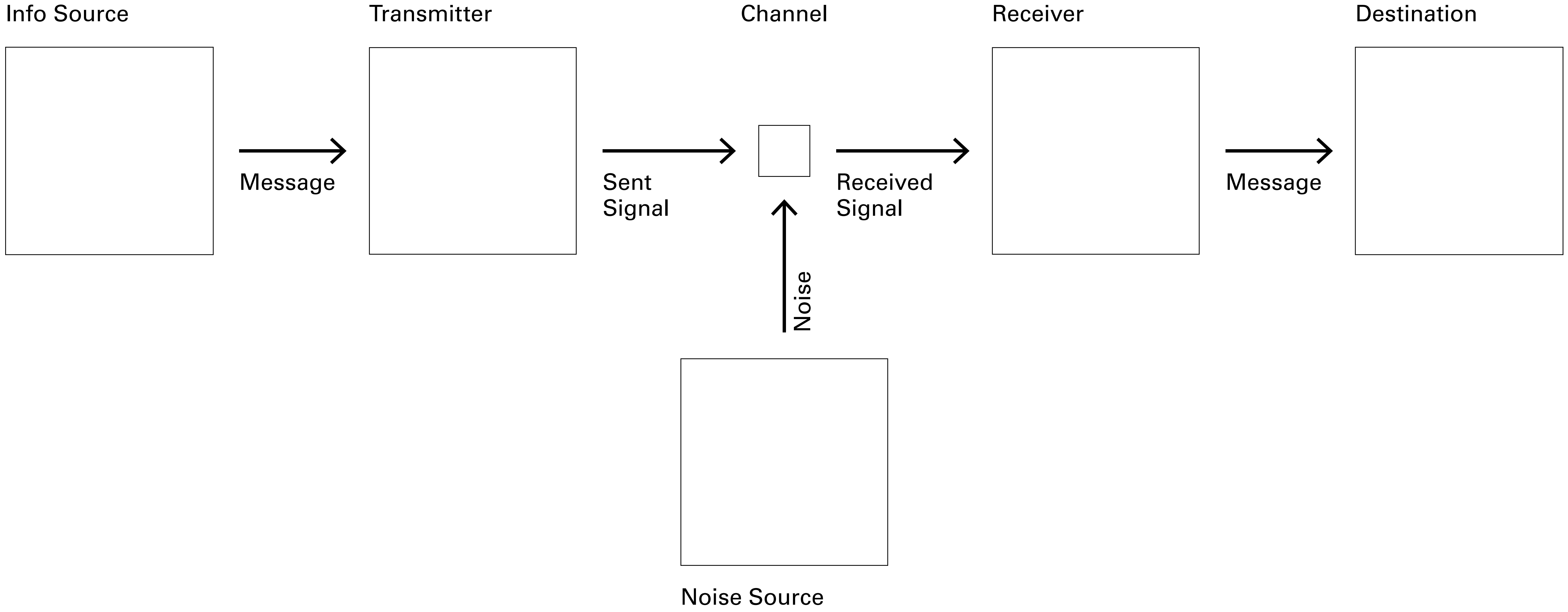
University of California, Berkeley Jacobs Institute for Design Innovation MDes Program

Information (bits)
A material of design

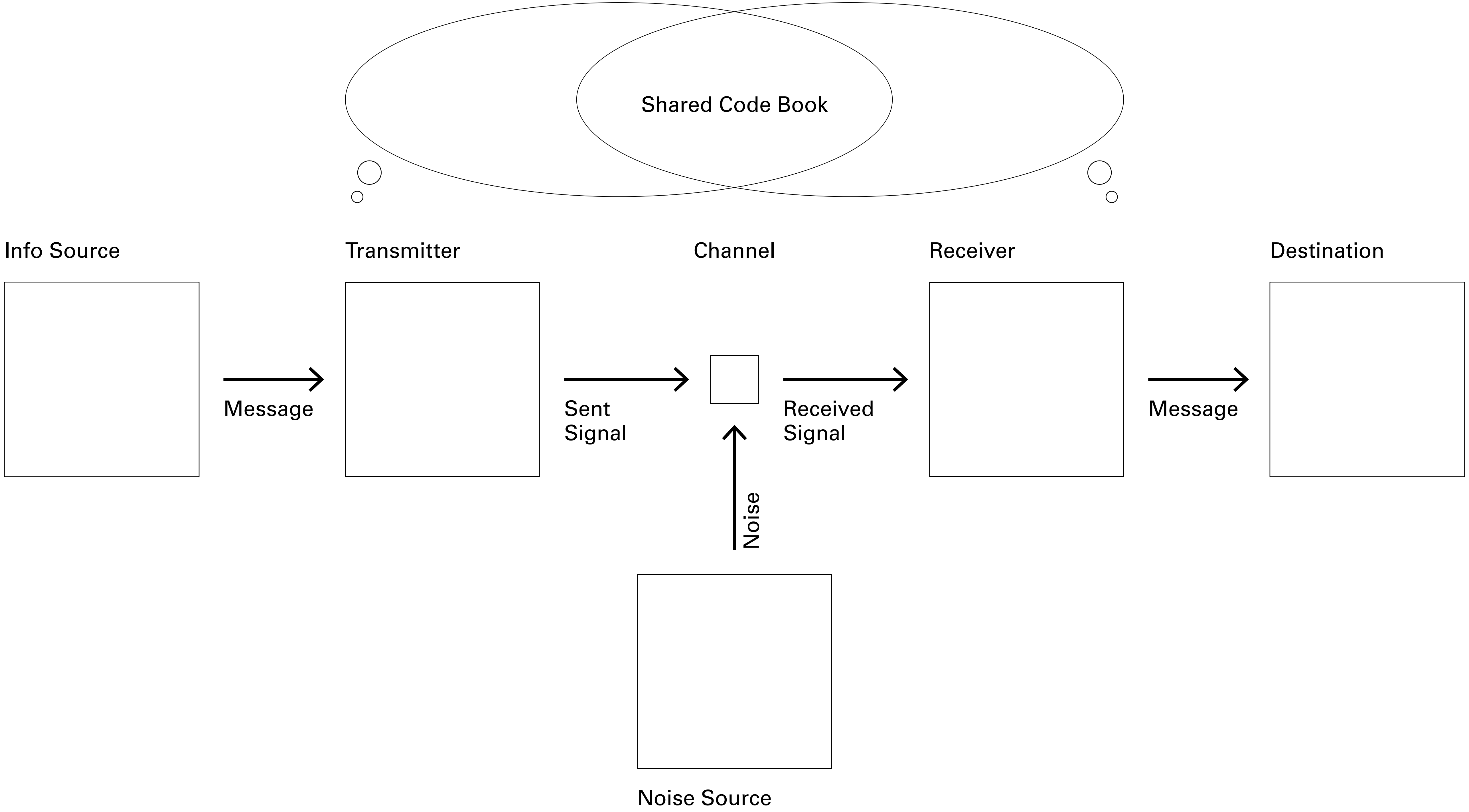
September 12, 2025

Shannon’s basic model of the **communications process.**



See Claude Shannon, “A Mathematical Theory of Communication,” 1948

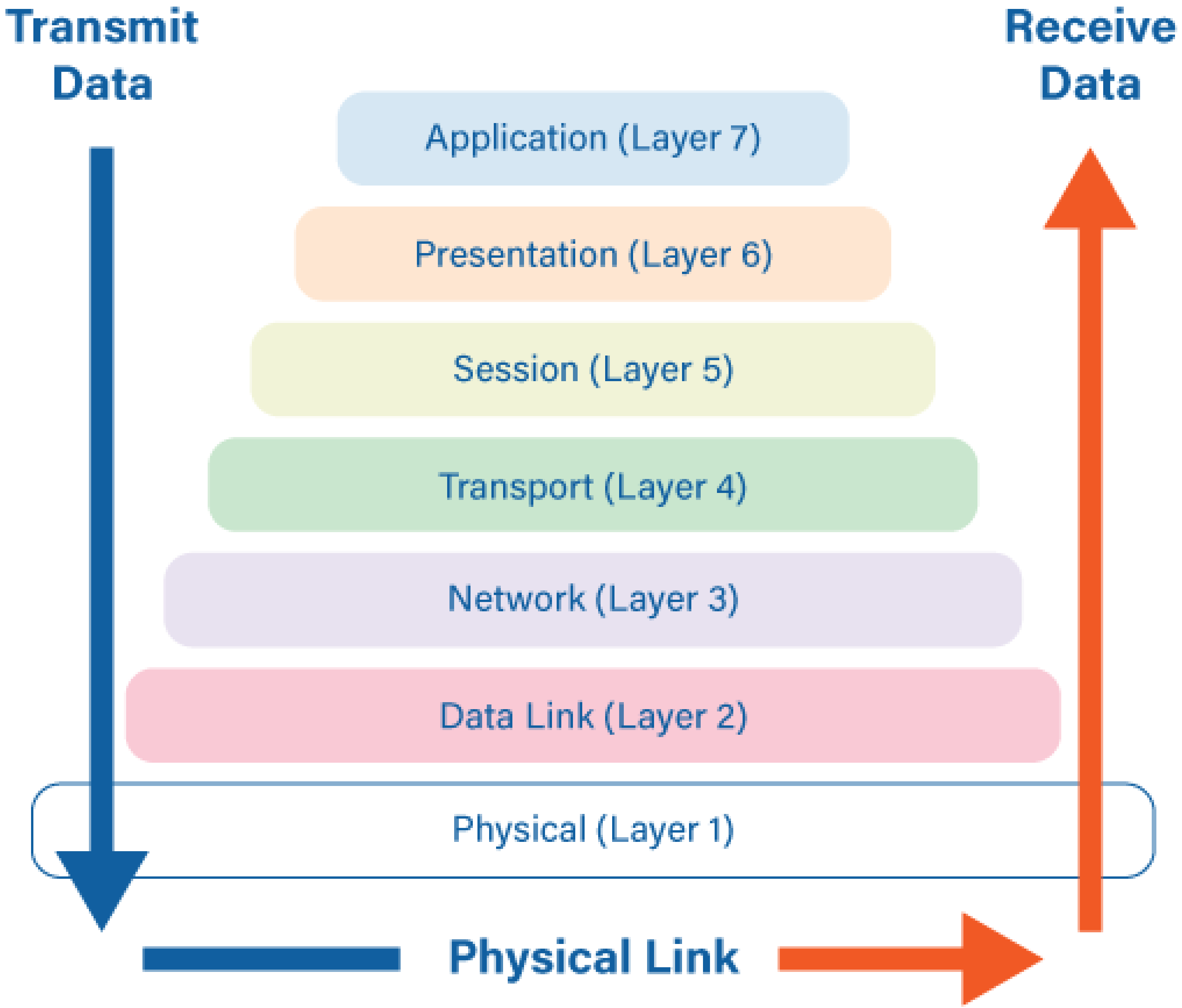
Encoding and decoding require a **shared code book.**



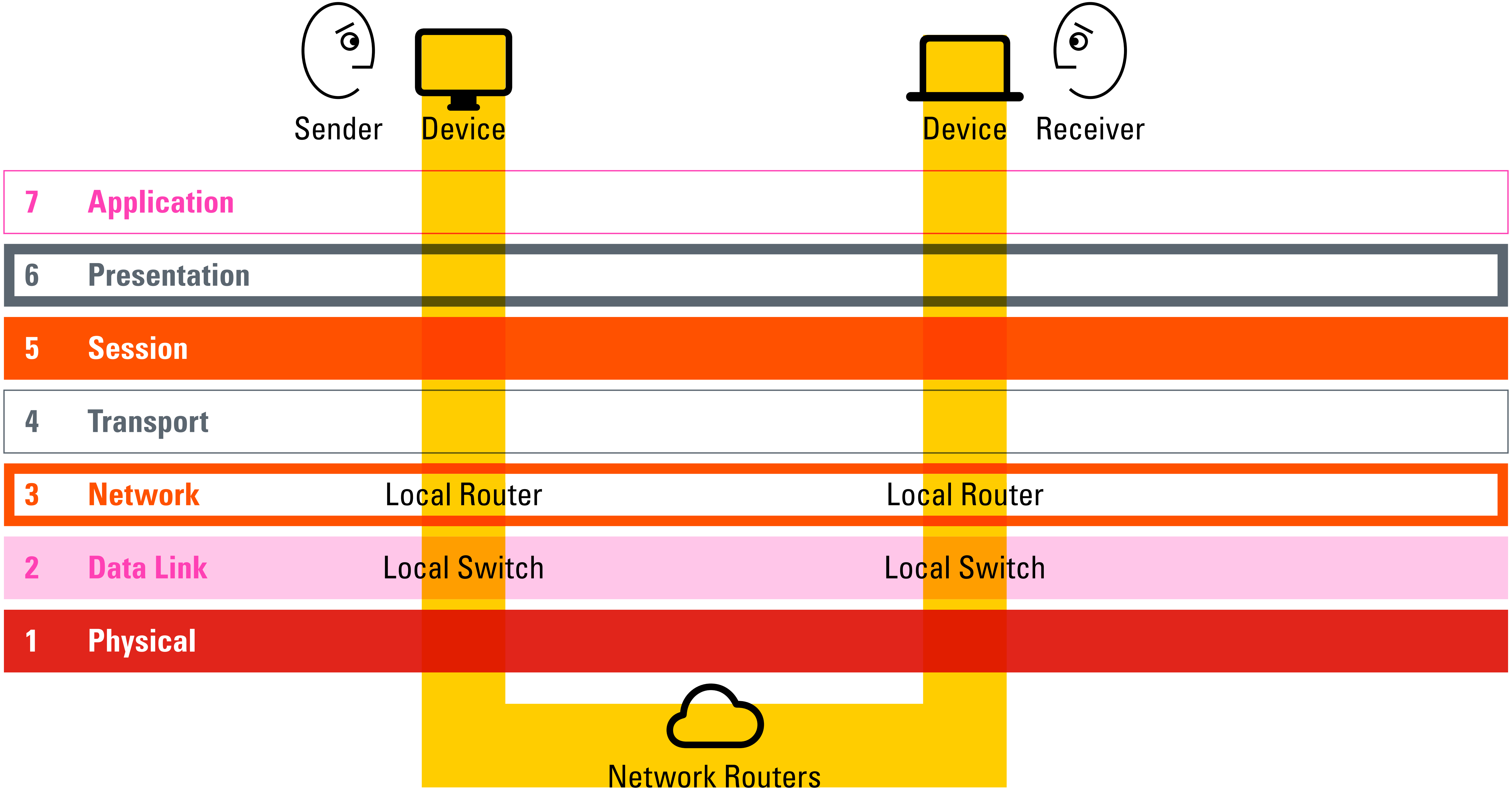
See Claude Shannon, “A Mathematical Theory of Communication,” 1948

Open Systems Interconnection (OSI) model,

which explains how today’s computer networks function.



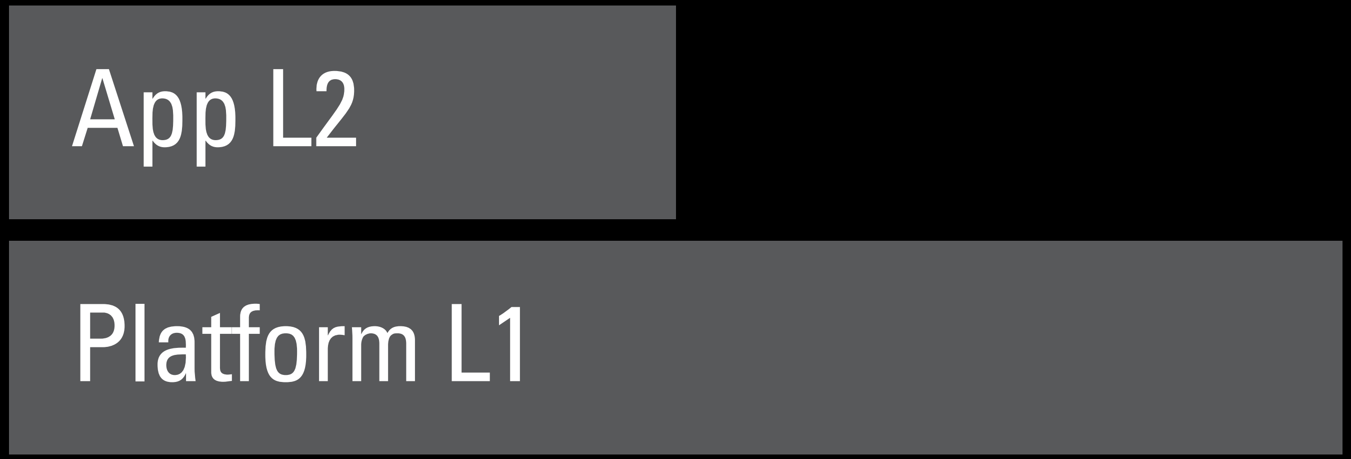
The OSI model is a “stack” — a series of platforms.



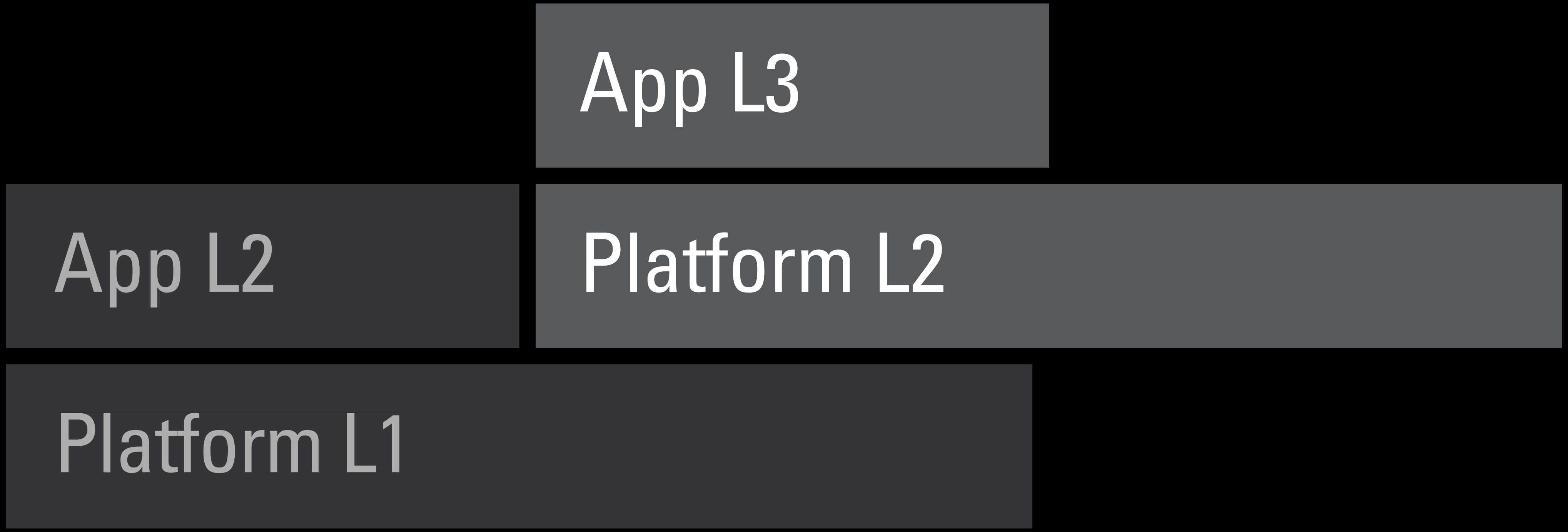
> *“A ‘platform’ is a system that can be programmed and therefore customized by outside developers—users—and in that way, adapted to countless needs and niches that the platform’s original developers could not have possibly contemplated, much less had time to accommodate.”*

—Marc Andreessen, founder of Netscape, Opsware, and Ning

A platform is a service on which others can build.



Applications rely on platforms; **but an app itself may be a platform** for another higher level app.



Microsoft made a fortune by controlling a “choke point”—the PC OS, linking apps and hardware.

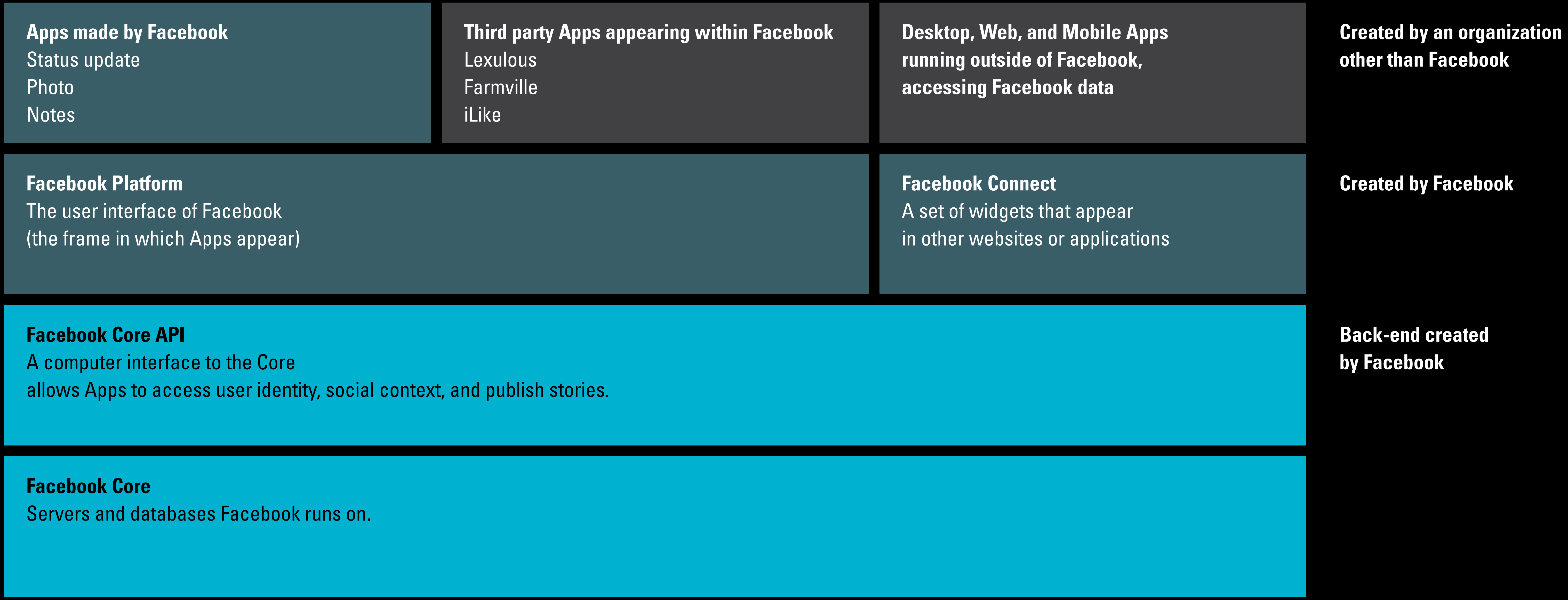
Microsoft’s monopoly has lasted more than 30 years.

Local Documents	.doc, .xls, .ppt, etc.
PC Apps	Word, Excel, PowerPoint, etc.
Operating System (OS)	Windows
Processor	8086, 80286, 80386, etc.

The web threatens Microsoft's monopoly by introducing a new layer.

Web Documents	A, B, C, etc.
Web-based Apps	Amazon, Google, Facebook, etc.
Browser	Chrome, Firefox, IE, Safari
Operating System (OS)	Windows, Mac, Linux, etc.
Processor	Intel, ARM, etc.

Facebook has turned itself into a platform, enabling developers to offer apps that run in Facebook’s site and providing access to user data to apps outside of Facebook.



Apps made by Facebook
Status update
Photo
Notes

Third party Apps appearing within Facebook
Lexulous
Farmville
iLike

Desktop, Web, and Mobile Apps running outside of Facebook, accessing Facebook data

Created by an organization other than Facebook

Facebook Platform
The user interface of Facebook
(the frame in which Apps appear)

Facebook Connect
A set of widgets that appear
in other websites or applications

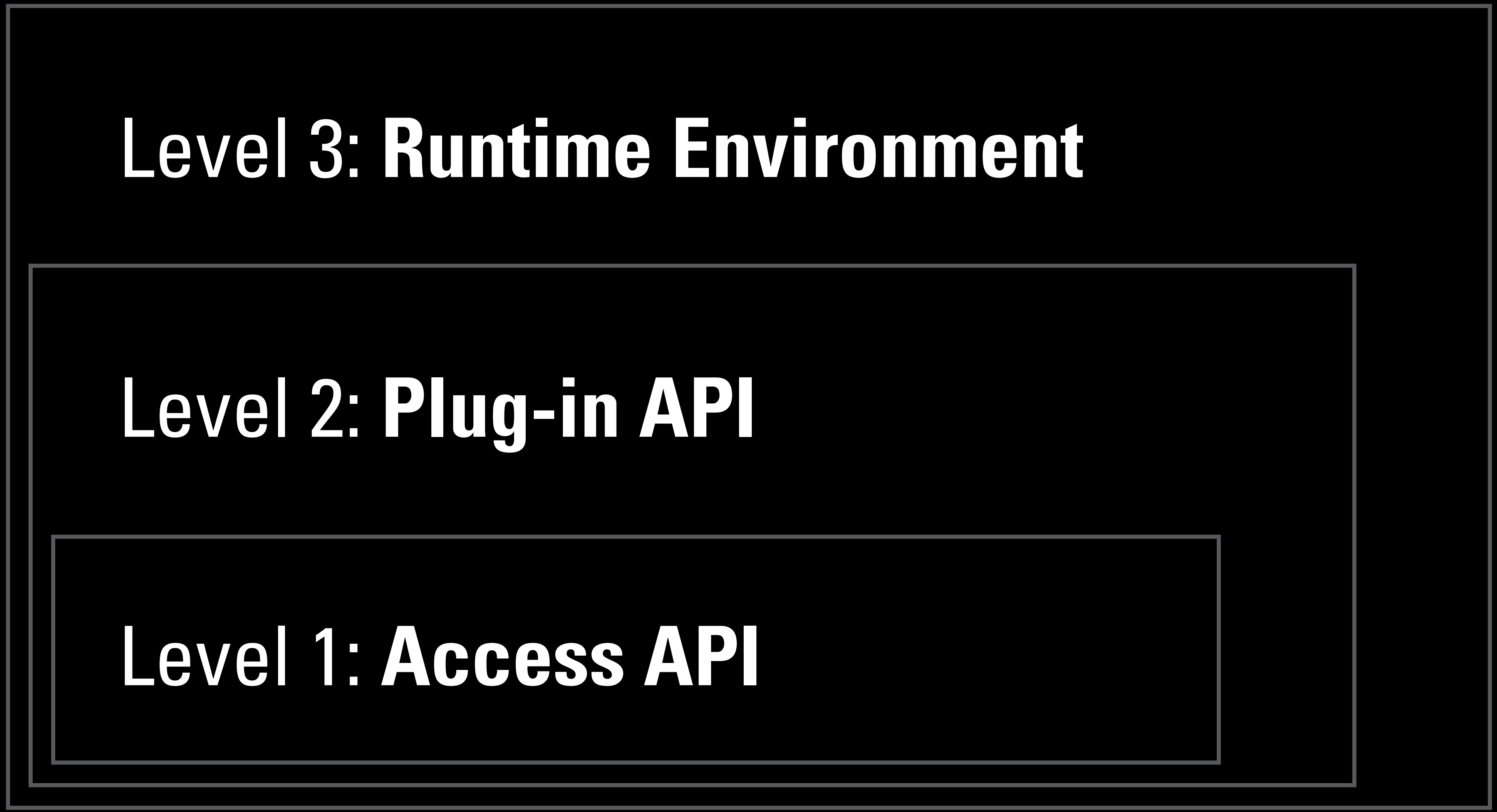
Created by Facebook

Facebook Core API
A computer interface to the Core
allows Apps to access user identity, social context, and publish stories.

Back-end created by Facebook

Facebook Core
Servers and databases Facebook runs on.

Marc Andreessen’s **“Three Kinds of Platforms”**



Level 3: **Runtime Environment**

Level 2: **Plug-in API**

Level 1: **Access API**

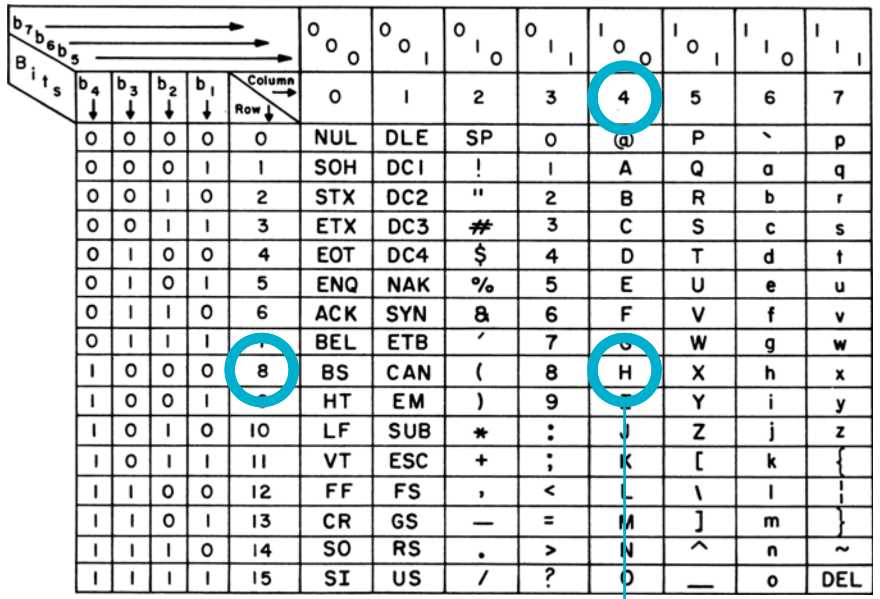
Online platform runs uploaded code
e.g., Ning, Salesforce, Amazon

Plug-in shows up within the platform
but runs elsewhere
e.g., Facebook

App runs elsewhere;
calls data from platform
e.g., eBay, PayPal, Flickr

How Is Type Input?

Key Strokes to Words

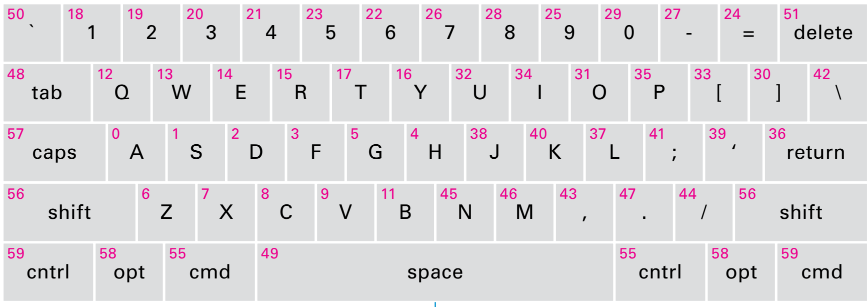


b7 b6 b5 →						0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits	b4	b3	b2	b1	Row ↓ / Column →	0	1	2	3	4	5	6	7
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	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

ASCII Table
Sub-set of Unicode

100,1000

Keyboard
The keycodes for each mechanical key are shown in magenta



Mechanical Keys	shift H	E	L	L	O	,	space	shift W	O	R	L	D	shift 1
Keycodes	56-4	14	37	37	31	43	49	56-13	31	15	37	2	56-18
Unicode Values This is how text information is stored in computer files	0048	0065	006C	006C	006F	002C	00A0	0057	006F	0072	006C	0064	0021
Characters	H	e	l	l	o	,		W	o	r	l	d	!

Mechanical keys pressed and raw keycodes sent to keyboard driver.

Keycodes translated into Unicode values by the functional layout.

Unicode values used to generate character string. (See page 48.)

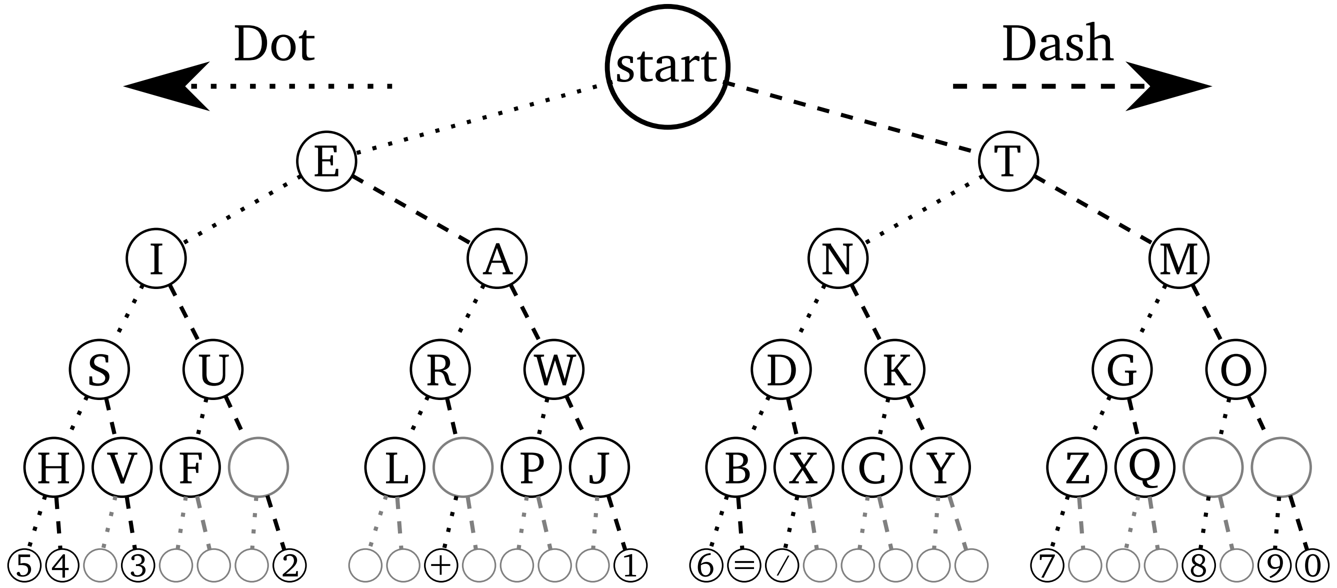
Displayed Words

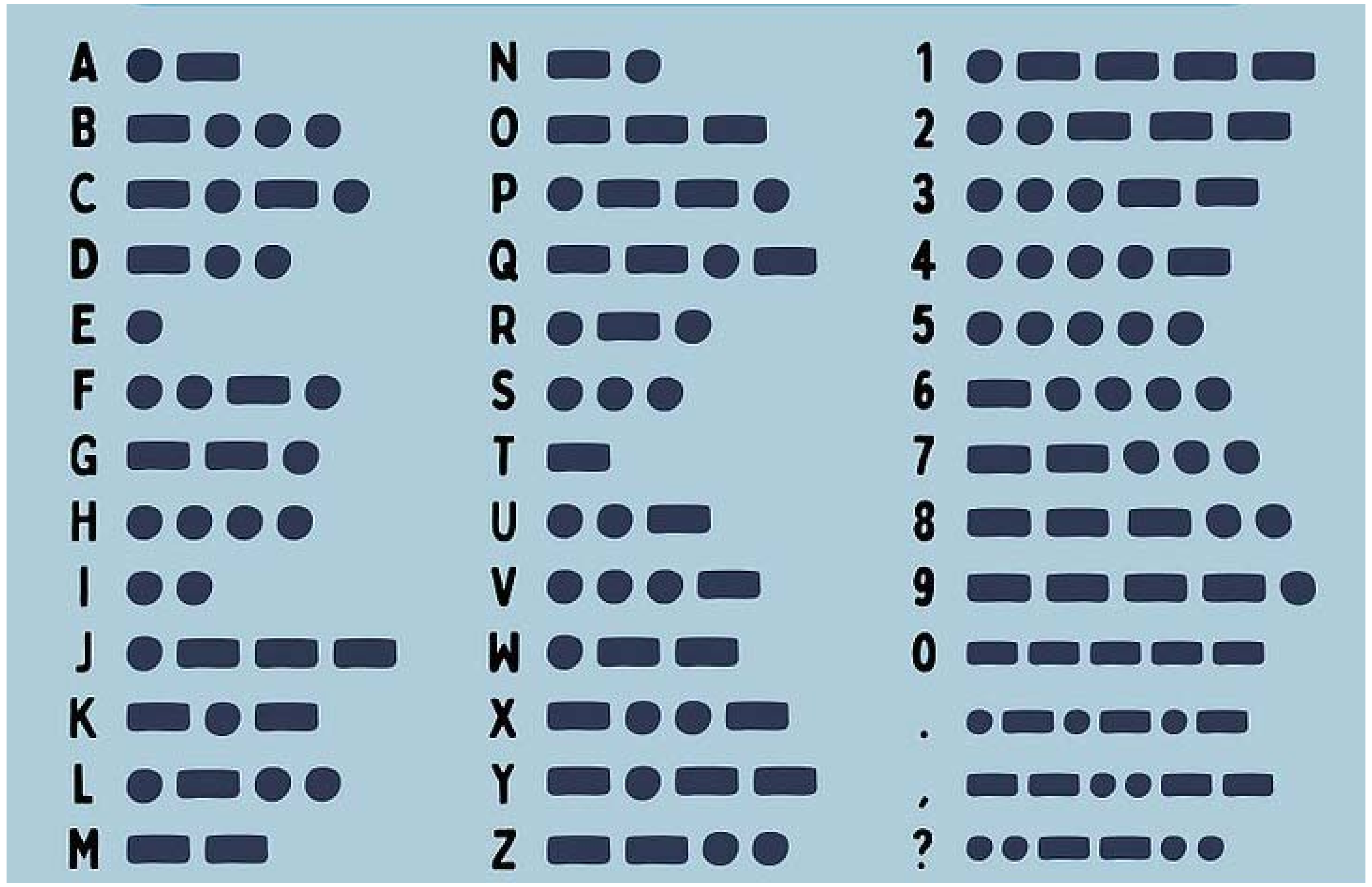
Hello, World!

Character string output on display (specific formatting choices such as font and type size are applied before final onscreen display).

Morse Code

Two ways of visualizing the same information.

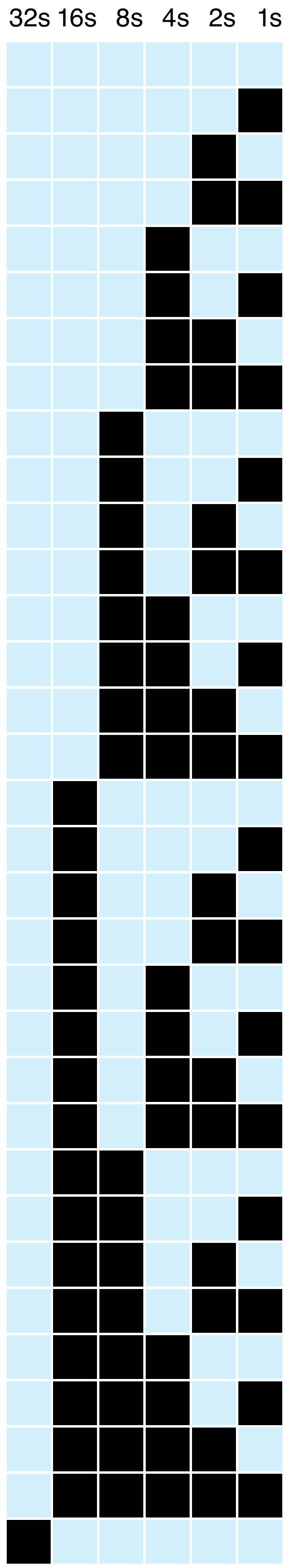




A	●▬	N	▬●	1	●▬▬▬▬
B	▬●●●	O	▬▬▬	2	●●▬▬▬
C	▬●▬●	P	●▬▬●	3	●●●▬▬
D	▬●●	Q	▬▬●▬	4	●●●●▬
E	●	R	●▬●	5	●●●●●
F	●●▬●	S	●●●	6	▬●●●●
G	▬▬●	T	▬	7	▬▬●●●
H	●●●●	U	●●▬	8	▬▬▬●●
I	●●	V	●●●▬	9	▬▬▬▬●
J	●▬▬▬	W	●▬▬	0	▬▬▬▬▬
K	▬●▬	X	▬●●▬	.	●▬●▬●▬
L	●▬●●	Y	▬●▬▬	,	▬▬●●▬▬
M	▬▬	Z	▬▬●●	?	●●▬▬●●

Binary (base 2)

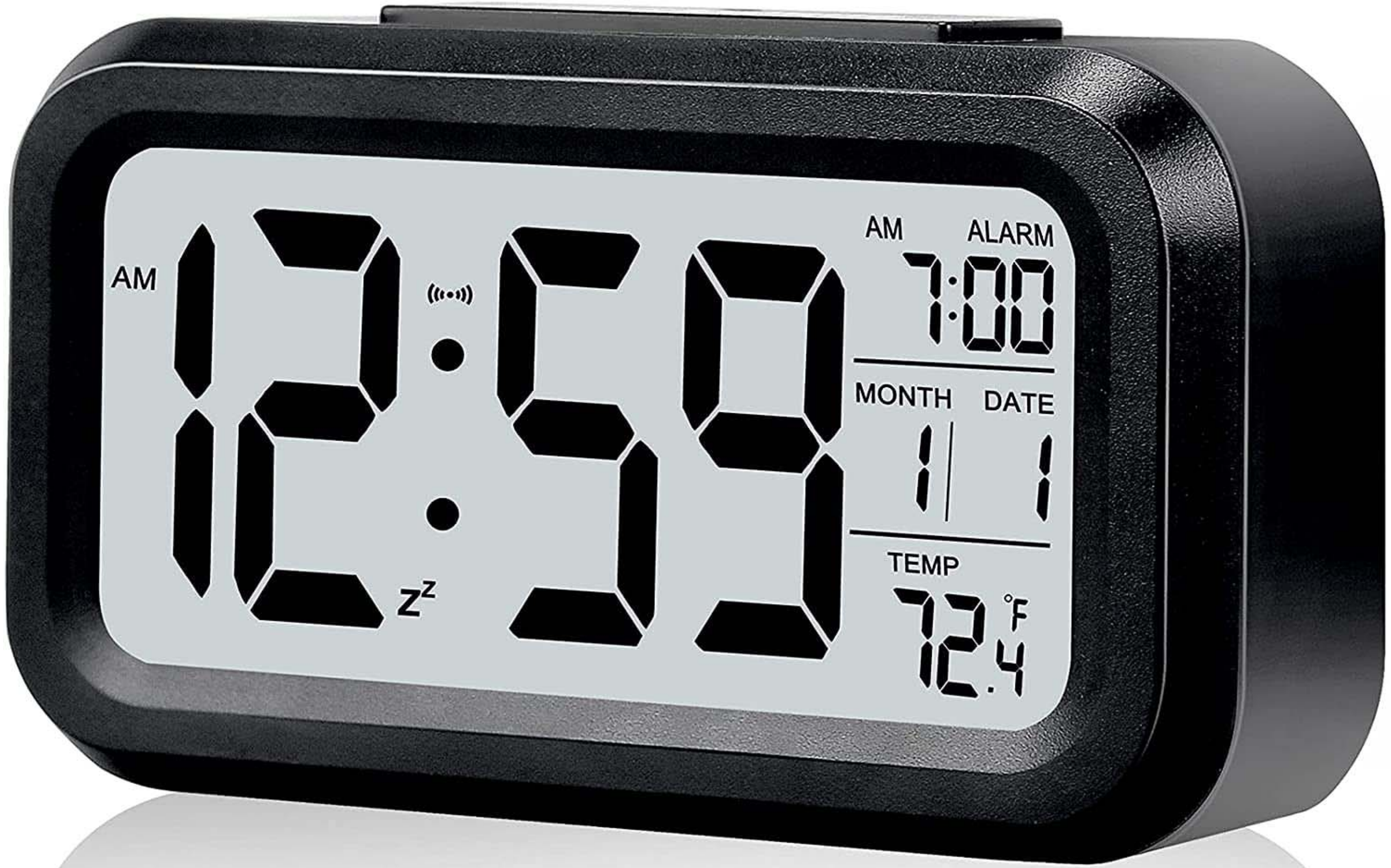
How do you count in binary?



00 0000	= 00 in base 10
00 0001	= 01
00 0010	= 02
00 0011	= 03
00 0100	= 04
00 0101	= 05
00 0110	= 06
00 0111	= 07
00 1000	= 08
00 1001	= 09
00 1010	= 10
00 1011	= 11
00 1100	= 12
00 1101	= 13
00 1110	= 14
00 1111	= 15
01 0000	= 16
01 0001	= 17
01 0010	= 18
01 0011	= 19
01 0100	= 20
01 0101	= 21
01 0110	= 22
01 0111	= 23
01 1000	= 24
01 1001	= 25
01 1010	= 26
01 1011	= 27
01 1100	= 28
01 1101	= 29
01 1110	= 30
01 1111	= 31
10 0000	= 32

Time

How do you count in hours, minutes, and seconds? What’s next?



Hexadecimal

How do you count in hex?

Hex is counting in 16,
but we only have 10 numbers, so we add letters.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1a, 1b, 1c, 1d, 1e, 1f ...ff
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 ...255

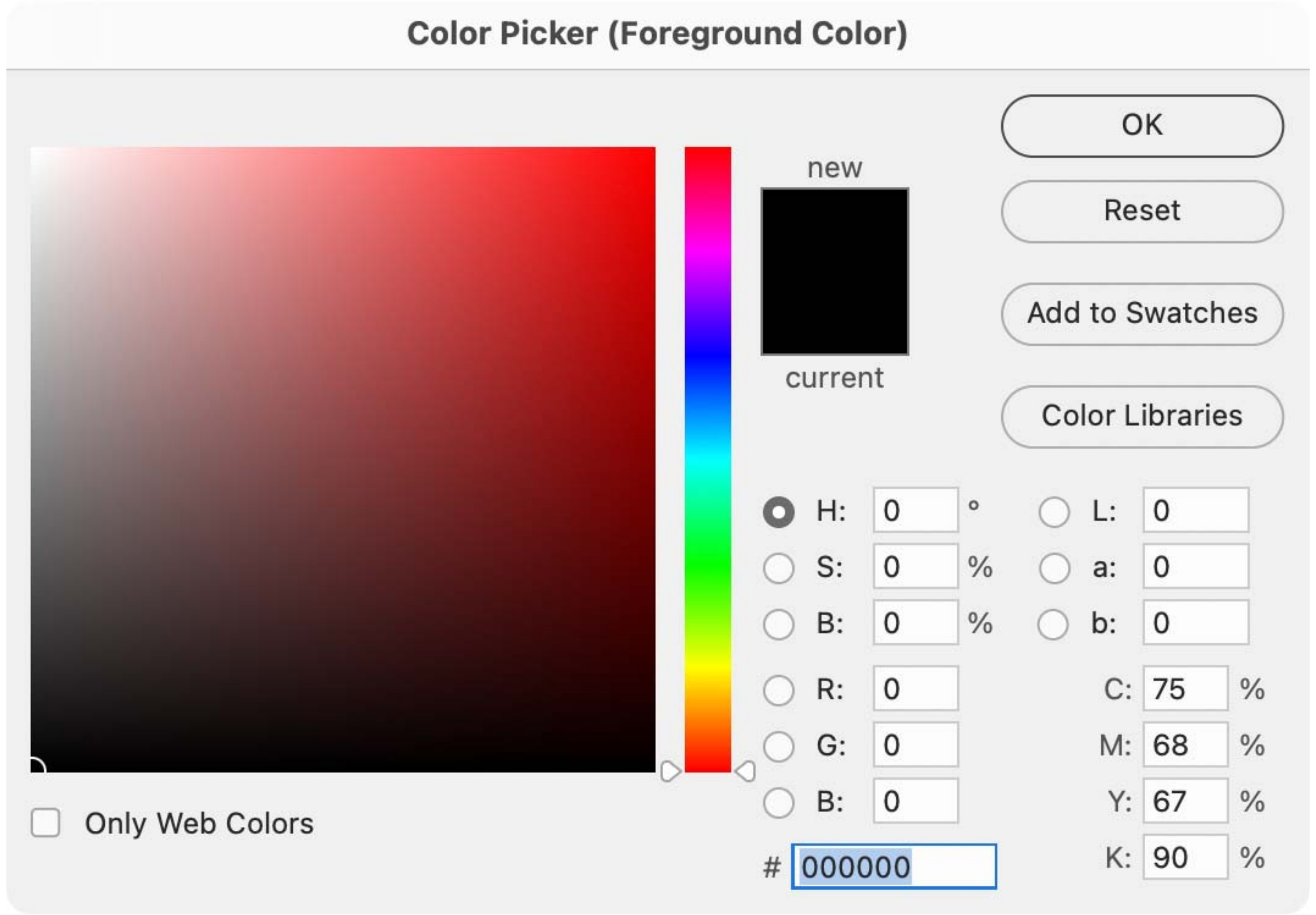
Hex is a short-hand for writing long strings.

15 = f = 1111 = (8 + 4 + 2 + 1)
255 = ff = 1111,1111 = (128 + 64 + 32 + 16 + 8 + 4 + 2 + 1)
255 = ff = (240 + 15) = 15 in the 16s column + 15 in the 1s column

ff ff ff = 255, 255, 255 = white

Photoshop’s color picker

How does hex explain this interface?



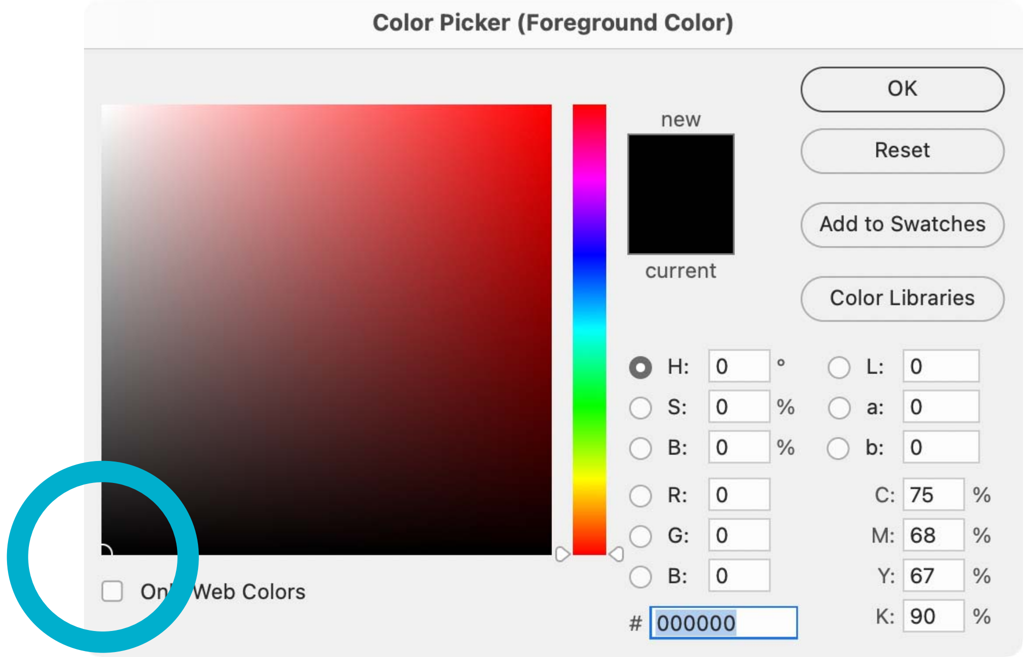
□ Only Web Colors

● H: 0 °
○ S: 0 %
○ B: 0 %
○ R: 0
○ G: 0
○ B: 0
000000

○ L: 0
○ a: 0
○ b: 0
C: 75 %
M: 68 %
Y: 67 %
K: 90 %

000000 is black

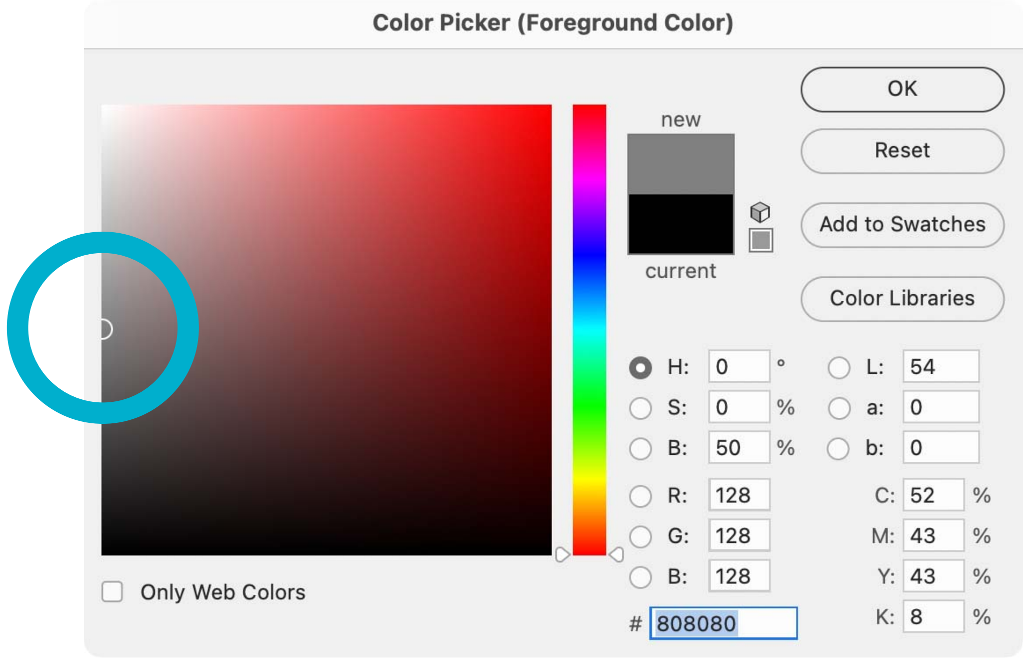
everything off.



● H: 0 ° ○ L: 0
○ S: 0 % ○ a: 0
○ B: 0 % ○ b: 0
○ R: 0 C: 75 %
○ G: 0 M: 68 %
○ B: 0 Y: 67 %
000000 K: 90 %
□ Only Web Colors

808080 is gray

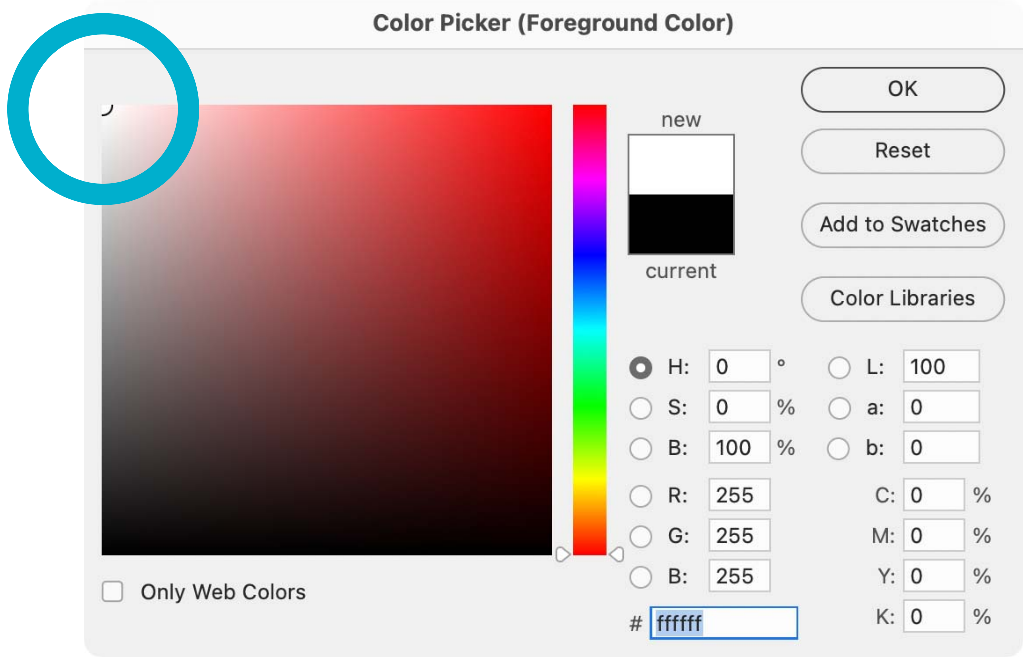
everything half-on.



● H: 0 ° ○ L: 54
○ S: 0 % ○ a: 0
○ B: 50 % ○ b: 0
○ R: 128 C: 52 %
○ G: 128 M: 43 %
○ B: 128 Y: 43 %
808080 K: 8 %
□ Only Web Colors

ffffff is white

everything on.



● H: 0 ° ○ L: 100
○ S: 0 % ○ a: 0
○ B: 100 % ○ b: 0
○ R: 255 C: 0 %
○ G: 255 M: 0 %
○ B: 255 Y: 0 %
# ffffff K: 0 %
□ Only Web Colors

The selected dimension has **256 possible colors.**

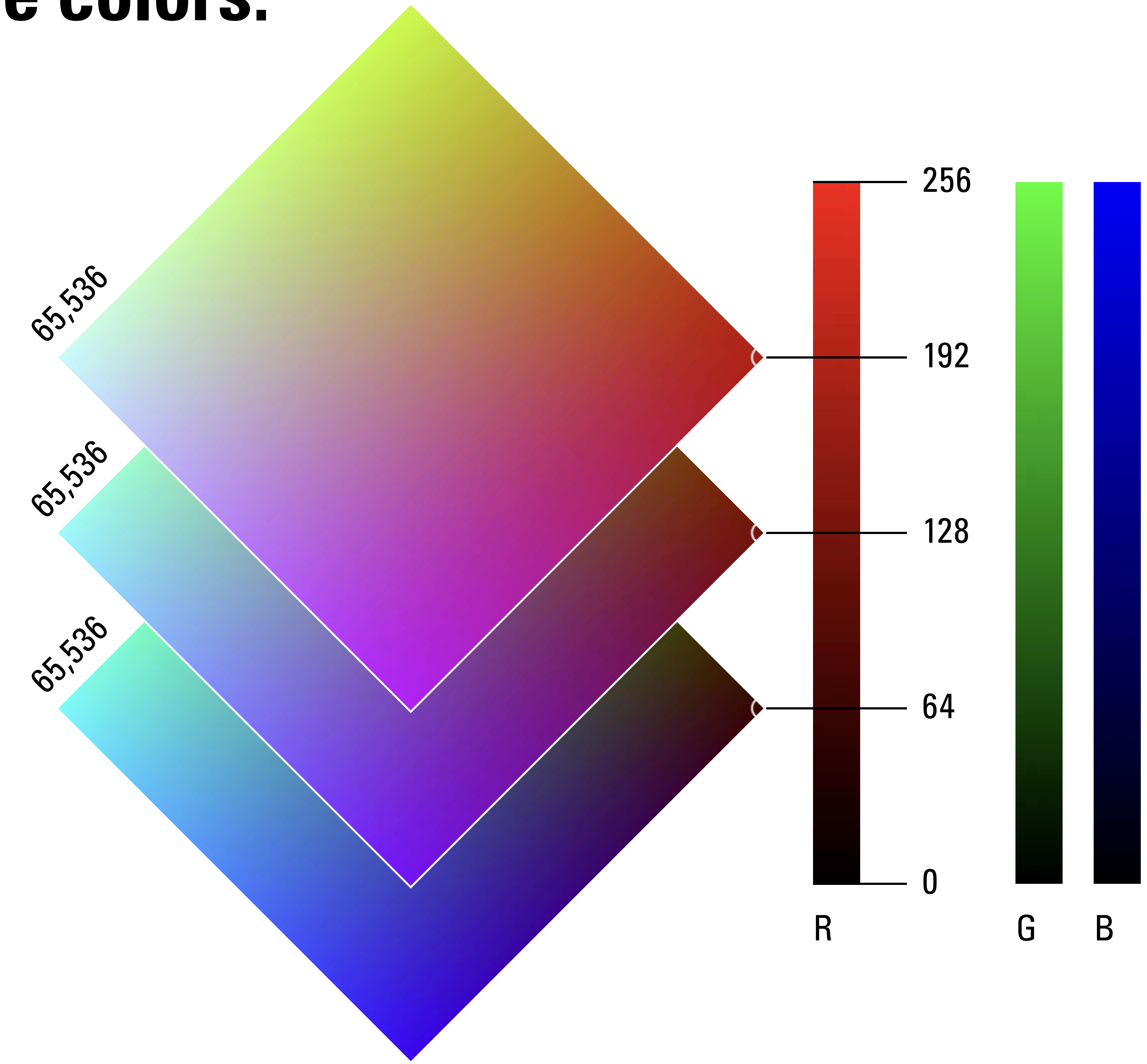
Each plane has **65,536 possible colors.**

The total cube has **16,777,216 possible colors.**

256 × 256 × 256 = 16,777,216

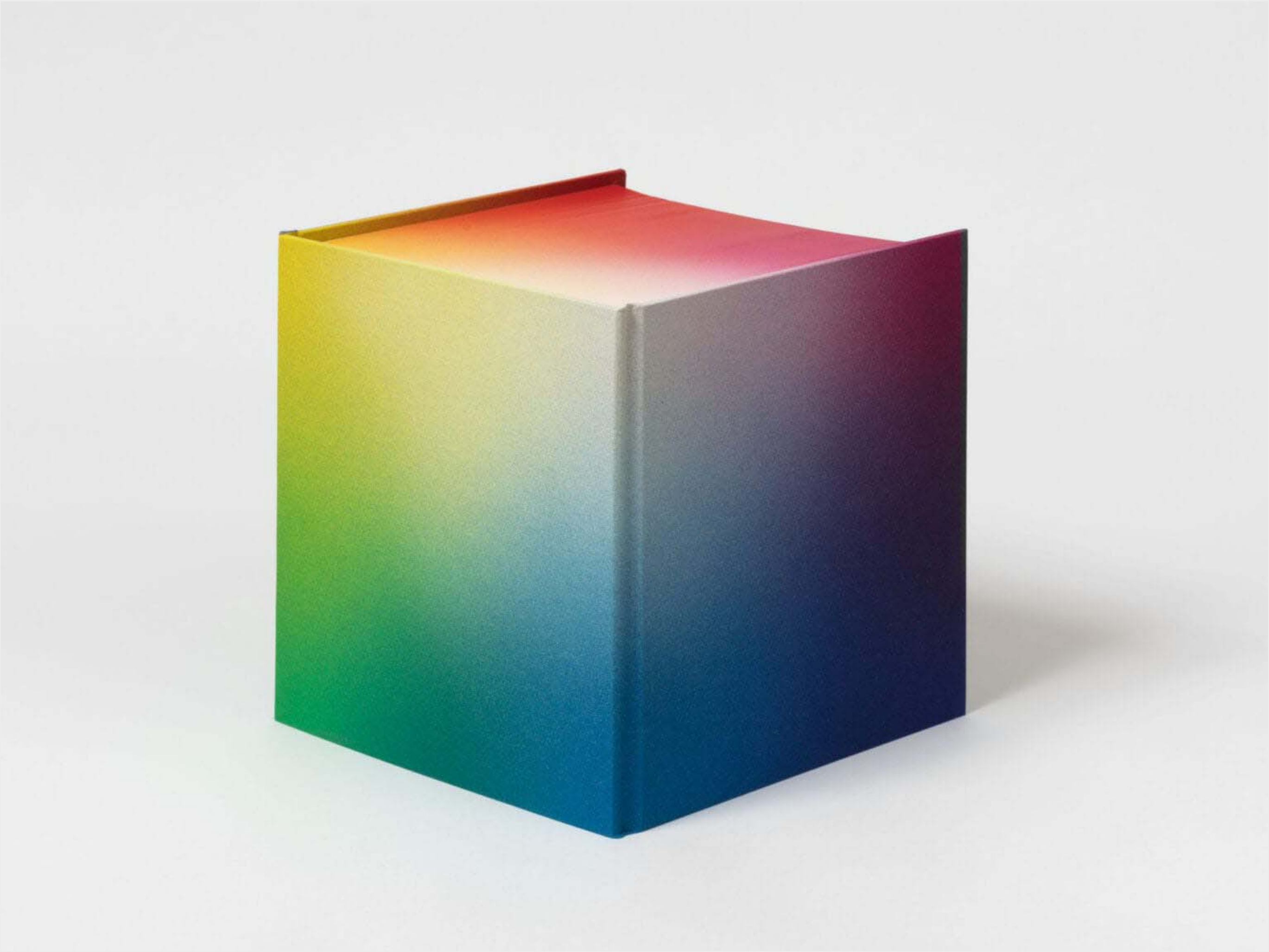
8-bit × 8-bit × 8-bit = 24-bit color

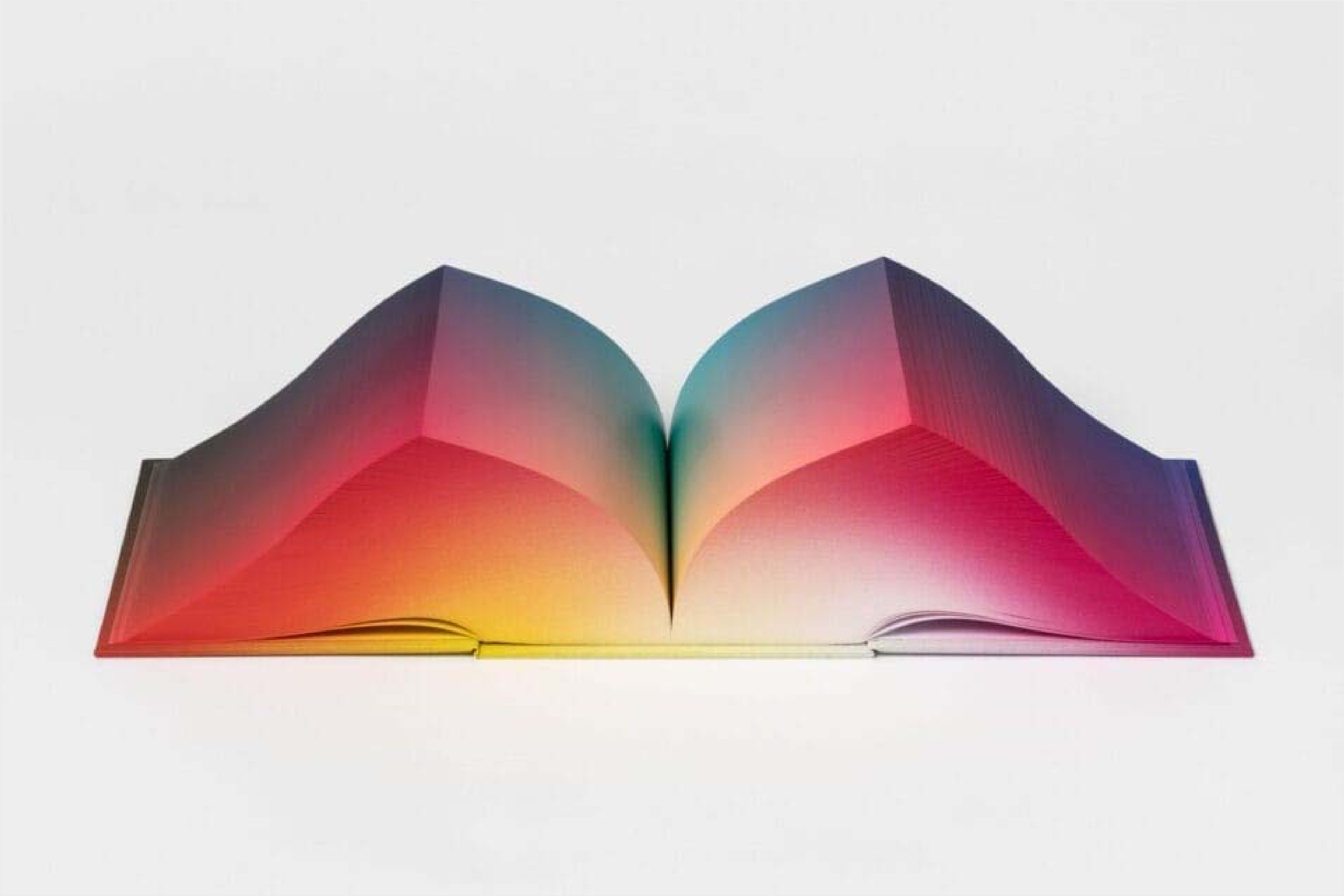
2^8 × 2^8 × 2^8 = 2^24



The space of possible colors can be visualized as a cube.

Opening the cube reveals planes of colors.

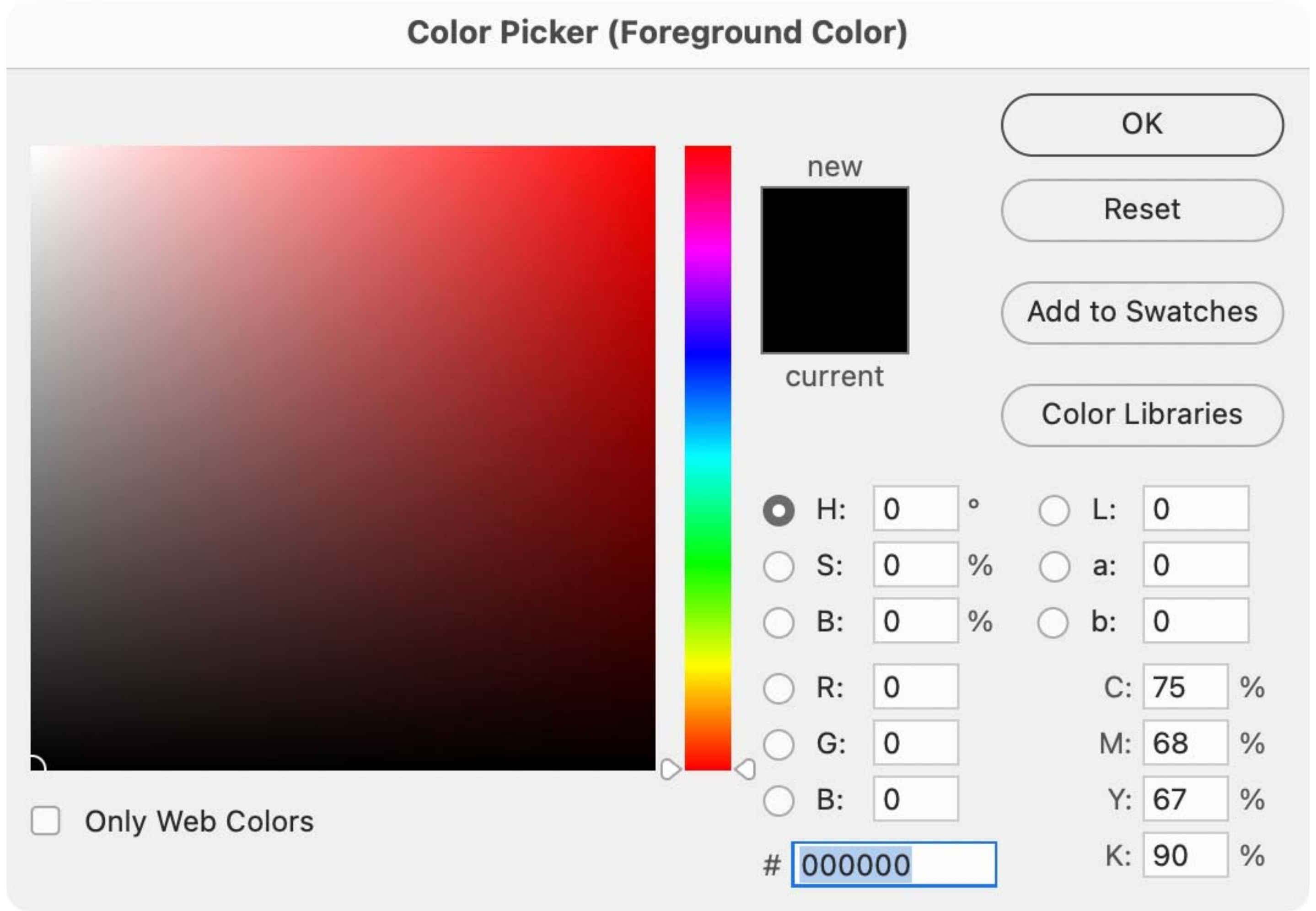




https://taubaauerbach.com/view.php?id=286

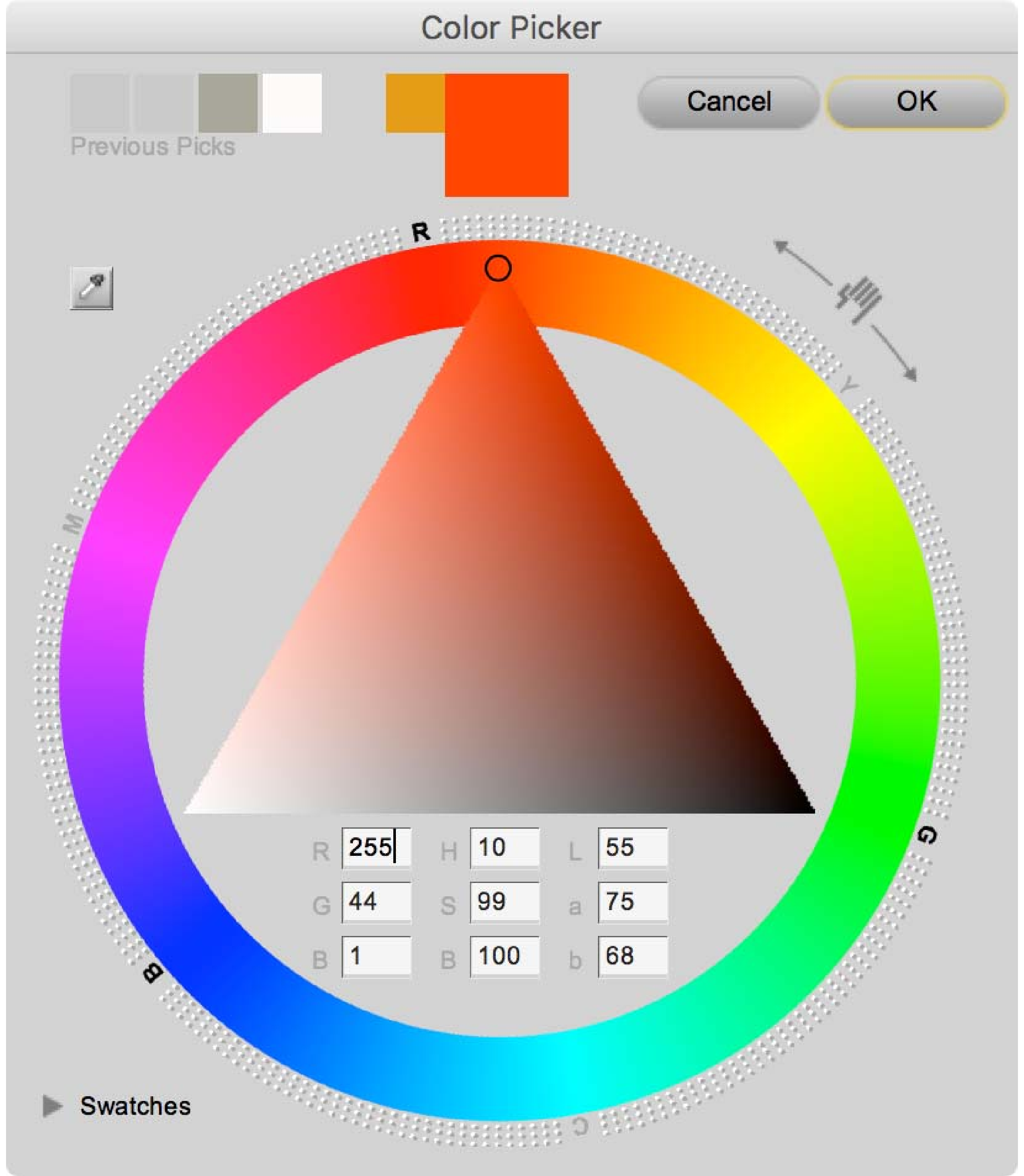
Color Space

Two ways of visualizing the same information.



● H: 0 °
○ S: 0 %
○ B: 0 %
○ R: 0
○ G: 0
○ B: 0
000000
○ L: 0
○ a: 0
○ b: 0
C: 75 %
M: 68 %
Y: 67 %
K: 90 %
□ Only Web Colors

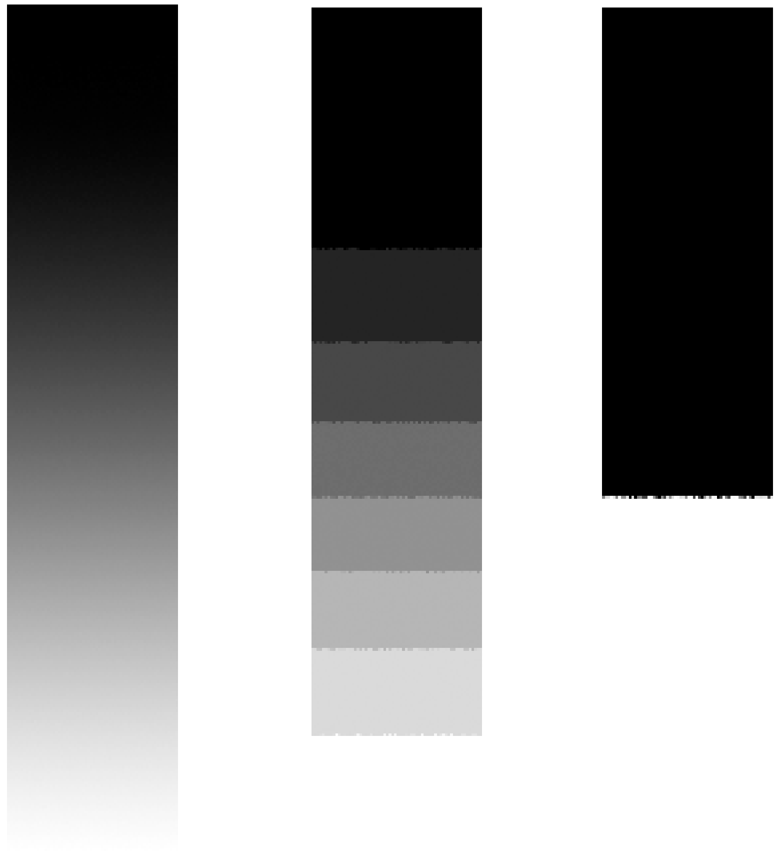
Adobe Photoshop



R 255 H 10 L 55
G 44 S 99 a 75
B 1 B 100 b 68

Nikon Capture NX

Grayscale Ramps



8 bit
256 steps

4 bit
8 steps

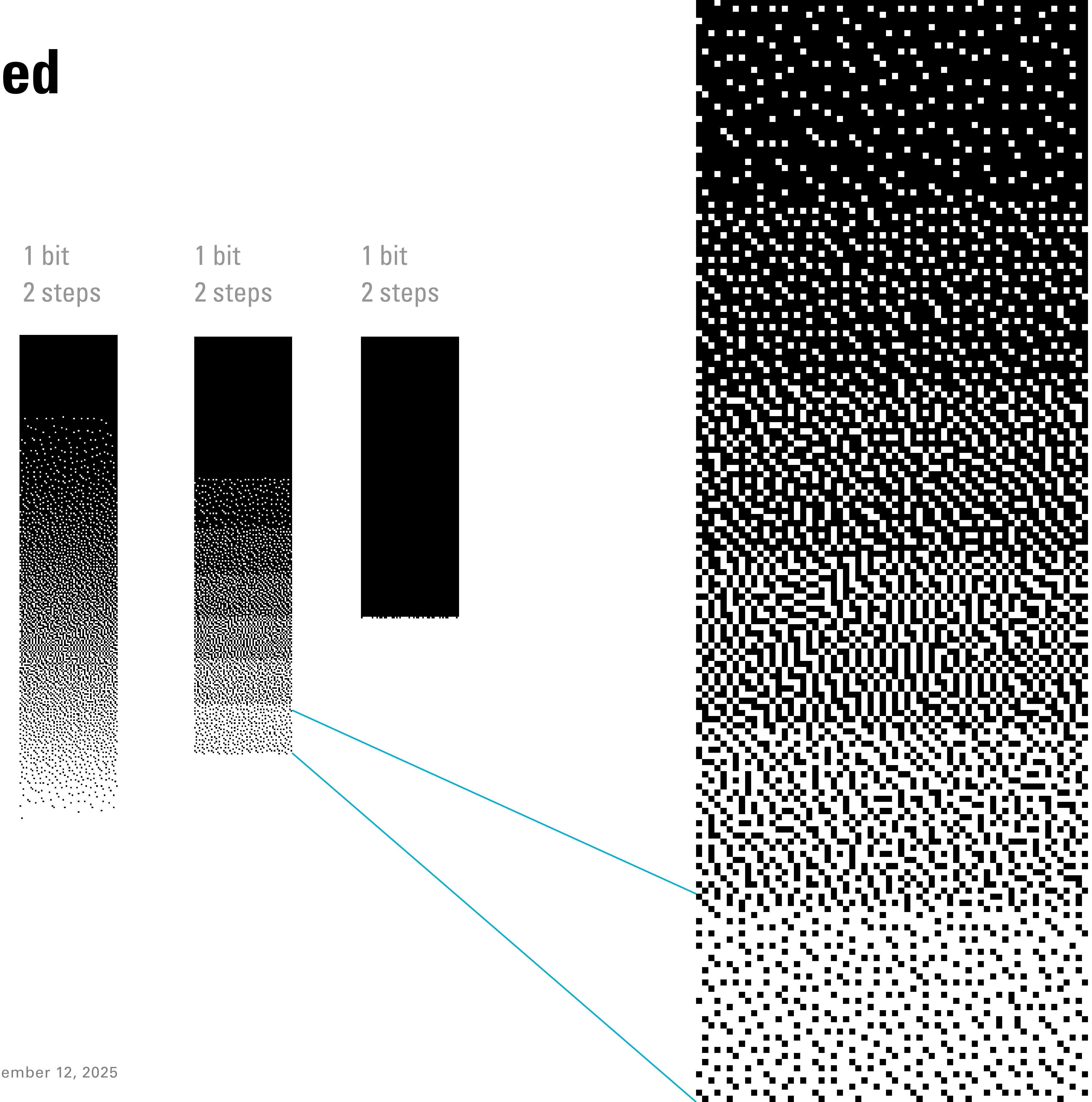
1 bit
2 steps

Grayscale Ramps, **Posterized**

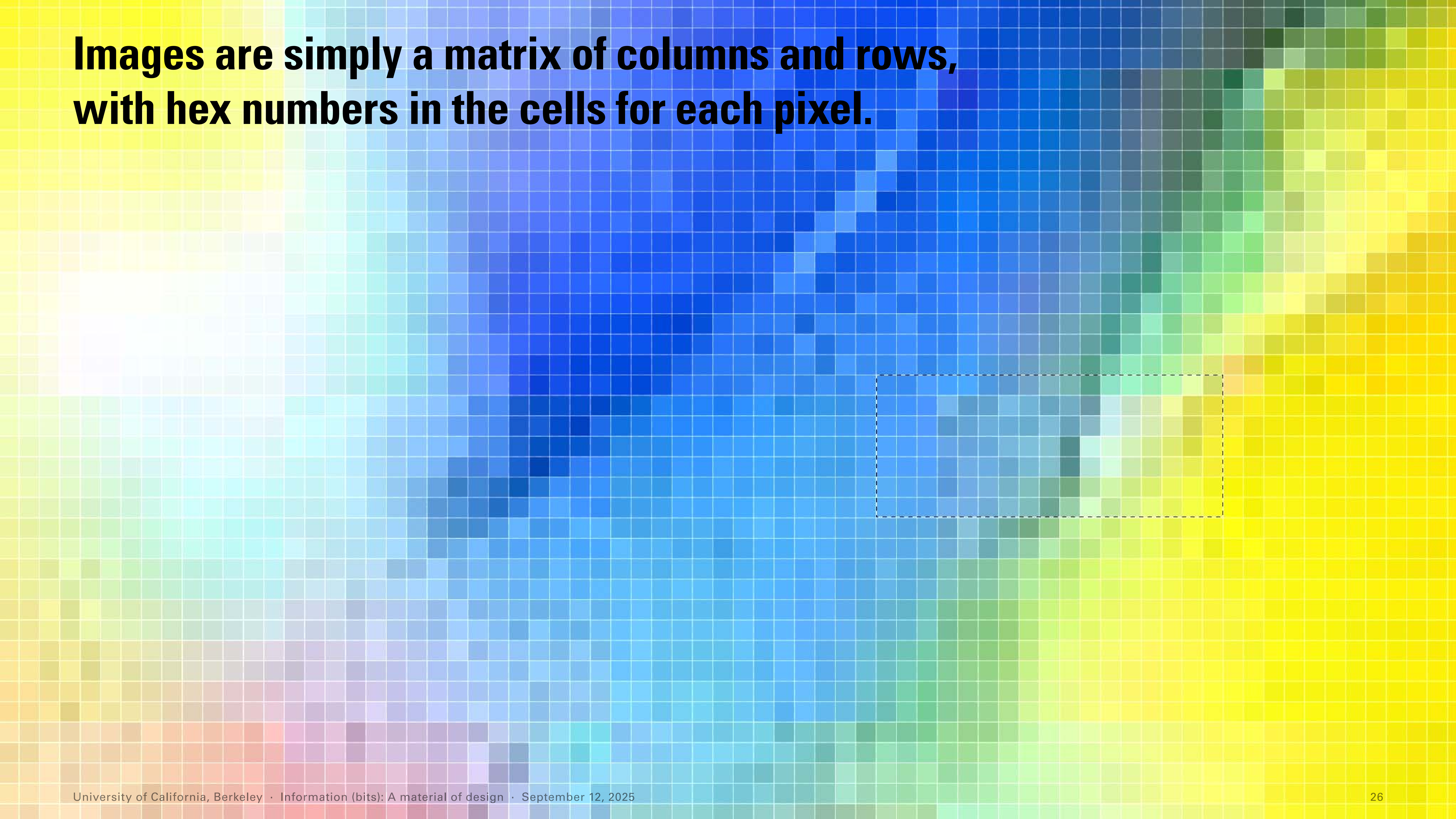
1 bit
2 steps

1 bit
2 steps

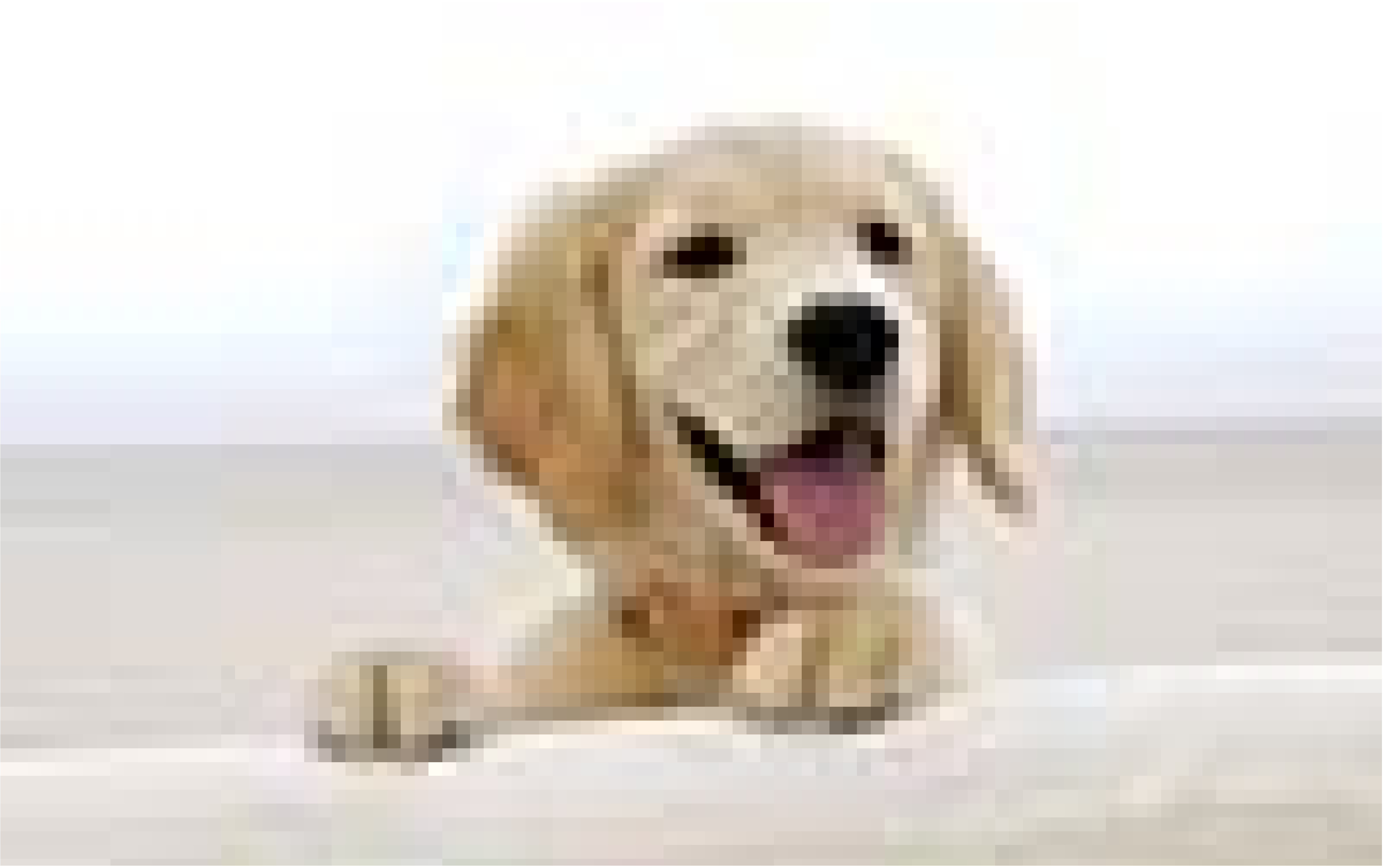
1 bit
2 steps



Images are simply a matrix of columns and rows, with hex numbers in the cells for each pixel.

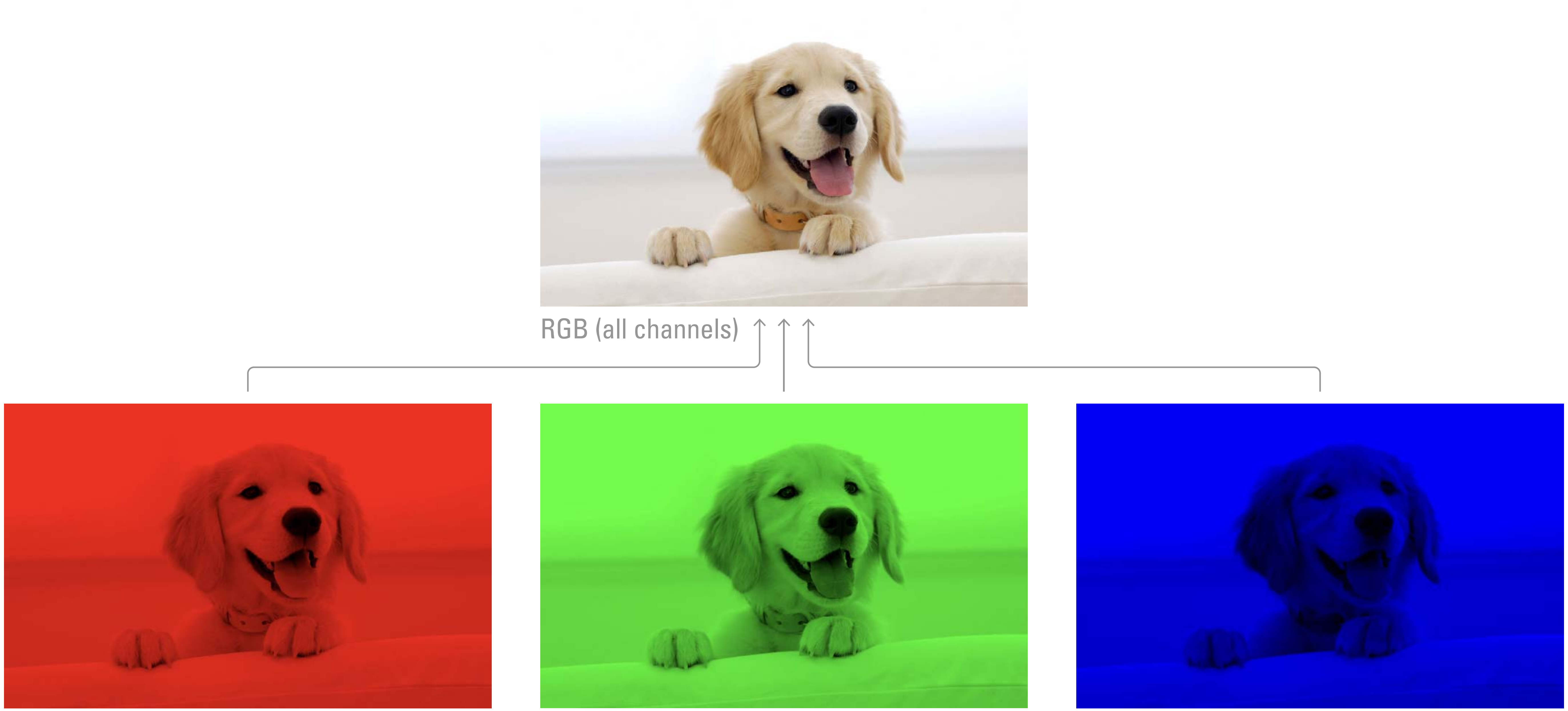


Small images are a smaller matrix with **less data** = smaller file size.



Large images are a larger matrix with **more data** = larger file size.

Splitting the channels of the image into R, G, and B.



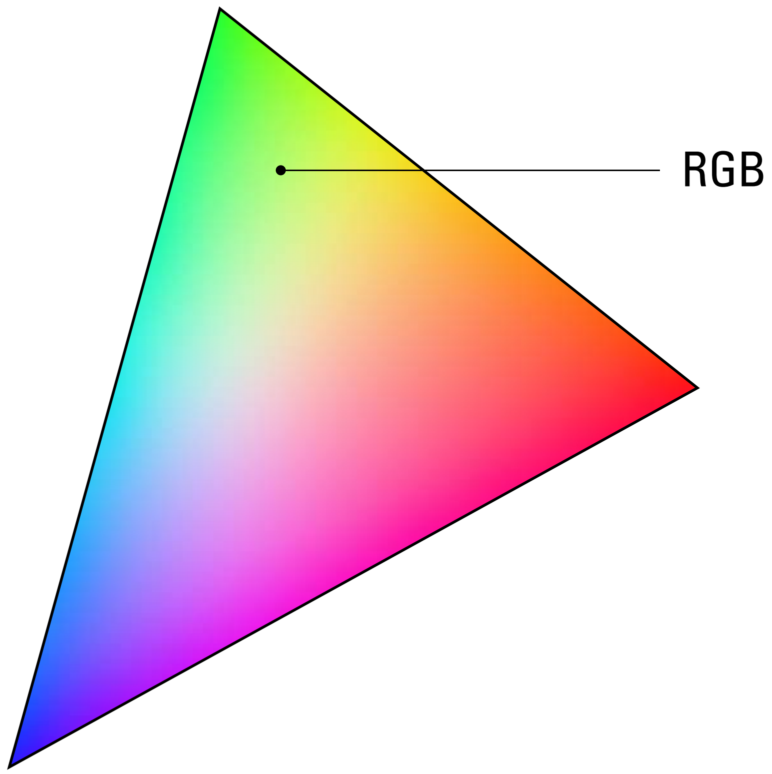
RGB (all channels)

Red channel only

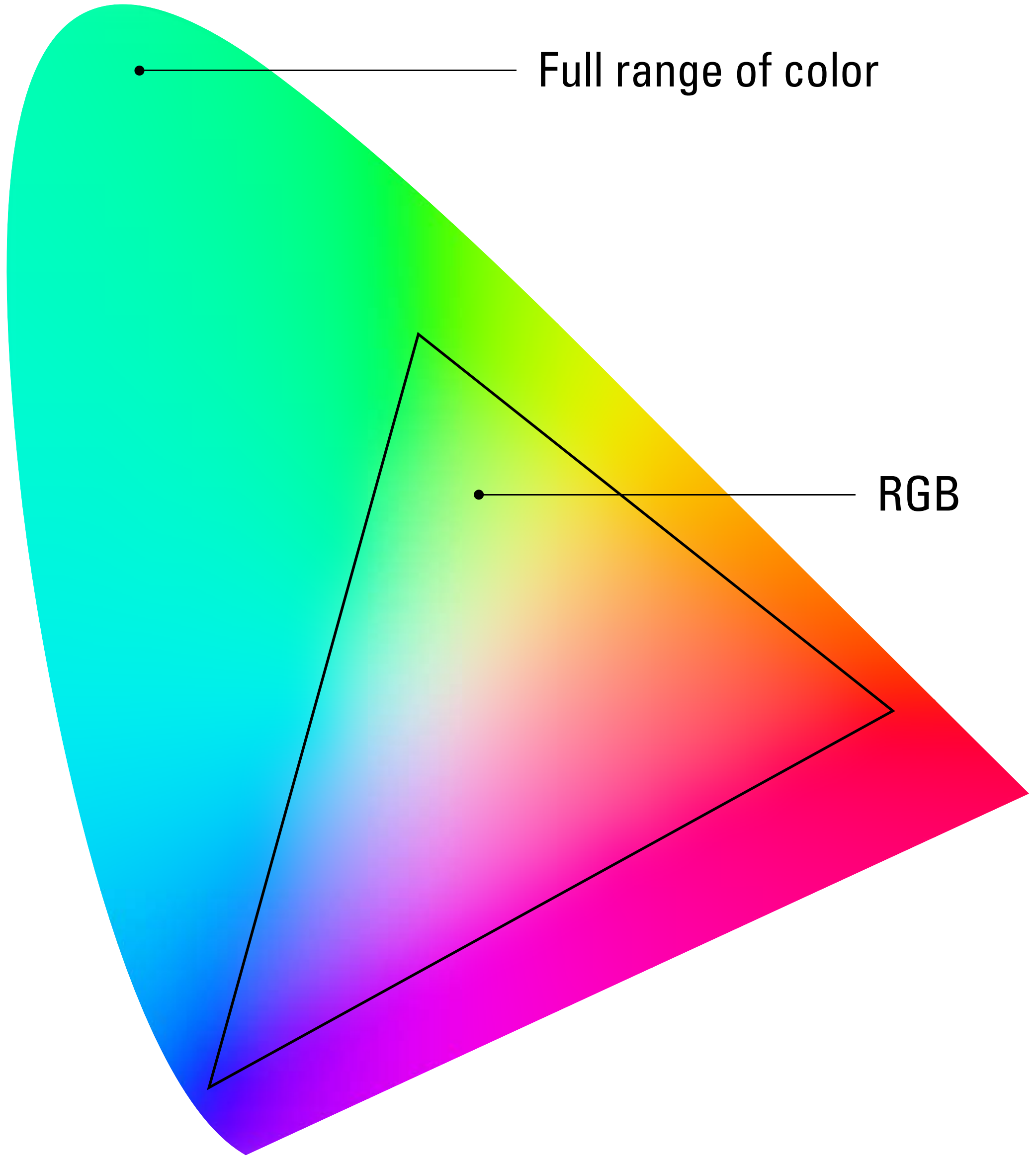
Green channel only

Blue channel only

sRGB includes a large range of colors.



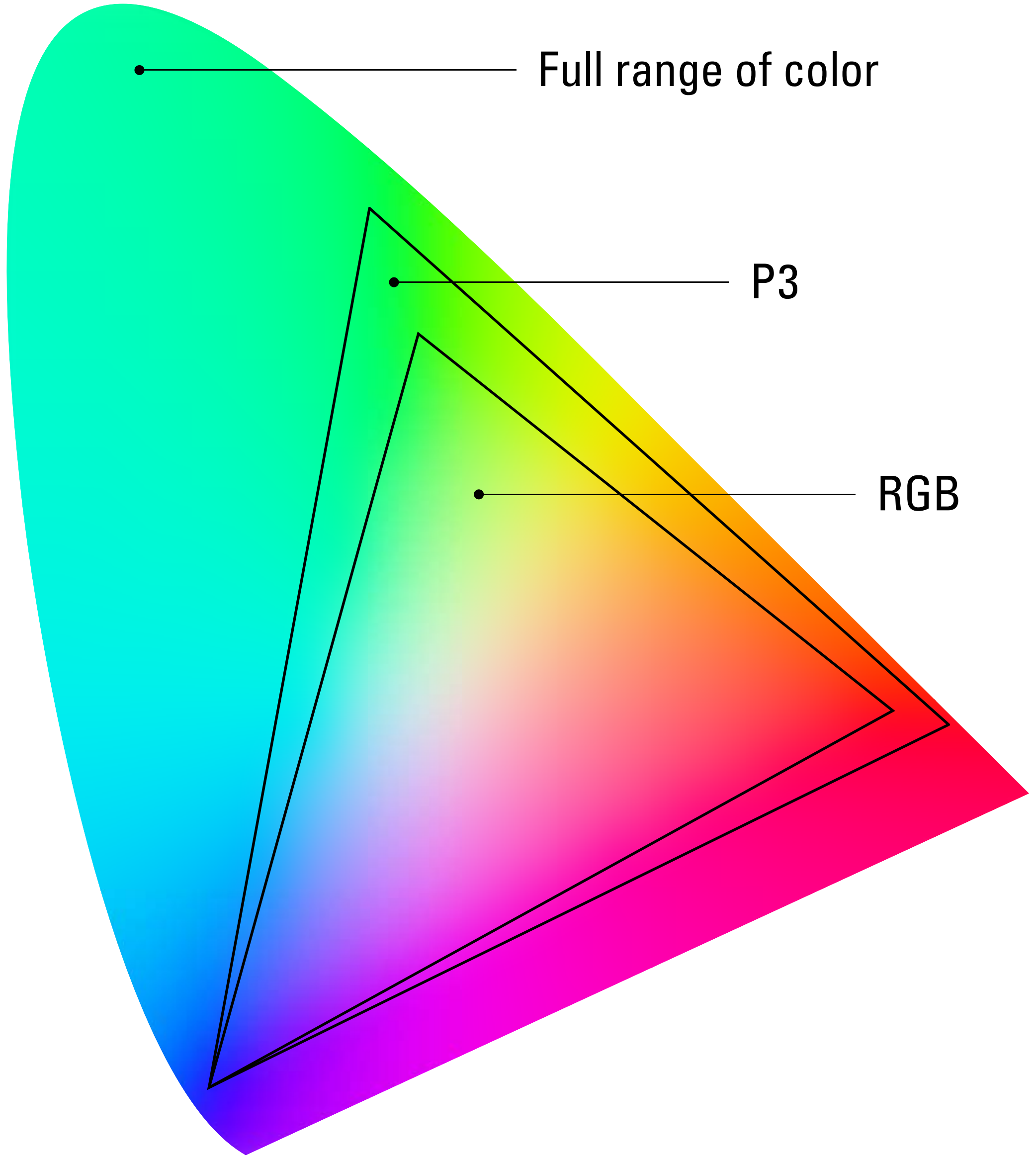
The full range of colors is much larger.



Full range of color

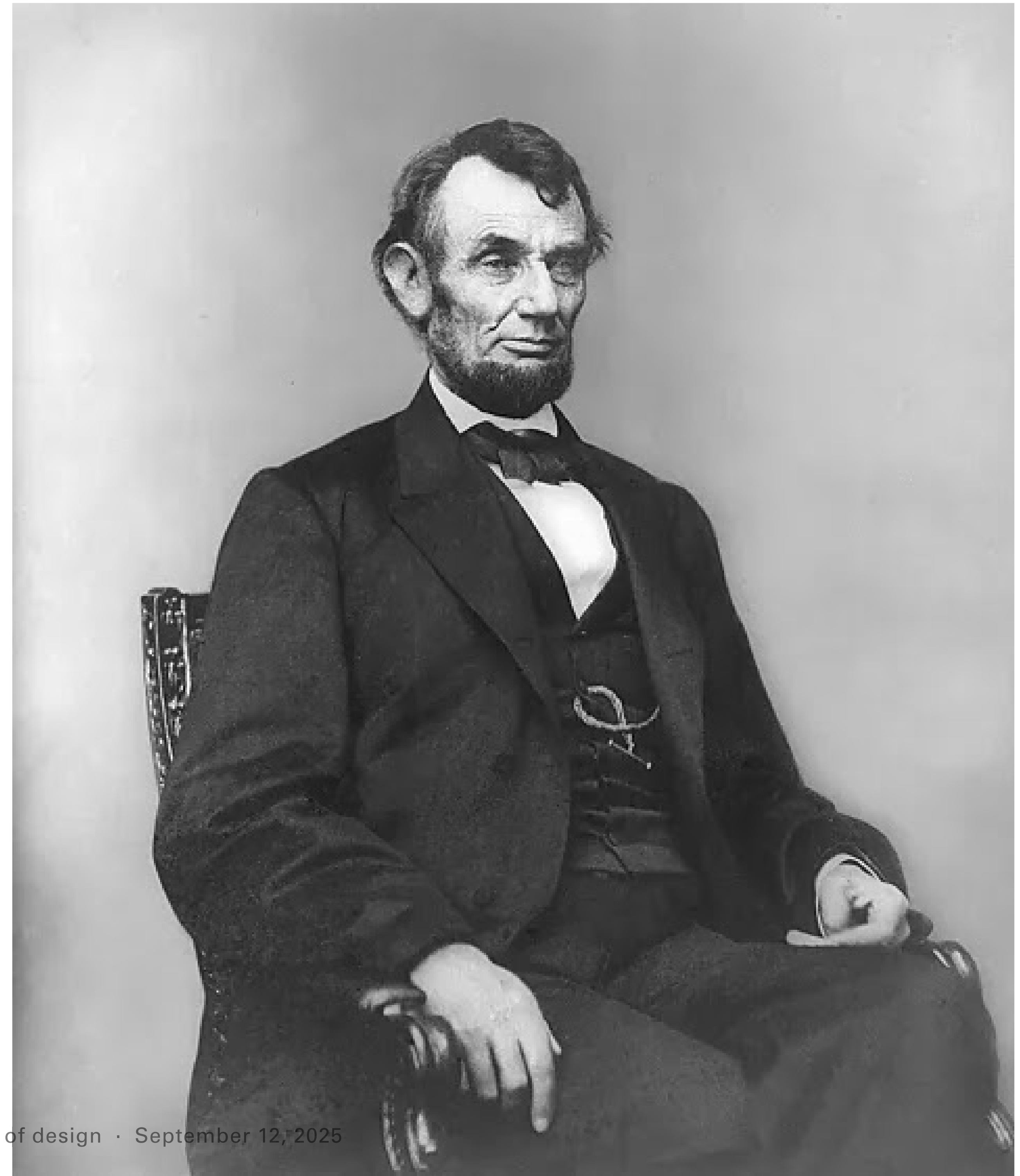
RGB

P3 includes more colors than RGB,
expanding the range of colors that can be displayed on screen.



Original analog photograph digitized

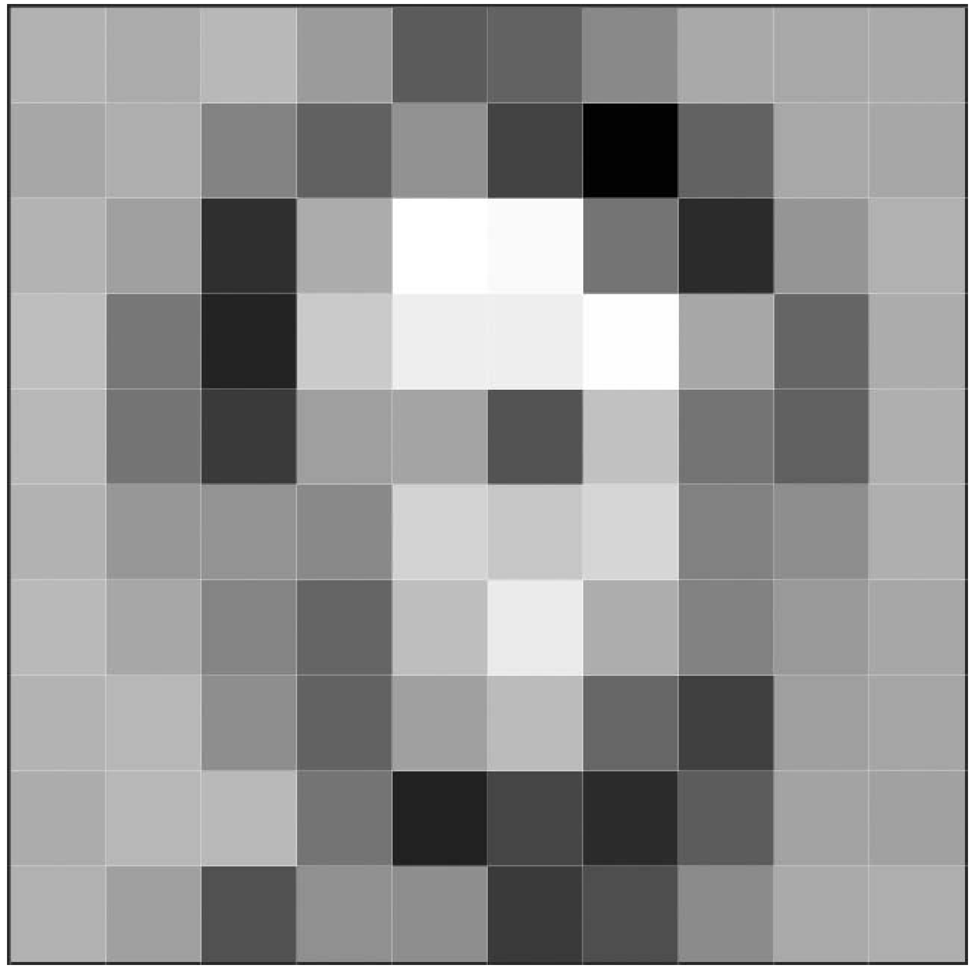
Abraham Lincoln, by Anthony Berger, February, 1864



Digital photo cropped

Cropped downsampled file enlarged,

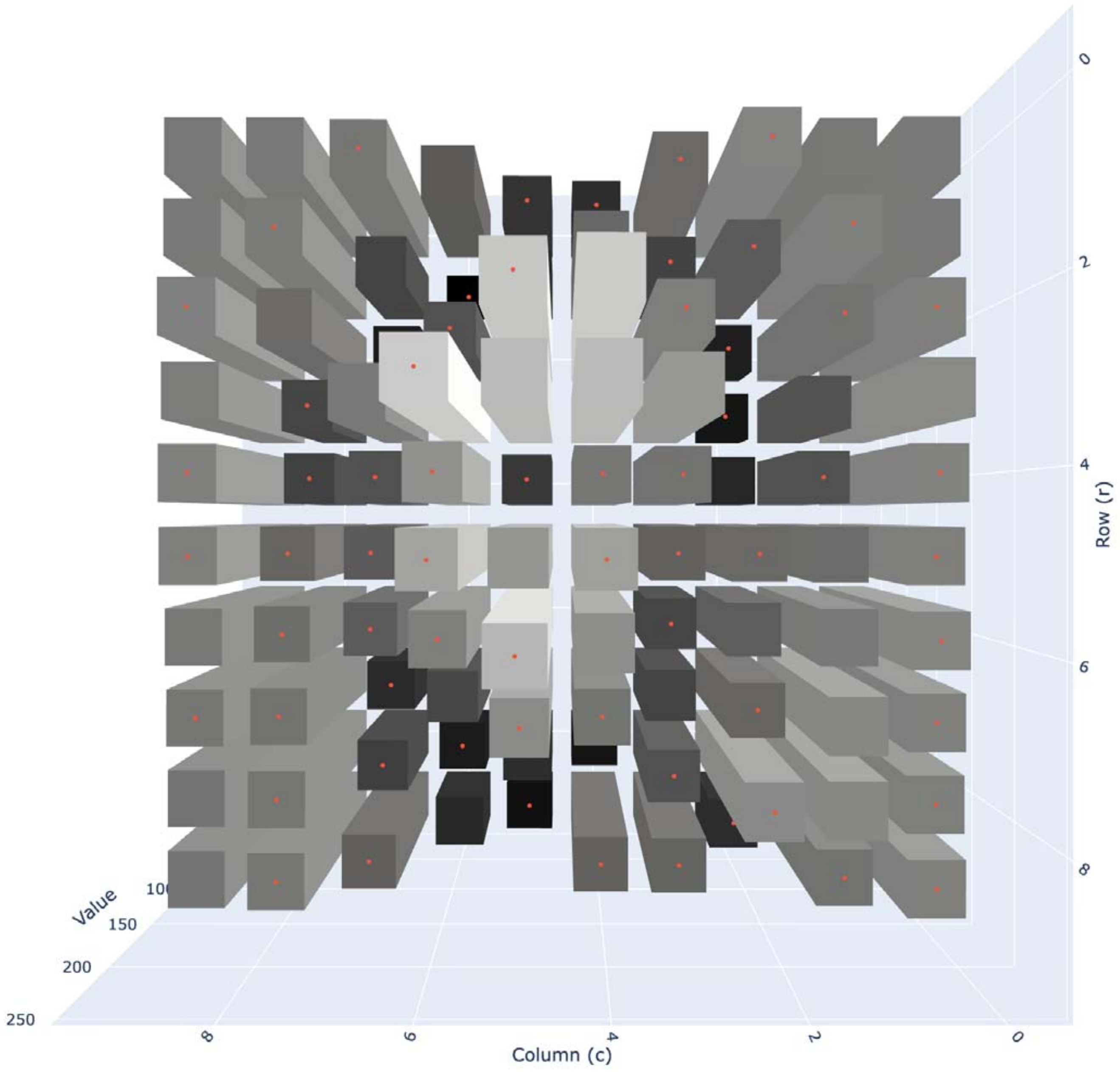
image pixels made of many screen pixels.



Cropped photo downsampled to 10x10 pixels, thus very very small (original file).

Downsampled file with each pixel as a bar (in a bar chart).

Height = gray value, black = 0; white = 255



Three-quarter view of the bar chart.

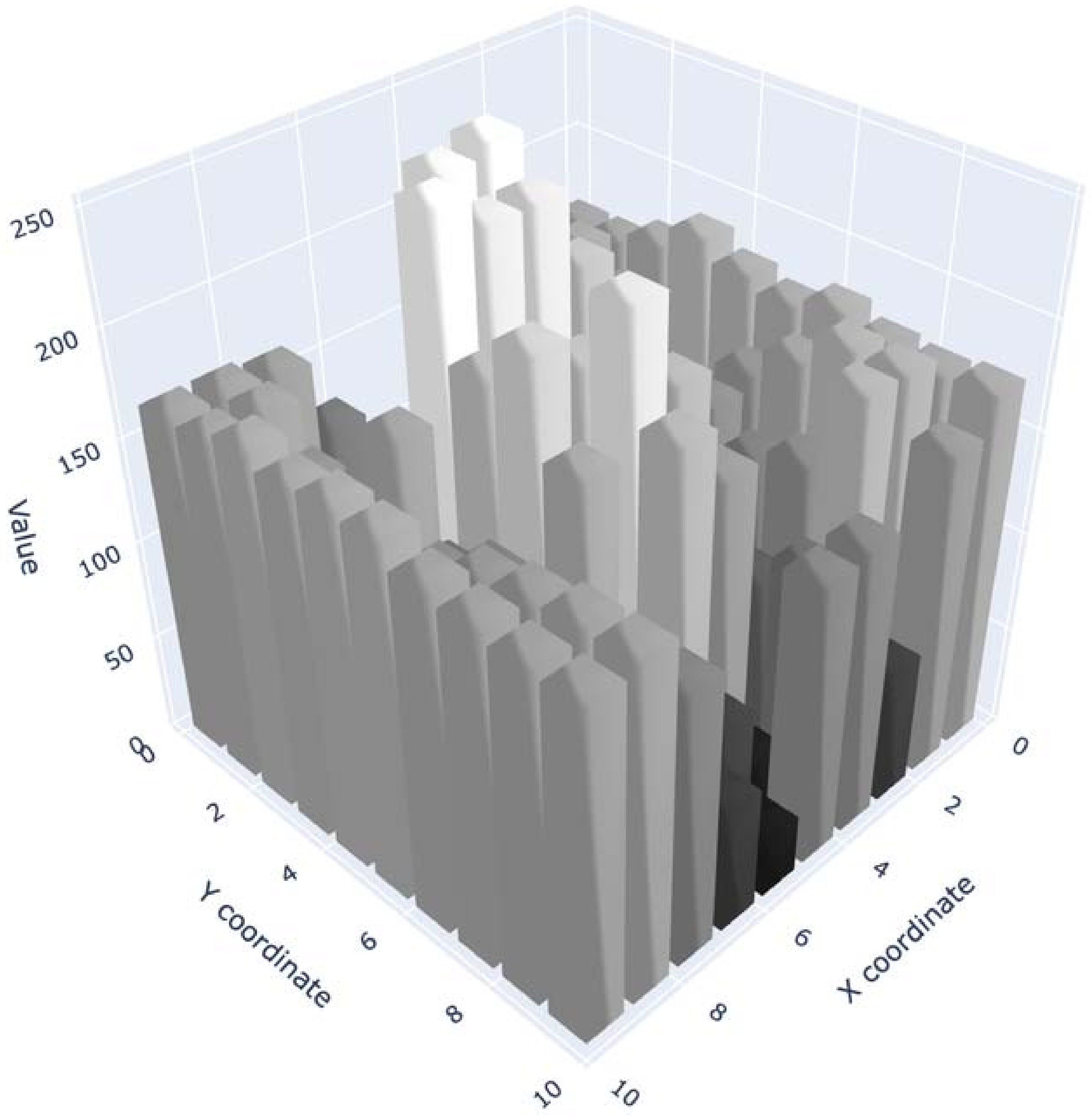
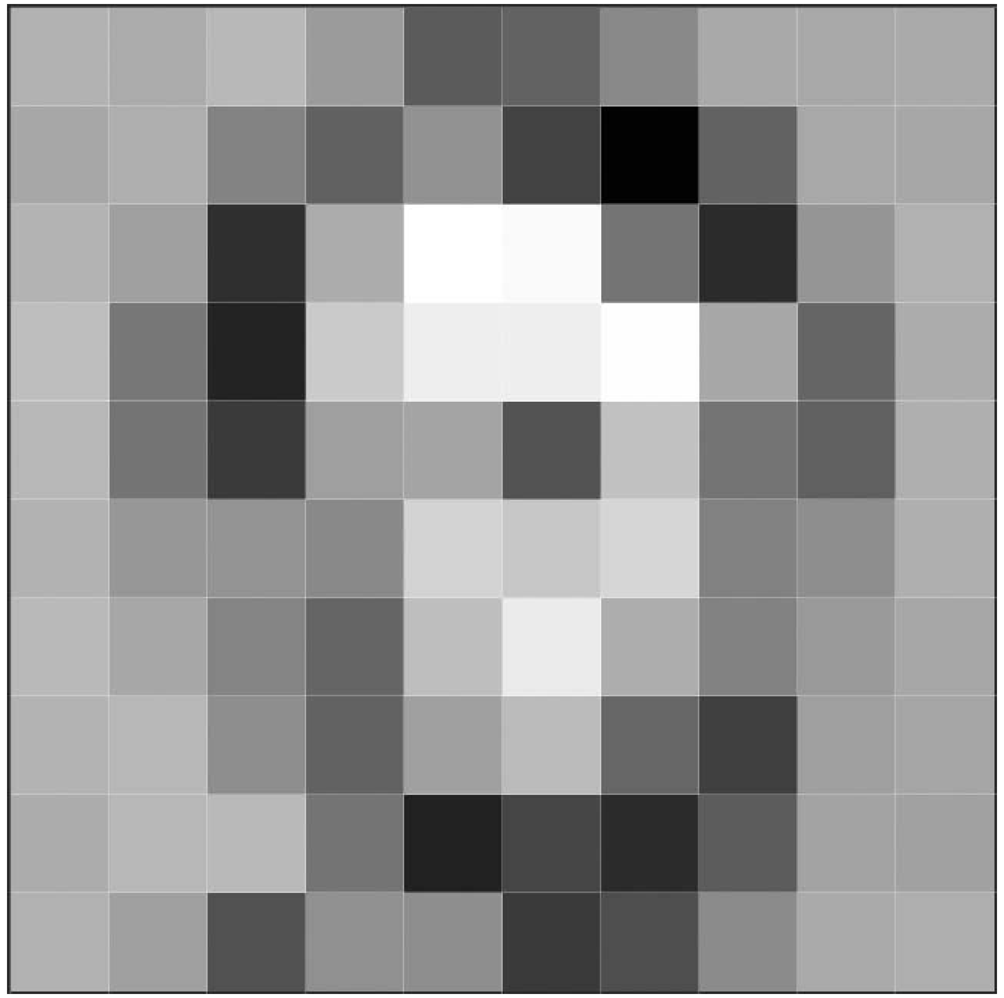


Table with the actual values.

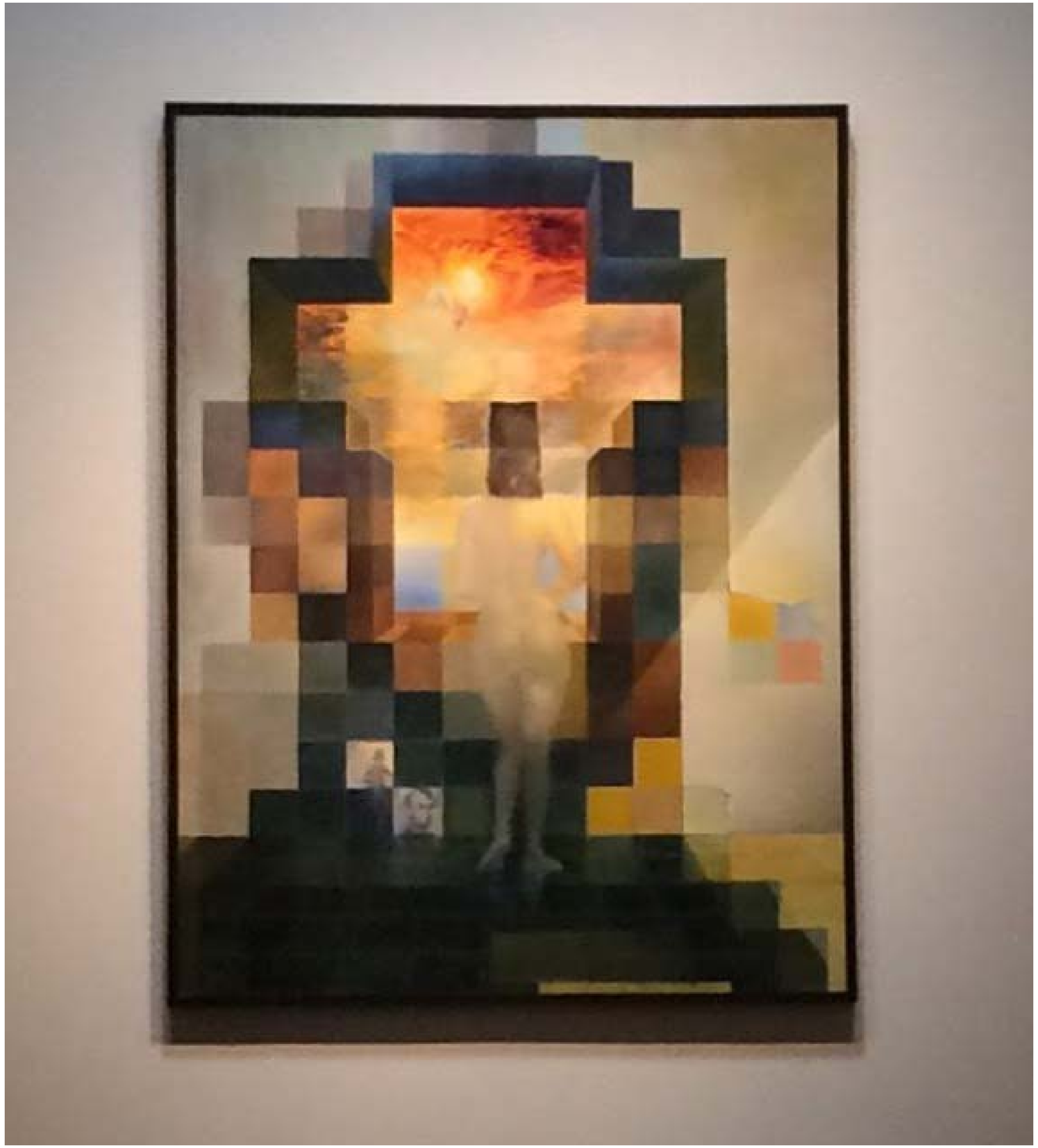
167	165	179	147	68	77	127	163	166	166
167	174	131	97	146	66	2	98	168	167
174	159	47	172	255	250	116	43	149	175
186	119	35	202	238	238	254	167	101	170
178	116	58	159	164	83	193	116	96	174
173	151	147	137	210	198	214	129	142	173
180	167	132	101	190	234	173	130	153	165
174	183	142	98	160	187	102	63	159	163
172	183	185	116	33	69	43	92	163	161
174	158	66	140	138	29	61	134	164	161



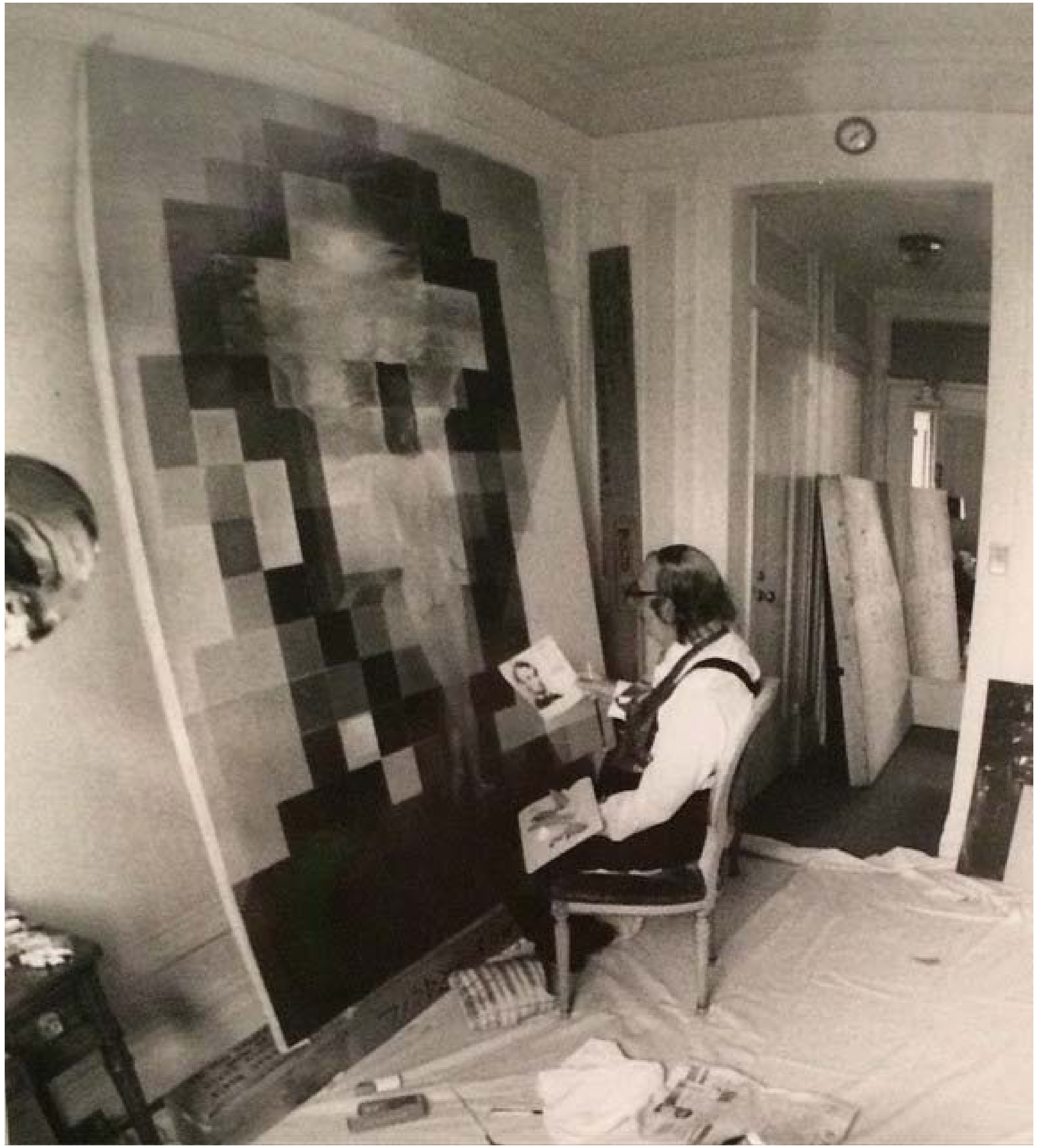
Salvador Dalí painting based on the downsampled photo.

Gala Contemplating the Mediterranean Sea
which at a distance of 20 meters is transformed into
the portrait of Abraham Lincoln (Homage to Rothko), 1974.

Later reprised in a lithograph as “Lincoln in Dalivision”.



Dalí working on the painting.



All applications may be understood in relation to their data types.

An “application” is a tool for **selecting** and **editing** the information in a file.

Understanding the **information’s structure**
helps you understand how it might be edited,
which helps you understand how the app’s interface might “work.”

…making you a more effective interaction designer,

…so that you can make the world a better place ;-)

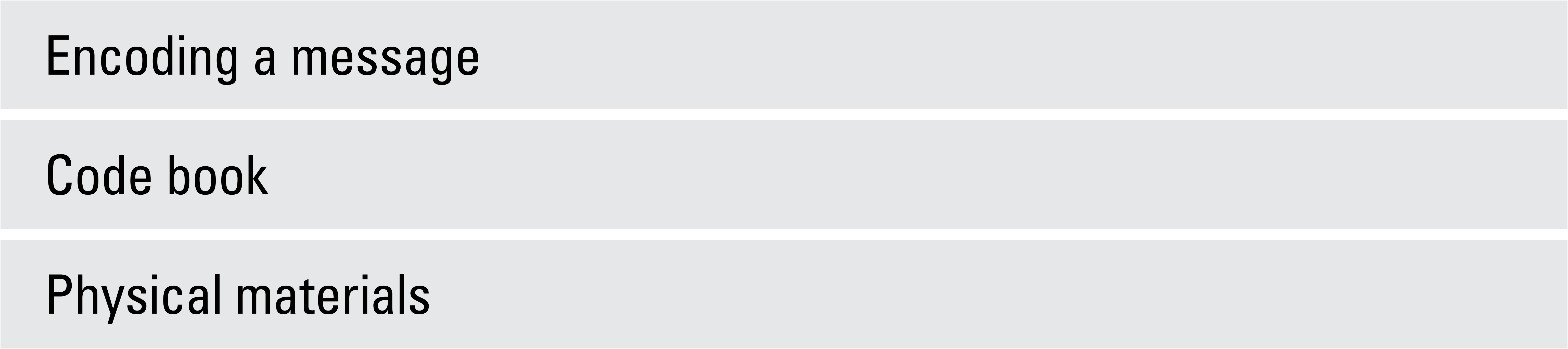
The industry jargon for “editing” is “CRUD”: Create, Read, Update, Delete.

For the last 25 years interaction design has **converged** on “normal”, but normal design is in crisis.

The material of design is once again **expanding.**

We are adding a **new layer** (or layers) to the stack.

Last week you made a simple stack.

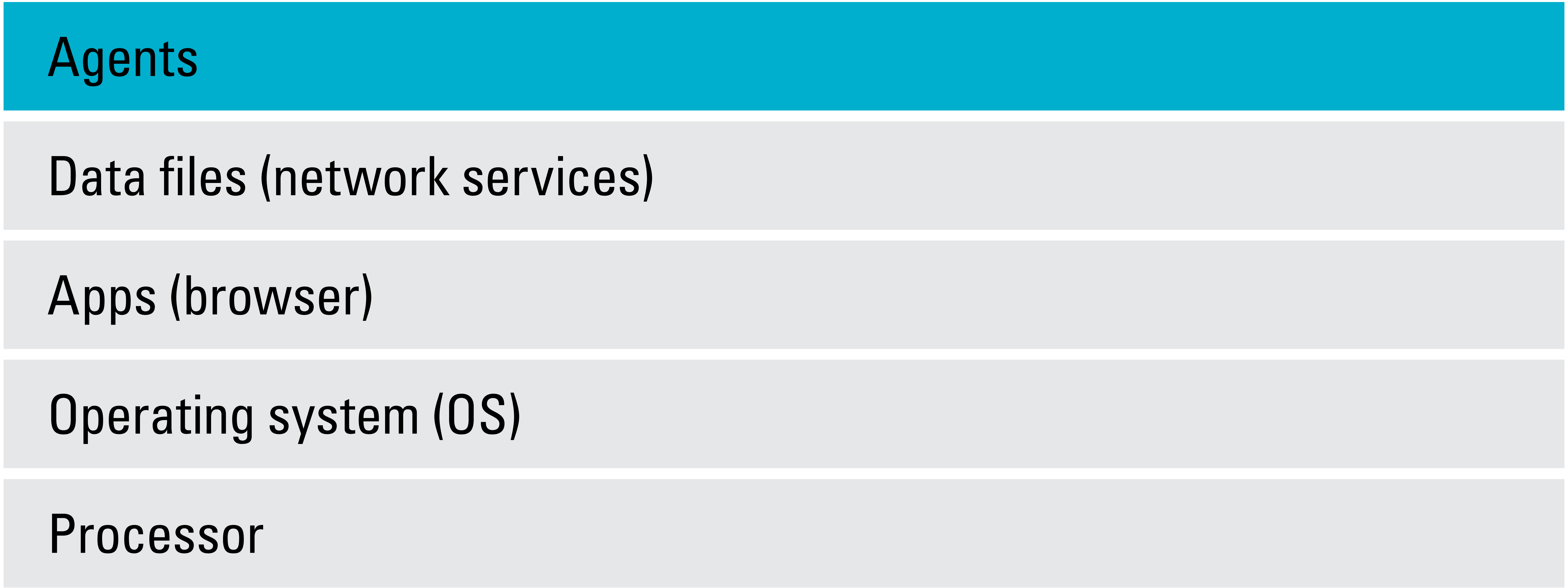


Encoding a message

Code book

Physical materials

Agentic systems are **a new layer atop** the tech stack.



Agents

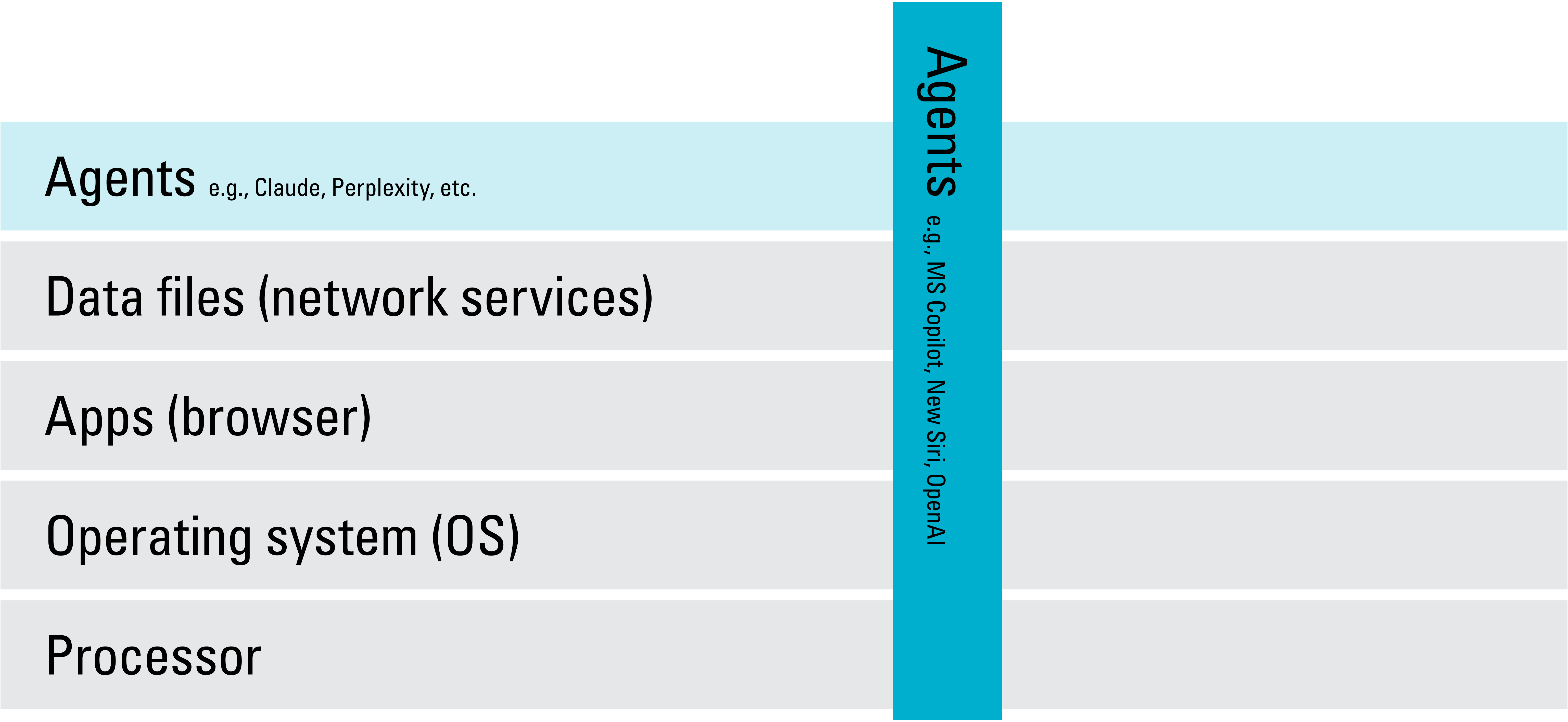
Data files (network services)

Apps (browser)

Operating system (OS)

Processor

Agentic systems are beginning to be **deeply integrated into** the stack.



Agents e.g., Claude, Perplexity, etc.

Data files (network services)

Apps (browser)

Operating system (OS)

Processor

Agents e.g., MS Copilot, New Siri, OpenAI

Here’s another way to look at the stack.

Vibe-coding with human-language prompts

Writing high-level languages (interpreted)

Writing assembly language (compiled)

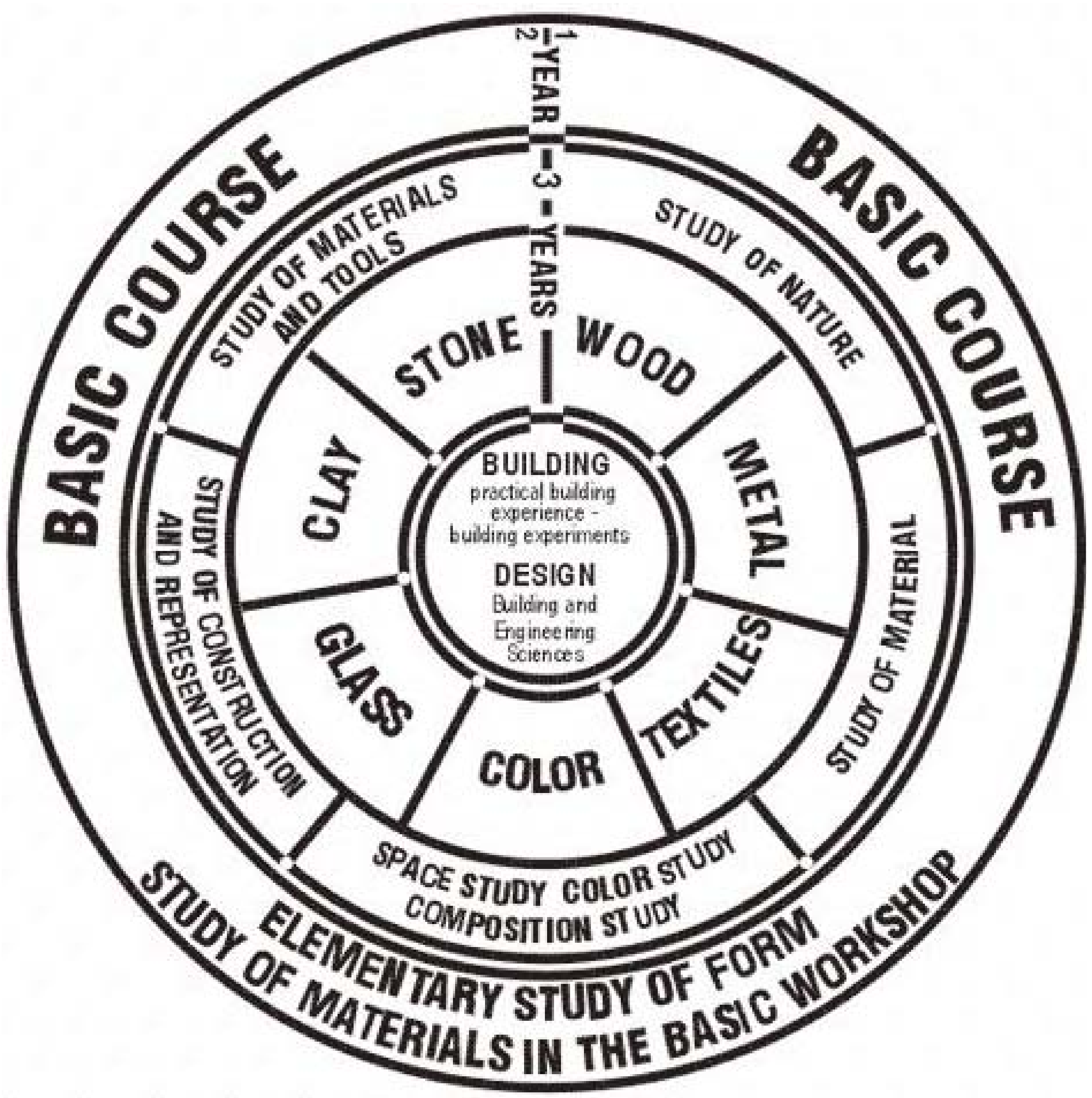
Writing machine code

Flipping switches + wiring

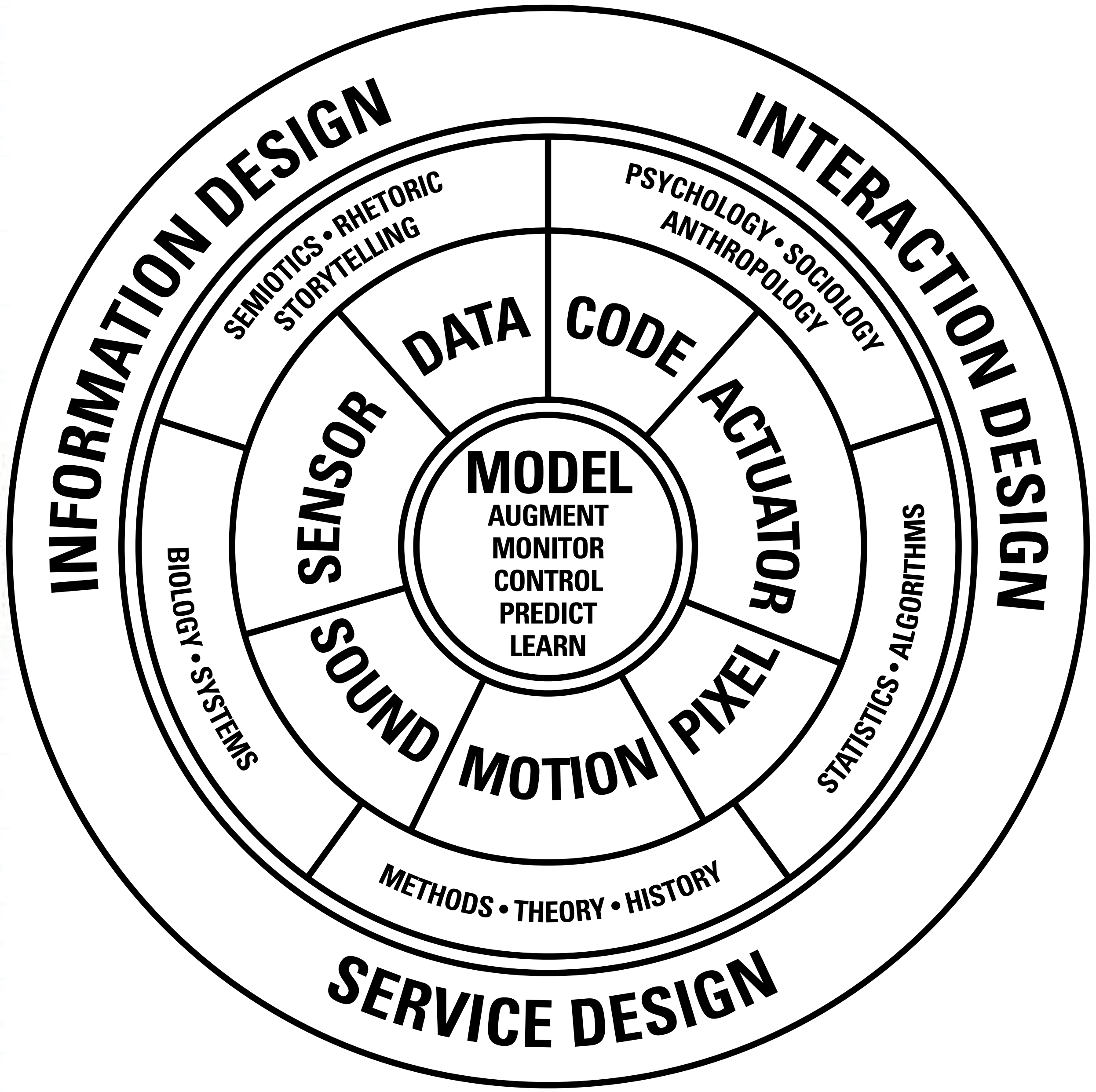
Physical current or magnetic charges

If Walter Gropius were to reform the Bauhaus School curriculum for today,
he might replace

the old 20th century materials… with the new 21st century materials.



1/2 YEAR
3 YEARS
BASIC COURSE
BASIC COURSE
STUDY OF MATERIALS AND TOOLS
STUDY OF NATURE
STONE
WOOD
CLAY
METAL
GLASS
TEXTILES
COLOR
BUILDING
practical building experience - building experiments
DESIGN
Building and Engineering Sciences
STUDY OF CONSTRUCTION AND REPRESENTATION
STUDY OF MATERIAL
SPACE STUDY COLOR STUDY
COMPOSITION STUDY
ELEMENTARY STUDY OF FORM
STUDY OF MATERIALS IN THE BASIC WORKSHOP



INFORMATION DESIGN
INTERACTION DESIGN
SERVICE DESIGN
SEMIOTICS • RHETORIC
STORYTELLING
PSYCHOLOGY • SOCIOLOGY
ANTHROPOLOGY
BIOLOGY • SYSTEMS
STATISTICS • ALGORITHMS
METHODS • THEORY • HISTORY
DATA
CODE
SENSOR
ACTUATOR
SOUND
PIXEL
MOTION
MODEL
AUGMENT
MONITOR
CONTROL
PREDICT
LEARN