Viewing and Interpreting Binary Data

Peter Bubestinger-Steindl
(`pb @ ArkThis.com`)

Speaker notes

No notes on this slide.

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 ...
```

Speaker notes

No notes on this slide.

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

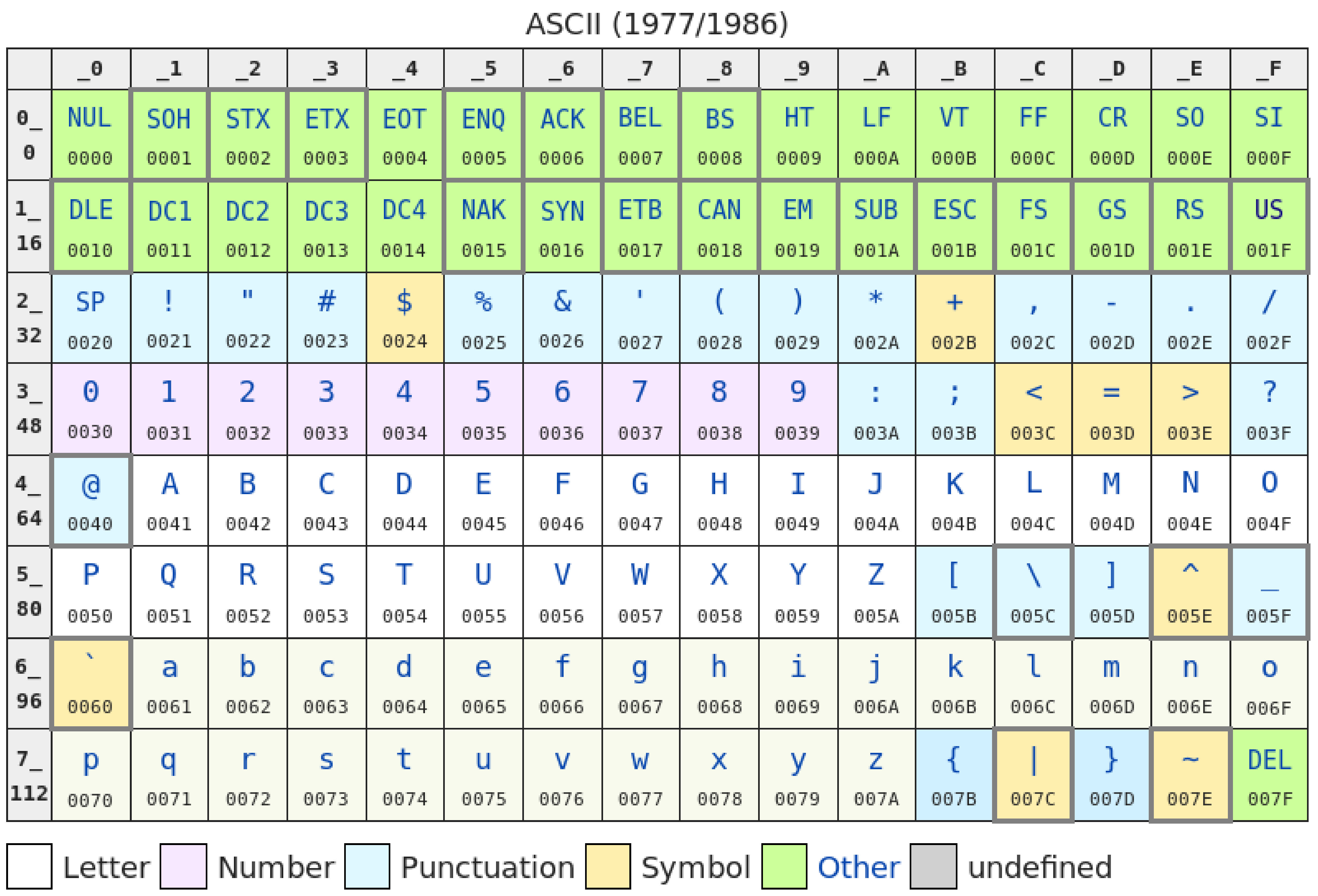
Speaker notes

Short: It makes (bit-)patterns in a Byte easier to visually read/understand/see.

Sounds complicated, but it’s easier than you think and a matter of practice.

What’s important for you is that you’ve heard about this way of viewing/editing data.

Character encoding



ASCII (1977/1986)

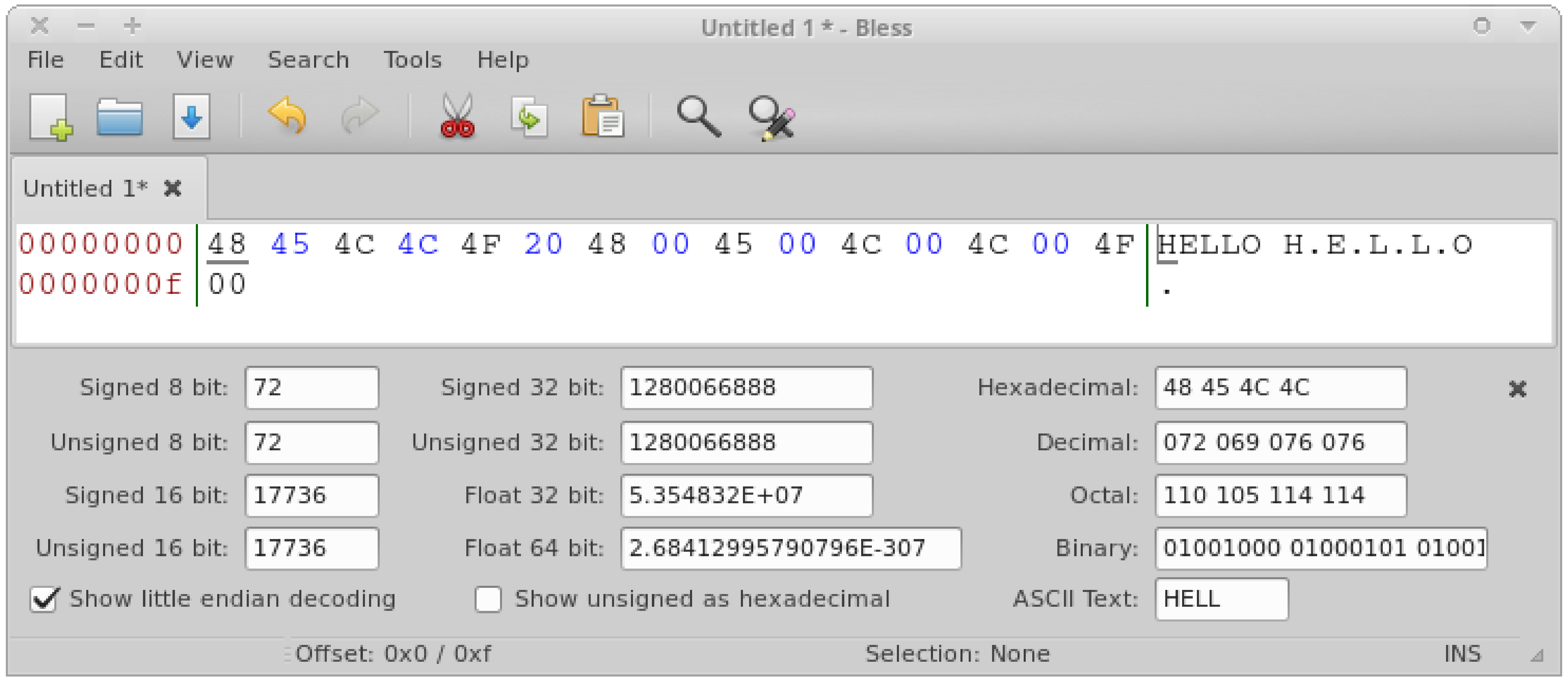
	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_ 0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
1_ 16	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	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?



```
00000000 48 45 4C 4C 4F 20 48 00 45 00 4C 00 4C 00 4F HELLO H.E.L.L.O
0000000f 00                                            .
```

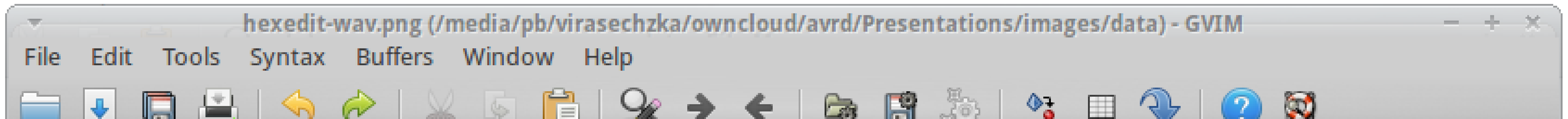
Signed 8 bit:	72	Signed 32 bit:	1280066888	Hexadecimal:	48 45 4C 4C
Unsigned 8 bit:	72	Unsigned 32 bit:	1280066888	Decimal:	072 069 076 076
Signed 16 bit:	17736	Float 32 bit:	5.354832E+07	Octal:	110 105 114 114
Unsigned 16 bit:	17736	Float 64 bit:	2.68412995790796E-307	Binary:	01001000 01000101 01001
[✓] Show little endian decoding		[ ] Show unsigned as hexadecimal		ASCII Text:	HELL

Offset: 0x0 / 0xf Selection: None INS

Speaker notes

No notes on this slide.

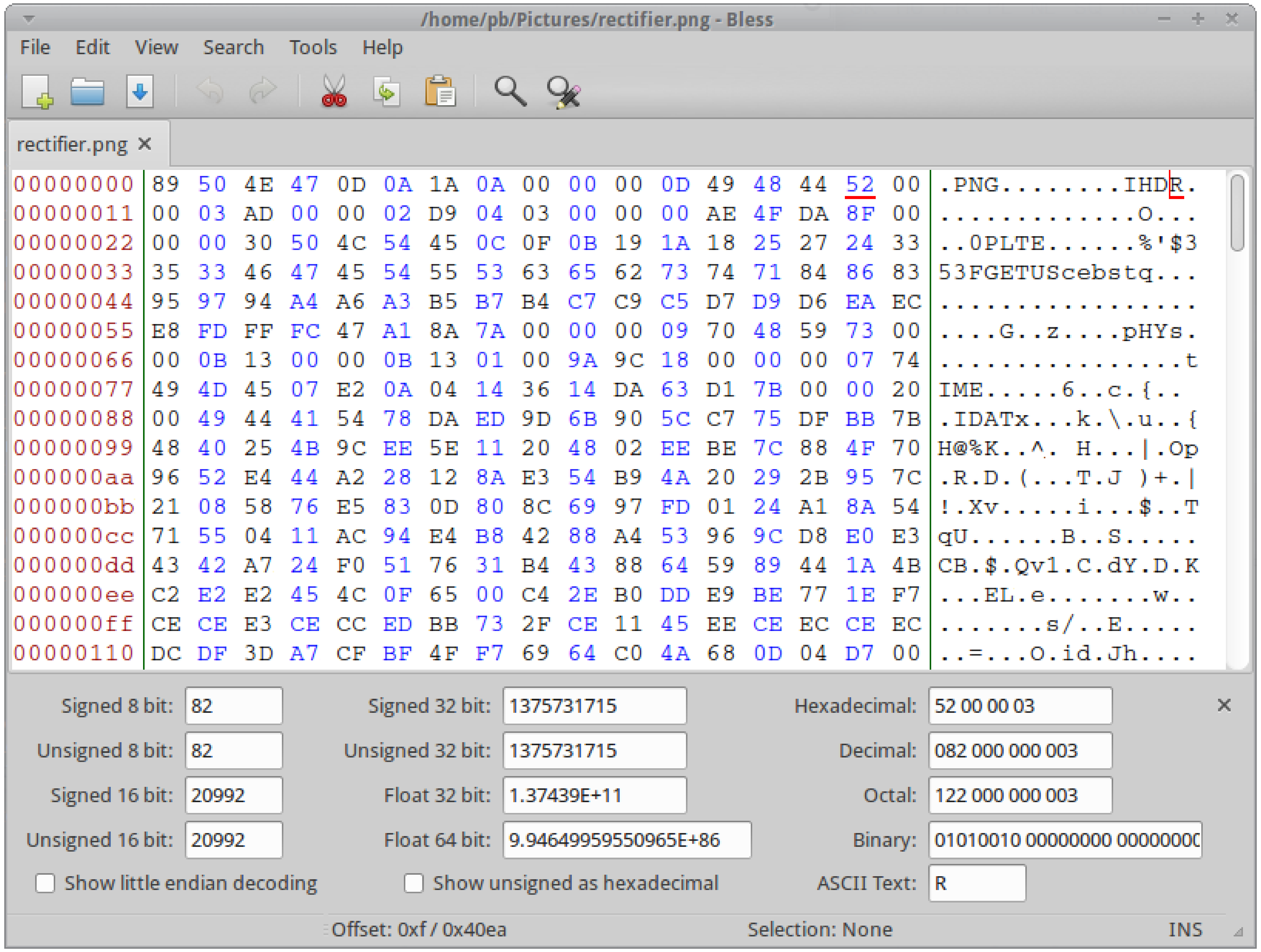
Data as Text?


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

Speaker notes

No notes on this slide.

Hex editing!



/home/pb/Pictures/rectifier.png - Bless
File Edit View Search Tools Help
rectifier.png

```
00000000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 .PNG........IHDR.
00000011 00 03 AD 00 00 02 D9 04 03 00 00 00 AE 4F DA 8F 00 .............O...
00000022 00 00 30 50 4C 54 45 0C 0F 0B 19 1A 18 25 27 24 33 ..0PLTE......%'$3
00000033 35 33 46 47 45 54 55 53 63 65 62 73 74 71 84 86 83 53FGETUScebstq...
00000044 95 97 94 A4 A6 A3 B5 B7 B4 C7 C9 C5 D7 D9 D6 EA EC .................
00000055 E8 FD FF FC 47 A1 8A 7A 00 00 00 09 70 48 59 73 00 ....G..z....pHYs.
00000066 00 0B 13 00 00 0B 13 01 00 9A 9C 18 00 00 00 07 74 ................t
00000077 49 4D 45 07 E2 0A 04 14 36 14 DA 63 D1 7B 00 00 20 IME.....6..c.{..
00000088 00 49 44 41 54 78 DA ED 9D 6B 90 5C C7 75 DF BB 7B .IDATx...k.\.u..{
00000099 48 40 25 4B 9C EE 5E 11 20 48 02 EE BE 7C 88 4F 70 H@%K..^. H...|.Op
000000aa 96 52 E4 44 A2 28 12 8A E3 54 B9 4A 20 29 2B 95 7C .R.D.(...T.J )+.|
000000bb 21 08 58 76 E5 83 0D 80 8C 69 97 FD 01 24 A1 8A 54 !.Xv.....i...$..T
000000cc 71 55 04 11 AC 94 E4 B8 42 88 A4 53 96 9C D8 E0 E3 qU......B..S.....
000000dd 43 42 A7 24 F0 51 76 31 B4 43 88 64 59 89 44 1A 4B CB.$.Qv1.C.dY.D.K
000000ee C2 E2 E2 45 4C 0F 65 00 C4 2E B0 DD E9 BE 77 1E F7 ...EL.e.......w..
000000ff CE CE E3 CE CC ED BB 73 2F CE 11 45 EE CE EC CE EC .......s/..E.....
00000110 DC DF 3D A7 CF BF 4F F7 69 64 C0 4A 68 0D 04 D7 00 ..=...O.id.Jh....
```

Signed 8 bit:	82	Signed 32 bit:	1375731715	Hexadecimal:	52 00 00 03
Unsigned 8 bit:	82	Unsigned 32 bit:	1375731715	Decimal:	082 000 000 003
Signed 16 bit:	20992	Float 32 bit:	1.37439E+11	Octal:	122 000 000 003
Unsigned 16 bit:	20992	Float 64 bit:	9.94649959550965E+86	Binary:	01010010 00000000 0000000
☐ Show little endian decoding		☐ Show unsigned as hexadecimal		ASCII Text:	R

Offset: 0xf / 0x40ea   Selection: None   INS

“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

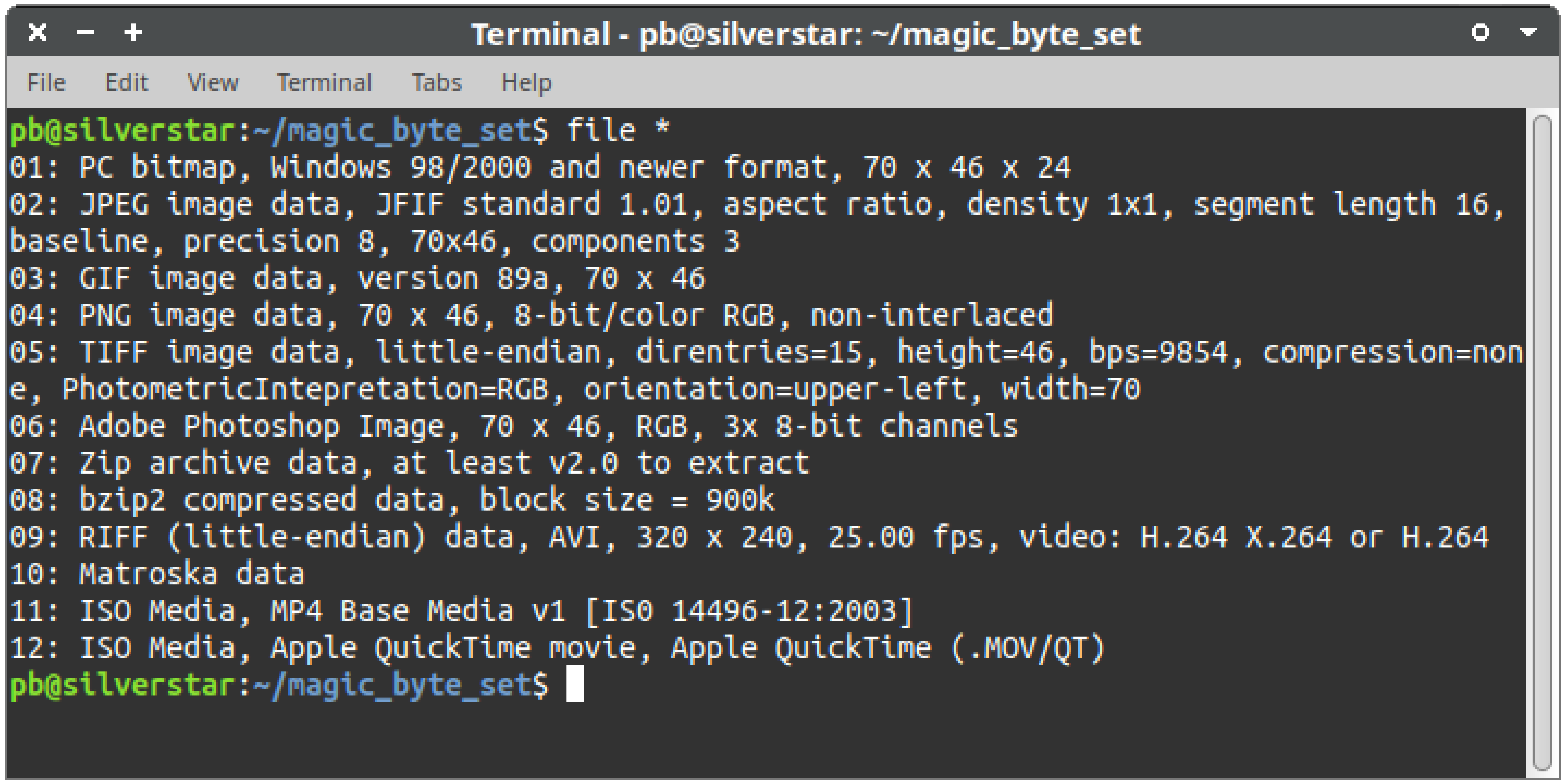
Identify the file types in the given set, using a Hexeditor and the "Magic Byte" list on Wikipedia.

See: List of File Signatures (Wikipedia)

Speaker notes

No notes on this slide.

Unix “file” command


Terminal - pb@silverstar: ~/magic_byte_set
File Edit View Terminal Tabs Help

```
pb@silverstar:~/magic_byte_set$ file *
01: PC bitmap, Windows 98/2000 and newer format, 70 x 46 x 24
02: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16,
baseline, precision 8, 70x46, components 3
03: GIF image data, version 89a, 70 x 46
04: PNG image data, 70 x 46, 8-bit/color RGB, non-interlaced
05: TIFF image data, little-endian, direntries=15, height=46, bps=9854, compression=non
e, PhotometricIntepretation=RGB, orientation=upper-left, width=70
06: Adobe Photoshop Image, 70 x 46, RGB, 3x 8-bit channels
07: Zip archive data, at least v2.0 to extract
08: bzip2 compressed data, block size = 900k
09: RIFF (little-endian) data, AVI, 320 x 240, 25.00 fps, video: H.264 X.264 or H.264
10: Matroska data
11: ISO Media, MP4 Base Media v1 [IS0 14496-12:2003]
12: ISO Media, Apple QuickTime movie, Apple QuickTime (.MOV/QT)
pb@silverstar:~/magic_byte_set$
```

See Wikipedia: File (command)

Speaker notes

No notes on this slide.

Speaker notes

Exercise / Puzzle

The file “whatami”:

- Try to identify what it is.
- Try to find out what is wrong with it.

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*

Speaker notes

No notes on this slide.

MIME Type Examples

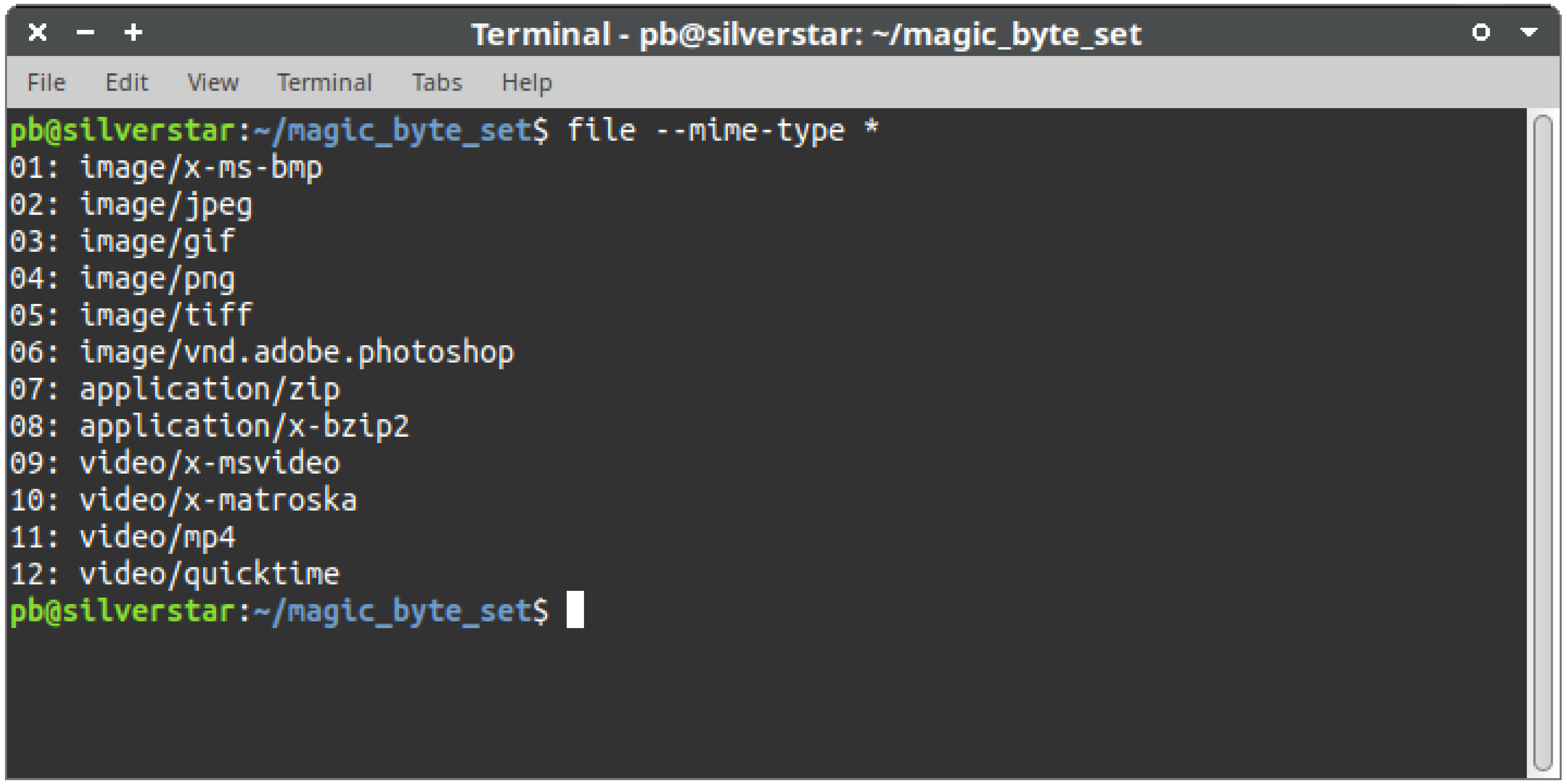
- application/zip
- application/pdf
- text/html
- text/xml
- text/csv
- text/plain
- image/png
- image/jpeg
- image/gif
- audio/aac
- audio/mpeg
- video/DV
- video/H264
- video/mp4

Complete List (IANA), 2019-10-16

Speaker notes

No notes on this slide.

Remember our "no suffix" file set?

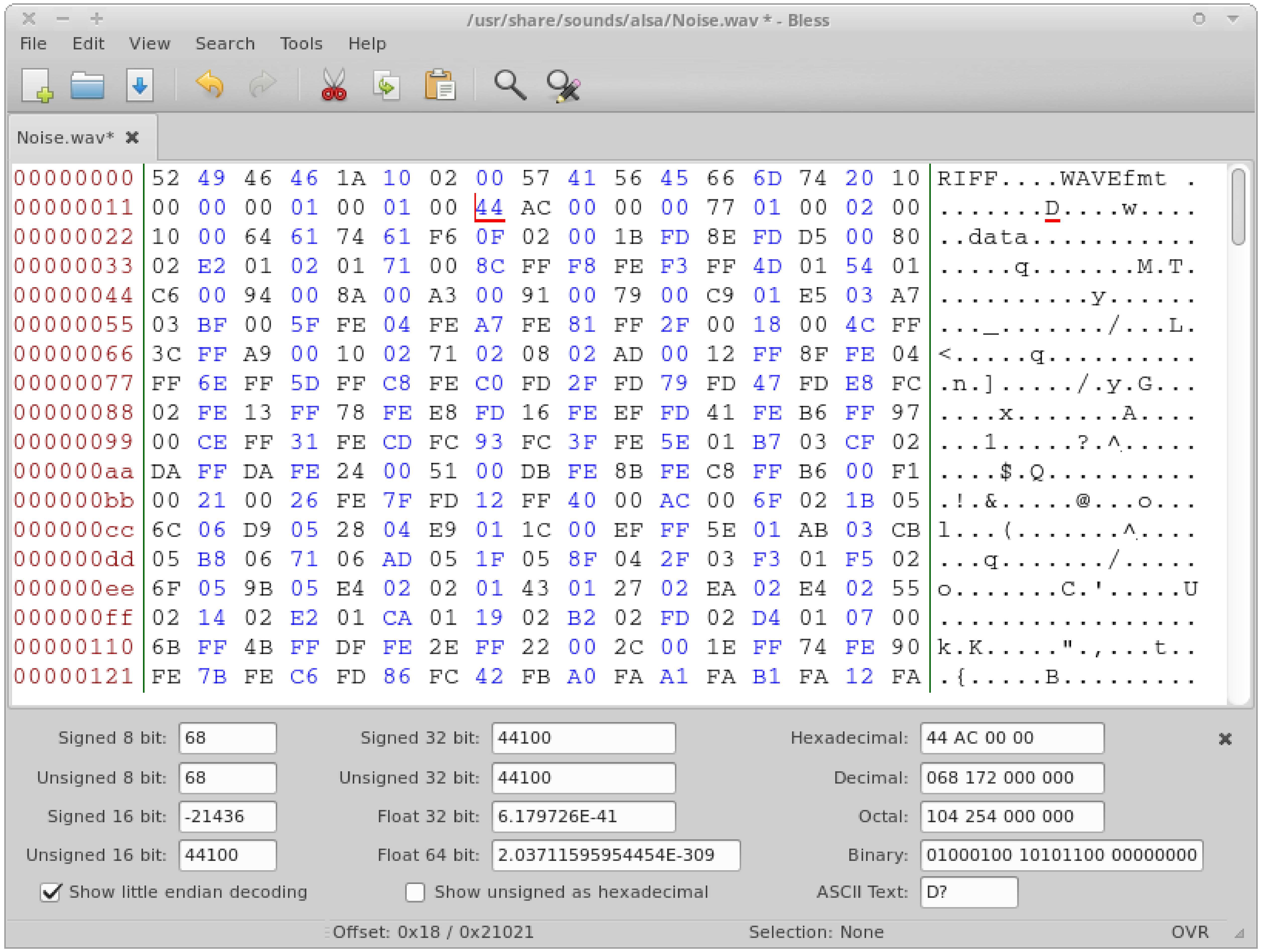

Terminal - pb@silverstar: ~/magic_byte_set
File Edit View Terminal Tabs Help

```
pb@silverstar:~/magic_byte_set$ file --mime-type *
01: image/x-ms-bmp
02: image/jpeg
03: image/gif
04: image/png
05: image/tiff
06: image/vnd.adobe.photoshop
07: application/zip
08: application/x-bzip2
09: video/x-msvideo
10: video/x-matroska
11: video/mp4
12: video/quicktime
pb@silverstar:~/magic_byte_set$
```

Speaker notes

No notes on this slide.

Binary Data?



```
00000000 52 49 46 46 1A 10 02 00 57 41 56 45 66 6D 74 20 10 RIFF....WAVEfmt .
00000011 00 00 00 01 00 01 00 44 AC 00 00 00 77 01 00 02 00 .......D....w....
00000022 10 00 64 61 74 61 F6 0F 02 00 1B FD 8E FD D5 00 80 ..data...........
00000033 02 E2 01 02 01 71 00 8C FF F8 FE F3 FF 4D 01 54 01 .....q.......M.T.
00000044 C6 00 94 00 8A 00 A3 00 91 00 79 00 C9 01 E5 03 A7 ..........y......
00000055 03 BF 00 5F FE 04 FE A7 FE 81 FF 2F 00 18 00 4C FF ..._......./...L.
00000066 3C FF A9 00 10 02 71 02 08 02 AD 00 12 FF 8F FE 04 <.....q..........
00000077 FF 6E FF 5D FF C8 FE C0 FD 2F FD 79 FD 47 FD E8 FC .n.]...../.y.G...
00000088 02 FE 13 FF 78 FE E8 FD 16 FE EF FD 41 FE B6 FF 97 ....x.......A....
00000099 00 CE FF 31 FE CD FC 93 FC 3F FE 5E 01 B7 03 CF 02 ...1.....?.^.....
000000aa DA FF DA FE 24 00 51 00 DB FE 8B FE C8 FF B6 00 F1 ....$.Q..........
000000bb 00 21 00 26 FE 7F FD 12 FF 40 00 AC 00 6F 02 1B 05 .!.&.....@...o...
000000cc 6C 06 D9 05 28 04 E9 01 1C 00 EF FF 5E 01 AB 03 CB l...(.......^....
000000dd 05 B8 06 71 06 AD 05 1F 05 8F 04 2F 03 F3 01 F5 02 ...q......./.....
000000ee 6F 05 9B 05 E4 02 02 01 43 01 27 02 EA 02 E4 02 55 o.......C.'.....U
000000ff 02 14 02 E2 01 CA 01 19 02 B2 02 FD 02 D4 01 07 00 .................
00000110 6B FF 4B FF DF FE 2E FF 22 00 2C 00 1E FF 74 FE 90 k.K.....".,...t..
00000121 FE 7B FE C6 FD 86 FC 42 FB A0 FA A1 FA B1 FA 12 FA .{.....B.........
```

```
  Signed 8 bit: 68          Signed 32 bit: 44100                   Hexadecimal: 44 AC 00 00
Unsigned 8 bit: 68        Unsigned 32 bit: 44100                       Decimal: 068 172 000 000
 Signed 16 bit: -21436      Float 32 bit: 6.179726E-41                   Octal: 104 254 000 000
Unsigned 16 bit: 44100      Float 64 bit: 2.03711595954454E-309         Binary: 01000100 10101100 00000000
[✓] Show little endian decoding   [ ] Show unsigned as hexadecimal   ASCII Text: D?
                     Offset: 0x18 / 0x21021          Selection: None          OVR
```

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)*

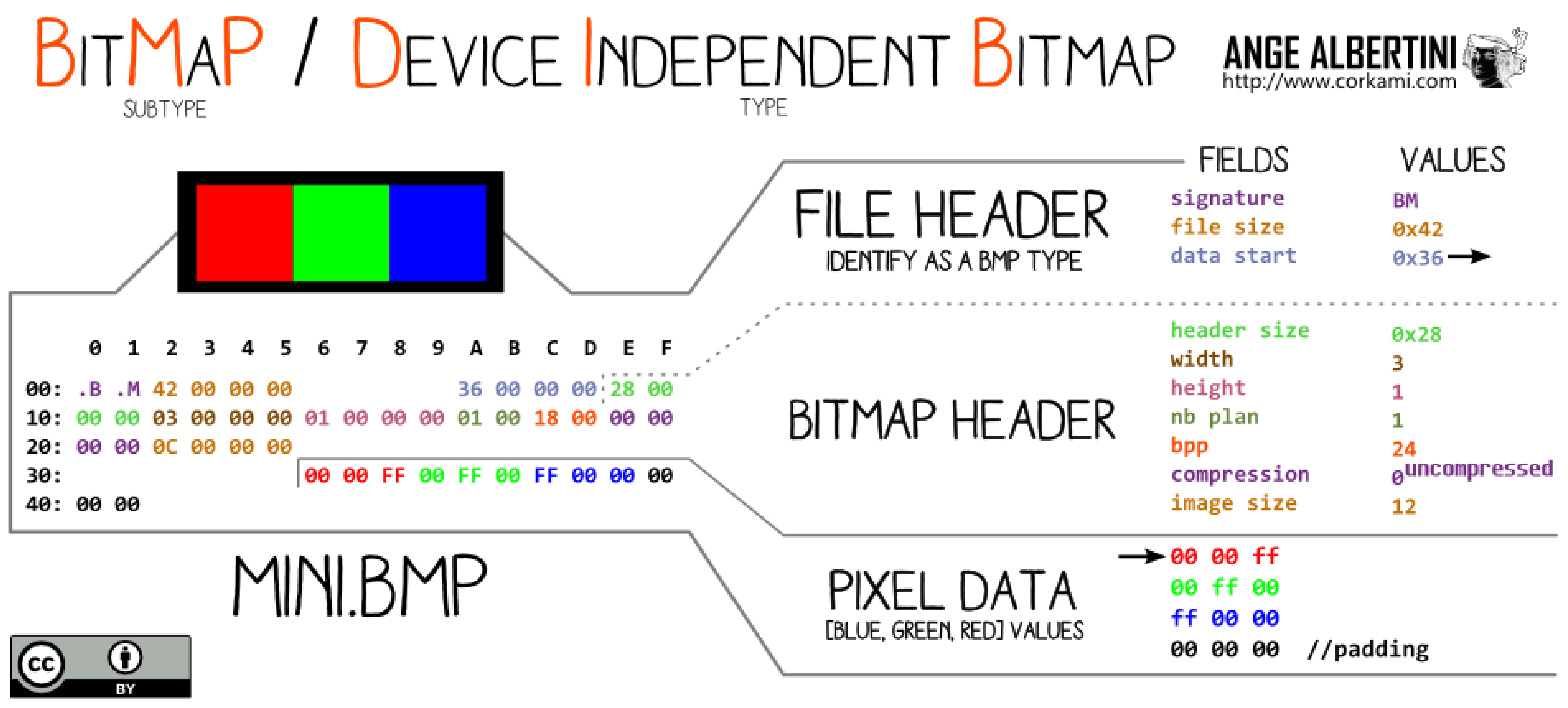
Speaker notes

No notes on this slide.

Examples

Speaker notes

No notes on this slide.



```
     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
```

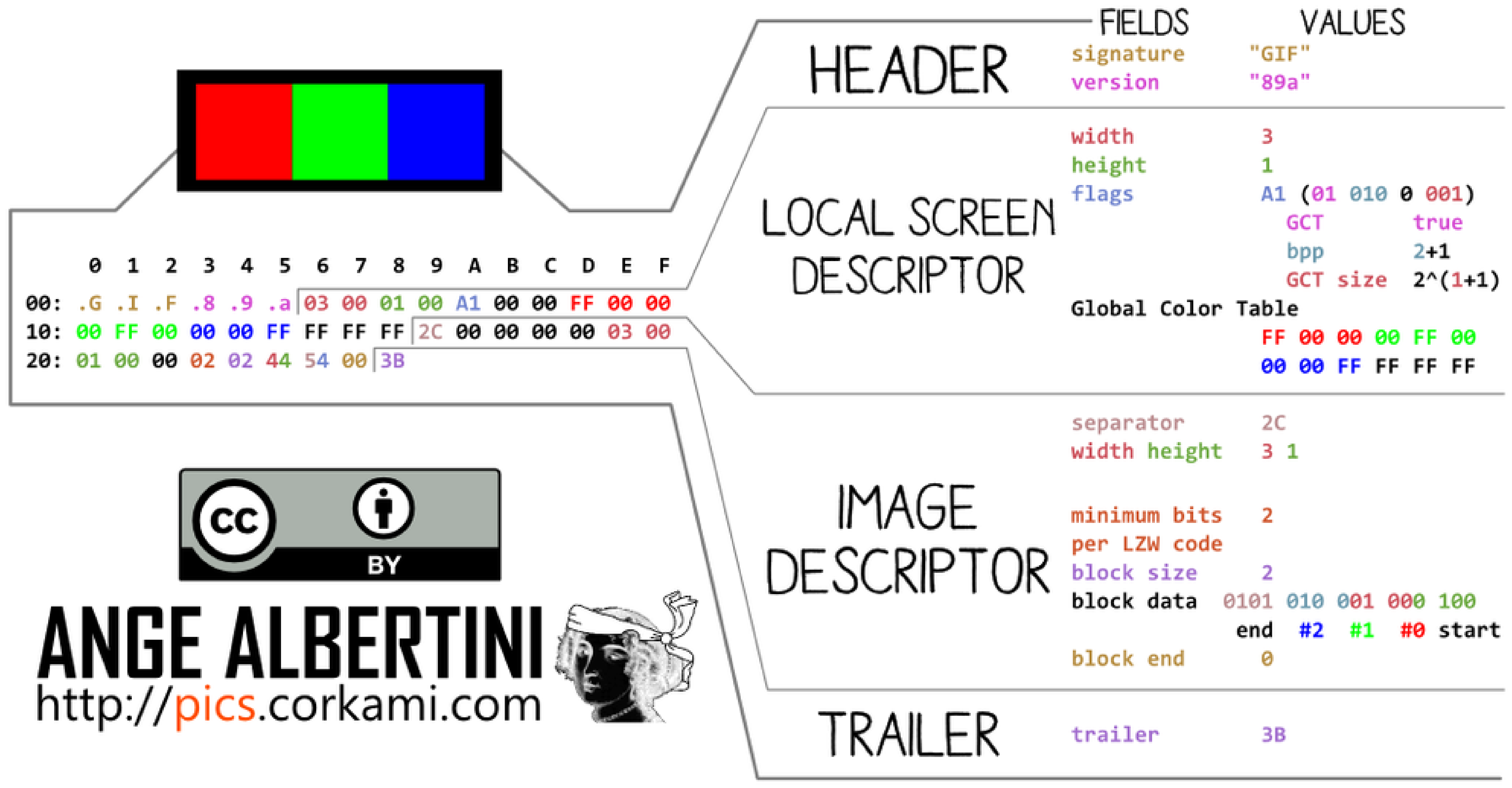
FIELDS	VALUES
signature	BM
file size	0x42
data start	0x36
header size	0x28
width	3
height	1
nb plan	1
bpp	24
compression	0 uncompressed
image size	12

```
00 00 ff
00 ff 00
ff 00 00
00 00 00  //padding
```

Speaker notes

No notes on this slide.

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
```

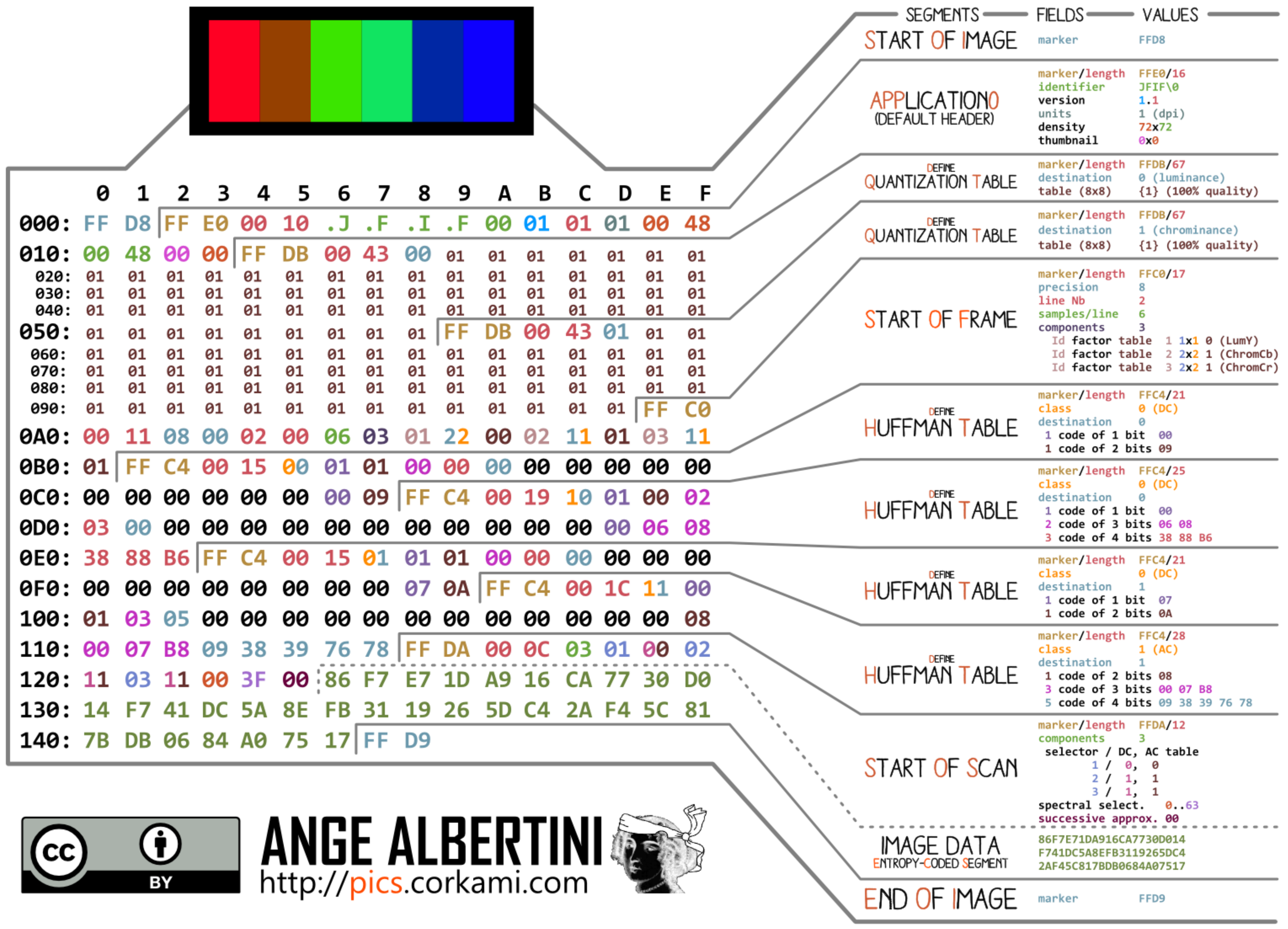
```
                     FIELDS          VALUES
HEADER               signature       "GIF"
                     version         "89a"

LOCAL SCREEN         width           3
DESCRIPTOR           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                separator       2C
DESCRIPTOR           width height    3 1

                     minimum bits    2
                     per LZW code
                     block size      2
                     block data    0101 010 001 000 100
                                    end  #2  #1  #0  start
                     block end       0

TRAILER              trailer         3B
```

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.

Speaker notes

No notes on this slide.

JOINT PHOTOGRAPHIC EXPERT GROUP FILE INTERCHANGE FORMAT



```
      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

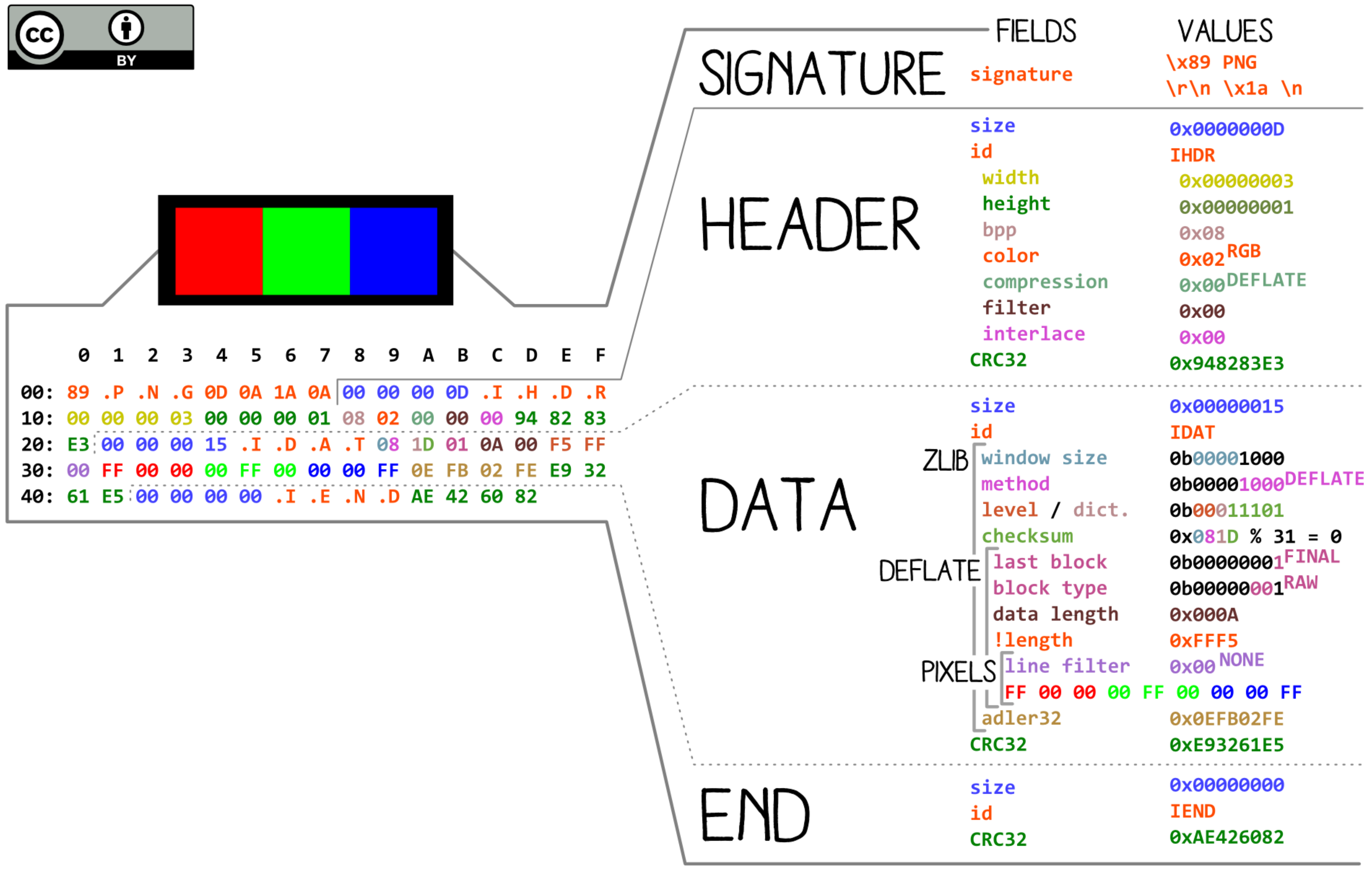
JPEG IS THE ENCODING STANDARD, JFIF IS THE FILE FORMAT

Speaker notes

No notes on this slide.

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

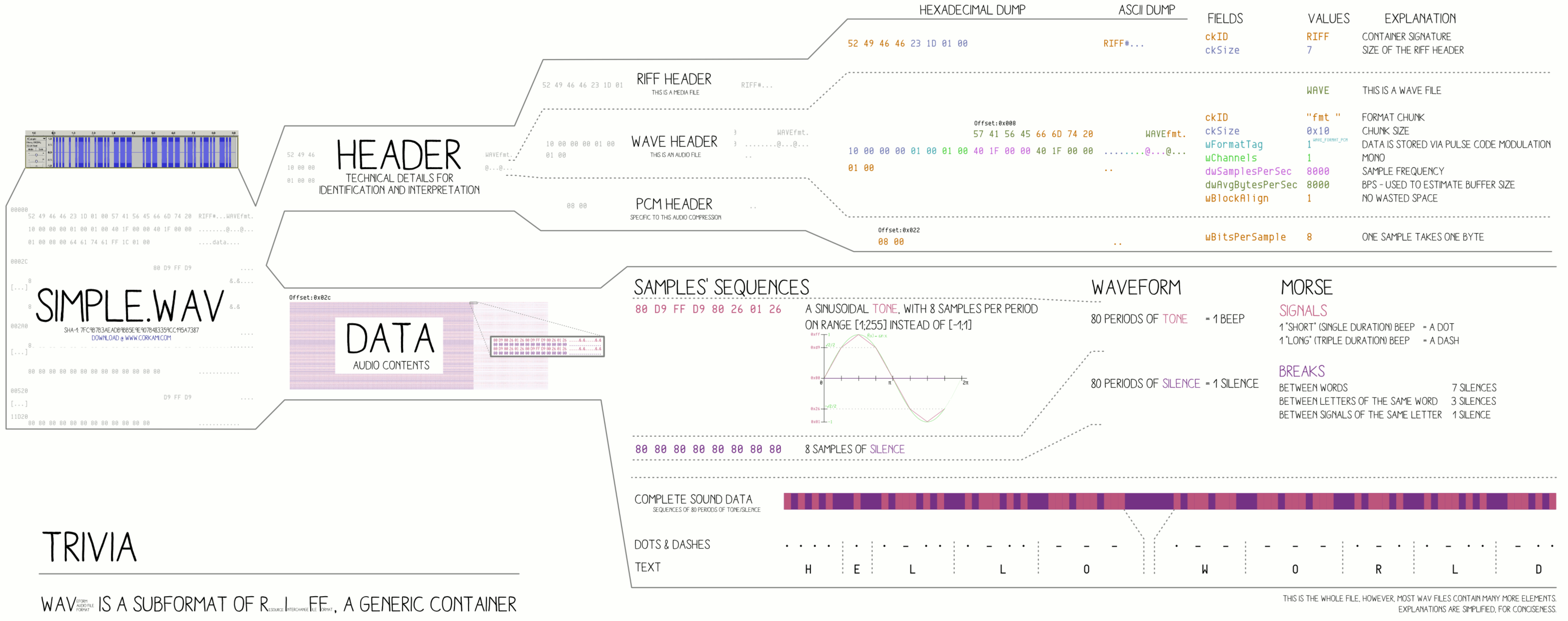
Speaker notes

No notes on this slide.

WAV101 an audio file walkthrough

ANGE ALBERTINI
CORKAMI.COM

DISSECTED FILE



```
00000
  52 49 46 46 23 1D 01 00 57 41 56 45 66 6D 74 20  RIFF#...WAVEfmt.
  10 00 00 00 01 00 01 00 40 1F 00 00 40 1F 00 00  ........@...@...
  01 00 08 00 64 61 74 61 FF 1C 01 00              ....data....
0002C
                                      80 D9 FF D9  ....
[...]
002A0
[...]
  80 80 80 80 80 80 80 80 80 80 80 80 80           .............
00520
                                         D9 FF D9  ....
[...]
11D20
  80 80 80 80 80 80 80 80 80 80 80 80              ............
```

HEXADECIMAL DUMP	ASCII DUMP	FIELDS	VALUES	EXPLANATION
52 49 46 46 23 1D 01 00	RIFF#...	ckID	RIFF	CONTAINER SIGNATURE
		ckSize	7	SIZE OF THE RIFF HEADER
57 41 56 45 66 6D 74 20	WAVEfmt.		WAVE	THIS IS A WAVE FILE
		ckID	"fmt "	FORMAT CHUNK
10 00 00 00 01 00 01 00 40 1F 00 00 40 1F 00 00	........@...@...	ckSize	0x10	CHUNK SIZE
		wFormatTag	1 WAVE_FORMAT_PCM	DATA IS STORED VIA PULSE CODE MODULATION
		wChannels	1	MONO
		dwSamplesPerSec	8000	SAMPLE FREQUENCY
		dwAvgBytesPerSec	8000	BPS - USED TO ESTIMATE BUFFER SIZE
01 00	..	wBlockAlign	1	NO WASTED SPACE
08 00	..	wBitsPerSample	8	ONE SAMPLE TAKES ONE BYTE

SAMPLES' SEQUENCES

80 D9 FF D9 80 26 01 26 — A SINUSOIDAL TONE, WITH 8 SAMPLES PER PERIOD ON RANGE [1;255] INSTEAD OF [-1;1]

80 80 80 80 80 80 80 80 — 8 SAMPLES OF SILENCE

WAVEFORM

80 PERIODS OF TONE = 1 BEEP

80 PERIODS OF SILENCE = 1 SILENCE

MORSE

SIGNALS

1 "SHORT" (SINGLE DURATION) BEEP = A DOT
1 "LONG" (TRIPLE DURATION) BEEP = A DASH

BREAKS

BETWEEN WORDS	7 SILENCES
BETWEEN LETTERS OF THE SAME WORD	3 SILENCES
BETWEEN SIGNALS OF THE SAME LETTER	1 SILENCE

THIS IS THE WHOLE FILE, HOWEVER, MOST WAV FILES CONTAIN MANY MORE ELEMENTS.
EXPLANATIONS ARE SIMPLIFIED, FOR CONCISENESS.

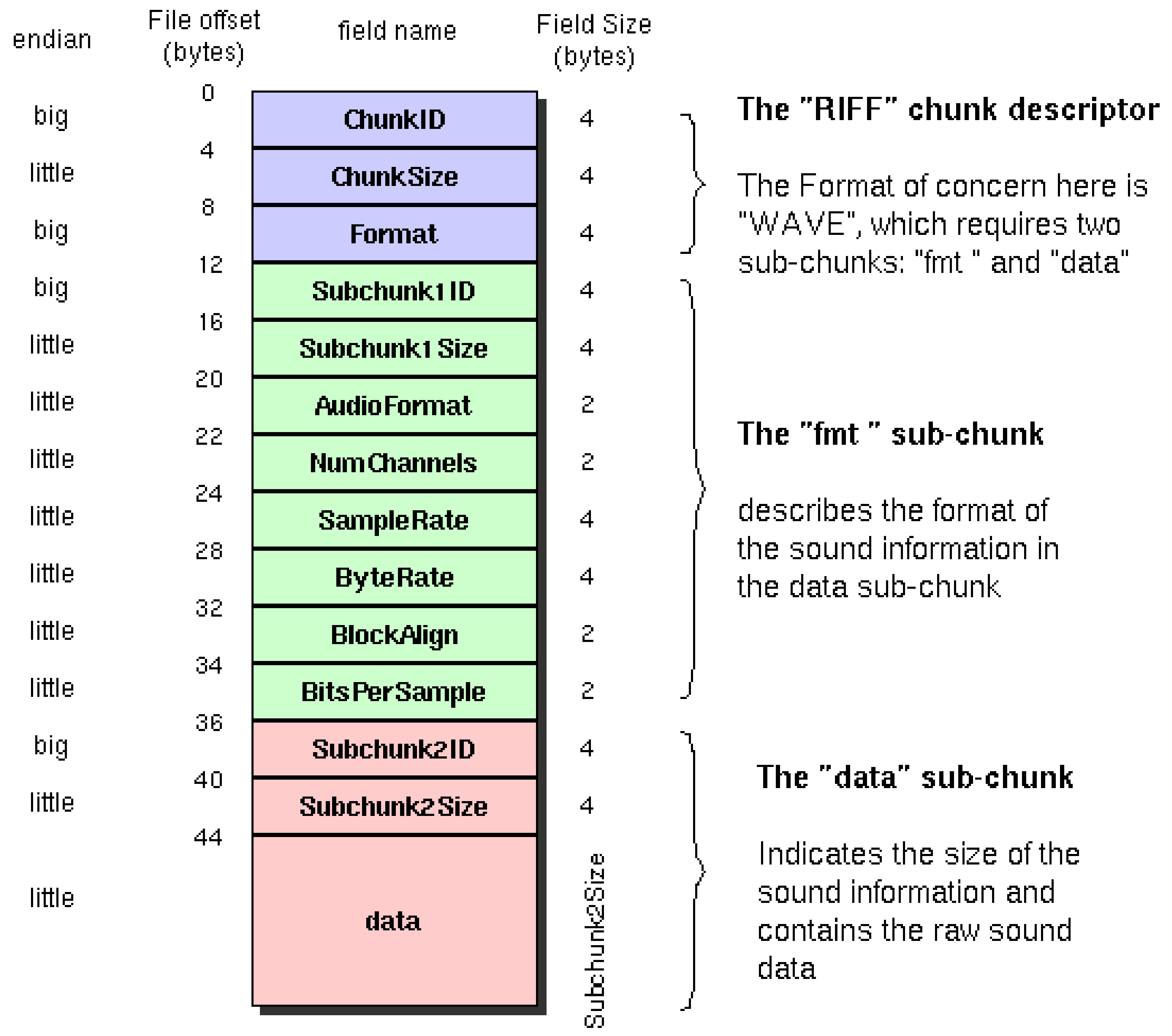
TRIVIA

WAV (EFORM AUDIO FILE FORMAT) IS A SUBFORMAT OF R(ESOURCE) I(NTERCHANGE) F(ILE) F(ORMAT), 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(NTERCHANGE) F(ILE) F(ORMAT),
CREATED BY E.A. IN 1985 FOR THE COMMODORE AMIGA

Speaker notes

No notes on this slide.



endian	File offset (bytes)	field name	Field Size (bytes)
big	0	ChunkID	4
little	4	ChunkSize	4
big	8	Format	4
big	12	Subchunk1ID	4
little	16	Subchunk1Size	4
little	20	AudioFormat	2
little	22	NumChannels	2
little	24	SampleRate	4
little	28	ByteRate	4
little	32	BlockAlign	2
little	34	BitsPerSample	2
big	36	Subchunk2ID	4
little	40	Subchunk2Size	4
little	44	data	Subchunk2Size

The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"

The "fmt " sub-chunk

describes the format of the sound information in the data sub-chunk

The "data" sub-chunk

Indicates the size of the sound information and contains the raw sound data

Speaker notes

No notes on this slide.

Speaker notes

No notes on this slide.

Comments?

Questions?