filetypes

operating systems & open source group

sep 15 2025

c wiebe

what are some filetypes?

🟩 audio

- mp3
- flac
- wav
- m4a
- ogg
- opus

🟩 image

- png
- jpeg
- avif
- heic
- svg
- webp
- gif

🟩 video

- mp4
- mov

🟩 document

- pdf
- html
- docx
- pptx
- xlsx
- gdoc
- gslides
- gsheet
- epub
- mobi

🟩 container

- mkv
- zip
- tar
- riff

🟩 program

- jar
- exe
- apk
- apkm
- apks

🟩 text

- txt
- md
- adoc

🟩 code

- c
- java
- ...

🟩 data

- json

🟩 ???

- xml

common types

riff

- full of "blocks" of data (bits)
- each block header has it's type and size

xml

- full of "tags" which contain data (strings)
- tags open and close themselves and have attributes

zip

- compresses data using one of many algorithms
- primarily deflate (also used in pngs!)

how does your computer know a file's type?

file extension

- easily fooled

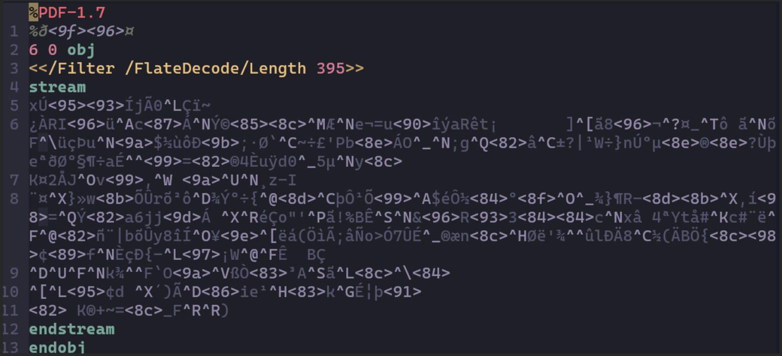
guess

- based on the file contents
- not always accurate or possible

type signatures

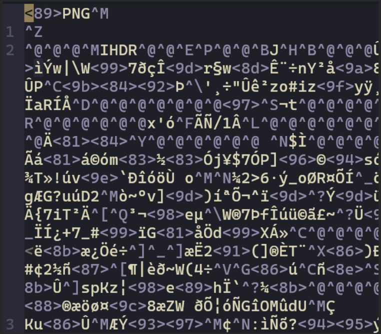
- also called magic numbers
- opening x bits of the file

inside a pdf



```
%PDF-1.7
%ð<9f><96>¤
6 0 obj
<</Filter /FlateDecode/Length 395>>
stream
xÚ<95><93>ÍjÃ0^LÇï~
¿ÀRI<96>ü^Ac<87>Á^NÝ©<85><8c>^MÆ^Ne¬=u<90>îýaRêt¡        ]^[ã8<96>¬^?¤_^Tô ã^Nõ
F^\üçÞu^N<9a>$½ùôÐ<9b>;·Ø`^C~÷£'Pb<8e>ÁO^_^N;g^Q<82>â^C±?|¹W÷}nÚ°µ<8e>®<8e>?Ùþ
eªðØ°§¶÷aÉ^^<99>=<82>®4ÈuÿdO^_5µ^Ny<8c>
K¤2ÅJ^Ov<99>¸^W <9a>^U^N¸z-I
¨¤^X}»w<8b>ÕÛrõ²ô^D¾Ý°÷{^@<8d>^CþÔ¹Õ<99>^A$éÔ½<84>°<8f>^O^_¾}¶R-<8d><8b>^X,í<9
8>=^QÝ<82>a6jj<9d>Á ^X^RéÇo"'^Pã!%BÊ^S^N&<96>R<93>3<84><84>c^Nxâ 4ªYtå#^Kc#¨ë^
F^@<82>ñ¨|bõÛy8îÍ^O¥<9e>^[ëá(ÖìÃ;âÑo>Ó7ÛÉ^_®æn<8c>^HØë'¾^^ûlÐÄ8^C½(ÄBÖ{<8c><98
>¢<89>f^NÈçÐ{-^L<97>¡W^@^FÊ  BÇ
^D^U^F^Nk¾^^F`O<9a>^VßÒ<83>³A^Sã^L<8c>^\<84>
^[^L<95>¢d ^X´)Ã^D<86>ie¹^H<83>k^GÉ¦þ<91>
<82> K®+~=<8c>_F^R^R)
endstream
endobj
```

inside a png



```
<89>PNG^M
^Z
^@^@^@^MIHDR^@^@^E^P^@^@^BJ^H^B^@^@^@
>ìÝw|\W<99>7ðçÎ<9d>r§w<8d>Ê¨÷nY²å<9a>
ÛP^C<9b><84><92>Þ^\',÷"Ûê²zo#iz<9f>yÿ
ÏaRÍÅ^D^@^@^@^@^@^@<97>^S¬t^@^@^@^@
R^@^@^@^@^@^@x'ó^FÃÑ/1Â^L^@^@^@^@^@
^@Ä<81><84>^Y^@^@^@^@^@ ^N$Ì^@^@^@
Ãá<81>á©óm<83>½<83>Ój¥$7ÓP]<96>©<94>s
¾T»!úv<9e>`ÐîóöÙ o^M^N¾2>6·ý_oØR¤ÕÍ^_
gÆG?uúD2^Mò~ºv]<9d>)íªÕ¬^ï<9d>^?Ý<9d>
Ä{7iT²Ä^[^Q³¬<98>eµ^\W®7Þfîúü©ã£~^?Ü<
_ÏÍ¿+7_#<99>ïG<81>åÖd<99>XÁ»^C^@^@^@
<ë<8b>æ¿Öé÷^]^_^]æË2<91>(]®ÈT¨^X<86>)
#¢2½ñ<87>^[¶|èð~W(4÷^V^G<86>ú^Cñ<8e>^
8b>Û^]spKz¦<98>e<89>hÏ`^?¾<8b>^@^@^@
<88>®æöø¤<9c>8æZW ðÕ¦óÑGîOMûdU^MÇ
Ku<86>Û^MÆÝ<93><97>^M¢^N:ìÑõ?<94><95>
```

0x89 :	0b10001001
0x504e47 :	ascii for png
0x0d0a :	dos-style line ending
0x1a :	dos-style end-of-file
0x0a :	unix-style line ending

- eight byte signature
- 0x89504e470d0a1a0a

files that lie to you

how

- file extensions that don't match the file signature
- file extensions that do match, but take advantage of how a parser works

why

- malicious intent
- completely normal reasons

examples

- arbitrarily rename text files
- html vs xml
- svg vs xml
- wav vs riff
- jar vs zip
- gdoc vs json
- docx vs xml
- docx vs zip
- pptx vs xml
- pptx vs zip
- opf vs xml
- opf vs zip
- epub vs xml
- epub vs zip
- mobi vs xml
- mobi vs zip
- azw vs xml
- azw vs zip
- vs xml
- kf8 vs zip

everything is a zip file of xml

- turns out making your file format is hard
- using an existing one is easy
- but, you don't want your files getting read by the wrong software
- the practice goes back to at least 1991 w/ wav
- most filetypes are like this

real filetypes

🟩 mkv

- container for unlimited audio, video, images, and subtitles
- commonly used as output from video/audio creation software, and you then strip out what you need
- from matroska (pronounced matryoshka, the word for russian nesting doll)
- honorary zip file of xml, but for extensible binary meta language
- ebml is xml but binary, with opening/closing tags, created for mkvs
- released 2002
- one i know the least about, but want to know more about

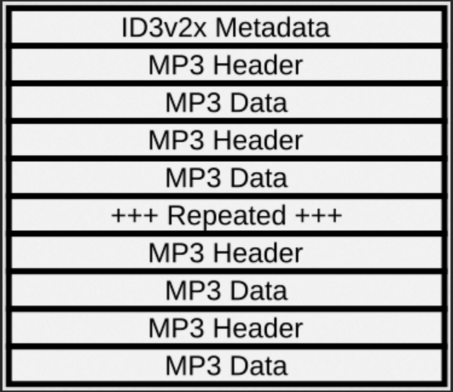
real filetypes

🟩 mp3

- lossy compression format that commonly achieves 75%--95% reduction in size
- revolutionized music distribution
- designed by the moving picture experts group, developed largely in germany by the fraunhofer society
- released 1991

real filetypes

mp3 file structure



mp3 encoding

1. divide the audio into small, overlapping pieces
1. convert each piece into a sum of cos functions
1. perform the fourier transform on each piece
1. remove sounds humans can't hear
1. encode each piece according to the bitrate
1. format each piece into an mp3 header/data block

real filetypes

🟩 png

- lossless compression format
- pronounced "ping" apparently
- created to be an improved version of gif (pronounced "jif" apparently)
- designed by the portable network graphics development group
- released 1996

real filetypes

png file structure

- type signature: 0x89504e470d0a1a0a
- made up of "chunks", like mp3s and riffs
- but png's headers are more fun
  - 4 bytes for size
  - 4 bytes for type
  - x bytes for data
  - 4 bytes for checksum

png headers

- chunk types are given as ascii words
  - IHDR
  - PLTE
  - IDAT
  - IEND
  - bKGD
  - gAMA
  - eXIf
- the case of each letter gives additional info
  - 1st = critical
  - 2nd = public
  - 3rd = reserved
  - 4th = dependent

real filetypes

🟩 png compression

- uses deflate
- first uses one of a number of prediction methods
  - that number is 1
- called method 0
  1. do nothing
  1. next = prev
  1. next = upper
  1. next = prev upper
  1. next = closest to prev + upper - prev upper
- intended to get everything close to 0 for deflates benefit
- basically, take the derivative of the image

thanks!