



Base64 Encoding

In a previous article on MIME, you have met the Base64 format used to encode mail attachments. This post explores what Base64 encoding is, why it was created, and how it is used in email and web protocols.

What is base64 encoding?

Base64 is a binary-to-text encoding scheme that converts binary data into an ASCII string format. In essence, it encodes raw binary data into a set of 64 characters from the ASCII table: letters, numbers, and two special characters (typically + and /), with = used for padding.

For example:

Input (binary): 01001000 01100101 01101100 01101100 01101111 ("Hello" in ASCII)

Encoded Output: SGVsbG8=

Each Base64-encoded string represents three bytes of binary data split into four 6-bit groups, mapped to a character in the 64-character set.

Why was base64 created?

Base64 encoding was introduced to address a key problem: many communication protocols (like email) were designed to handle text, not raw binary data. Sending binary data (e.g., images, documents) through text-based protocols could corrupt the data or render it unreadable. Base64 was designed to:



- **Ensure compatibility:** By converting binary data into a readable and safe text format, Base64 ensures compatibility with systems that handle text-based data.
- **Preserve integrity:** Encoding binary data prevents misinterpretation or corruption caused by control characters or non-printable bytes.

How base64 works

To encode data with the Base64 encoding scheme:

1. Split the binary data into chunks of three bytes (24 bits);
2. Divide the 24 bits into four 6-bit segments;
3. Map each 6-bit segment to a character in the Base64 alphabet;
4. Add padding (=) if the input isn't a multiple of three bytes.

Here is the Base64 alphabet table:

Value (Decimal)	Value (Binary)	Base64 Character
0	000000	A
1	000001	B
2	000010	C
3	000011	D
4	000100	E
5	000101	F
6	000110	G
7	000111	H
8	001000	I
9	001001	J
10	001010	K
11	001011	L
12	001100	M
13	001101	N
14	001110	O
15	001111	P
16	010000	Q
17	010001	R
18	010010	S
19	010011	T



20	010100	U
21	010101	V
22	010110	W
23	010111	X
24	011000	Y
25	011001	Z
26	011010	a
27	011011	b
28	011100	c
29	011101	d
30	011110	e
31	011111	f
32	100000	g
33	100001	h
34	100010	i
35	100011	j
36	100100	k
37	100101	l
38	100110	m
39	100111	n
40	101000	o
41	101001	p
42	101010	q
43	101011	r
44	101100	s
45	101101	t
46	101110	u
47	101111	v
48	110000	w
49	110001	x
50	110010	y
51	110011	z

52	110100	0
53	110101	1
54	110110	2
55	110111	3
56	111000	4
57	111001	5
58	111010	6
59	111011	7
60	111100	8
61	111101	9
62	111110	+
63	111111	/

Example of encoding a binary string

Let's walk through an example of encoding a binary string:

1. **Binary Input:** 01000001 01000010 01000011 (which represents the ASCII characters "ABC").
2. **Split into 6-bit Groups:** 010000 010100 001001 000011.
3. **Convert to Decimal:** 16, 20, 9, 3.
4. **Map to Base64 Alphabet:** Using the table above, 16 = Q, 20 = U, 9 = J, 3 = D.
5. **Encoded String:** The result is QUJD.

If the input length isn't a multiple of 3 bytes, padding characters (=) are added to ensure proper encoding.

Decoding

Decoding reverses the process above, transforming the Base64 text back into its original binary format.

To decode a Base64-encoded string:

1. Map each character in the Base64 alphabet to its 6-bit representation;
2. Split the resulting binary string into 8-bit bytes.

Example of decoding a base64 string

1. **Base64 Input:** UHJldHR5.
2. **Map to Base64 Alphabet:** 010100 000111 001001 100101 011101 000111 010001 111001.
3. **Split into 8-bit bytes:** 01010000 01110010 01100101 01110100 01110100 01111001.
4. This is the original binary value. If we try to decode it in ASCII, this gives:
 1. 01010000 = 80 = 'P'
 2. 01110010 = 114 = 'r'
 3. 01100101 = 101 = 'e'
 4. 01110100 = 116 = 't'
 5. 01110100 = 116 = 't'
 6. 01111001 = 121 = 'y'
5. **Decoded String:** The result is Pretty.

Base64 in email communication

In email systems, Base64 is widely used in **MIME (Multipurpose Internet Mail Extensions)**. MIME extends the original email protocol (SMTP) to handle non-text content like attachments, audio, video, and images.

When you attach a file to an email:

- The binary file (e.g., an image) is encoded into Base64.
- The encoded data is included in the email as part of the message body or a MIME section.
- A header specifies the encoding type, such as Content-Transfer-Encoding: base64.

This ensures that the file is transmitted reliably, even if the email passes through systems that can't handle raw binary data.

Base64 in web communication

In web development, Base64 encoding is frequently used to:

1. **Embed Images in HTML or CSS:** Images can be encoded into Base64 and embedded directly into HTML or CSS as data URIs. For example:

```

```

This is particularly useful for small images, as it eliminates the need for separate HTTP requests.

2. **API Authentication:** Base64 is often used to encode credentials in HTTP Basic Authentication. For instance:
Username:Password pair (e.g., admin:password123) is encoded into Base64 (YWRtaW46cGFzc3dvcmQxMjM=) and sent in the Authorization header.
3. **Encoding Binary Data in JSON:** APIs that return or accept binary data (e.g., file uploads or downloads) may use Base64 to encode the binary payload into a JSON-compatible format.

Advantages and limitations of base64

Advantages:

- **Portability:** Ensures compatibility with text-based protocols.
- **Simplicity:** Easy to implement and understand.

Limitations:

- **Increased Size:** Base64 encoding increases the size of the data by approximately 33%, which can lead to inefficiencies when transferring large files.
- **Security Risks:** Base64 encoding is not encryption. Sensitive data encoded in Base64 can be easily decoded, so always pair it with encryption for secure communication.

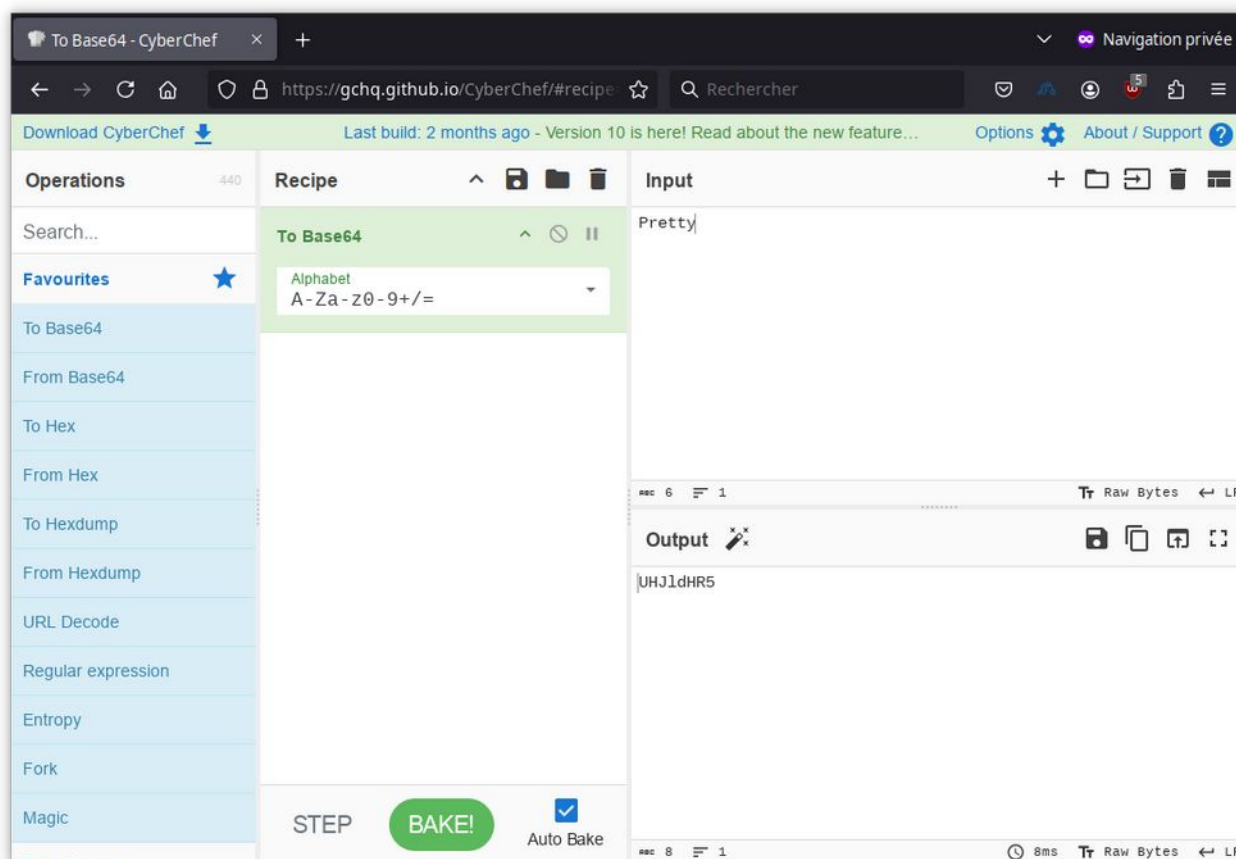
Useful tools for base64 encoding and decoding

Cyberchef

A practical tool for working with Base64 is [CyberChef](#), an intuitive web application for encoding, decoding, and manipulating data. CyberChef simplifies the process of converting to and from Base64.

Here is an example of encoding an ASCII string into Base64:

1. Type your input in the Input pane;
2. In the left Operations pane, click on the *Favourites* title, select *To Base64* and drag it to the *Recipe* center pane;
3. The converted string will appear in the *Output* pane.



To decode a Base64 string, type/paste it into the Input pane and drag the *From Base64* operation in the Recipe pane.

Code

You can find a didactic example of Python code implementing Base64 encoding and decoding in [my Codeberg repository](#).

Note: this code is for education purpose only; it is neither optimized nor hardened. Don't use it for production work.

Conclusion

Base64 may not be the most efficient way to handle binary data, but its universality and reliability make it indispensable for email and web communication. Understanding its role can help you troubleshoot issues more effectively, whether you're addressing broken email attachments, decoding API payloads, or embedding assets into web pages. As an IT professional, a solid grasp of Base64 can enhance your ability to navigate the complex interplay of modern communication protocols.