# Viewing and Interpreting Binary Data

## Hexadecimal

```
Decimal: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 ...
Hex:     0 1 2 3 4 5 6 7 8 9 A  B  C  D  E  F  10 11 12 13 ...
```

# Hexadecimal

Why is it useful to use the base 16?

# Hexadecimal

Why is it useful to use the base 16?

- 0-15 = 16 possibilities.
- 8 Bit = 1 Byte
- 4 Bit = 1/2 Byte
- 4 Bit = 2^4 = 16 possibilities

# Character encoding

ASCII (1977/1986)

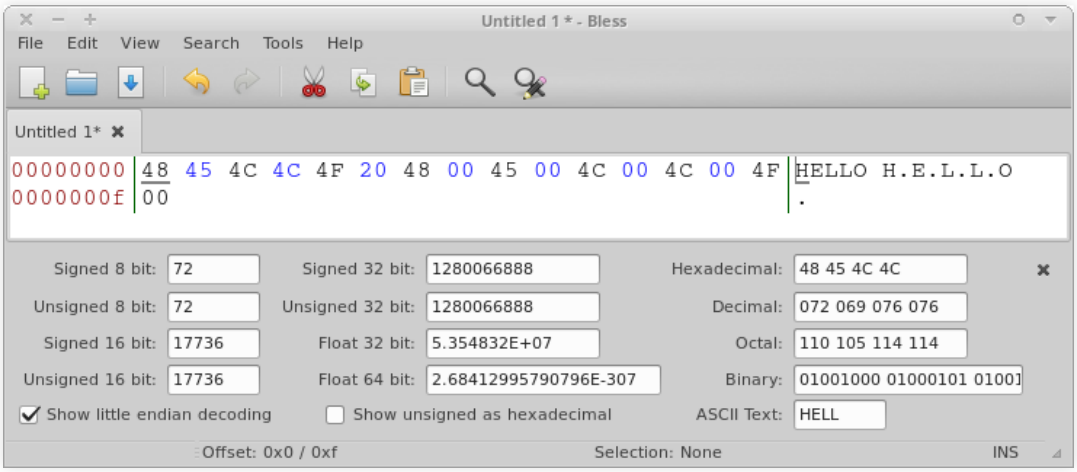
| | _0 | _1 | _2 | _3 | _4 | _5 | _6 | _7 | _8 | _9 | _A | _B | _C | _D | _E | _F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0_ 0 | NUL 0000 | SOH 0001 | STX 0002 | ETX 0003 | EOT 0004 | ENQ 0005 | ACK 0006 | BEL 0007 | BS 0008 | HT 0009 | LF 000A | VT 000B | FF 000C | CR 000D | SO 000E | SI 000F |
| 1_ 16 | DLE 0010 | DC1 0011 | DC2 0012 | DC3 0013 | DC4 0014 | NAK 0015 | SYN 0016 | ETB 0017 | CAN 0018 | EM 0019 | SUB 001A | ESC 001B | FS 001C | GS 001D | RS 001E | US 001F |
| 2_ 32 | SP 0020 | ! 0021 | " 0022 | # 0023 | $ 0024 | % 0025 | & 0026 | ' 0027 | ( 0028 | ) 0029 | * 002A | + 002B | , 002C | - 002D | . 002E | / 002F |
| 3_ 48 | 0 0030 | 1 0031 | 2 0032 | 3 0033 | 4 0034 | 5 0035 | 6 0036 | 7 0037 | 8 0038 | 9 0039 | : 003A | ; 003B | < 003C | = 003D | > 003E | ? 003F |
| 4_ 64 | @ 0040 | A 0041 | B 0042 | C 0043 | D 0044 | E 0045 | F 0046 | G 0047 | H 0048 | I 0049 | J 004A | K 004B | L 004C | M 004D | N 004E | O 004F |
| 5_ 80 | P 0050 | Q 0051 | R 0052 | S 0053 | T 0054 | U 0055 | V 0056 | W 0057 | X 0058 | Y 0059 | Z 005A | [ 005B | \ 005C | ] 005D | ^ 005E | _ 005F |
| 6_ 96 | ` 0060 | a 0061 | b 0062 | c 0063 | d 0064 | e 0065 | f 0066 | g 0067 | h 0068 | i 0069 | j 006A | k 006B | l 006C | m 006D | n 006E | o 006F |
| 7_ 112 | p 0070 | q 0071 | r 0072 | s 0073 | t 0074 | u 0075 | v 0076 | w 0077 | x 0078 | y 0079 | z 007A | { 007B | \| 007C | } 007D | ~ 007E | DEL 007F |

Letter Number Punctuation Symbol Other undefined

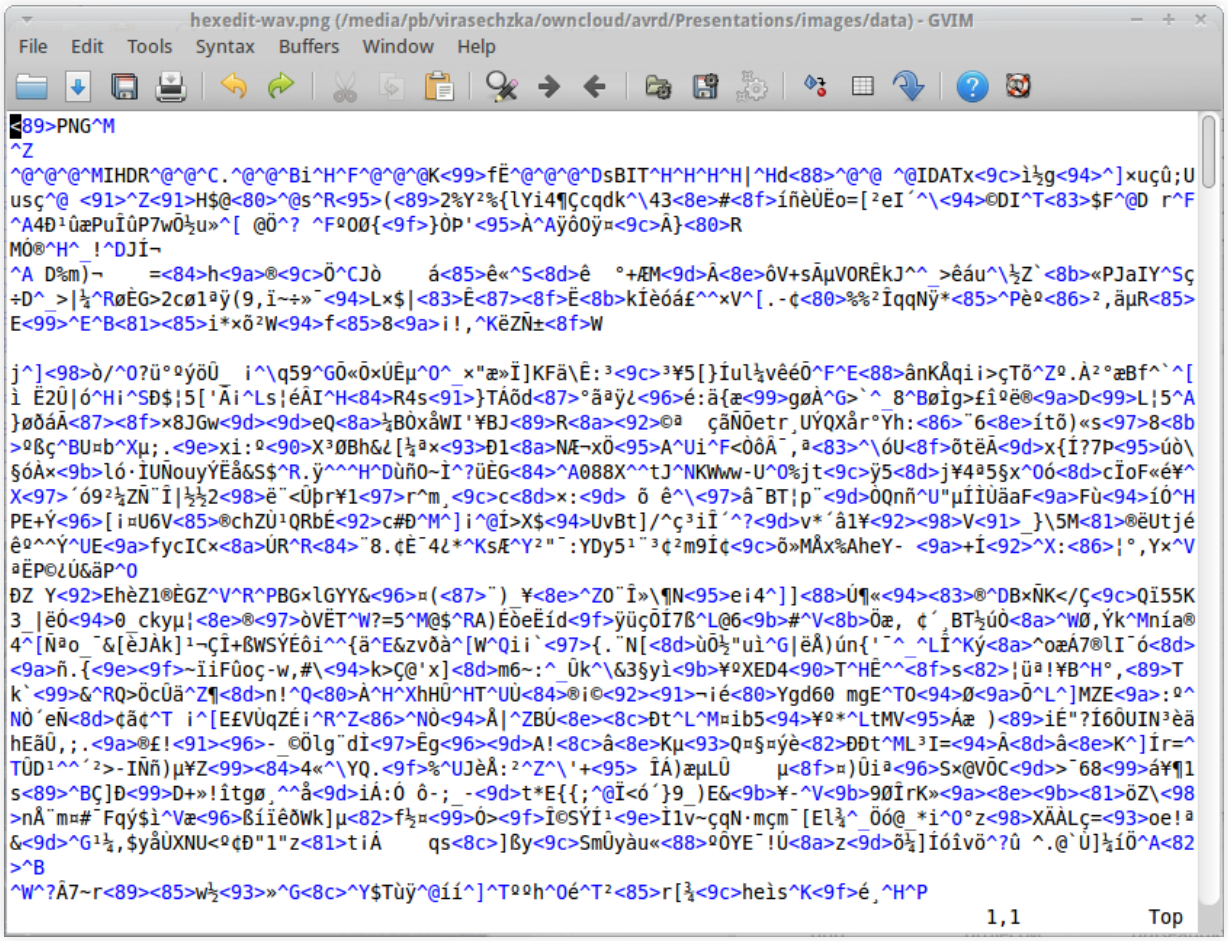
Speaker notes

A nicer ASCII table, but this time only with hex values.

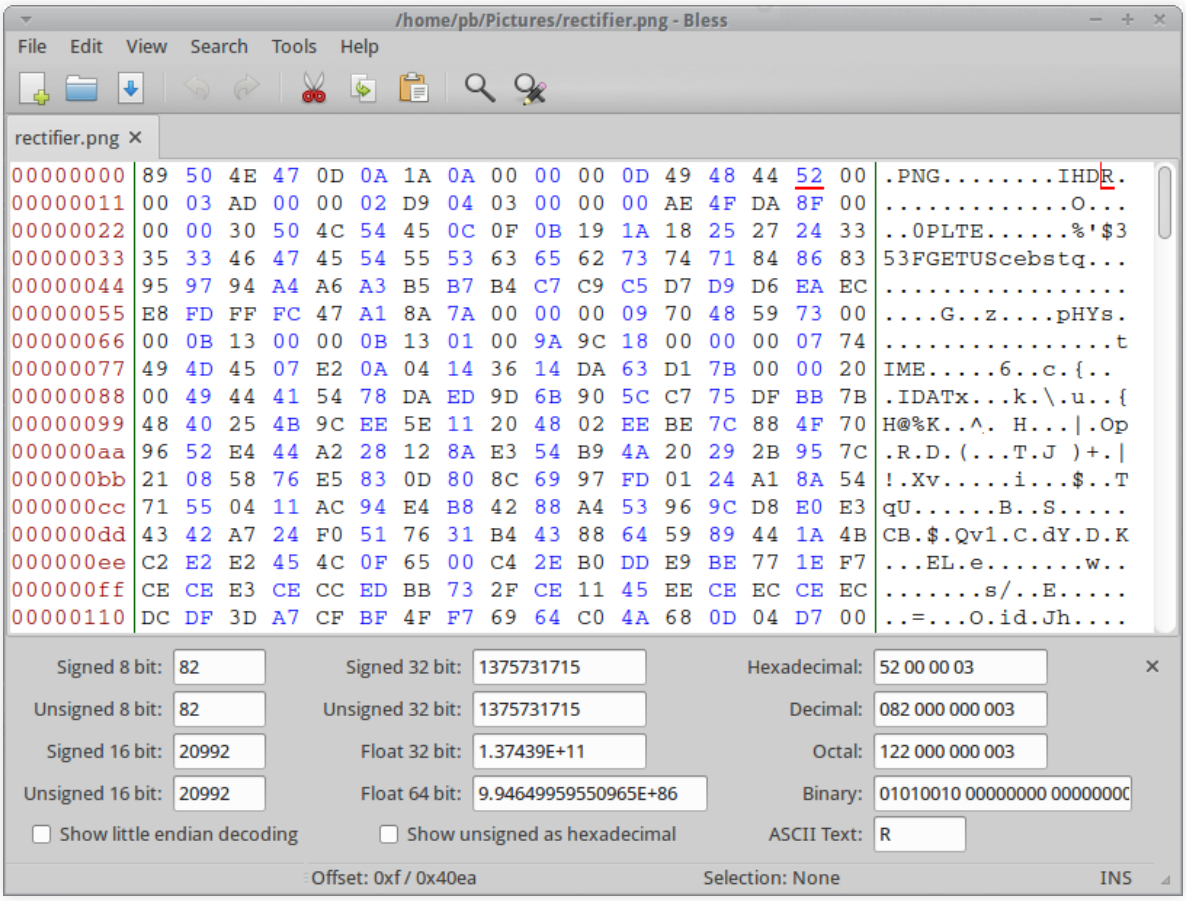
# Text as Data?




# Data as Text?


hexedit-wav.png (/media/pb/virasechzka/owncloud/avrd/Presentations/images/data) - GVIM
File Edit Tools Syntax Buffers Window Help
1,1 Top

# Hex editing!




# “Magic bytes”

- .PNG
- RIFF
- PK..
- JFIF
- AIFF
- .Eß£
- %PDF-
- 8BPS
- ...

See: List of File Signatures (Wikipedia)

Speaker notes

File formats are often defined so, that the first few characters contain a distinguishable sequence. This is called a “file signature”, “magic numbers” or “magic bytes”.

They can be used to quickly identify the filetype, regardless of the filename(-ending). Very useful for recovering deleted files or identifying wrongly renamed files.

# Exercise

*Identify the file types in the given set, using a Hexeditor and the “Magic Byte” list on Wikipedia.*

See: List of File Signatures (Wikipedia)

# Unix “file” command

```
Terminal - pb@pb-S21E: ~/magic_byte_set
File  Edit  View  Terminal  Tabs  Help
pb@pb-S21E:~/magic_byte_set$ file *
1:  PC bitmap, Windows 98/2000 and newer format, 70 x 46 x 24
10: Matroska data
11: ISO Media, MP4 Base Media v1 [ISO 14496-12:2003]
12: ISO Media, Apple QuickTime movie, Apple QuickTime (.MOV/QT)
2:  JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16,
baseline, precision 8, 70x46, frames 3
3:  GIF image data, version 89a, 70 x 46
4:  PNG image data, 70 x 46, 8-bit/color RGB, non-interlaced
5:  TIFF image data, little-endian, direntries=15, height=46, bps=9854, compression=non
e, PhotometricIntepretation=RGB, orientation=upper-left, width=70
6:  Adobe Photoshop Image, 70 x 46, RGB, 3x 8-bit channels
7:  Zip archive data, at least v2.0 to extract
8:  bzip2 compressed data, block size = 900k
9:  RIFF (little-endian) data, AVI, 320 x 240, 25.00 fps, video: H.264 X.264 or H.264
pb@pb-S21E:~/magic_byte_set$
```

See Wikipedia: File (command)

# MIME Type

*“**Multipurpose Internet Mail Extensions** (MIME) is an Internet standard that extends the format of email messages to support text in character sets other than ASCII, as well attachments of audio, video, images, and application programs.”*

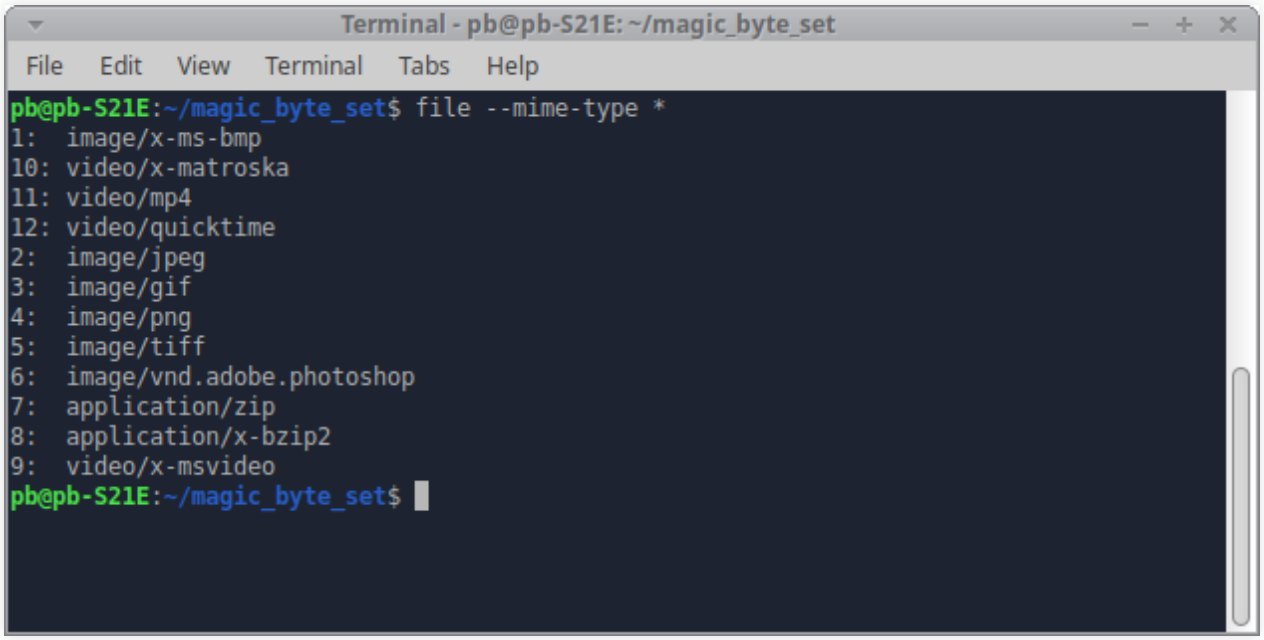
*– Wikipedia: Media Type*

# MIME Type Examples

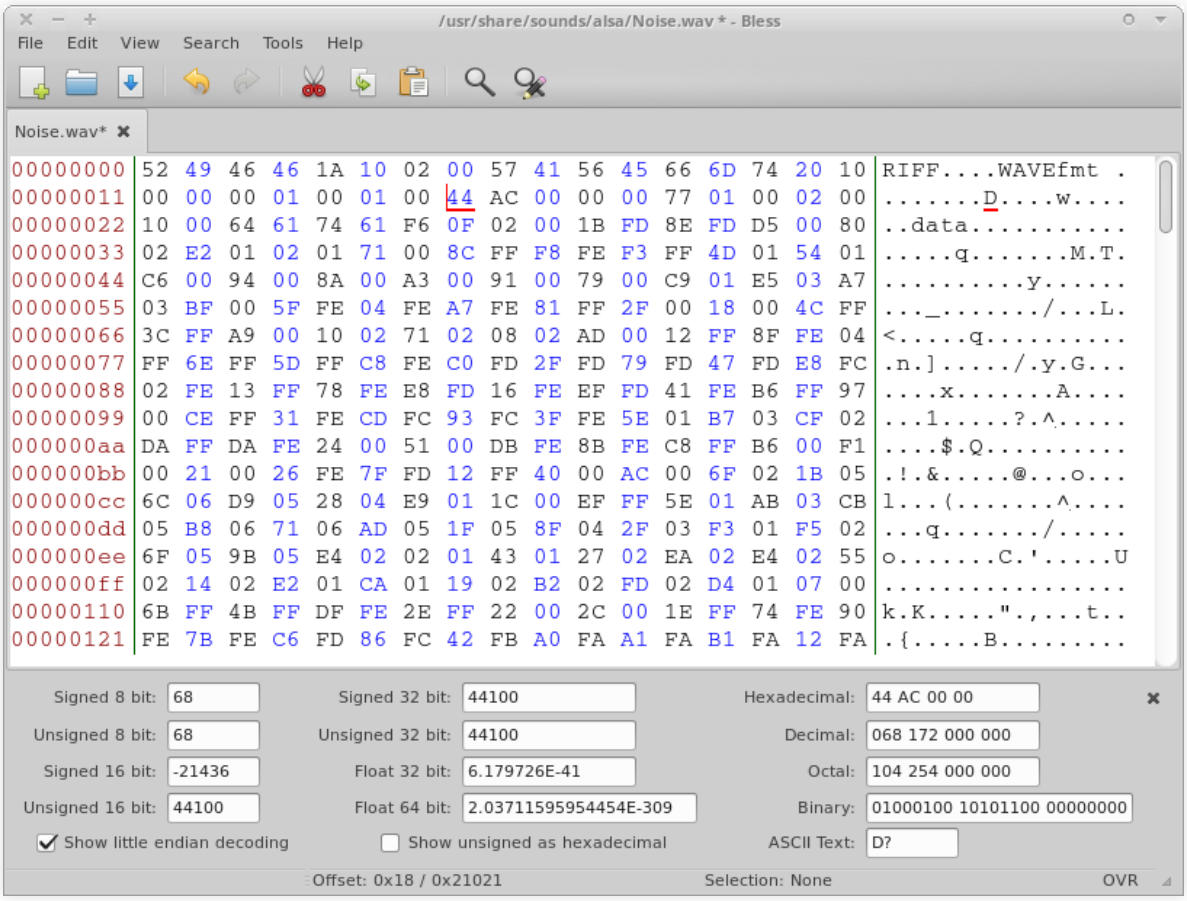
- application/zip
- application/pdf
- text/html
- text/xml
- text/csv
- text/plain
- image/png
- image/jpeg
- image/gif
- audio/aac
- audio/mpeg
- video/DV
- video/H264
- video/mp4

Complete List (IANA), 2019-10-16

# Remember our “no suffix” file set?



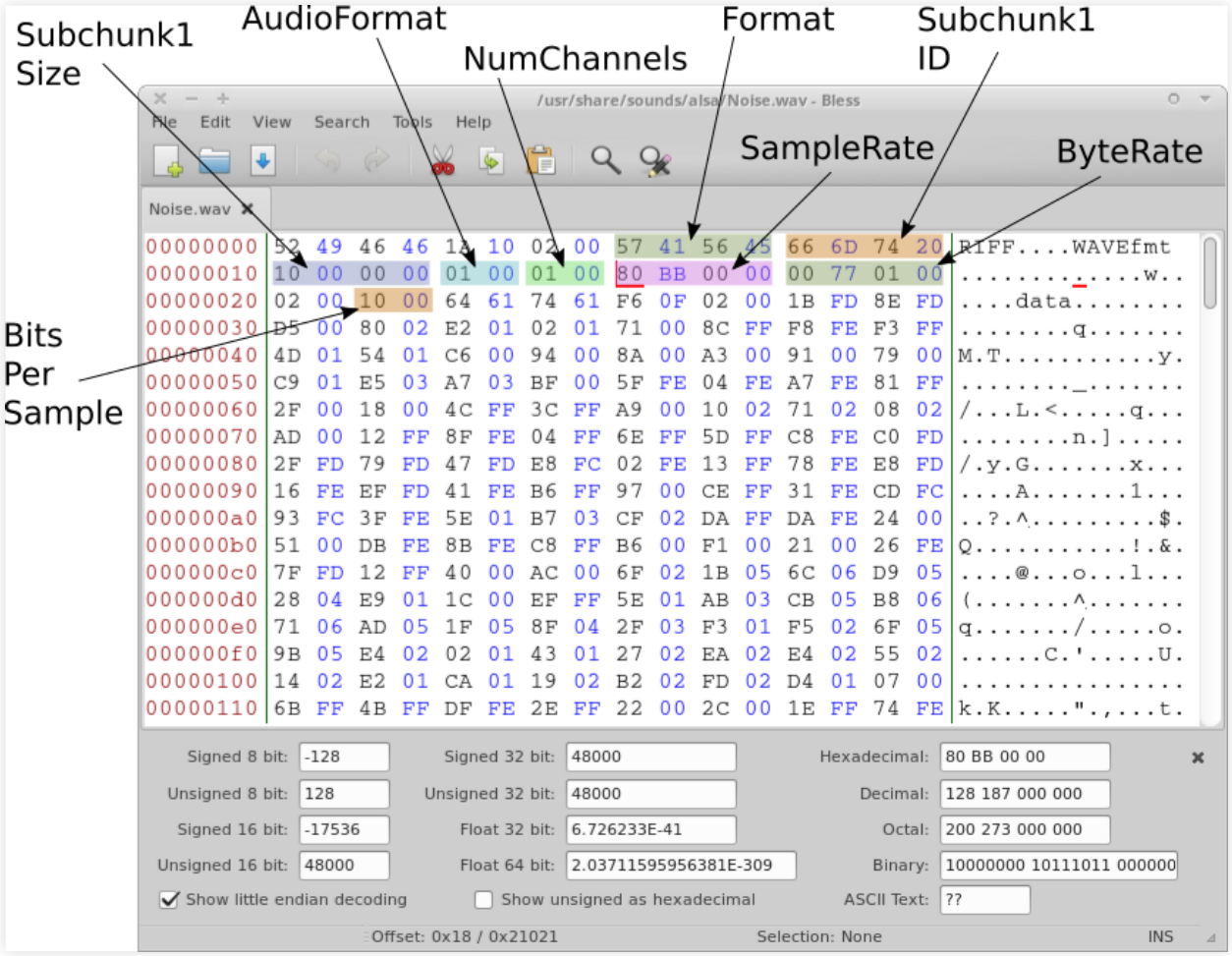

# Binary Data?




Speaker notes

For more information about binary data in media files, here's an introduction to hex & hex editing: "Hex Editing for Archivists"

# Data Structure



Speaker notes

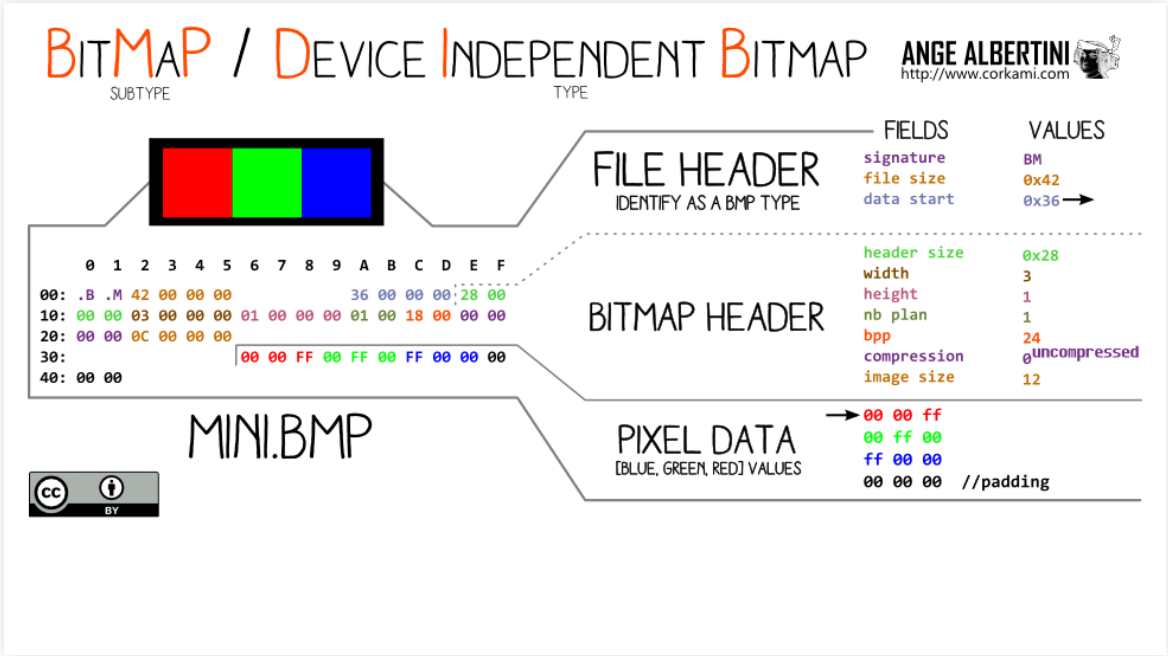
This structural information is called “header”, because it’s usually found on top of a binary file, since it’s the very first thing that needs to be read in order to make sense of the bytes that are coming.

# Header? Payload?

*“header refers to supplemental data placed at the beginning of a block of data being stored or transmitted. In data transmission, the data following the header is sometimes called the payload or body.”*

– *Wikipedia: Header (computing)*

# Examples


BITMAP / DEVICE INDEPENDENT BITMAP
SUBTYPE
TYPE
ANGE ALBERTINI
http://www.corkami.com
FIELDS
VALUES
FILE HEADER
IDENTIFY AS A BMP TYPE
signature
BM
file size
0x42
data start
0x36
0 1 2 3 4 5 6 7 8 9 A B C D E F
00: .B .M 42 00 00 00 36 00 00 00 28 00
10: 00 00 03 00 00 00 01 00 00 00 01 00 18 00 00 00
20: 00 00 0C 00 00 00
30: 00 00 FF 00 FF 00 FF 00 00 00
40: 00 00
BITMAP HEADER
header size
0x28
width
3
height
1
nb plan
1
bpp
24
compression
0 uncompressed
image size
12
MINI.BMP
PIXEL DATA
[BLUE, GREEN, RED] VALUES
00 00 ff
00 ff 00
ff 00 00
00 00 00 //padding

# GRAPHICS INTERCHANGE FORMAT

```
     0 1 2 3 4 5 6 7 8 9 A B C D E F
00: .G .I .F .8 .9 .a 03 00 01 00 A1 00 00 FF 00 00
10: 00 FF 00 00 00 FF FF FF FF 2C 00 00 00 00 03 00
20: 01 00 00 02 02 44 54 00 3B
```

| | FIELDS | VALUES |
|---|---|---|
| HEADER | signature | "GIF" |
| | version | "89a" |
| LOCAL SCREEN DESCRIPTOR | width | 3 |
| | height | 1 |
| | flags | A1 (01 010 0 001) |
| | GCT | true |
| | bpp | 2+1 |
| | GCT size | 2^(1+1) |
| | Global Color Table | FF 00 00 00 FF 00 00 00 FF FF FF FF |
| IMAGE DESCRIPTOR | separator | 2C |
| | width height | 3 1 |
| | minimum bits per LZW code | 2 |
| | block size | 2 |
| | block data | 0101 010 001 000 100 |
| | | end #2 #1 #0 start |
| | block end | 0 |
| TRAILER | trailer | 3B |



ANGE ALBERTINI
http://pics.corkami.com

THE GIF WAS CREATED BY COMPUSERVE IN 1987.
IT'S PALETTE BASED: EACH BLOCK IS LIMITED TO 256 COLORS.
IT USES THE LEMPEL-ZIV-WELCH ALGORITHM, WHICH WAS PATENTED UNTIL 2004.

# JPEG FILE INTERCHANGE FORMAT

JOINT PHOTOGRAPHIC EXPERT GROUP

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000: FF D8 FF E0 00 10 .J .F .I .F 00 01 01 01 00 48
010: 00 48 00 00 FF DB 00 43 00 01 01 01 01 01 01 01
020: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
030: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
040: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
050: 01 01 01 01 01 01 01 01 01 FF DB 00 43 01 01 01
060: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
070: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
080: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
090: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 FF C0
0A0: 00 11 08 00 02 00 06 03 01 22 00 02 11 01 03 11
0B0: 01 FF C4 00 15 00 01 01 00 00 00 00 00 00 00 00
0C0: 00 00 00 00 00 00 00 09 FF C4 00 19 10 01 00 02
0D0: 03 00 00 00 00 00 00 00 00 00 00 00 00 00 06 08
0E0: 38 88 B6 FF C4 00 15 01 01 01 00 00 00 00 00 00
0F0: 00 00 00 00 00 00 00 00 07 0A FF C4 00 1C 11 00
100: 01 03 05 00 00 00 00 00 00 00 00 00 00 00 00 08
110: 00 07 B8 09 38 39 76 78 FF DA 00 0C 03 01 00 02
120: 11 03 11 00 3F 00 86 F7 E7 1D A9 16 CA 77 30 D0
130: 14 F7 41 DC 5A 8E FB 31 19 26 5D C4 2A F4 5C 81
140: 7B DB 06 84 A0 75 17 FF D9
```

| SEGMENTS | FIELDS | VALUES |
|---|---|---|
| START OF IMAGE | marker | FFD8 |
| APPLICATION0 (DEFAULT HEADER) | marker/length | FFE0/16 |
| | identifier | JFIF\0 |
| | version | 1.1 |
| | units | 1 (dpi) |
| | density | 72X72 |
| | thumbnail | 0x0 |
| DEFINE QUANTIZATION TABLE | marker/length | FFDB/67 |
| | destination | 0 (luminance) |
| | table (8x8) | {1} (100% quality) |
| DEFINE QUANTIZATION TABLE | marker/length | FFDB/67 |
| | destination | 1 (chrominance) |
| | table (8x8) | {1} (100% quality) |
| START OF FRAME | marker/length | FFC0/17 |
| | precision | 8 |
| | line Nb | 2 |
| | samples/line | 6 |
| | components | 3 |
| | Id factor table | 1 1x1 0 (LumY) |
| | Id factor table | 2 2x2 1 (ChromCb) |
| | Id factor table | 3 2x2 1 (ChromCr) |
| DEFINE HUFFMAN TABLE | marker/length | FFC4/21 |
| | class | 0 (DC) |
| | destination | 0 |
| | 1 code of 1 bit | 00 |
| | 1 code of 2 bits | 09 |
| DEFINE HUFFMAN TABLE | marker/length | FFC4/25 |
| | class | 0 (DC) |
| | destination | 0 |
| | 1 code of 1 bit | 00 |
| | 2 code of 3 bits | 06 08 |
| | 3 code of 4 bits | 38 88 B6 |
| DEFINE HUFFMAN TABLE | marker/length | FFC4/21 |
| | class | 0 (DC) |
| | destination | 1 |
| | 1 code of 1 bit | 07 |
| | 1 code of 2 bits | 0A |
| DEFINE HUFFMAN TABLE | marker/length | FFC4/28 |
| | class | 1 (AC) |
| | destination | 1 |
| | 1 code of 2 bits | 08 |
| | 3 code of 3 bits | 00 07 B8 |
| | 5 code of 4 bits | 09 38 39 76 78 |
| START OF SCAN | marker/length | FFDA/12 |
| | components | 3 |
| | selector / DC, AC table | 1 / 0, 0 |
| | | 2 / 1, 1 |
| | | 3 / 1, 1 |
| | spectral select. | 0..63 |
| | successive approx. | 00 |
| IMAGE DATA ENTROPY-CODED SEGMENT | | 86F7E71DA916CA7730D014 F741DC5A8EFB3119265DC4 2AF45C817BDB0684A07517 |
| END OF IMAGE | marker | FFD9 |



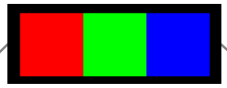
ANGE ALBERTINI
http://pics.corkami.com

JPEG IS THE ENCODING STANDARD, JFIF IS THE FILE FORMAT

# PORTABLE NETWORK GRAPHICS

ANGE ALBERTINI
http://www.corkami.com





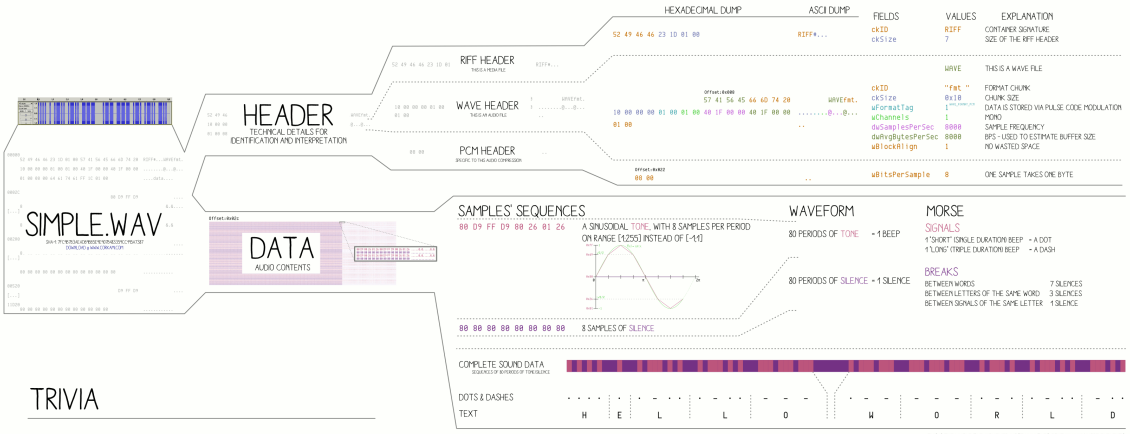
```
     0 1 2 3 4 5 6 7 8 9 A B C D E F
00: 89 .P .N .G 0D 0A 1A 0A 00 00 00 0D .I .H .D .R
10: 00 00 00 03 00 00 00 01 08 02 00 00 00 94 82 83
20: E3 00 00 00 15 .I .D .A .T 08 1D 01 0A 00 F5 FF
30: 00 FF 00 00 00 FF 00 00 00 FF 0E FB 02 FE E9 32
40: 61 E5 00 00 00 00 .I .E .N .D AE 42 60 82
```

| | FIELDS | VALUES |
|---|---|---|
| SIGNATURE | signature | \x89 PNG \r\n \x1a \n |
| HEADER | size | 0x0000000D |
| | id | IHDR |
| | width | 0x00000003 |
| | height | 0x00000001 |
| | bpp | 0x08 |
| | color | 0x02 RGB |
| | compression | 0x00 DEFLATE |
| | filter | 0x00 |
| | interlace | 0x00 |
| | CRC32 | 0x948283E3 |
| DATA | size | 0x00000015 |
| | id | IDAT |
| ZLIB | window size | 0b00001000 |
| | method | 0b00001000 DEFLATE |
| | level / dict. | 0b00011101 |
| | checksum | 0x081D % 31 = 0 |
| DEFLATE | last block | 0b00000001 FINAL |
| | block type | 0b00000001 RAW |
| | data length | 0x000A |
| | !length | 0xFFF5 |
| PIXELS | line filter | 0x00 NONE |
| | FF 00 00 00 FF 00 00 00 FF | |
| | adler32 | 0x0EFB02FE |
| | CRC32 | 0xE93261E5 |
| END | size | 0x00000000 |
| | id | IEND |
| | CRC32 | 0xAE426082 |

# WAV 101 an audio file walk-through

ANGE ALBERTINI
CORKAMI.COM

## DISSECTED FILE

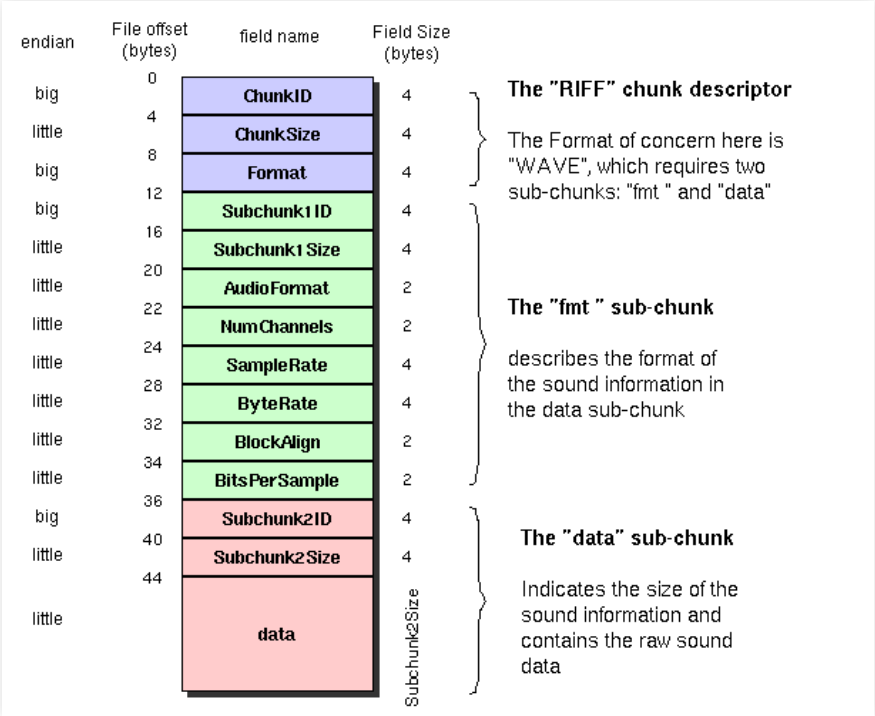



## TRIVIA

WAV IS A SUBFORMAT OF R.....I.....FF..., A GENERIC CONTAINER THAT CAN ALSO CONTAIN AVI (VIDEOS), ANI (CURSORS)...

RIFF WAS CREATED IN 1991 BY MICROSOFT & I.B.M.
AND IS BASED ON I.....FF...,
CREATED BY E.A. IN 1985 FOR THE COMMODORE AMIGA

VERSION 1.00
2014/01/08




# Comments?

# Questions?