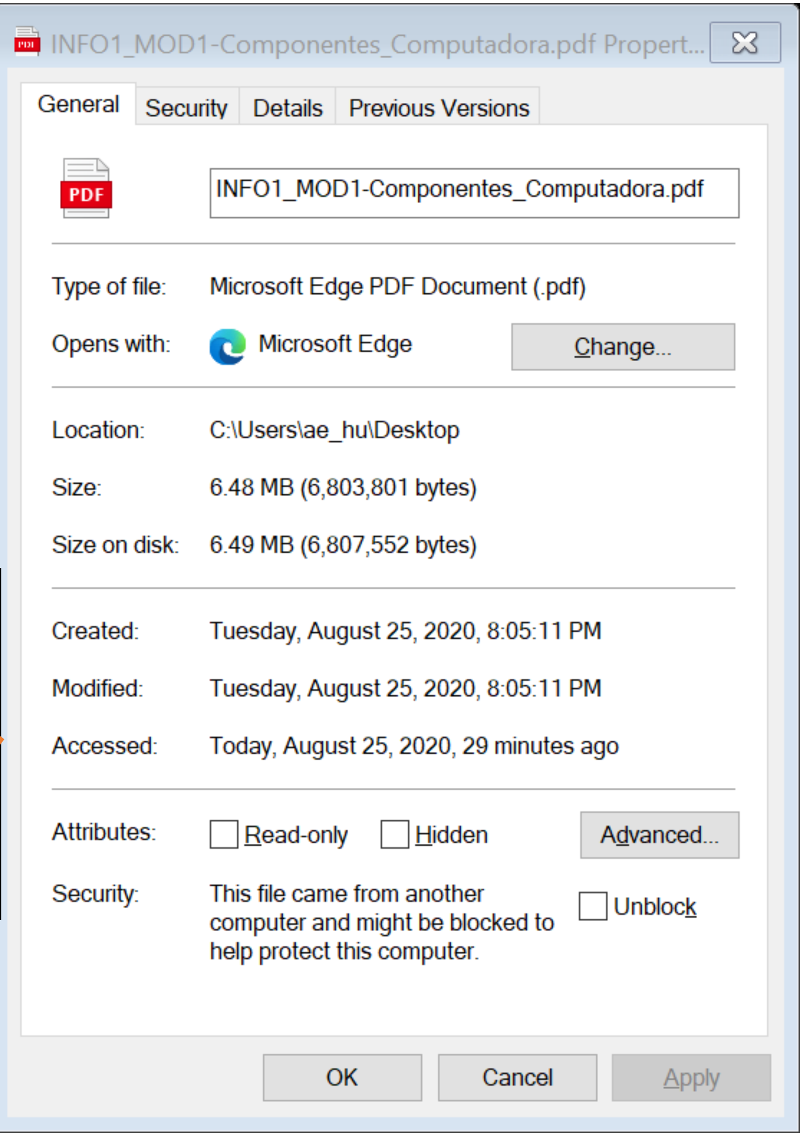


Kind: PDF document
Size: 5,762,018 bytes (5.8 MB on disk)
Where: Macintosh HD ▸ Users ▸ Omar ▸



Size: 6.48 MB (6,803,801 bytes)
Size on disk: 6.49 MB (6,807,552 bytes)

Windows uses JEDEC binary system (powers of 2) to measure file sizes



INFO1_MOD1-Componentes_Computadora.pdf Propert...

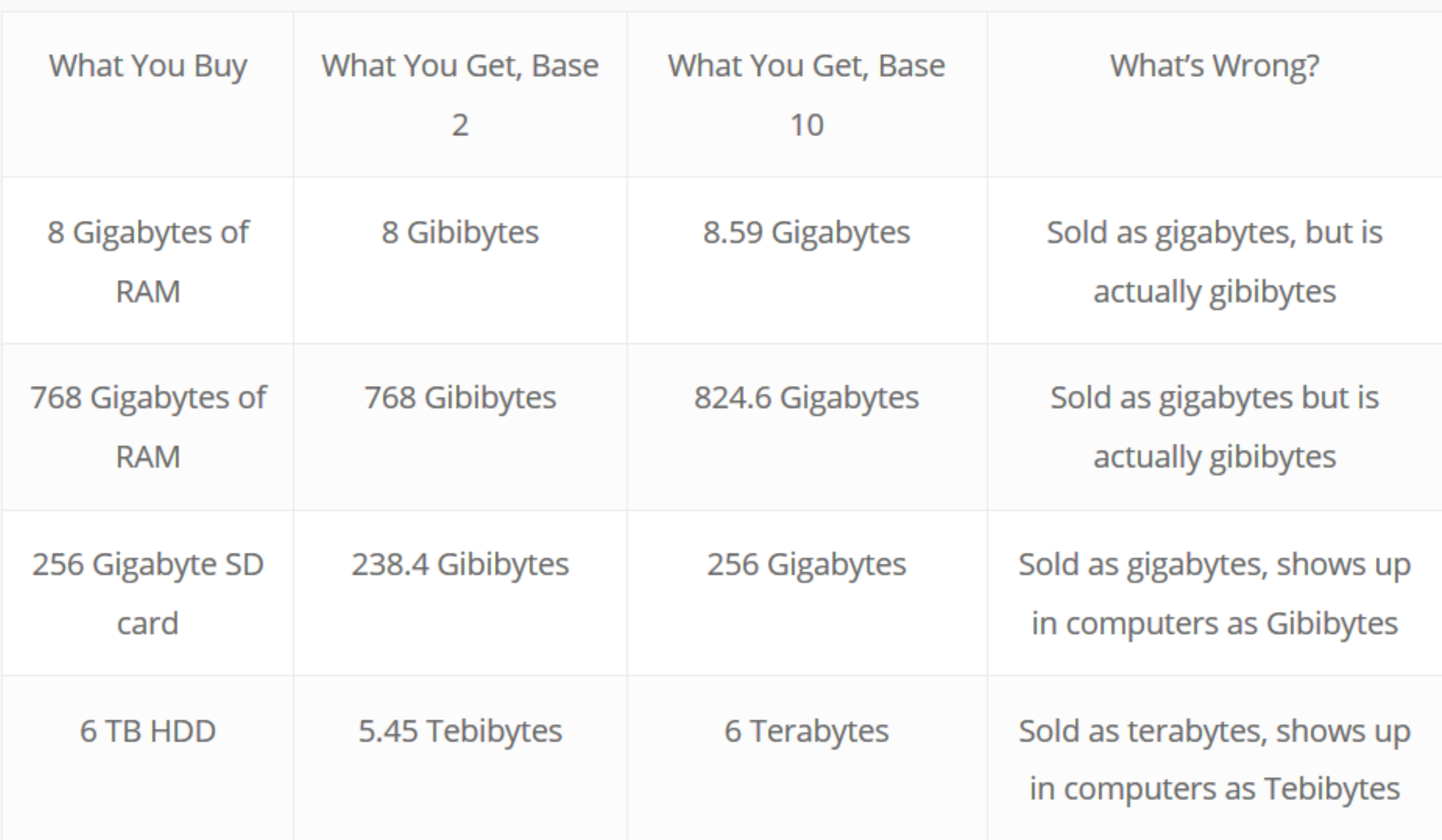
INFO1_MOD1-Componentes_Computadora.pdf

Type of file: Microsoft Edge PDF Document (.pdf)
Opens with: Microsoft Edge Change...

Location: C:\Users\ae_hu\Desktop
Size: 6.48 MB (6,803,801 bytes)
Size on disk: 6.49 MB (6,807,552 bytes)

Created: Tuesday, August 25, 2020, 8:05:11 PM
Modified: Tuesday, August 25, 2020, 8:05:11 PM
Accessed: Today, August 25, 2020, 29 minutes ago

Attributes: □ Read-only □ Hidden Advanced...
Security: This file came from another computer and might be blocked to help protect this computer. □ Unblock



What You Buy	What You Get, Base 2	What You Get, Base 10	What's Wrong?
8 Gigabytes of RAM	8 Gibibytes	8.59 Gigabytes	Sold as gigabytes, but is actually gibibytes
768 Gigabytes of RAM	768 Gibibytes	824.6 Gigabytes	Sold as gigabytes but is actually gibibytes
256 Gigabyte SD card	238.4 Gibibytes	256 Gigabytes	Sold as gigabytes, shows up in computers as Gibibytes
6 TB HDD	5.45 Tebibytes	6 Terabytes	Sold as terabytes, shows up in computers as Tebibytes

Characters

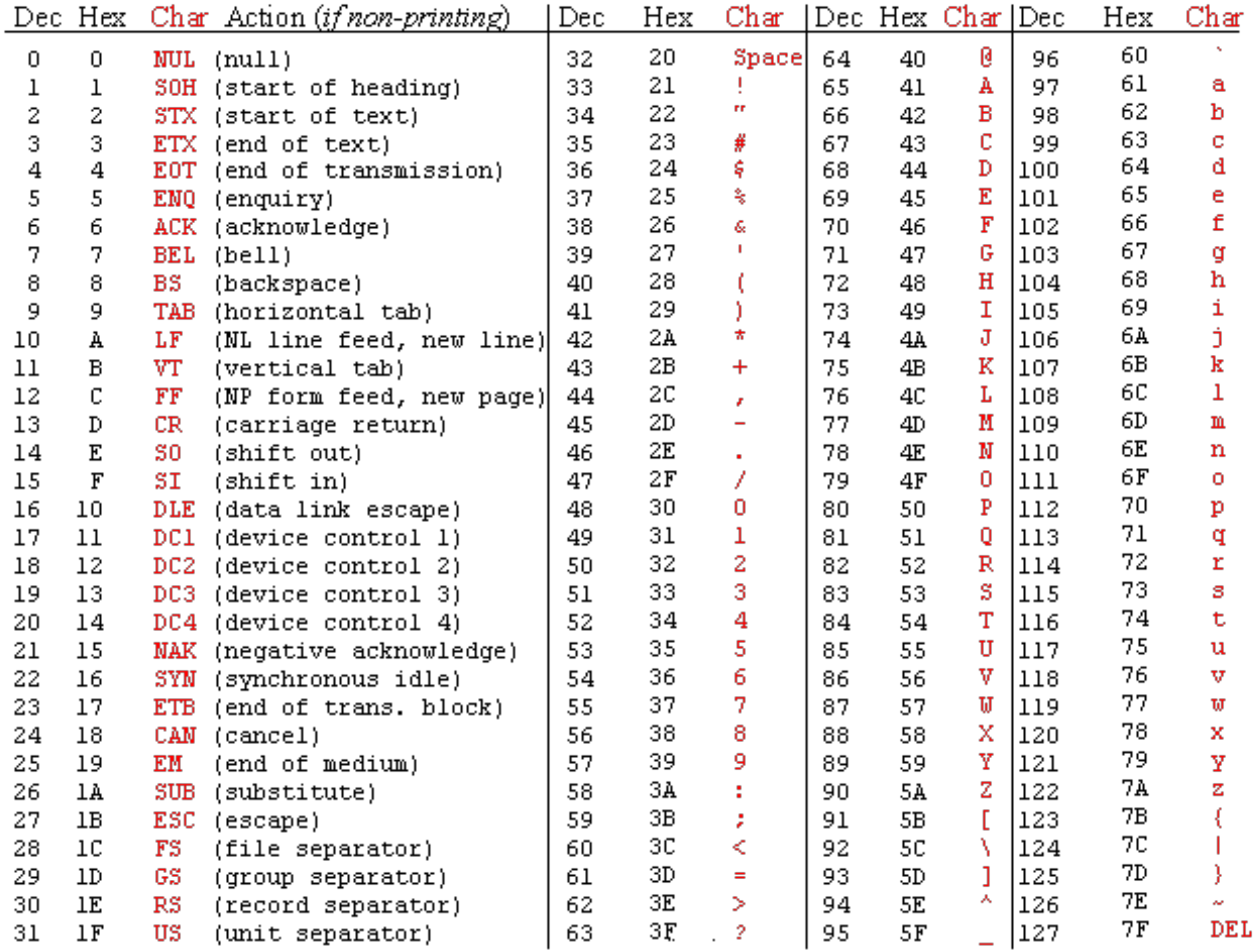
Characters

Computers use **character sets** to transmit, store, and process information. Each symbol within a character set is denominated a **character**.

Each character represents something different:

- Sound
- Pause
- Number
- Feeling
- etc.

Characters



Dec	Hex	Char	Action (*if non-printing*)	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	(null)	32	20	Space	64	40	@	96	60	`
1	1	SOH	(start of heading)	33	21	!	65	41	A	97	61	a
2	2	STX	(start of text)	34	22	"	66	42	B	98	62	b
3	3	ETX	(end of text)	35	23	#	67	43	C	99	63	c
4	4	EOT	(end of transmission)	36	24	$	68	44	D	100	64	d
5	5	ENQ	(enquiry)	37	25	%	69	45	E	101	65	e
6	6	ACK	(acknowledge)	38	26	&	70	46	F	102	66	f
7	7	BEL	(bell)	39	27	'	71	47	G	103	67	g
8	8	BS	(backspace)	40	28	(	72	48	H	104	68	h
9	9	TAB	(horizontal tab)	41	29	)	73	49	I	105	69	i
10	A	LF	(NL line feed, new line)	42	2A	*	74	4A	J	106	6A	j
11	B	VT	(vertical tab)	43	2B	+	75	4B	K	107	6B	k
12	C	FF	(NP form feed, new page)	44	2C	,	76	4C	L	108	6C	l
13	D	CR	(carriage return)	45	2D	-	77	4D	M	109	6D	m
14	E	SO	(shift out)	46	2E	.	78	4E	N	110	6E	n
15	F	SI	(shift in)	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	(data link escape)	48	30	0	80	50	P	112	70	p
17	11	DC1	(device control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2	(device control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3	(device control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4	(device control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK	(negative acknowledge)	53	35	5	85	55	U	117	75	u
22	16	SYN	(synchronous idle)	54	36	6	86	56	V	118	76	v
23	17	ETB	(end of trans. block)	55	37	7	87	57	W	119	77	w
24	18	CAN	(cancel)	56	38	8	88	58	X	120	78	x
25	19	EM	(end of medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB	(substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	(escape)	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	(file separator)	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	(group separator)	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	(record separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US	(unit separator)	63	3F	?	95	5F	_	127	7F	DEL

Alphabetic Characters

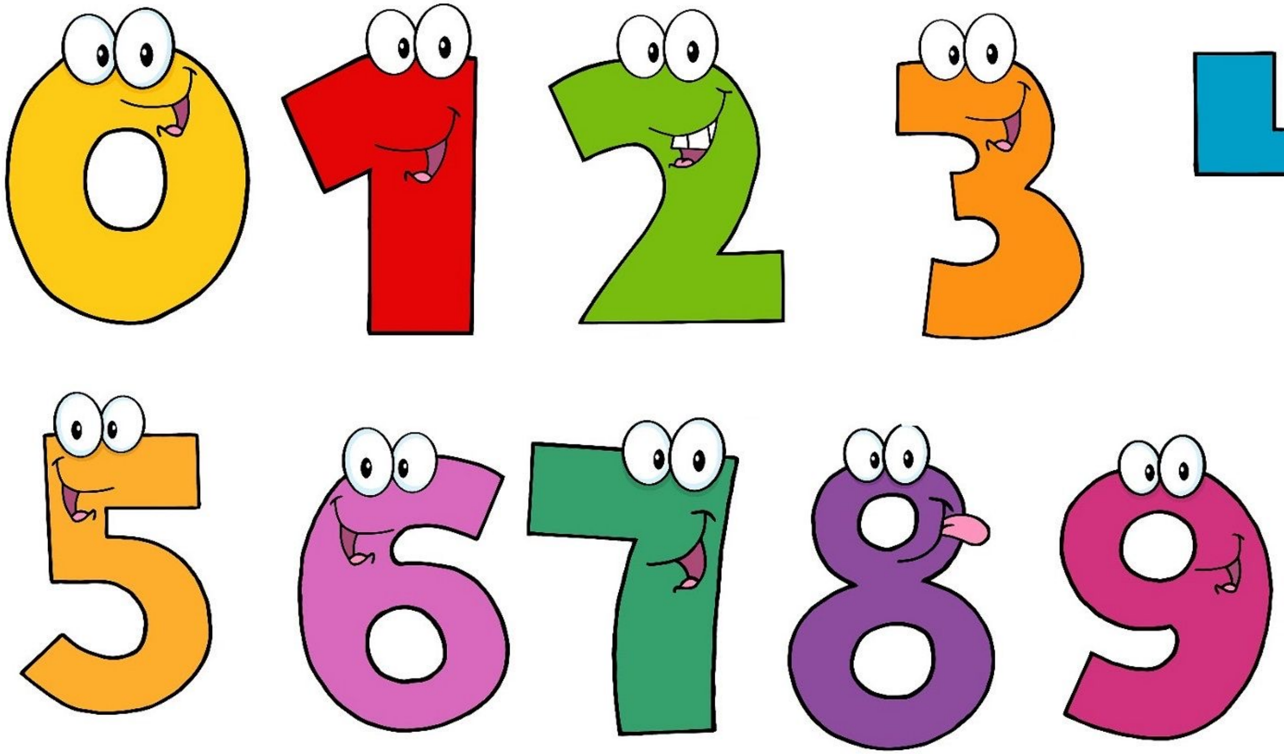
Represents all 26 letters of the English alphabet, in upper and lower case.

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

a b c d e f g h i j k l m n
o p q r s t u v w x y z

Numeric Characters

Represents numbers using the decimal base. From “0” to “9”.



Special Characters

Special characters from other languages (i.e., Spanish, French, German). They also include characters that transmit emphasis, questions, enumerations, among others.

- SPACE
- () {} [] < >
- @
- ¡ ! ¿ ?
- , . ;
- = + - / * %
- Ñ ß á ë î

Control Characters

Characters with no graphic representation serve as control characters to indicate tabs, line feeds, segments, etc.

- TAB
- NULL
- BELL
- Backspace
- Line Feed
- Carriage Return

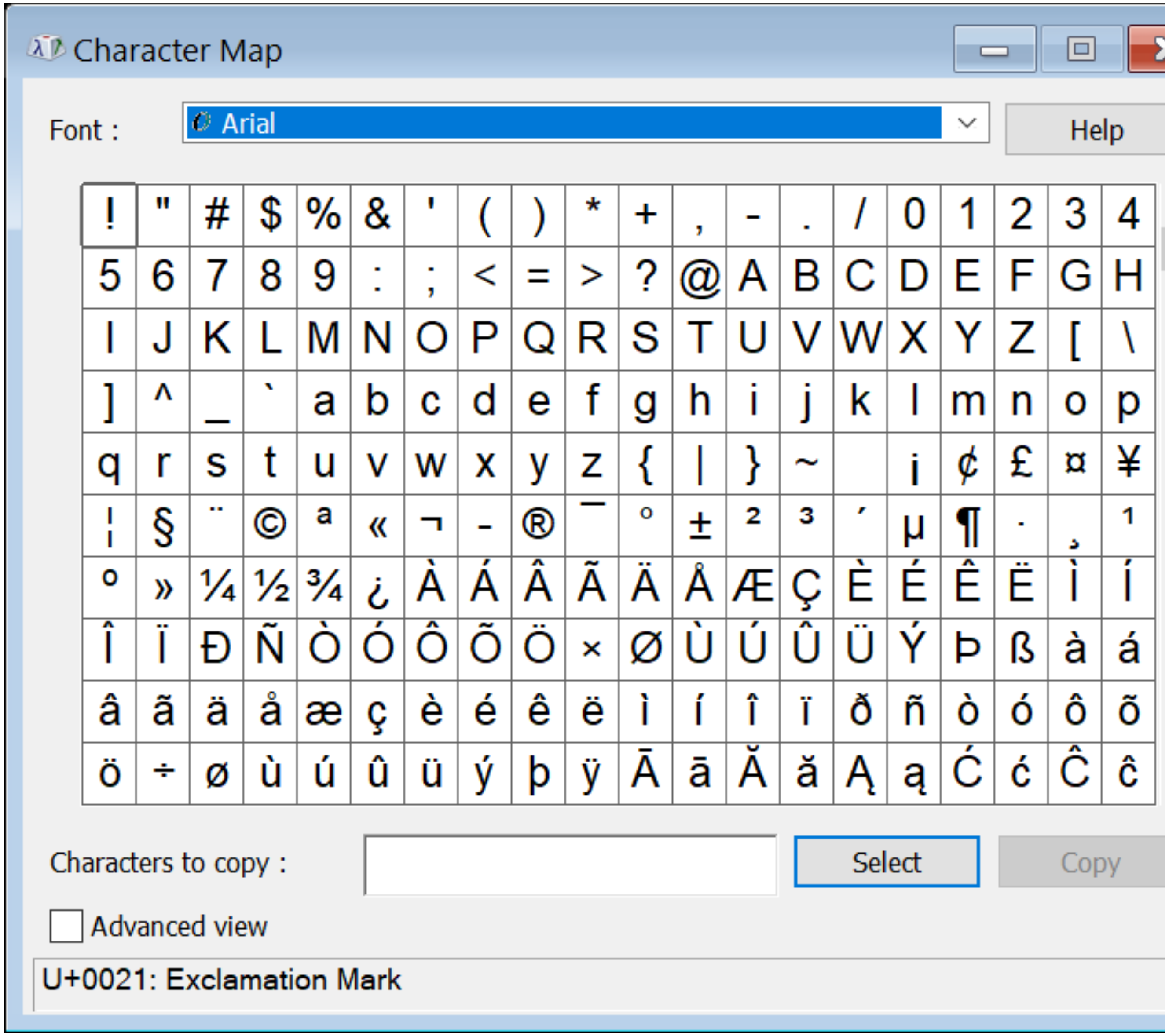
Graphic Characters

- ® ฿ ℗ ☺
- →𣩠☞⇏➴⤧☃️★
- Emojis

😀 😃 😄 😁 😆 😅 😂
😌 😍 🥰 😘 😗 😙 😚
🤓 😎 🤩 🥳 😏 😒 😞
😫 😩 🥺 😢 😭 😤 😠
😨 😰 😥 😓 🤗 🤔 🤭
😯 😦 😧 😮 😲 🥱 😴

Character Map

In Windows, you can open Character Map to view the different characters



Font : Arial

!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4
5	6	7	8	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\
]	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x	y	z	{	|	}	~		¡	¢	£	¤	¥
¦	§	¨	©	ª	«	¬	-	®	¯	°	±	²	³	´	µ	¶	·	¸	¹
º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í
Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á
â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ	ò	ó	ô	õ
ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ

Characters to copy :

☐ Advanced view

U+0021: Exclamation Mark

Decimal System

Decimal System

To represent numbers, us humans use the decimal system

This means each position can represent up to **10 different possible values**.

Smallest number	Largest number
0	9

Decimal System

To indicate a number is using the decimal system, we can add 10 as a subscript.

$$7_{10}$$
$$257_{10}$$
$$100_{10}$$

0 ← **Start with smallest number that we can represent with 1 digit**

1

2

3

4

5

6

7

8

9 ← **We reach the largest number that can be represented with 1 digit**

10 ← **We add 1 more digit and start from the smallest number we can represent using two digits**

12

13

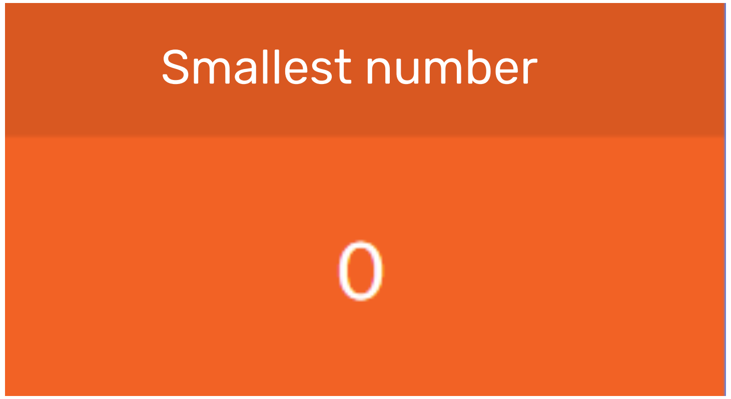
14

15

Binary System

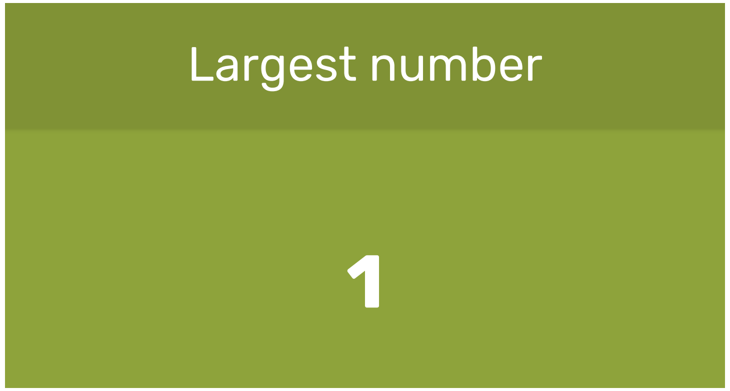
Binary System

Computers represent information using the binary system. Binary has a base of 2, which means each digit can represent **two possible values**.



Smallest number

0



Largest number

1

Binary System

To indicate we are using binary, we specify the base 2 as a subscript.

$$10100010101_2$$
$$11111111_2$$
$$1010_2$$

0 ← Start with one digit, with the smallest value in binary

1 ← Largest binary number using one digit

10 ← Increase total number of digits to two

11

100 ← Increment total number of digits to three

101

110

111

1000 ← Increment total number of digits to four

1001

1010

1011

1100

1101

1110

1111

Binary to Decimal Conversions

Binary to Decimal Conversion

1. Given a binary number:

$$110100_2$$

2. We begin by numbering the positions of each digit right to left, starting from 0:

digits	1	1	0	1	0	0
positions	5	4	3	2	1	0

3. Now we add each digit multiplied by the base (2) elevated to its position

$$110100_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1$$

$$110100_2 = 32_{10} + 16_{10} + 0_{10} + 4_{10} + 0_{10} + 0_{10}$$

$$\boxed{\mathbf{110100_2 = 52_{10}}}$$

Another Example:

$$10101011_2$$

1	0	1	0	1	0	1	1
7	6	5	4	3	2	1	0

$$10101011_2 = 1 \cdot 2^7 + \cancel{0} \cdot 2^6 + 1 \cdot 2^5 + \cancel{0} \cdot 2^4 + 1 \cdot 2^3 + \cancel{0} \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$10101011_2 = 1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$10101011_2 = 128_{10} + 32_{10} + 8_{10} + 2_{10} + 1_{10}$$

$$\boxed{10101011_2 = 171_{10}}$$

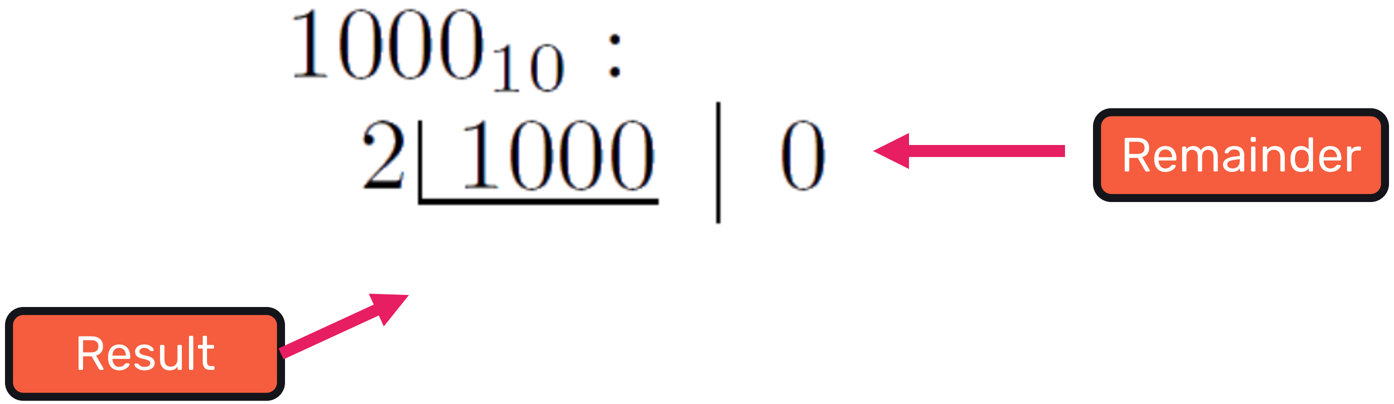
Decimal to Binary Conversion

1000_{10} :

Decimal to Binary Conversion

We divide the decimal number by 2, which is the desired base (base).

Decimal to Binary Conversion



$1000_{10}:$

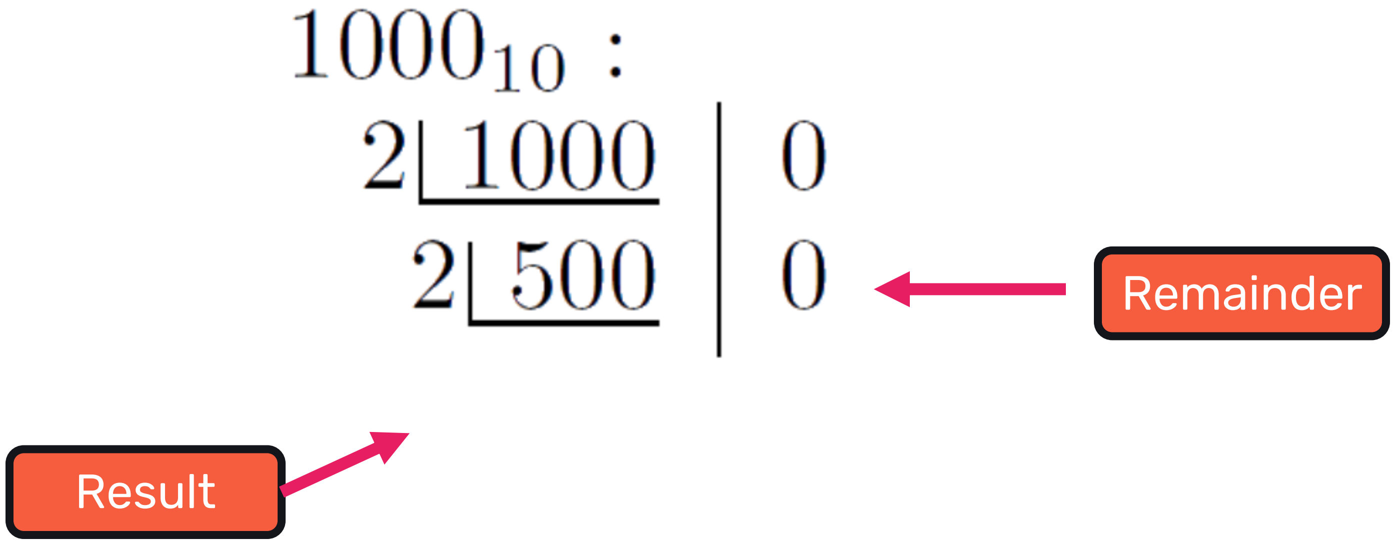
$$2\underline{\lfloor 1000} \;\big|\; 0$$

Remainder

Result

We write down the remainder and the result of the division

Decimal to Binary Conversion



$1000_{10}:$

$$\begin{array}{r|l|c} 2 & 1000 & 0 \\ 2 & 500 & 0 \end{array}$$

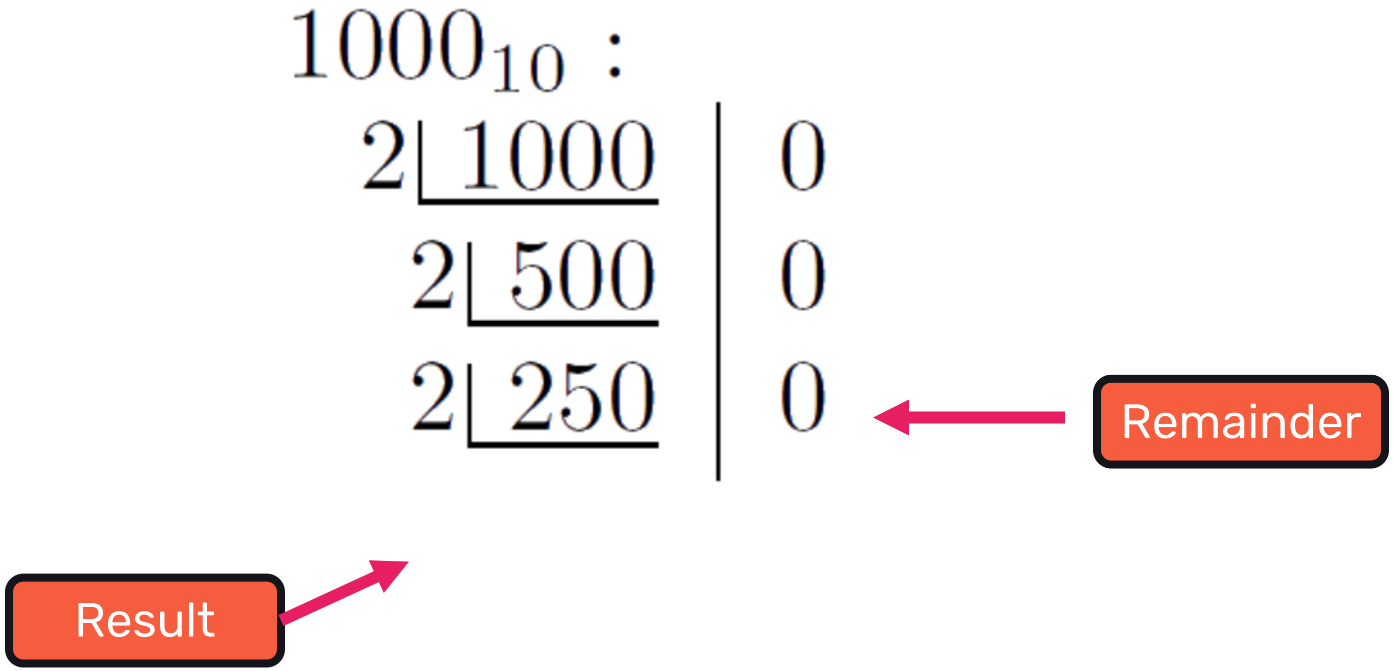
Remainder

Result

We repeat the operation, we write down the remainder and the result

Decimal to Binary Conversion

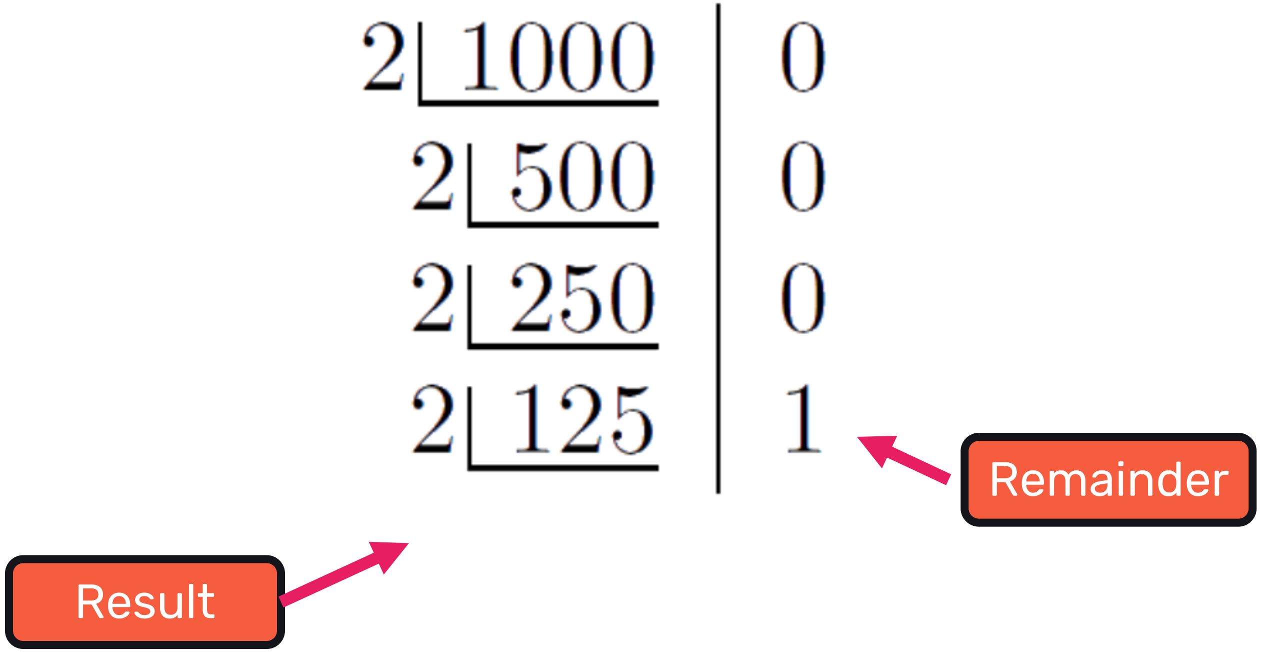
1000_{10} :



$$\begin{array}{r|l} 2 \,\underline{\big|\, 1000} & 0 \\ 2 \,\underline{\big|\, 500} & 0 \\ 2 \,\underline{\big|\, 250} & 0 \end{array}$$

Decimal to Binary Conversion

$1000_{10}:$



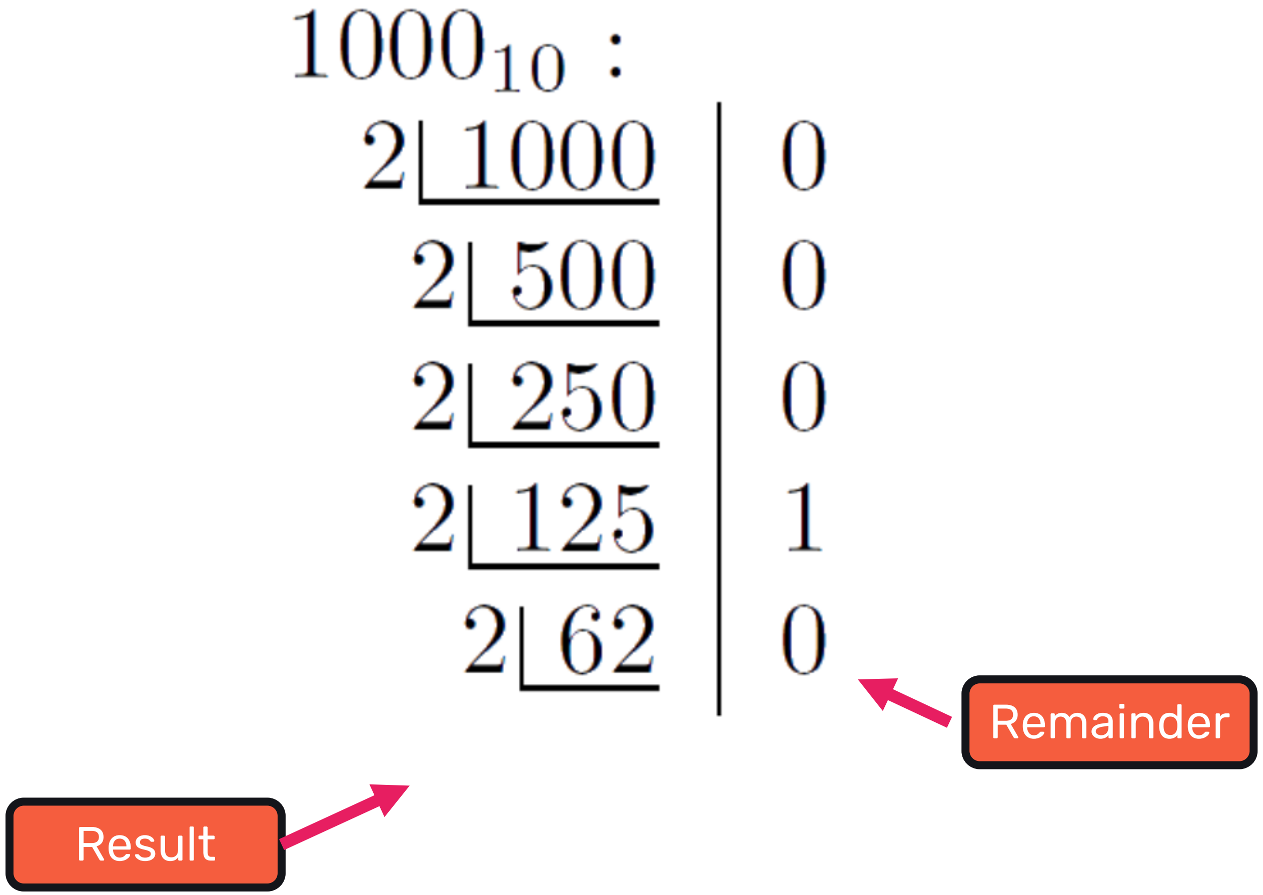
$2\lfloor 1000$	0
$2\lfloor 500$	0
$2\lfloor 250$	0
$2\lfloor 125$	1

Remainder

Result

The result of the integer division 125/2 is 62, with a remainder of 1.

Decimal to Binary Conversion



$1000_{10}:$

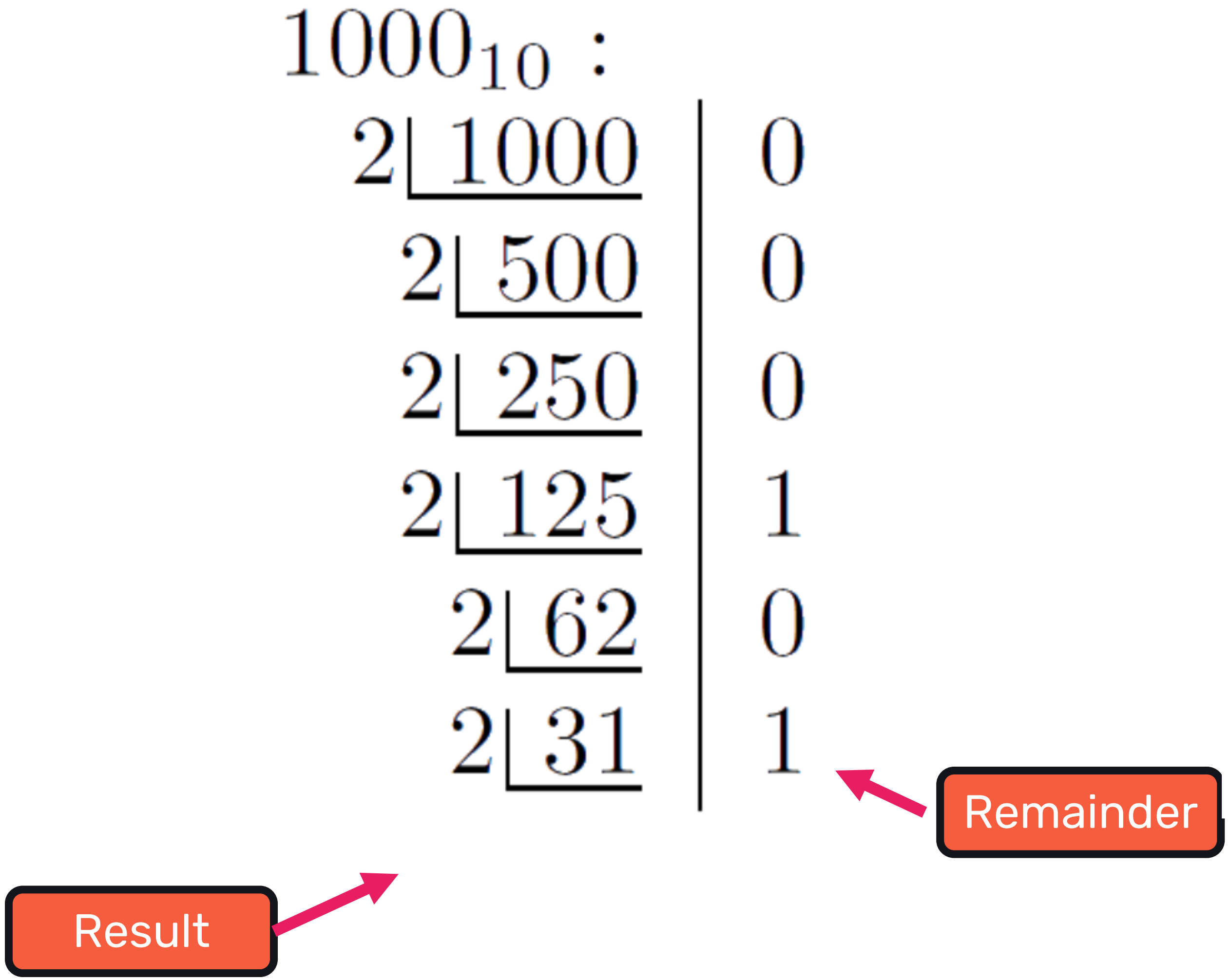
$$\begin{array}{r|c}
2 \,\lfloor\, 1000 & 0 \\
2 \,\lfloor\, 500 & 0 \\
2 \,\lfloor\, 250 & 0 \\
2 \,\lfloor\, 125 & 1 \\
2 \,\lfloor\, 62 & 0
\end{array}$$

Remainder

Result

We keep repeating this operation

Decimal to Binary Conversion



$1000_{10}:$

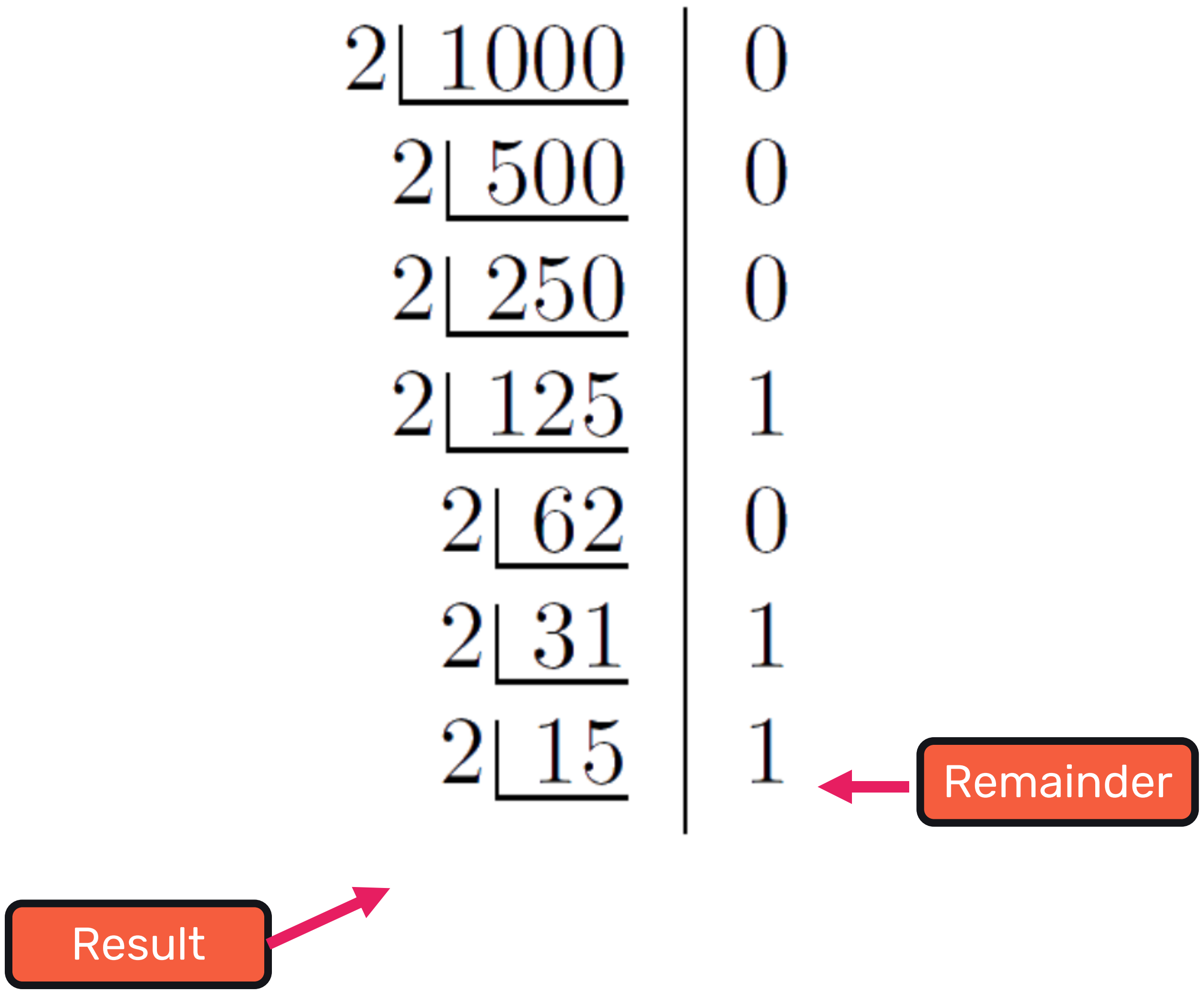
2	1000	0
2	500	0
2	250	0
2	125	1
2	62	0
2	31	1

Remainder

Result

Decimal to Binary Conversion

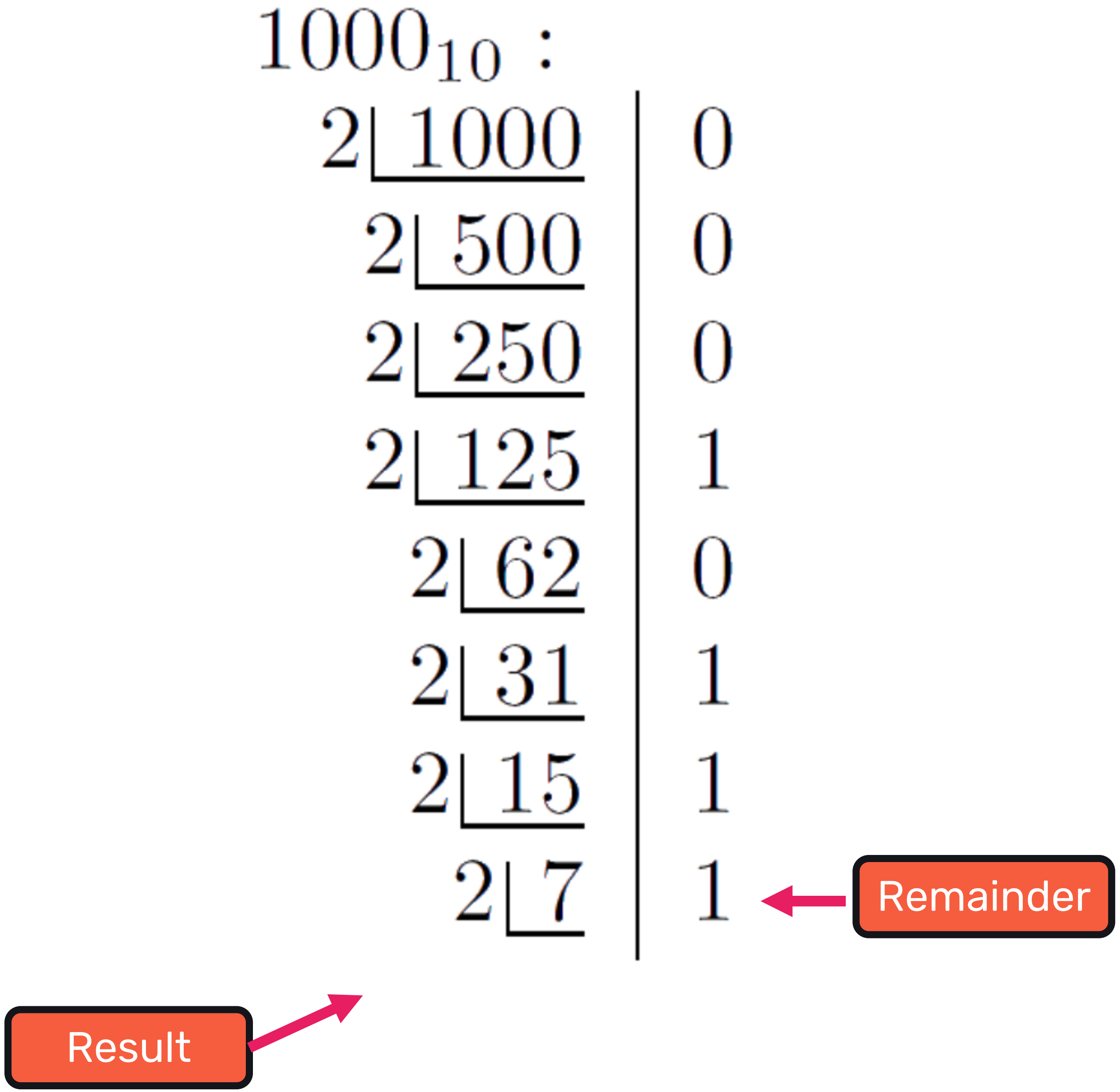
1000_{10} :



$2 \lfloor 1000$	0
$2 \lfloor 500$	0
$2 \lfloor 250$	0
$2 \lfloor 125$	1
$2 \lfloor 62$	0
$2 \lfloor 31$	1
$2 \lfloor 15$	1

Decimal to Binary Conversion

$1000_{10}:$



$$\begin{array}{r|c}
2\underline{\lfloor\,1000} & 0 \\
2\underline{\lfloor\,500} & 0 \\
2\underline{\lfloor\,250} & 0 \\
2\underline{\lfloor\,125} & 1 \\
2\underline{\lfloor\,62} & 0 \\
2\underline{\lfloor\,31} & 1 \\
2\underline{\lfloor\,15} & 1 \\
2\underline{\lfloor\,7} & 1
\end{array}$$

Decimal to Binary Conversion

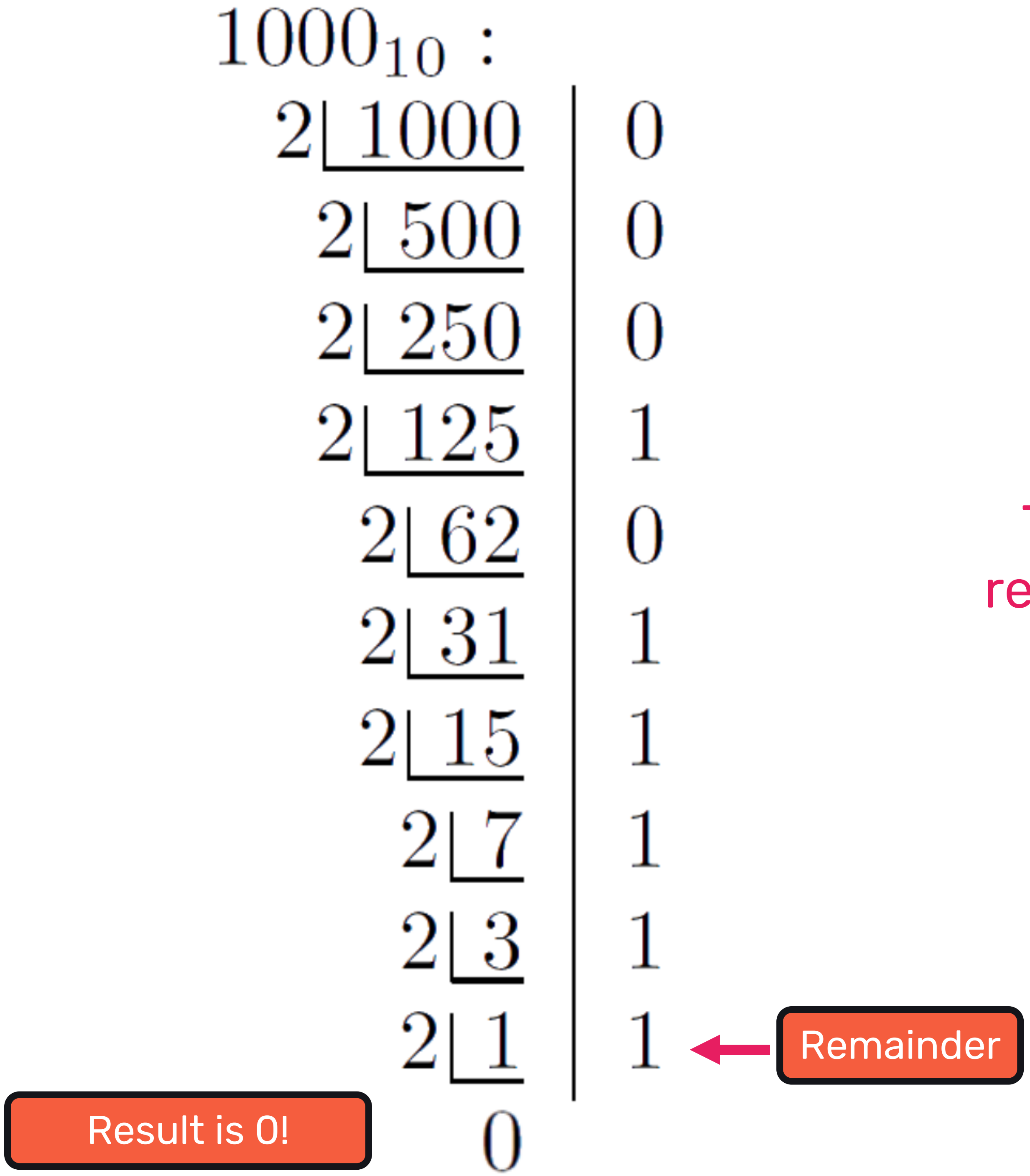
$$1000_{10}:$$

$$\begin{array}{r|l}
2\underline{\lfloor 1000} & 0 \\
2\underline{\lfloor 500} & 0 \\
2\underline{\lfloor 250} & 0 \\
2\underline{\lfloor 125} & 1 \\
2\underline{\lfloor 62} & 0 \\
2\underline{\lfloor 31} & 1 \\
2\underline{\lfloor 15} & 1 \\
2\underline{\lfloor 7} & 1 \\
2\underline{\lfloor 3} & 1
\end{array}$$

Remainder

Result

Decimal to Binary Conversion



1000_{10} :

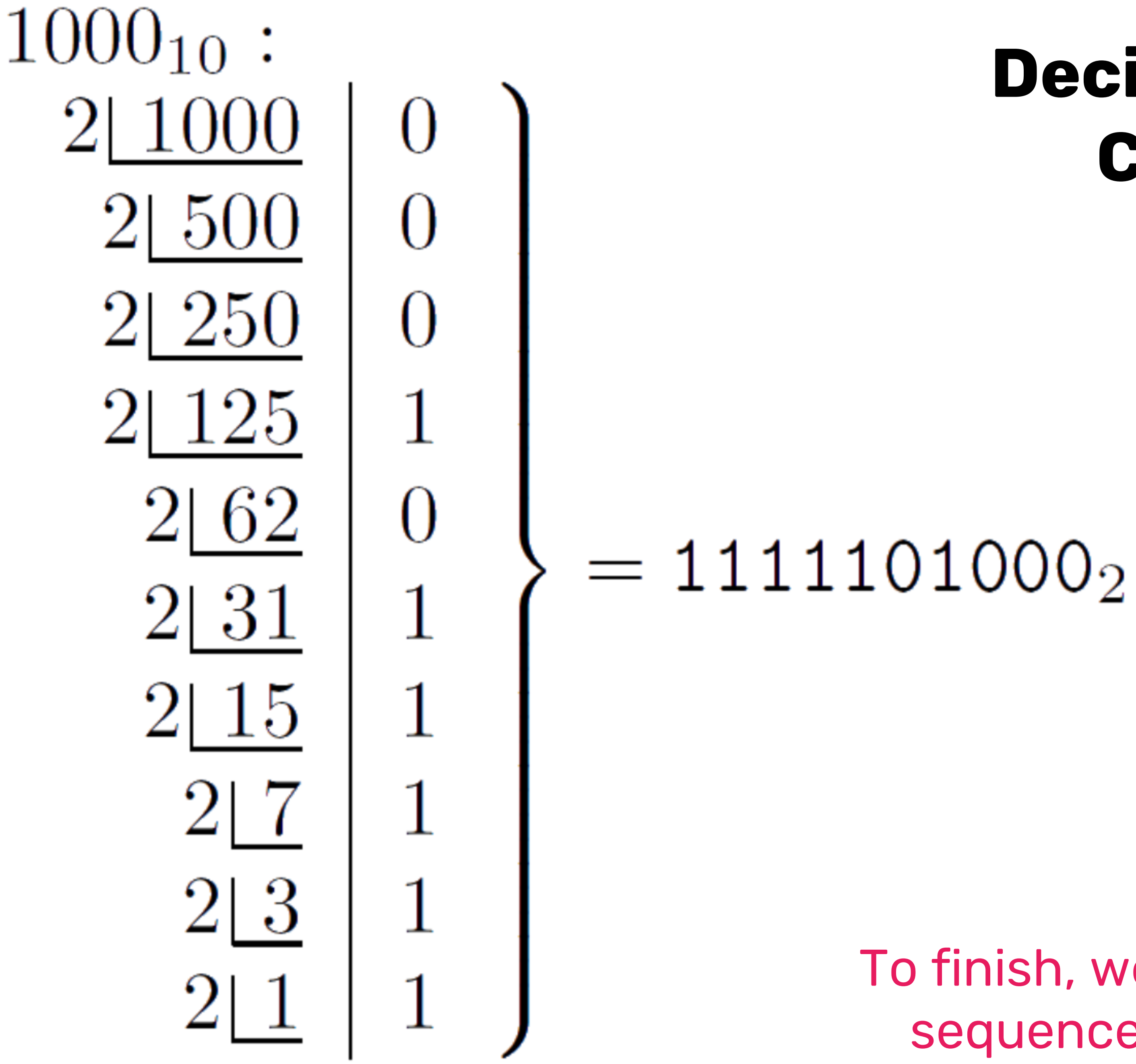
2	1000	0
2	500	0
2	250	0
2	125	1
2	62	0
2	31	1
2	15	1
2	7	1
2	3	1
2	1	1
	0	

Remainder

Result is 0!

The integer division 1/2 = 0, with a remainder of 1. We write the remainer and finish.

Decimal to Binary Conversion



$$1000_{10}:$$

$$\left.\begin{array}{r|c}
2\,\underline{\lfloor\, 1000} & 0 \\
2\,\underline{\lfloor\, 500} & 0 \\
2\,\underline{\lfloor\, 250} & 0 \\
2\,\underline{\lfloor\, 125} & 1 \\
2\,\underline{\lfloor\, 62} & 0 \\
2\,\underline{\lfloor\, 31} & 1 \\
2\,\underline{\lfloor\, 15} & 1 \\
2\,\underline{\lfloor\, 7} & 1 \\
2\,\underline{\lfloor\, 3} & 1 \\
2\,\underline{\lfloor\, 1} & 1
\end{array}\right\} = 1111101000_2$$

To finish, we copy the numbers in sequence from bottom to top

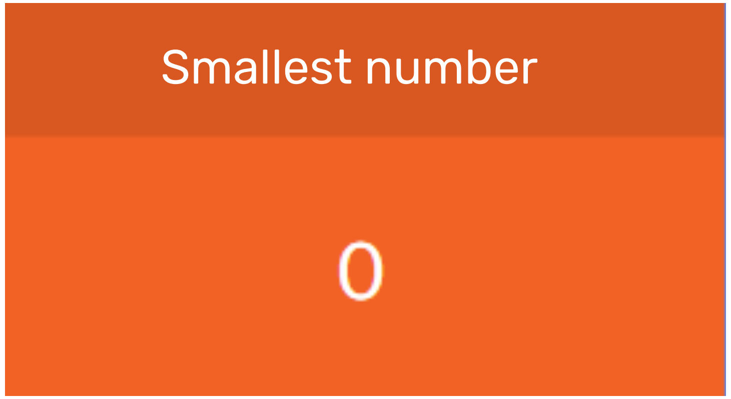
Individual Work

$$100001_2 \rightarrow x_{10}$$
$$11111111_2 \rightarrow x_{10}$$
$$254_{10} \rightarrow x_2$$
$$17_{10} \rightarrow x_2$$

Hexadecimal System

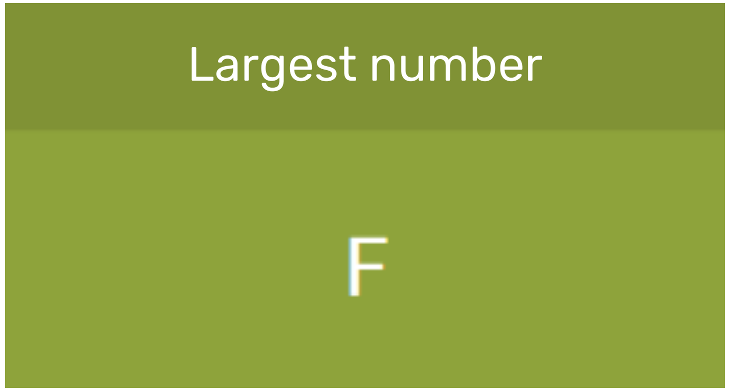
Hexadecimal System

In hexadecimal system, each digit can represent 16 possible values. For this, the characters used go from 0 to 9 and letters from A to F.



Smallest number

0

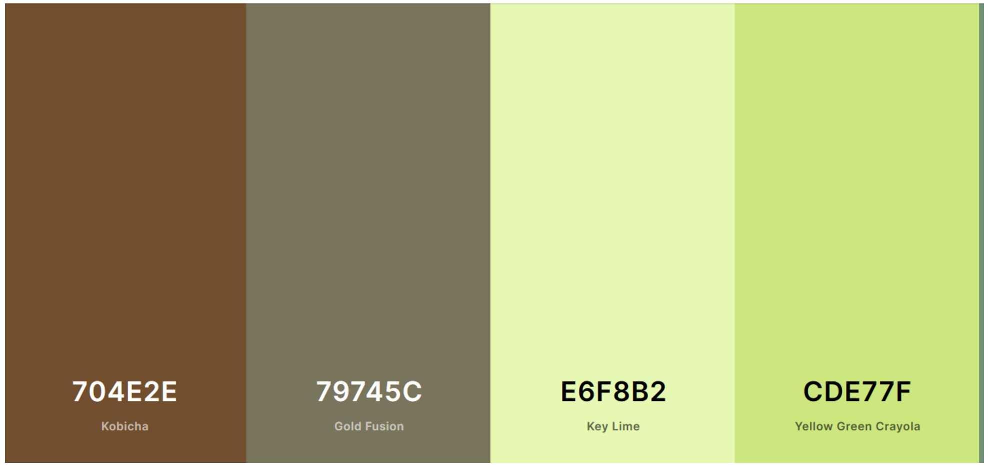


Largest number

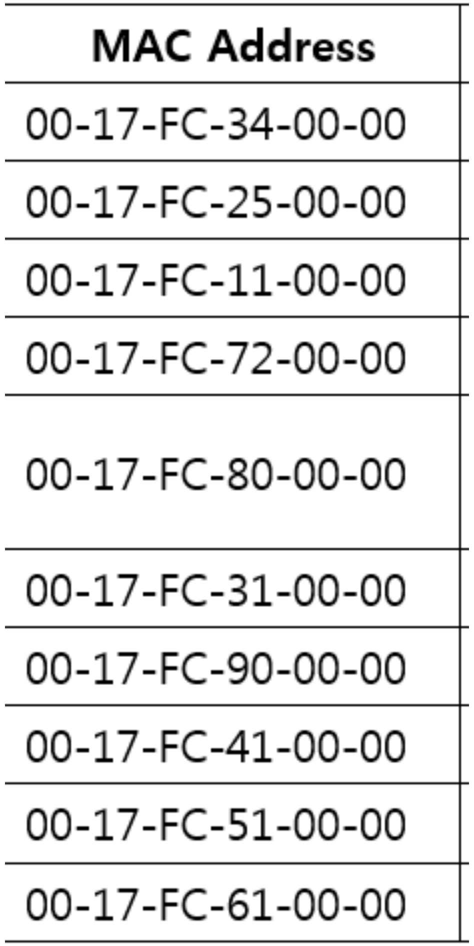
F

Where do we use hexadecimal numbers?

To represent RGB colors



In networks, for computer MAC Addresses



MAC Address
00-17-FC-34-00-00
00-17-FC-25-00-00
00-17-FC-11-00-00
00-17-FC-72-00-00
00-17-FC-80-00-00
00-17-FC-31-00-00
00-17-FC-90-00-00
00-17-FC-41-00-00
00-17-FC-51-00-00
00-17-FC-61-00-00

Hexadecimal System

In computer science, a hexadecimal base is represented using the following prefix: “**`0x`**”.

- **`0xF3`**
- **`0x6A23`**
- **`0x1745`**

Another way of representing it is by including 16 as a subscript:

- $F3_{16}$
- $6A23_{16}$
- 1745_{16}

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F ← Largest value with 1 digit
10
11
12
13
14
15
16
17
18
19
1A
1B
1C
1D
1E
1F
20
…
FE
FF ← Largest value with two digits
100

Binary to Hex Conversion

1. Given a binary number:

$$110000101101_2$$

2. From right to left, we begin grouping groups of 4 bits

$$\boxed{1100}\boxed{0010}\boxed{1101}_2 \rightarrow x_{16}$$

3. We convert each group to hexadecimal. Remember each digit goes from 0 to F

$$\boxed{1100}_2 = 12_{10} = C_{16}$$

$$\boxed{0010}_2 = 2_{10} = 2_{16}$$

$$\boxed{1101}_2 = 13_{10} = D_{16}$$

4. Now we put the numbers together in the proper order.

$$110000101101_2 = C2D_{16}$$

Hex to Decimal Conversions

1. Given a hexadecimal number:

$$23E_{16}$$

2. We extend the polinomious by multiplying each digit (in decimal) and its base to the power of its position:

Remember that: $E_{16} = 14_{10}$

$$23E_{16} = 2 \cdot 16^2 + 3 \cdot 16^1 + 14 \cdot 16^0$$

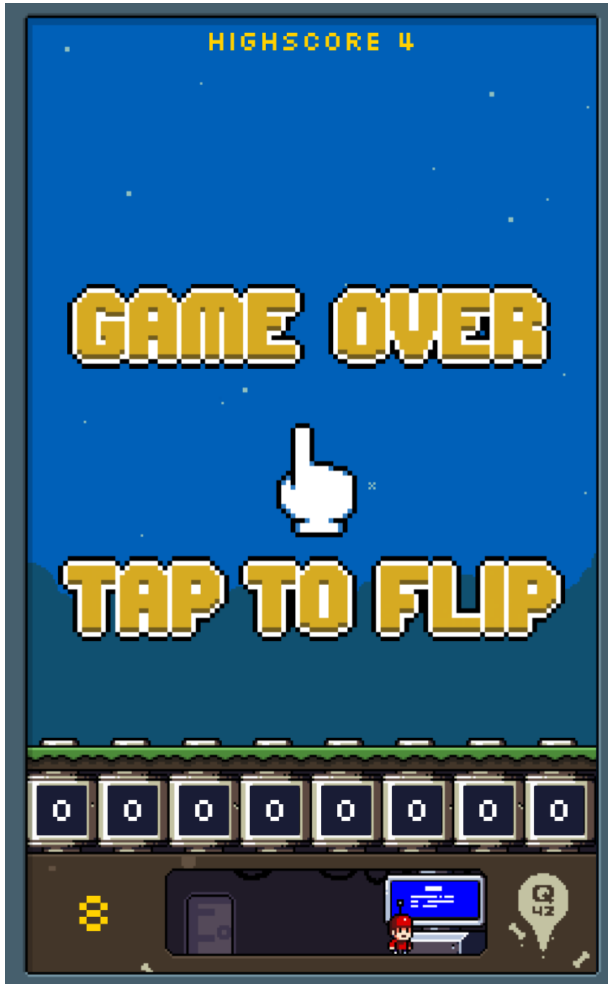
$$23E_{16} = 512_{10} + 48_{10} + 14_{10}$$

3. Add every factor:

$$23E_{16} = 574_{10}$$

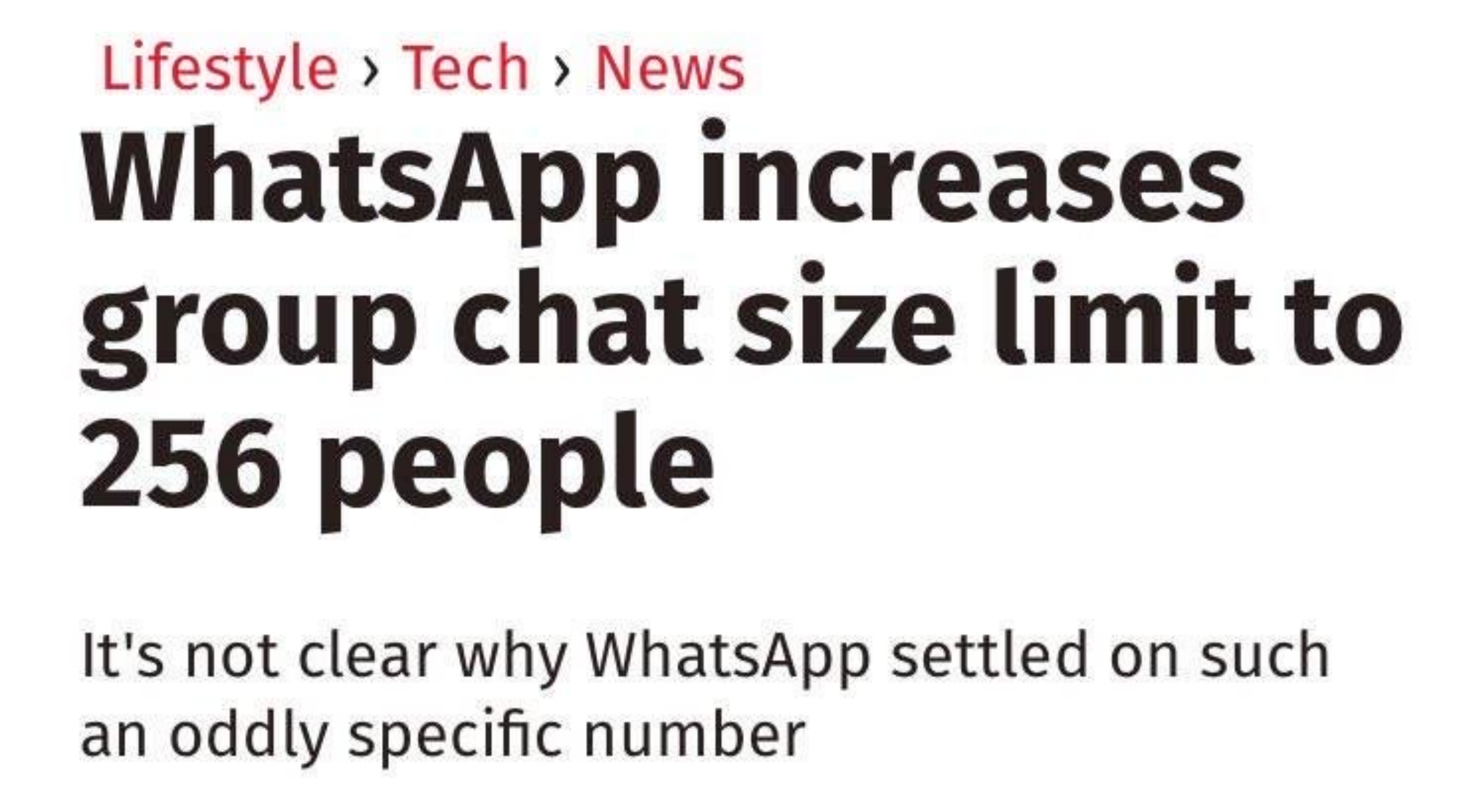
Individual work

https://flippybitandtheattackofthehexadecimalsfrombase16.com/



iOS

Data Encoding



Lifestyle › Tech › News

WhatsApp increases group chat size limit to 256 people

It's not clear why WhatsApp settled on such an oddly specific number

Data Encoding

We can express the amount of representable values “M” as a function of the amount of bits “n”.

$$M(n) = 2^n$$

For example, if we use a single byte (8 bits) we can represent:

$$M(8) = 2^8 = 256$$

Data Encoding

Solving for n in the equation:

$$n \geq \log_2(M)$$

$$n \geq 3.32 \cdot \log_{10}(M)$$

For example, for an alphabet of 10 different values:

$$\{0,1,2,3,4,5,6,7,8,9\}$$

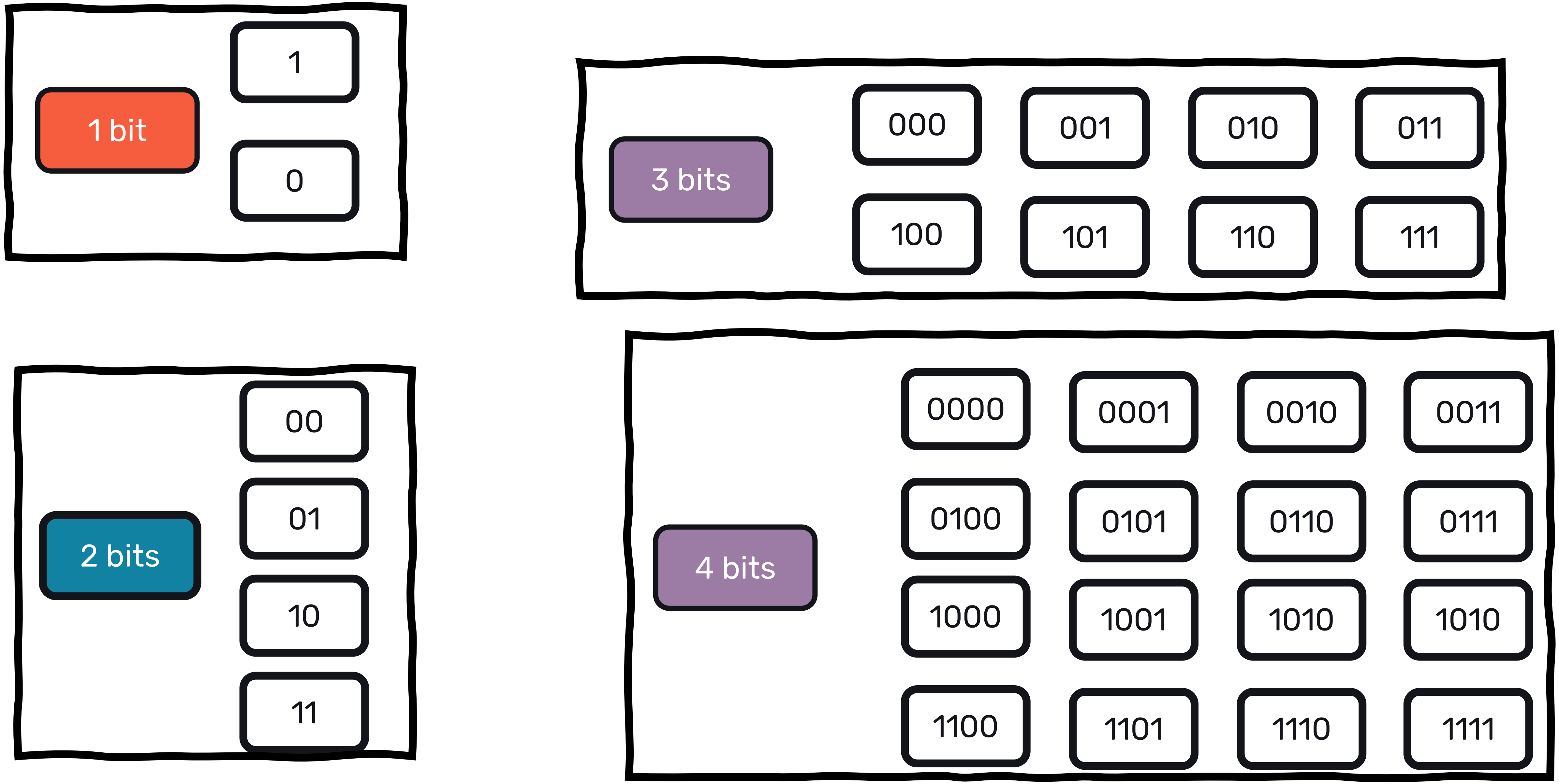
The minimum number of bits can be described by the next equation:

$$n \geq 3.32 \cdot \log(10)$$

$$n \geq 3.32$$

$$n \geq 4$$

Here is an alternative method.



1 bit

1
0

2 bits

00
01
10
11

3 bits

000 001 010 011
100 101 110 111

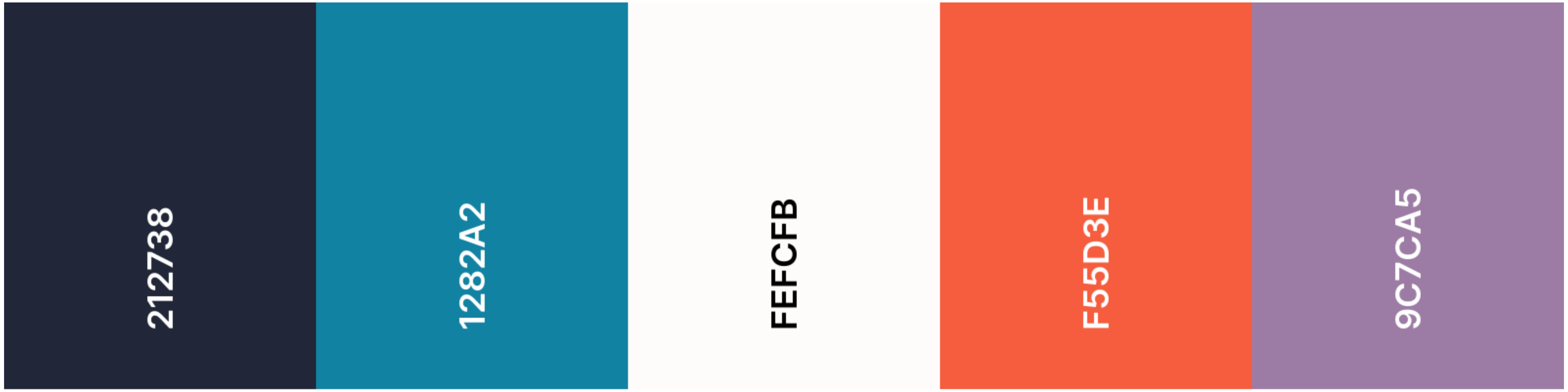
4 bits

0000 0001 0010 0011
0100 0101 0110 0111
1000 1001 1010 1010
1100 1101 1110 1111

Example

If we want to represent an alphabet of 700 different characters, we need to increment the bit count one by one until it

1 bit = 2^1 = 2 values
2 bits = 2^2 = 4 values
3 bits = 2^3 = 8 values
4 bits = 2^4 = 16 values
5 bits = 2^5 = 32 values
6 bits = 2^6 = 64 values
7 bits = 2^7 = 128 values
8 bits = 2^8 = 256 values
9 bits = 2^9 = 512 values
10 bits = 2^{10} = 1024 values



Modulo 1