Parsing binary formats with Kaitai Struct

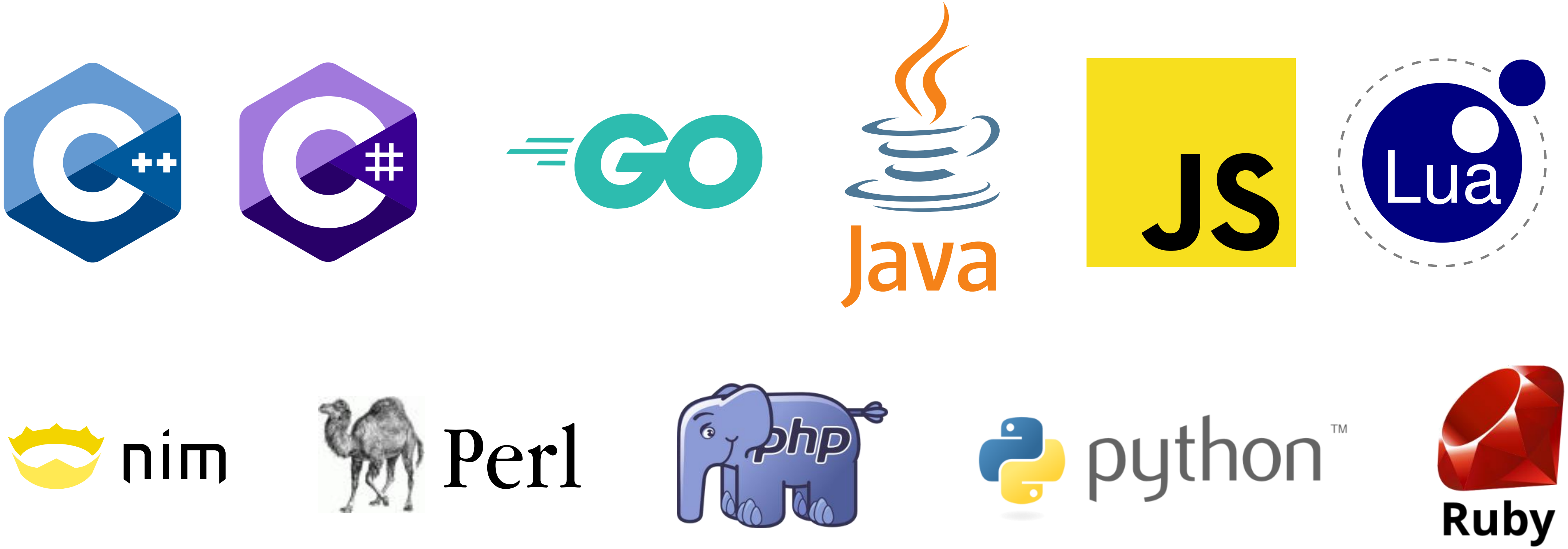
Petr Pucil

CZECH TECHNICAL UNIVERSITY IN PRAGUE

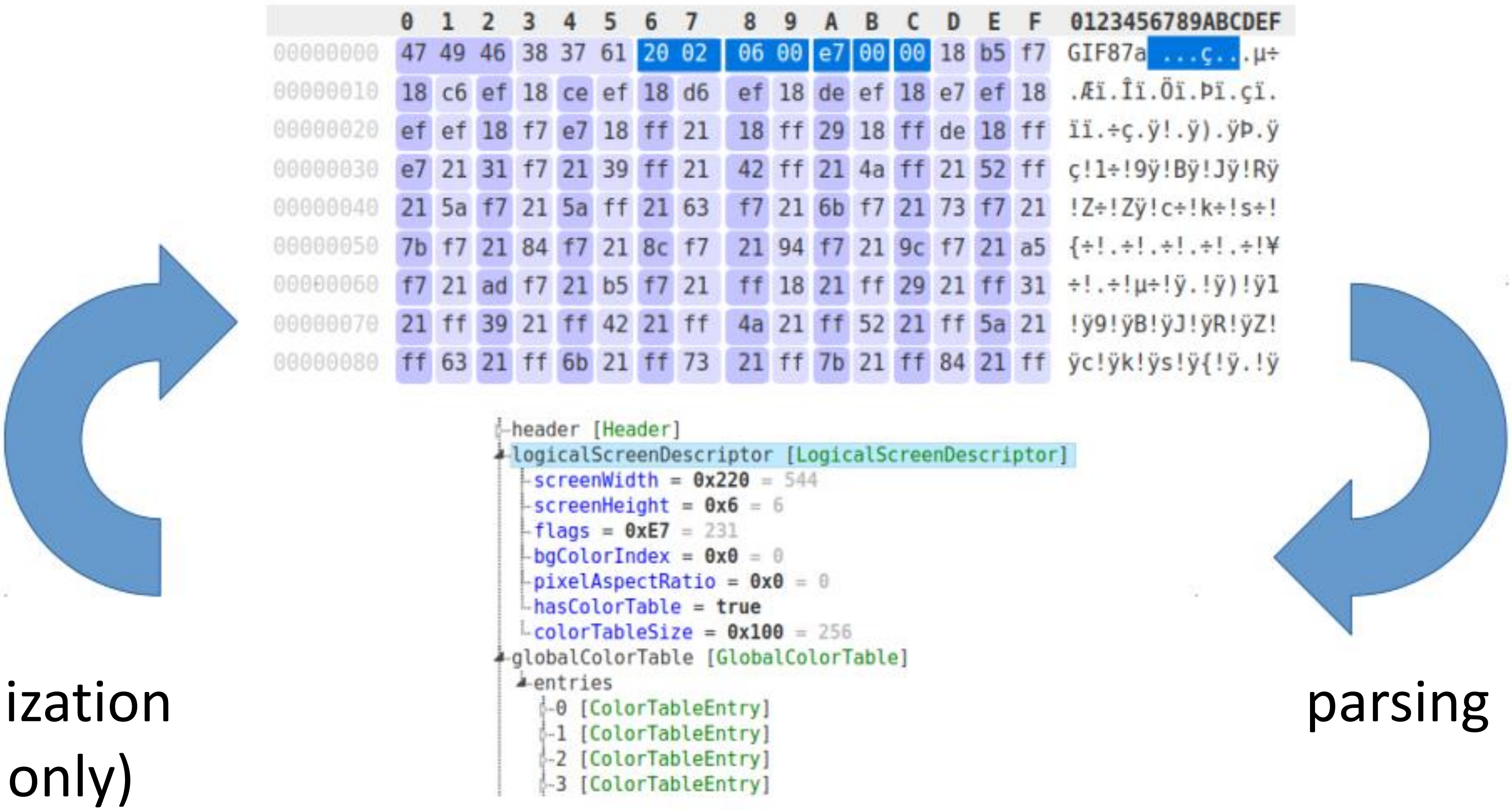
What is Kaitai Struct?

What is Kaitai Struct?

- tool for dealing with binary formats
- declarative language (.ksy) for specifying arbitrary binary formats
- parser generator for 11 programming languages



What is Kaitai Struct?



```
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F  0123456789ABCDEF
00000000 47 49 46 38 37 61 20 02  06 00 e7 00 00 18 b5 f7  GIF87a ...ç...µ÷
00000010 18 c6 ef 18 ce ef 18 d6  ef 18 de ef 18 e7 ef 18  .Æï.Îï.Öï.Þï.çï.
00000020 ef ef 18 f7 e7 18 ff 21  18 ff 29 18 ff de 18 ff  ïï.÷ç.ÿ!.ÿ).ÿÞ.ÿ
00000030 e7 21 31 f7 21 39 ff 21  42 ff 21 4a ff 21 52 ff  ç!1÷!9ÿ!Bÿ!Jÿ!Rÿ
00000040 21 5a f7 21 5a ff 21 63  f7 21 6b f7 21 73 f7 21  !Z÷!Zÿ!c÷!k÷!s÷!
00000050 7b f7 21 84 f7 21 8c f7  21 94 f7 21 9c f7 21 a5  {÷!.÷!.÷!.÷!.÷!¥
00000060 f7 21 ad f7 21 b5 f7 21  ff 18 21 ff 29 21 ff 31  ÷!.÷!µ÷!ÿ.!ÿ)!ÿ1
00000070 21 ff 39 21 ff 42 21 ff  4a 21 ff 52 21 ff 5a 21  !ÿ9!ÿB!ÿJ!ÿR!ÿZ!
00000080 ff 63 21 ff 6b 21 ff 73  21 ff 7b 21 ff 84 21 ff  ÿc!ÿk!ÿs!ÿ{!ÿ.!ÿ
```

```
-header [Header]
-logicalScreenDescriptor [LogicalScreenDescriptor]
  -screenWidth = 0x220 = 544
  -screenHeight = 0x6 = 6
  -flags = 0xE7 = 231
  -bgColorIndex = 0x0 = 0
  -pixelAspectRatio = 0x0 = 0
  -hasColorTable = true
  -colorTableSize = 0x100 = 256
-globalColorTable [GlobalColorTable]
  -entries
    -0 [ColorTableEntry]
    -1 [ColorTableEntry]
    -2 [ColorTableEntry]
    -3 [ColorTableEntry]
```

serialization
(Java only)

parsing

History

History

2014 ○ origin of the project (Mikhail Yakshin)

2016 ○ open-source release — Java, Ruby

2017 ○ FOSDEM 2017 presentation — 8 supported languages

☆ 400+

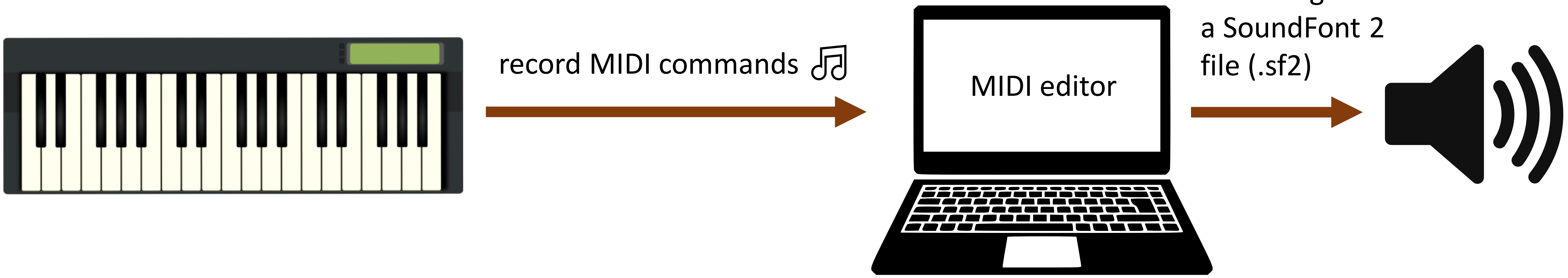
2023 ○ FOSDEM 2023 presentation, Java serialization — 11 supported languages

500+ ☆ 3300

How I discovered Kaitai Struct

How I discovered Kaitai Struct

→ My story



record MIDI commands ♫

MIDI editor

play
according to
a SoundFont 2
file (.sf2)

→ I became

- 2019 – Kaitai developer
- 2020 – admin

How to parse

How to parse

a) dedicated format library

b) own parser

c) parser combinator

d) parser generator

How to parse

a) dedicated format library

b) own parser

c) parser combinator

d) parser generator

How to parse

a) dedicated format library

b) own parser

c) parser combinator

d) parser generator

How to parse

a) dedicated format library

b) own parser

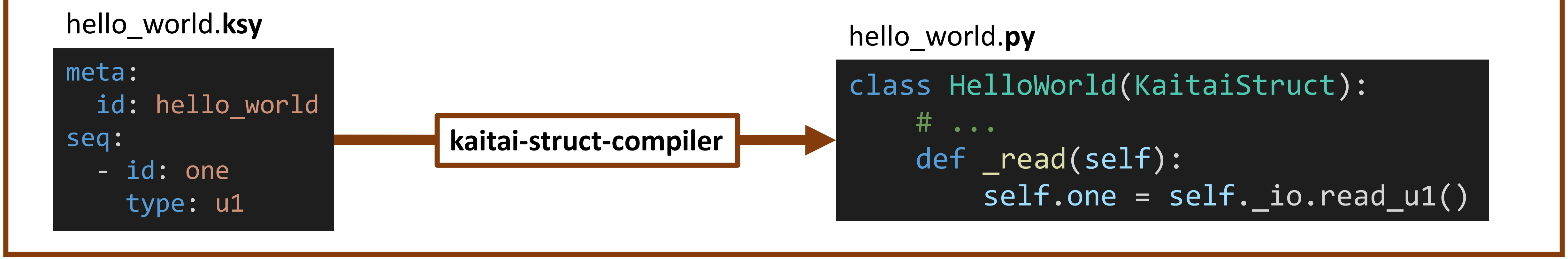
c) parser combinator

d) parser generator

Kaitai workflow

Kaitai workflow

1. Compilation



hello_world.**ksy**

```yaml
meta:
  id: hello_world
seq:
  - id: one
    type: u1
```

kaitai-struct-compiler

hello_world.**py**

```python
class HelloWorld(KaitaiStruct):
    # ...
    def _read(self):
        self.one = self._io.read_u1()
```

Kaitai workflow

1. Compilation

hello_world.**ksy**

```yaml
meta:
  id: hello_world
seq:
  - id: one
    type: u1
```

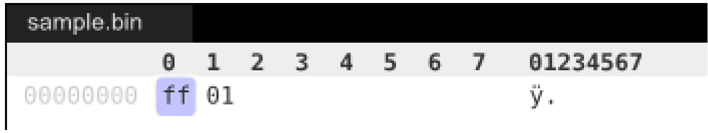
kaitai-struct-compiler

hello_world.**py**

```python
class HelloWorld(KaitaiStruct):
    # ...
    def _read(self):
        self.one = self._io.read_u1()
```

2. Parsing

input binary file



```
sample.bin
            0  1  2  3  4  5  6  7   01234567
00000000   ff 01                     ÿ.
```

hello_world.**py**

```python
class HelloWorld(KaitaiStruct):
    # ...
    def _read(self):
        self.one = self._io.read_u1()
```

parsed data

```
└─ one = 0xFF = 255
```

kaitaistruct.py (runtime library)

```python
class KaitaiStream:
    # ...
    def read_u1(self):
        return struct.unpack('B', self.read_bytes(1))[0]
```

Why Kaitai

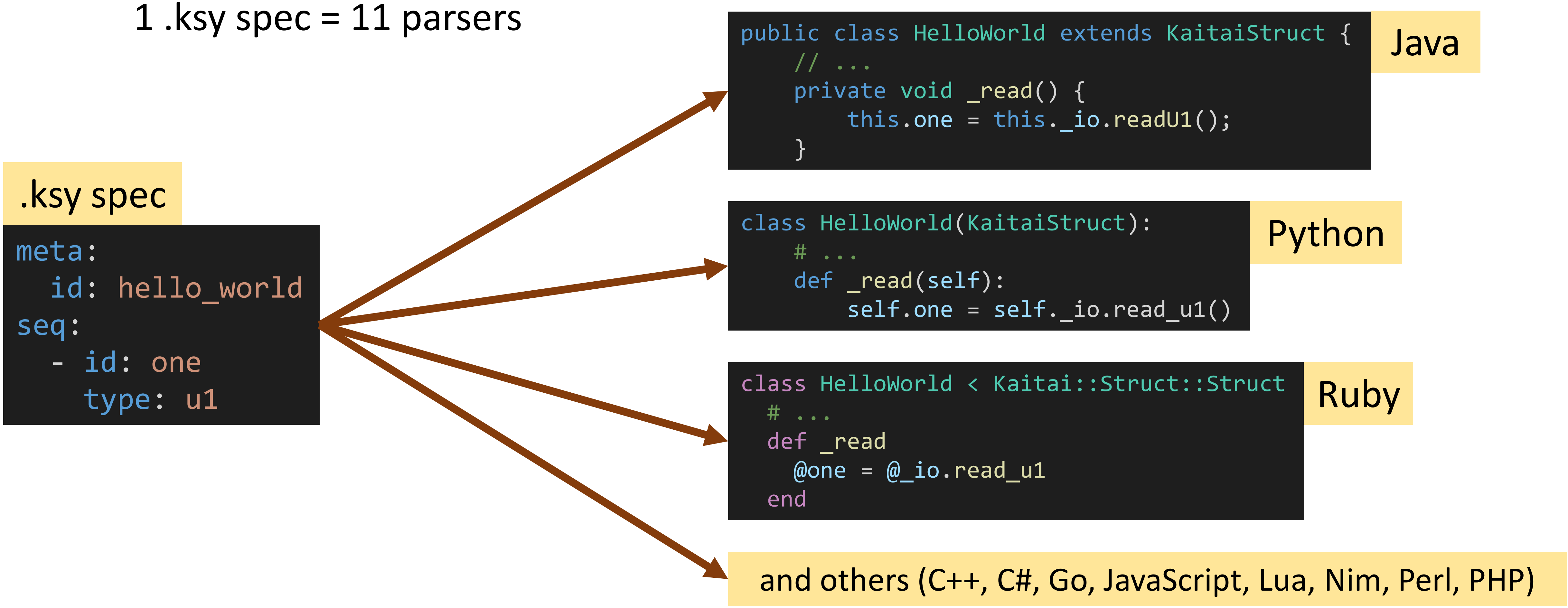
Why Kaitai

- write once, use everywhere
- standard way to describe binary formats
- library of format specifications
- GraphViz diagrams
- .ksy language simplicity
- visualization and dumping tools
    - console visualizer (ksv)
    - ksdump
    - Web IDE

Why Kaitai

→ write once, use everywhere

1 .ksy spec = 11 parsers



.ksy spec

```
meta:
  id: hello_world
seq:
  - id: one
    type: u1
```

```
public class HelloWorld extends KaitaiStruct {
    // ...
    private void _read() {
        this.one = this._io.readU1();
    }
```

Java

```
class HelloWorld(KaitaiStruct):
    # ...
    def _read(self):
        self.one = self._io.read_u1()
```

Python

```
class HelloWorld < Kaitai::Struct::Struct
  # ...
  def _read
    @one = @_io.read_u1
  end
```

Ruby

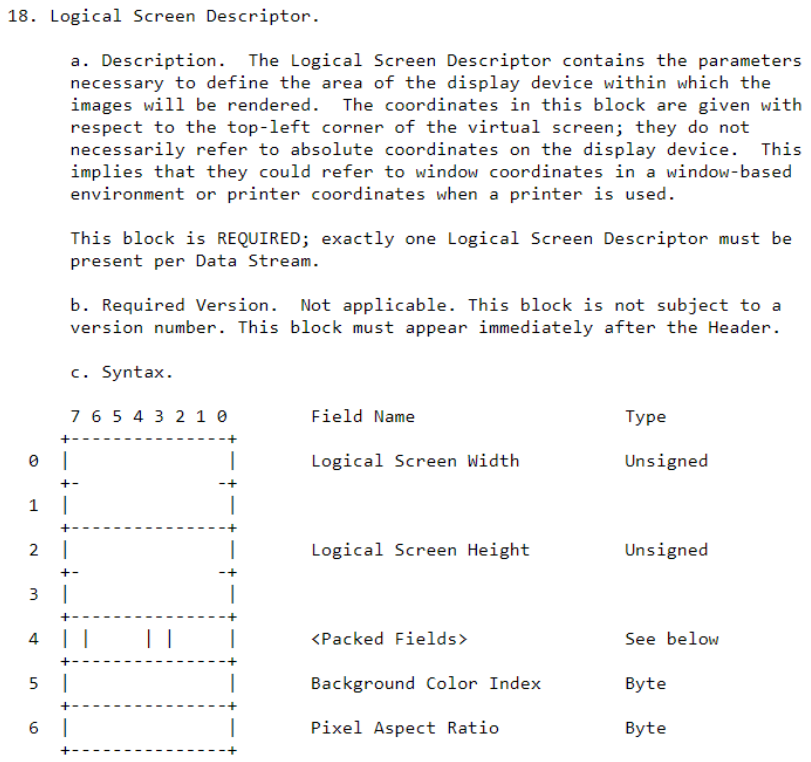
and others (C++, C#, Go, JavaScript, Lua, Nim, Perl, PHP)

Why Kaitai

→ standard way to describe binary formats

no single standard

GIF



18. Logical Screen Descriptor.

a. Description. The Logical Screen Descriptor contains the parameters necessary to define the area of the display device within which the images will be rendered. The coordinates in this block are given with respect to the top-left corner of the virtual screen; they do not necessarily refer to absolute coordinates on the display device. This implies that they could refer to window coordinates in a window-based environment or printer coordinates when a printer is used.

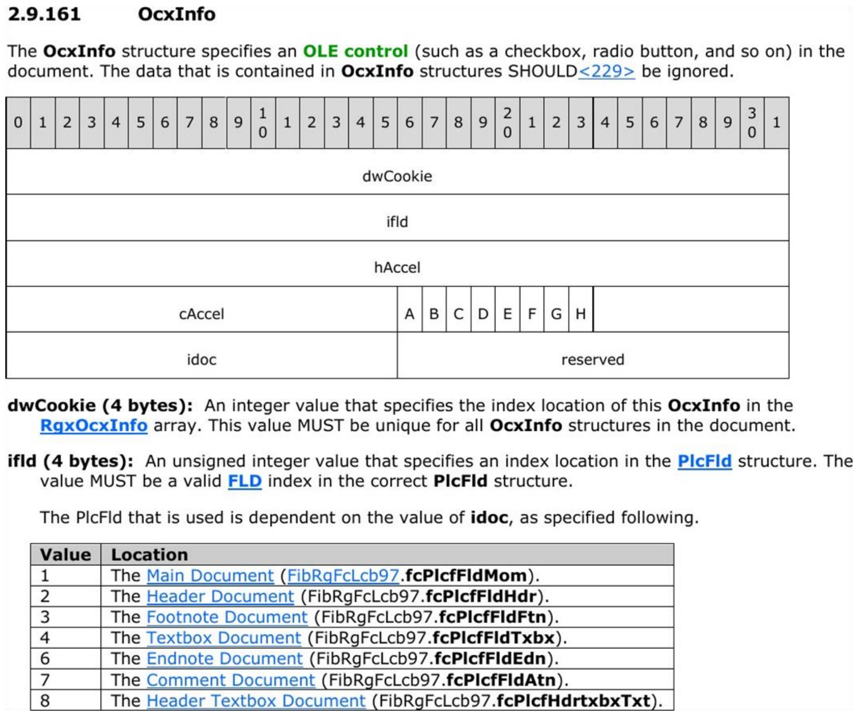
This block is REQUIRED; exactly one Logical Screen Descriptor must be present per Data Stream.

b. Required Version. Not applicable. This block is not subject to a version number. This block must appear immediately after the Header.

c. Syntax.

```
      7 6 5 4 3 2 1 0        Field Name                    Type
     +---------------+
  0  |               |       Logical Screen Width          Unsigned
     +-             -+
  1  |               |
     +---------------+
  2  |               |       Logical Screen Height         Unsigned
     +-             -+
  3  |               |
     +---------------+
  4  | |     | |     |       <Packed Fields>               See below
     +---------------+
  5  |               |       Background Color Index        Byte
     +---------------+
  6  |               |       Pixel Aspect Ratio            Byte
     +---------------+
```

Microsoft Word .doc



2.9.161 OcxInfo

The **OcxInfo** structure specifies an **OLE control** (such as a checkbox, radio button, and so on) in the document. The data that is contained in **OcxInfo** structures SHOULD<229> be ignored.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
dwCookie																															
ifld																															
hAccel																															
cAccel																A	B	C	D	E	F	G	H								
idoc																reserved															

dwCookie (4 bytes): An integer value that specifies the index location of this **OcxInfo** in the **RgxOcxInfo** array. This value MUST be unique for all **OcxInfo** structures in the document.

ifld (4 bytes): An unsigned integer value that specifies an index location in the **PlcFld** structure. The value MUST be a valid **FLD** index in the correct **PlcFld** structure.

The PlcFld that is used is dependent on the value of **idoc**, as specified following.

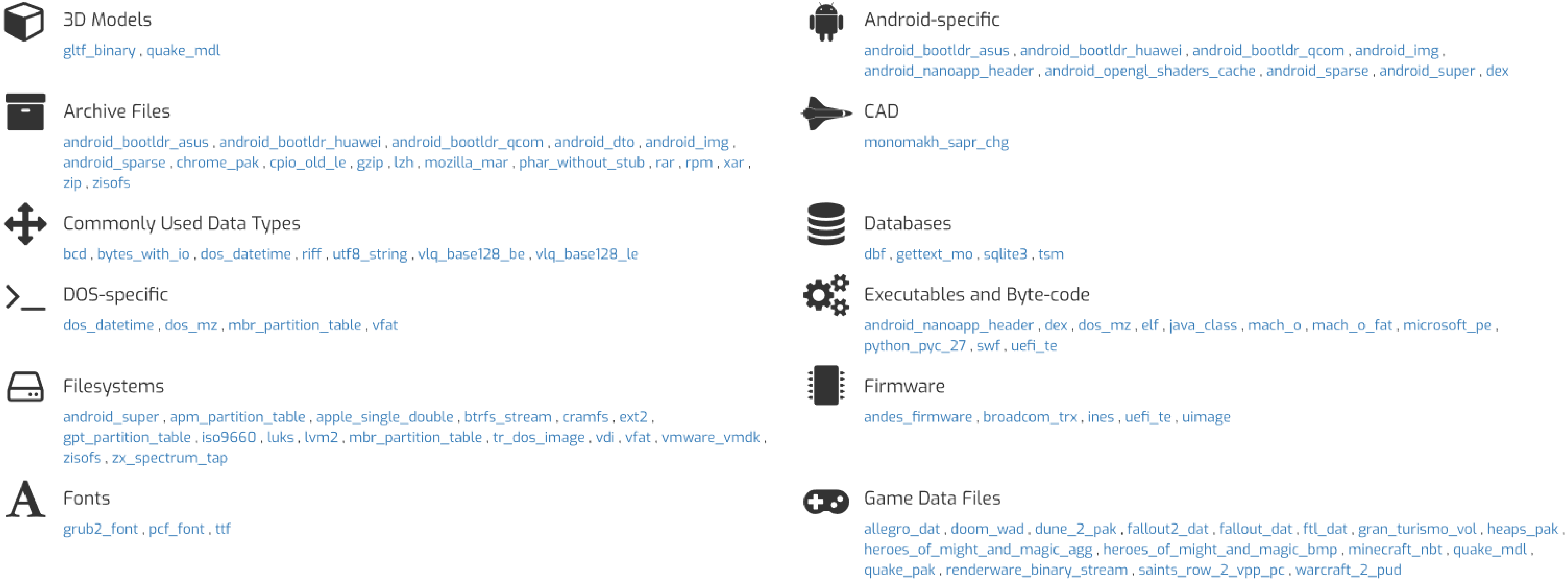
Value	Location
1	The Main Document (FibRgFcLcb97.**fcPlcfFldMom**).
2	The Header Document (FibRgFcLcb97.**fcPlcfFldHdr**).
3	The Footnote Document (FibRgFcLcb97.**fcPlcfFldFtn**).
4	The Textbox Document (FibRgFcLcb97.**fcPlcfFldTxbx**).
6	The Endnote Document (FibRgFcLcb97.**fcPlcfFldEdn**).
7	The Comment Document (FibRgFcLcb97.**fcPlcfFldAtn**).
8	The Header Textbox Document (FibRgFcLcb97.**fcPlcfHdrtxbxTxt**).

Why Kaitai

→ library of .ksy format specifications

181 specifications

formats.kaitai.io



3D Models

gltf_binary , quake_mdl

Archive Files

android_bootldr_asus , android_bootldr_huawei , android_bootldr_qcom , android_dto , android_img , android_sparse , chrome_pak , cpio_old_le , gzip , lzh , mozilla_mar , phar_without_stub , rar , rpm , xar , zip , zisofs

Commonly Used Data Types

bcd , bytes_with_io , dos_datetime , riff , utf8_string , vlq_base128_be , vlq_base128_le

DOS-specific

dos_datetime , dos_mz , mbr_partition_table , vfat

Filesystems

android_super , apm_partition_table , apple_single_double , btrfs_stream , cramfs , ext2 , gpt_partition_table , iso9660 , luks , lvm2 , mbr_partition_table , tr_dos_image , vdi , vfat , vmware_vmdk , zisofs , zx_spectrum_tap

Fonts

grub2_font , pcf_font , ttf

Android-specific

android_bootldr_asus , android_bootldr_huawei , android_bootldr_qcom , android_img , android_nanoapp_header , android_opengl_shaders_cache , android_sparse , android_super , dex

CAD

monomakh_sapr_chg

Databases

dbf , gettext_mo , sqlite3 , tsm

Executables and Byte-code

android_nanoapp_header , dex , dos_mz , elf , java_class , mach_o , mach_o_fat , microsoft_pe , python_pyc_27 , swf , uefi_te

Firmware

andes_firmware , broadcom_trx , ines , uefi_te , uimage

Game Data Files

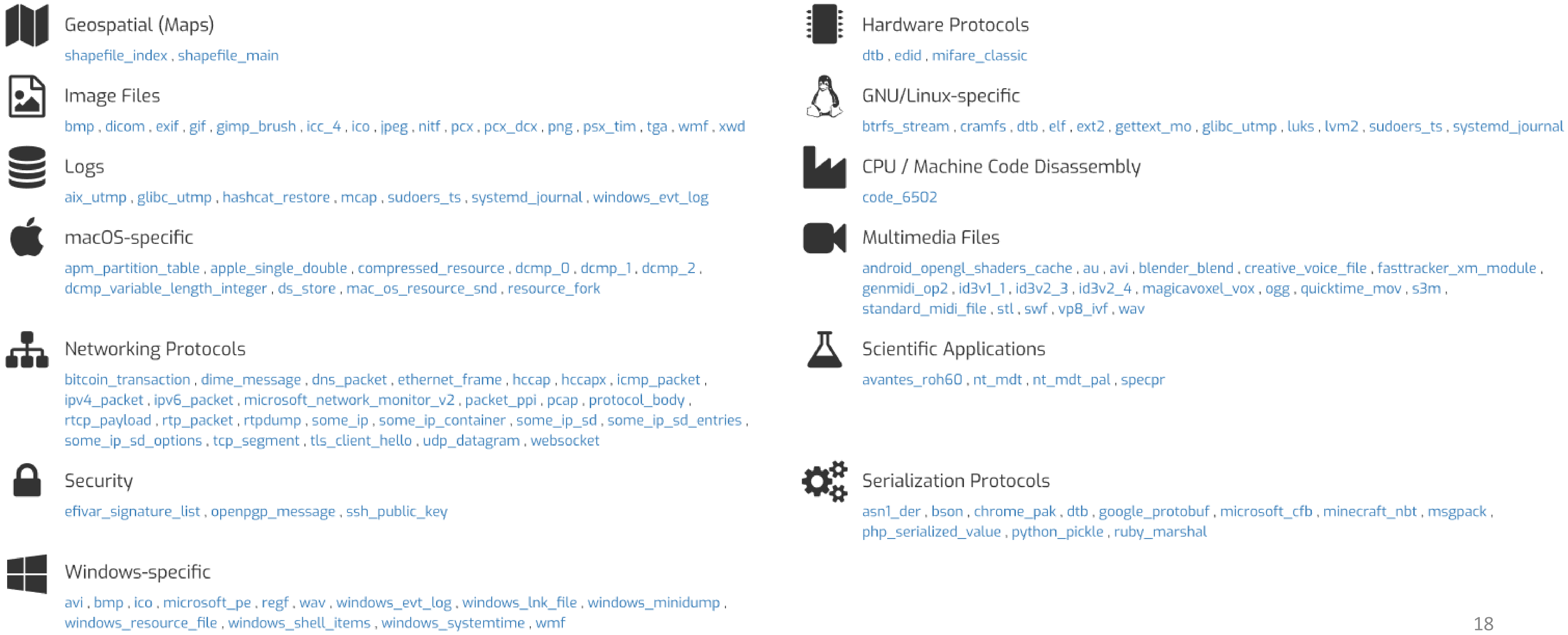
allegro_dat , doom_wad , dune_2_pak , fallout2_dat , fallout_dat , ftl_dat , gran_turismo_vol , heaps_pak , heroes_of_might_and_magic_agg , heroes_of_might_and_magic_bmp , minecraft_nbt , quake_mdl , quake_pak , renderware_binary_stream , saints_row_2_vpp_pc , warcraft_2_pud

Why Kaitai

→ library of .ksy format specifications

181 specifications

formats.kaitai.io



Geospatial (Maps)

shapefile_index , shapefile_main

Image Files

bmp , dicom , exif , gif , gimp_brush , icc_4 , ico , jpeg , nitf , pcx , pcx_dcx , png , psx_tim , tga , wmf , xwd

Logs

aix_utmp , glibc_utmp , hashcat_restore , mcap , sudoers_ts , systemd_journal , windows_evt_log

macOS-specific

apm_partition_table , apple_single_double , compressed_resource , dcmp_0 , dcmp_1 , dcmp_2 , dcmp_variable_length_integer , ds_store , mac_os_resource_snd , resource_fork

Networking Protocols

bitcoin_transaction , dime_message , dns_packet , ethernet_frame , hccap , hccapx , icmp_packet , ipv4_packet , ipv6_packet , microsoft_network_monitor_v2 , packet_ppi , pcap , protocol_body , rtcp_payload , rtp_packet , rtpdump , some_ip , some_ip_container , some_ip_sd , some_ip_sd_entries , some_ip_sd_options , tcp_segment , tls_client_hello , udp_datagram , websocket

Security

efivar_signature_list , openpgp_message , ssh_public_key

Windows-specific

avi , bmp , ico , microsoft_pe , regf , wav , windows_evt_log , windows_lnk_file , windows_minidump , windows_resource_file , windows_shell_items , windows_systemtime , wmf

Hardware Protocols

dtb , edid , mifare_classic

GNU/Linux-specific

btrfs_stream , cramfs , dtb , elf , ext2 , gettext_mo , glibc_utmp , luks , lvm2 , sudoers_ts , systemd_journal

CPU / Machine Code Disassembly

code_6502

Multimedia Files

android_opengl_shaders_cache , au , avi , blender_blend , creative_voice_file , fasttracker_xm_module , genmidi_op2 , id3v1_1 , id3v2_3 , id3v2_4 , magicavoxel_vox , ogg , quicktime_mov , s3m , standard_midi_file , stl , swf , vp8_ivf , wav

Scientific Applications

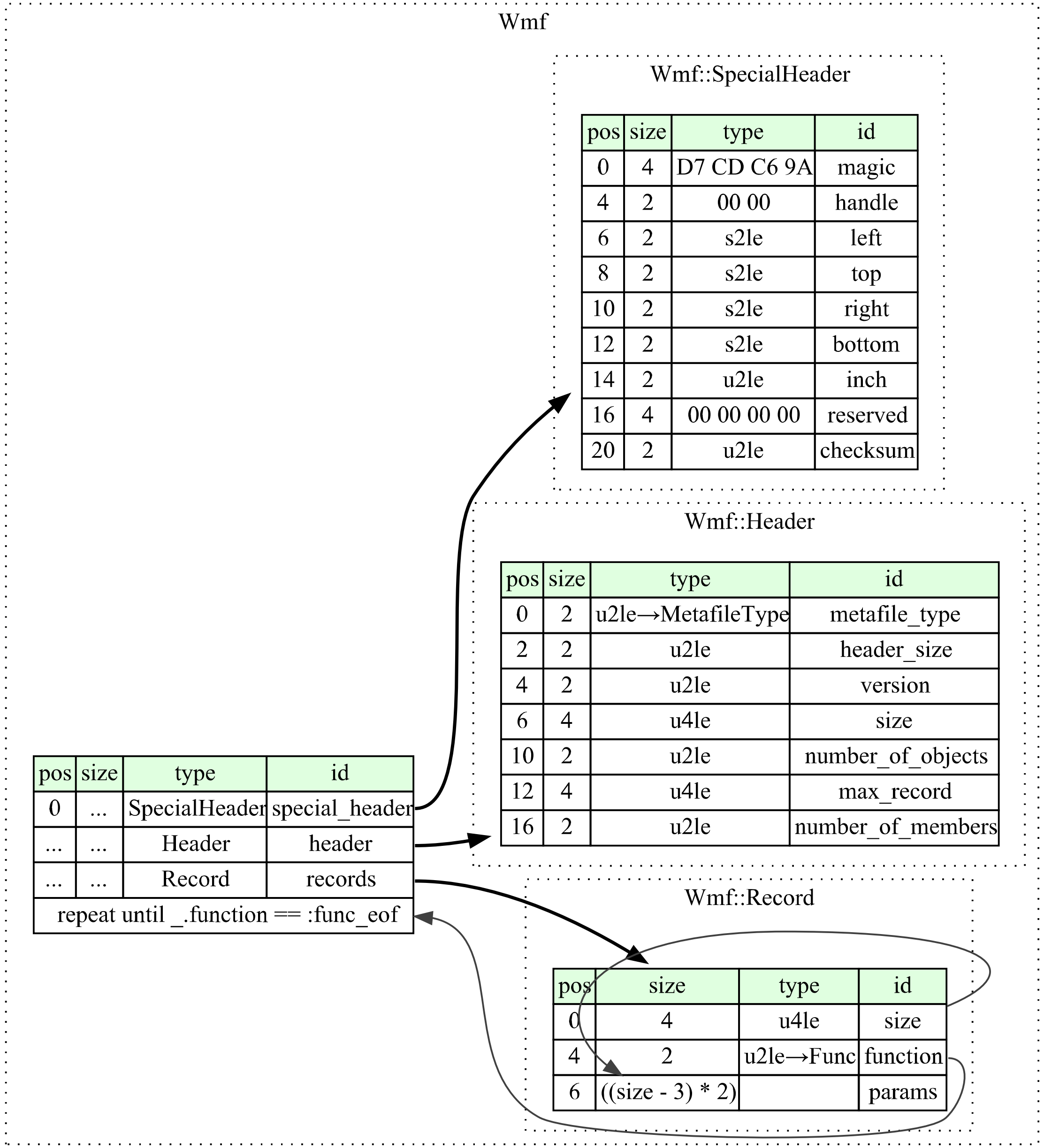
avantes_roh60 , nt_mdt , nt_mdt_pal , specpr

Serialization Protocols

asn1_der , bson , chrome_pak , dtb , google_protobuf , microsoft_cfb , minecraft_nbt , msgpack , php_serialized_value , python_pickle , ruby_marshal

Why Kaitai

→ GraphViz diagrams



Wmf::SpecialHeader

pos	size	type	id
0	4	D7 CD C6 9A	magic
4	2	00 00	handle
6	2	s2le	left
8	2	s2le	top
10	2	s2le	right
12	2	s2le	bottom
14	2	u2le	inch
16	4	00 00 00 00	reserved
20	2	u2le	checksum

Wmf::Header

pos	size	type	id
0	2	u2le→MetafileType	metafile_type
2	2	u2le	header_size
4	2	u2le	version
6	4	u4le	size
10	2	u2le	number_of_objects
12	4	u4le	max_record
16	2	u2le	number_of_members

Wmf

pos	size	type	id
0	...	SpecialHeader	special_header
...	...	Header	header
...	...	Record	records
repeat until _.function == :func_eof			

Wmf::Record

pos	size	type	id
0	4	u4le	size
4	2	u2le→Func	function
6	((size - 3) * 2)		params

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

```yaml
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

```yaml
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

```yaml
  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

little endian

```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

```

  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

attribute name

```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

```
  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4 unsigned 4-byte integer
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

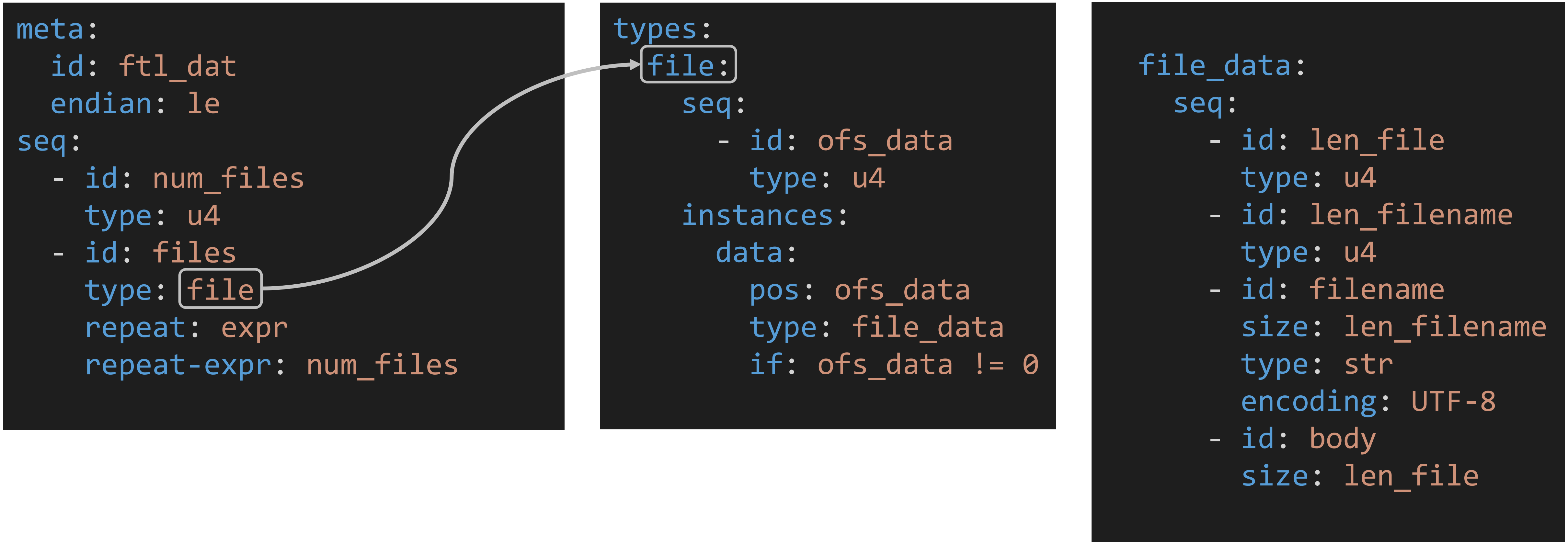
```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

```

  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …



```yaml
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

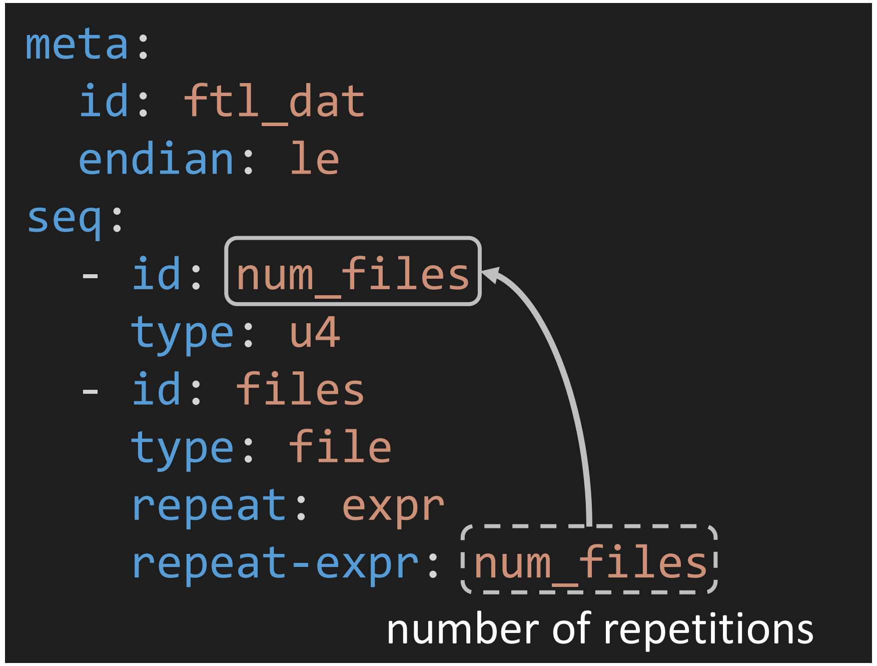
```yaml
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

```yaml

  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …



```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

number of repetitions

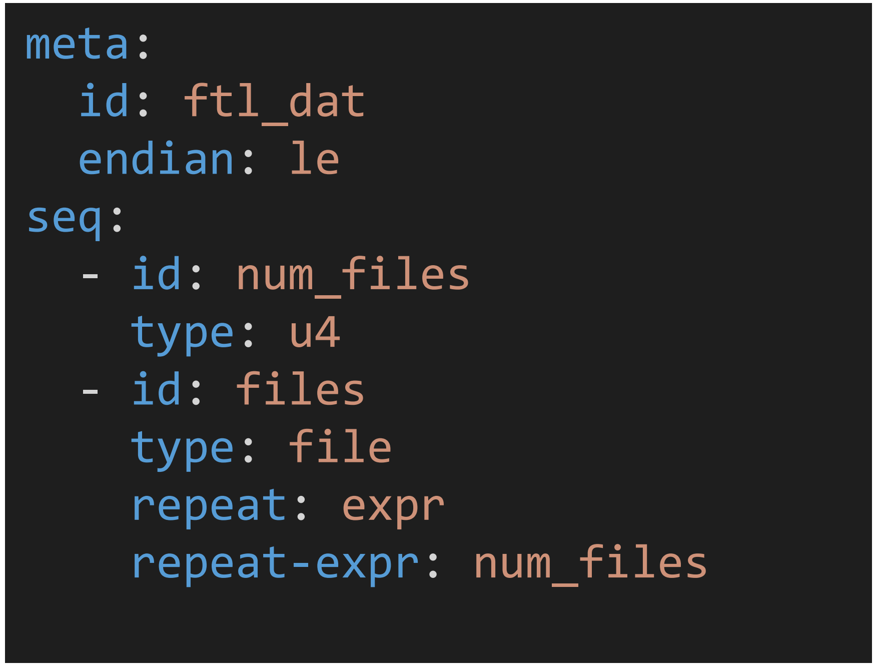
```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

```
  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

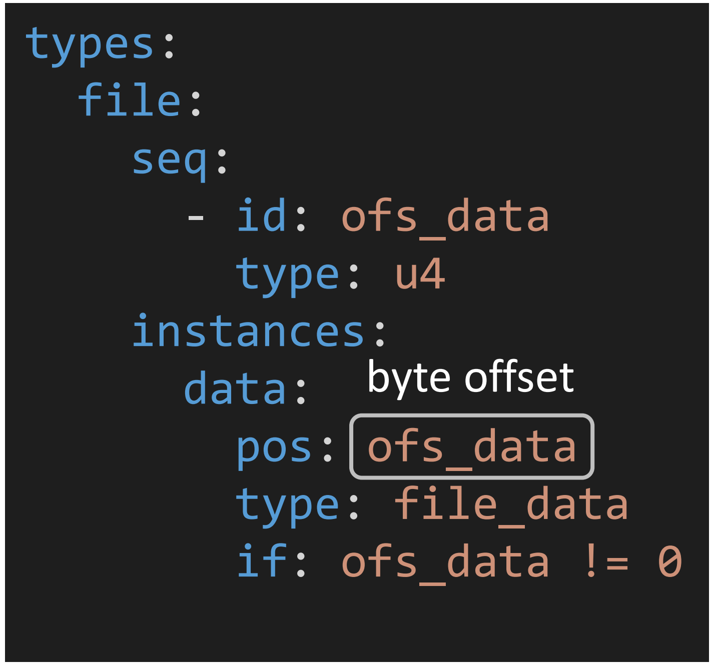
Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

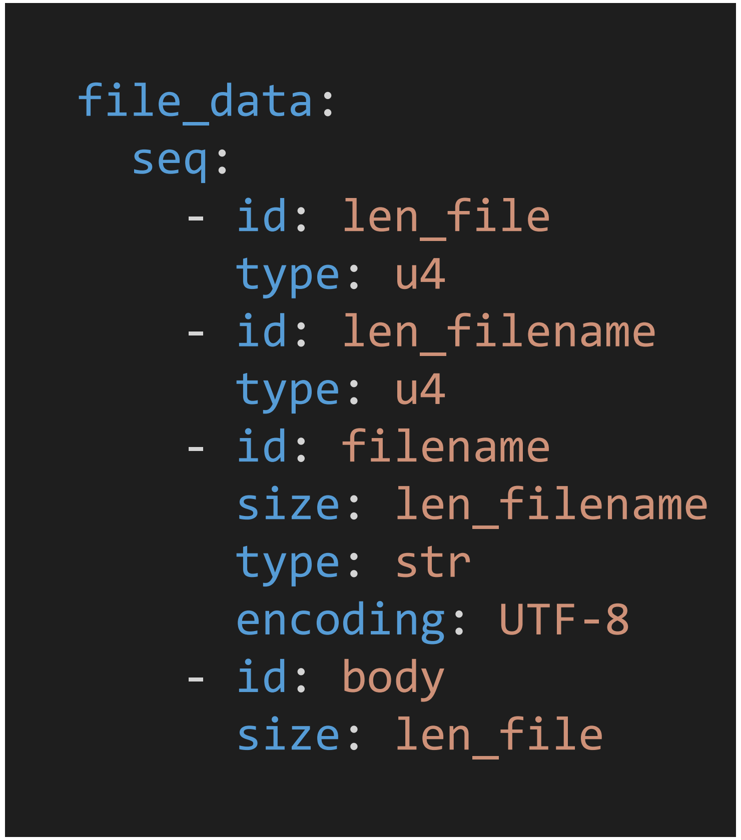


```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```



```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

byte offset



```
  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

expression language

```

  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

Why Kaitai

→ .ksy language simplicity

but powerful: unaligned bit types, type switch, byte processing, imports, …

```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```

```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

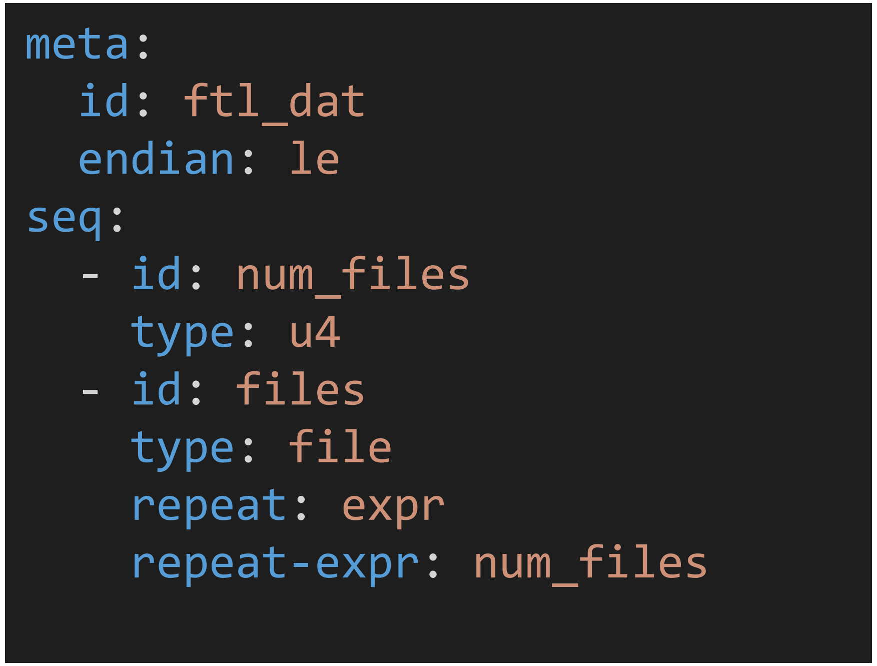
```
  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

char. string

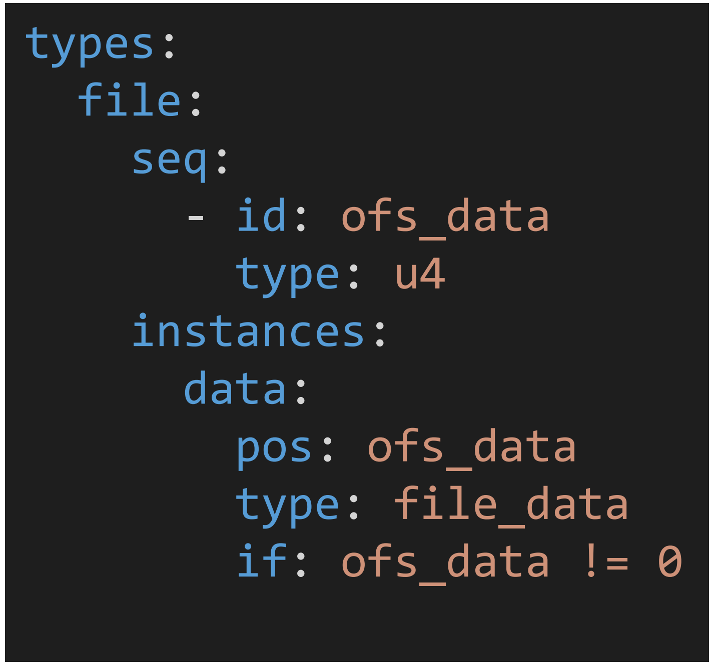
Why Kaitai

→ .ksy language simplicity

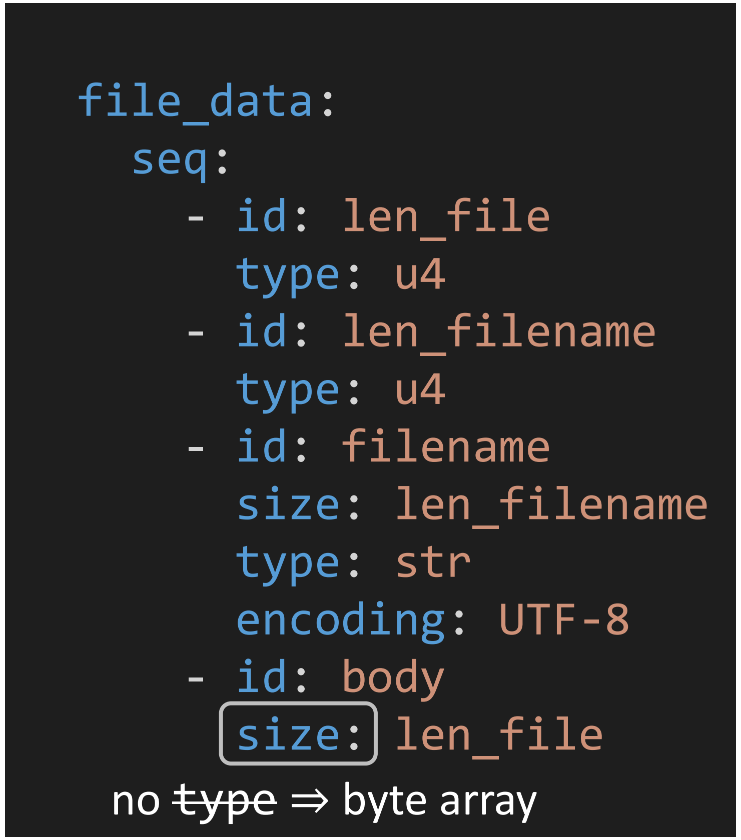
but powerful: unaligned bit types, type switch, byte processing, imports, …



```
meta:
  id: ftl_dat
  endian: le
seq:
  - id: num_files
    type: u4
  - id: files
    type: file
    repeat: expr
    repeat-expr: num_files
```



```
types:
  file:
    seq:
      - id: ofs_data
        type: u4
    instances:
      data:
        pos: ofs_data
        type: file_data
        if: ofs_data != 0
```

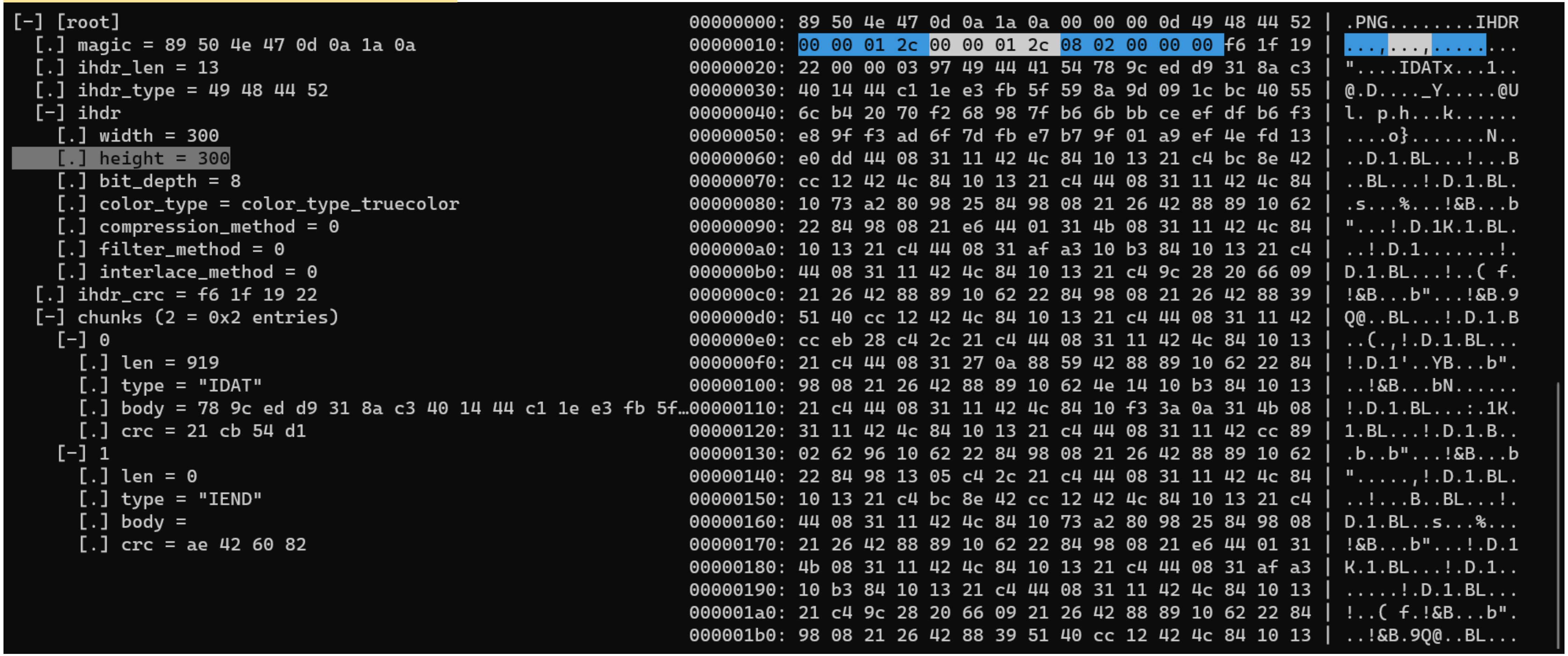


```
  file_data:
    seq:
      - id: len_file
        type: u4
      - id: len_filename
        type: u4
      - id: filename
        size: len_filename
        type: str
        encoding: UTF-8
      - id: body
        size: len_file
```

no ~~type~~ ⇒ byte array

→ visualization and dumping tools

Console visualizer (ksv)

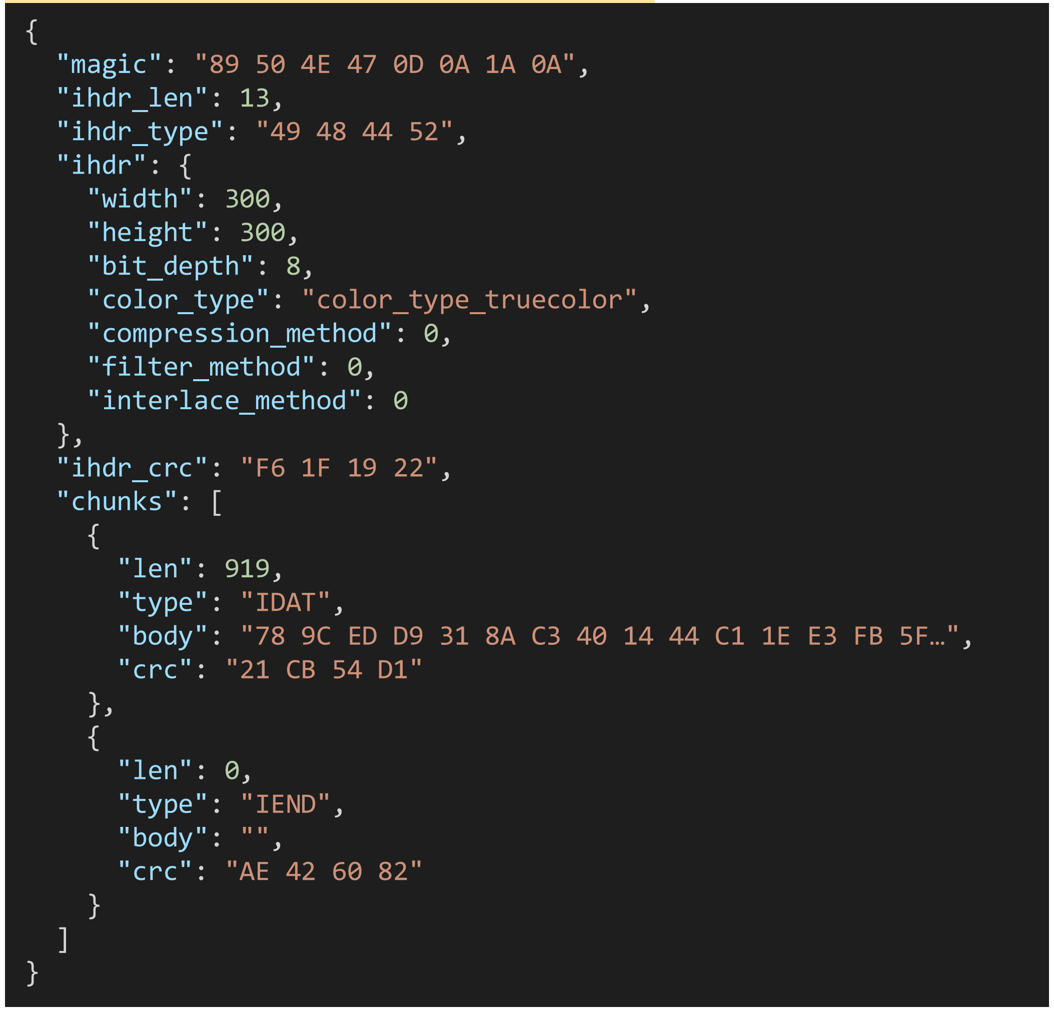


```
[-] [root]                                                    00000000: 89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 | .PNG........IHDR
  [.] magic = 89 50 4e 47 0d 0a 1a 0a                         00000010: 00 00 01 2c 00 00 01 2c 08 02 00 00 00 f6 1f 19 | ...,...,........
  [.] ihdr_len = 13                                           00000020: 22 00 00 03 97 49 44 41 54 78 9c ed d9 31 8a c3 | "....IDATx...1..
  [.] ihdr_type = 49 48 44 52                                 00000030: 40 14 44 c1 1e e3 fb 5f 59 8a 9d 09 1c bc 40 55 | @.D...._Y.....@U
  [-] ihdr                                                    00000040: 6c b4 20 70 f2 68 98 7f b6 6b bb ce ef df b6 f3 | l. p.h...k......
    [.] width = 300                                           00000050: e8 9f f3 ad 6f 7d fb e7 b7 9f 01 a9 ef 4e fd 13 | ....o}.......N..
    [.] height = 300                                          00000060: e0 dd 44 08 31 11 42 4c 84 10 13 21 c4 bc 8e 42 | ..D.1.BL...!...B
    [.] bit_depth = 8                                         00000070: cc 12 42 4c 84 10 13 21 c4 44 08 31 11 42 4c 84 | ..BL...!.D.1.BL.
    [.] color_type = color_type_truecolor                     00000080: 10 73 a2 80 98 25 84 98 08 21 26 42 88 89 10 62 | .s...%...!&B...b
    [.] compression_method = 0                                00000090: 22 84 98 08 21 e6 44 01 31 4b 08 31 11 42 4c 84 | "...!.D.1K.1.BL.
    [.] filter_method = 0                                     000000a0: 10 13 21 c4 44 08 31 af a3 10 b3 84 10 13 21 c4 | ..!.D.1.......!.
    [.] interlace_method = 0                                  000000b0: 44 08 31 11 42 4c 84 10 13 21 c4 9c 28 20 66 09 | D.1.BL...!..( f.
  [.] ihdr_crc = f6 1f 19 22                                  000000c0: 21 26 42 88 89 10 62 22 84 98 08 21 26 42 88 39 | !&B...b"...!&B.9
  [-] chunks (2 = 0x2 entries)                                000000d0: 51 40 cc 12 42 4c 84 10 13 21 c4 44 08 31 11 42 | Q@..BL...!.D.1.B
    [-] 0                                                     000000e0: cc eb 28 c4 2c 21 c4 44 08 31 11 42 4c 84 10 13 | ..(.,!.D.1.BL...
      [.] len = 919                                           000000f0: 21 c4 44 08 31 27 0a 88 59 42 88 89 10 62 22 84 | !.D.1'..YB...b".
      [.] type = "IDAT"                                       00000100: 98 08 21 26 42 88 89 10 62 4e 14 10 b3 84 10 13 | ..!&B...bN......
      [.] body = 78 9c ed d9 31 8a c3 40 14 44 c1 1e e3 fb 5f…00000110: 21 c4 44 08 31 11 42 4c 84 10 f3 3a 0a 31 4b 08 | !.D.1.BL...:.1K.
      [.] crc = 21 cb 54 d1                                   00000120: 31 11 42 4c 84 10 13 21 c4 44 08 31 11 42 cc 89 | 1.BL...!.D.1.B..
    [-] 1                                                     00000130: 02 62 96 10 62 22 84 98 08 21 26 42 88 89 10 62 | .b..b"...!&B...b
      [.] len = 0                                             00000140: 22 84 98 13 05 c4 2c 21 c4 44 08 31 11 42 4c 84 | ".....,!.D.1.BL.
      [.] type = "IEND"                                       00000150: 10 13 21 c4 bc 8e 42 cc 12 42 4c 84 10 13 21 c4 | ..!...B..BL...!.
      [.] body =                                              00000160: 44 08 31 11 42 4c 84 10 73 a2 80 98 25 84 98 08 | D.1.BL..s...%...
      [.] crc = ae 42 60 82                                   00000170: 21 26 42 88 89 10 62 22 84 98 08 21 e6 44 01 31 | !&B...b"...!.D.1
                                                              00000180: 4b 08 31 11 42 4c 84 10 13 21 c4 44 08 31 af a3 | K.1.BL...!.D.1..
                                                              00000190: 10 b3 84 10 13 21 c4 44 08 31 11 42 4c 84 10 13 | .....!.D.1.BL...
                                                              000001a0: 21 c4 9c 28 20 66 09 21 26 42 88 89 10 62 22 84 | !..( f.!&B...b".
                                                              000001b0: 98 08 21 26 42 88 39 51 40 cc 12 42 4c 84 10 13 | ..!&B.9Q@..BL...
```

Why Kaitai

→ visualization and dumping tools

ksdump (JSON output)

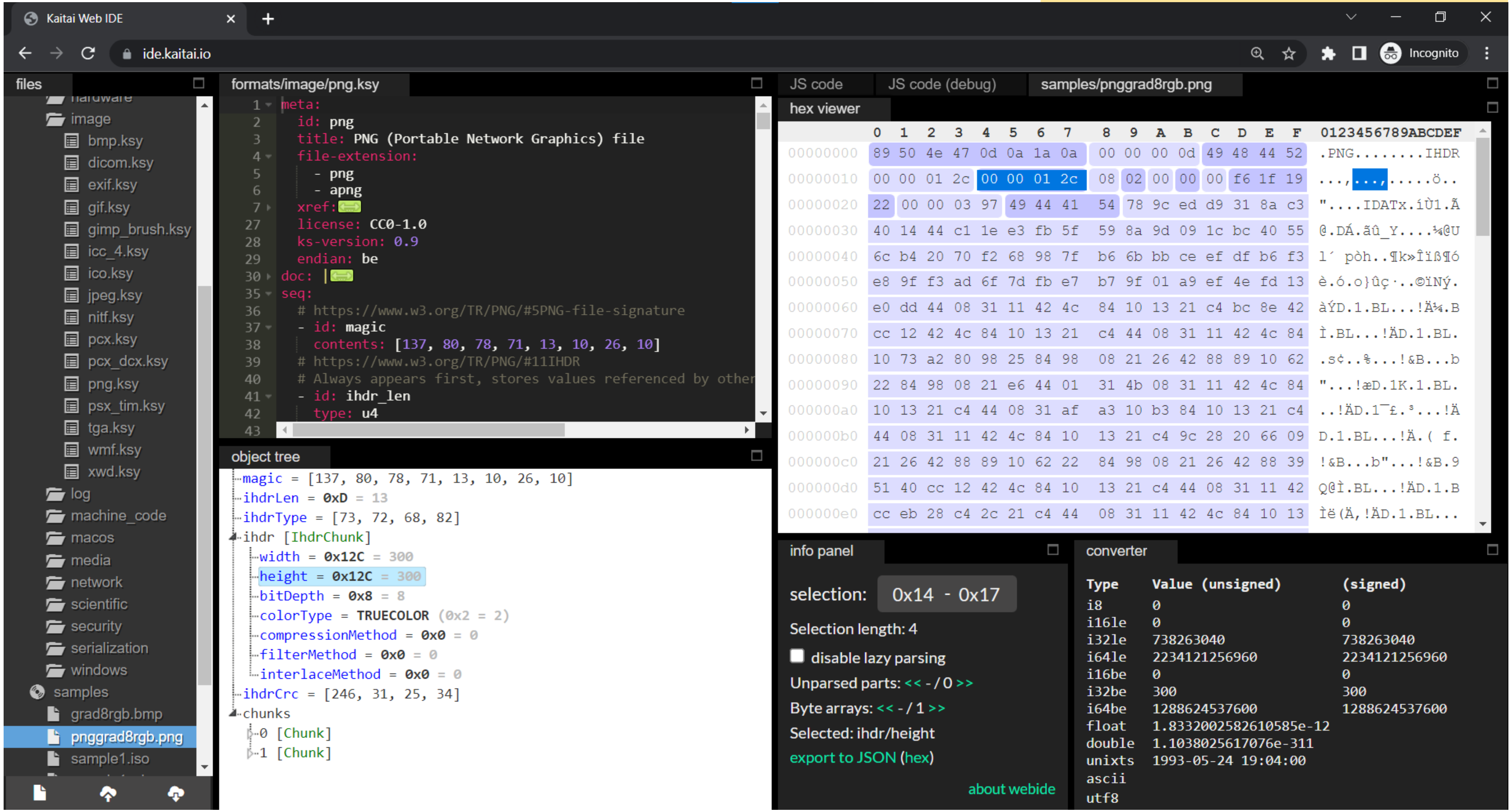


```
{
  "magic": "89 50 4E 47 0D 0A 1A 0A",
  "ihdr_len": 13,
  "ihdr_type": "49 48 44 52",
  "ihdr": {
    "width": 300,
    "height": 300,
    "bit_depth": 8,
    "color_type": "color_type_truecolor",
    "compression_method": 0,
    "filter_method": 0,
    "interlace_method": 0
  },
  "ihdr_crc": "F6 1F 19 22",
  "chunks": [
    {
      "len": 919,
      "type": "IDAT",
      "body": "78 9C ED D9 31 8A C3 40 14 44 C1 1E E3 FB 5F…",
      "crc": "21 CB 54 D1"
    },
    {
      "len": 0,
      "type": "IEND",
      "body": "",
      "crc": "AE 42 60 82"
    }
  ]
}
```

Why Kaitai

→ visualization and dumping tools

Web IDE (ide.kaitai.io)



hardware
- image
  - bmp.ksy
  - dicom.ksy
  - exif.ksy
  - gif.ksy
  - gimp_brush.ksy
  - icc_4.ksy
  - ico.ksy
  - jpeg.ksy
  - nitf.ksy
  - pcx.ksy
  - pcx_dcx.ksy
  - png.ksy
  - psx_tim.ksy
  - tga.ksy
  - wmf.ksy
  - xwd.ksy
- log
- machine_code
- macos
- media
- network
- scientific
- security
- serialization
- windows
- samples
  - grad8rgb.bmp
  - pnggrad8rgb.png
  - sample1.iso

formats/image/png.ksy

```
meta:
  id: png
  title: PNG (Portable Network Graphics) file
  file-extension:
    - png
    - apng
  xref:
  license: CC0-1.0
  ks-version: 0.9
  endian: be
doc: |
seq:
  # https://www.w3.org/TR/PNG/#5PNG-file-signature
  - id: magic
    contents: [137, 80, 78, 71, 13, 10, 26, 10]
  # https://www.w3.org/TR/PNG/#11IHDR
  # Always appears first, stores values referenced by other
  - id: ihdr_len
    type: u4
```

object tree

```
magic = [137, 80, 78, 71, 13, 10, 26, 10]
ihdrLen = 0xD = 13
ihdrType = [73, 72, 68, 82]
ihdr [IhdrChunk]
  width = 0x12C = 300
  height = 0x12C = 300
  bitDepth = 0x8 = 8
  colorType = TRUECOLOR (0x2 = 2)
  compressionMethod = 0x0 = 0
  filterMethod = 0x0 = 0
  interlaceMethod = 0x0 = 0
ihdrCrc = [246, 31, 25, 34]
chunks
  0 [Chunk]
  1 [Chunk]
```

hex viewer

```
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   0123456789ABCDEF
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  .PNG........IHDR
00000010  00 00 01 2c 00 00 01 2c  08 02 00 00 00 f6 1f 19  ...,...,.....ö..
00000020  22 00 00 03 97 49 44 41  54 78 9c ed d9 31 8a c3  "....IDATx.íÙ1.Ã
00000030  40 14 44 c1 1e e3 fb 5f  59 8a 9d 09 1c bc 40 55  @.DÁ.ãû_Y....¼@U
00000040  6c b4 20 70 f2 68 98 7f  b6 6b bb ce ef df b6 f3  l´ pòh..¶k»Îïß¶ó
00000050  e8 9f f3 ad 6f 7d fb e7  b7 9f 01 a9 ef 4e fd 13  è.ó.o}ûç·..©ïNý.
00000060  e0 dd 44 08 31 11 42 4c  84 10 13 21 c4 bc 8e 42  àÝD.1.BL...!Ä¼.B
00000070  cc 12 42 4c 84 10 13 21  c4 44 08 31 11 42 4c 84  Ì.BL...!ÄD.1.BL.
00000080  10 73 a2 80 98 25 84 98  08 21 26 42 88 89 10 62  .s¢..%...!&B...b
00000090  22 84 98 08 21 e6 44 01  31 4b 08 31 11 42 4c 84  "...!æD.1K.1.BL.
000000a0  10 13 21 c4 44 08 31 af  a3 10 b3 84 10 13 21 c4  ..!ÄD.1¯£.³...!Ä
000000b0  44 08 31 11 42 4c 84 10  13 21 c4 9c 28 20 66 09  D.1.BL...!Ä.( f.
000000c0  21 26 42 88 89 10 62 22  84 98 08 21 26 42 88 39  !&B...b"...!&B.9
000000d0  51 40 cc 12 42 4c 84 10  13 21 c4 44 08 31 11 42  Q@Ì.BL...!ÄD.1.B
000000e0  cc eb 28 c4 2c 21 c4 44  08 31 11 42 4c 84 10 13  Ìë(Ä,!ÄD.1.BL...
```

info panel

selection: 0x14 - 0x17

Selection length: 4

□ disable lazy parsing

Unparsed parts: << - / 0 >>

Byte arrays: << - / 1 >>

Selected: ihdr/height

export to JSON (hex)

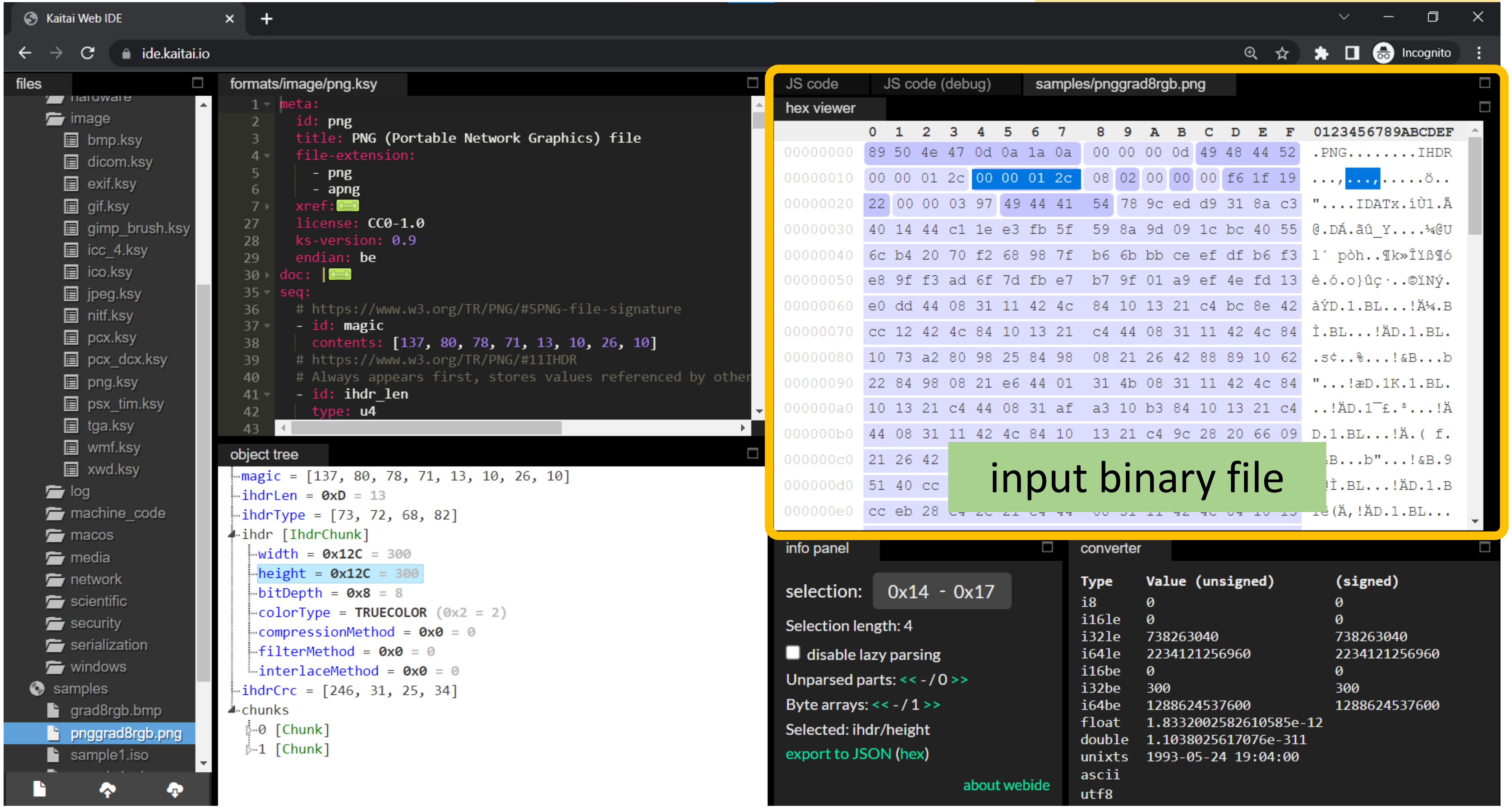
about webide

converter

```
Type    Value (unsigned)          (signed)
i8      0                         0
i16le   0                         0
i32le   738263040                 738263040
i64le   2234121256960             2234121256960
i16be   0                         0
i32be   300                       300
i64be   1288624537600             1288624537600
float   1.8332002582610585e-12
double  1.1038025617076e-311
unixts  1993-05-24 19:04:00
ascii
utf8
```

→ visualization and dumping tools

Web IDE (ide.kaitai.io)



```
hardware
image
  bmp.ksy
  dicom.ksy
  exif.ksy
  gif.ksy
  gimp_brush.ksy
  icc_4.ksy
  ico.ksy
  jpeg.ksy
  nitf.ksy
  pcx.ksy
  pcx_dcx.ksy
  png.ksy
  psx_tim.ksy
  tga.ksy
  wmf.ksy
  xwd.ksy
log
machine_code
macos
media
network
scientific
security
serialization
windows
samples
  grad8rgb.bmp
  pnggrad8rgb.png
  sample1.iso
```

```yaml
meta:
  id: png
  title: PNG (Portable Network Graphics) file
  file-extension:
    - png
    - apng
  xref:
  license: CC0-1.0
  ks-version: 0.9
  endian: be
doc: |
seq:
  # https://www.w3.org/TR/PNG/#5PNG-file-signature
  - id: magic
    contents: [137, 80, 78, 71, 13, 10, 26, 10]
  # https://www.w3.org/TR/PNG/#11IHDR
  # Always appears first, stores values referenced by other
  - id: ihdr_len
    type: u4
```

```
magic = [137, 80, 78, 71, 13, 10, 26, 10]
ihdrLen = 0xD = 13
ihdrType = [73, 72, 68, 82]
ihdr [IhdrChunk]
  width = 0x12C = 300
  height = 0x12C = 300
  bitDepth = 0x8 = 8
  colorType = TRUECOLOR (0x2 = 2)
  compressionMethod = 0x0 = 0
  filterMethod = 0x0 = 0
  interlaceMethod = 0x0 = 0
ihdrCrc = [246, 31, 25, 34]
chunks
  0 [Chunk]
  1 [Chunk]
```

```
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F  0123456789ABCDEF
00000000 89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  .PNG........IHDR
00000010 00 00 01 2c 00 00 01 2c  08 02 00 00 00 f6 1f 19  ...,...,.....ö..
00000020 22 00 00 03 97 49 44 41  54 78 9c ed d9 31 8a c3  "....IDATx.íÙ1.Ã
00000030 40 14 44 c1 1e e3 fb 5f  59 8a 9d 09 1c bc 40 55  @.DÁ.ãû_Y....¼@U
00000040 6c b4 20 70 f2 68 98 7f  b6 6b bb ce ef df b6 f3  l´ pòh..¶k»Îïß¶ó
00000050 e8 9f f3 ad 6f 7d fb e7  b7 9f 01 a9 ef 4e fd 13  è.ó.o}ûç·..©ïNý.
00000060 e0 dd 44 08 31 11 42 4c  84 10 13 21 c4 bc 8e 42  àÝD.1.BL...!Ä¼.B
00000070 cc 12 42 4c 84 10 13 21  c4 44 08 31 11 42 4c 84  Ì.BL...!ÄD.1.BL.
00000080 10 73 a2 80 98 25 84 98  08 21 26 42 88 89 10 62  .s¢..%...!&B...b
00000090 22 84 98 08 21 e6 44 01  31 4b 08 31 11 42 4c 84  "...!æD.1K.1.BL.
000000a0 10 13 21 c4 44 08 31 af  a3 10 b3 84 10 13 21 c4  ..!ÄD.1¯£.³...!Ä
000000b0 44 08 31 11 42 4c 84 10  13 21 c4 9c 28 20 66 09  D.1.BL...!Ä.( f.
000000c0 21 26 42                                           B...b"...!&B.9
000000d0 51 40 cc                                           Ì.BL...!ÄD.1.B
000000e0 cc eb 28                                           ë(Ä,!ÄD.1.BL...
```

input binary file

selection: 0x14 - 0x17

Selection length: 4

☐ disable lazy parsing

Unparsed parts: << - / 0 >>

Byte arrays: << - / 1 >>

Selected: ihdr/height

export to JSON (hex)

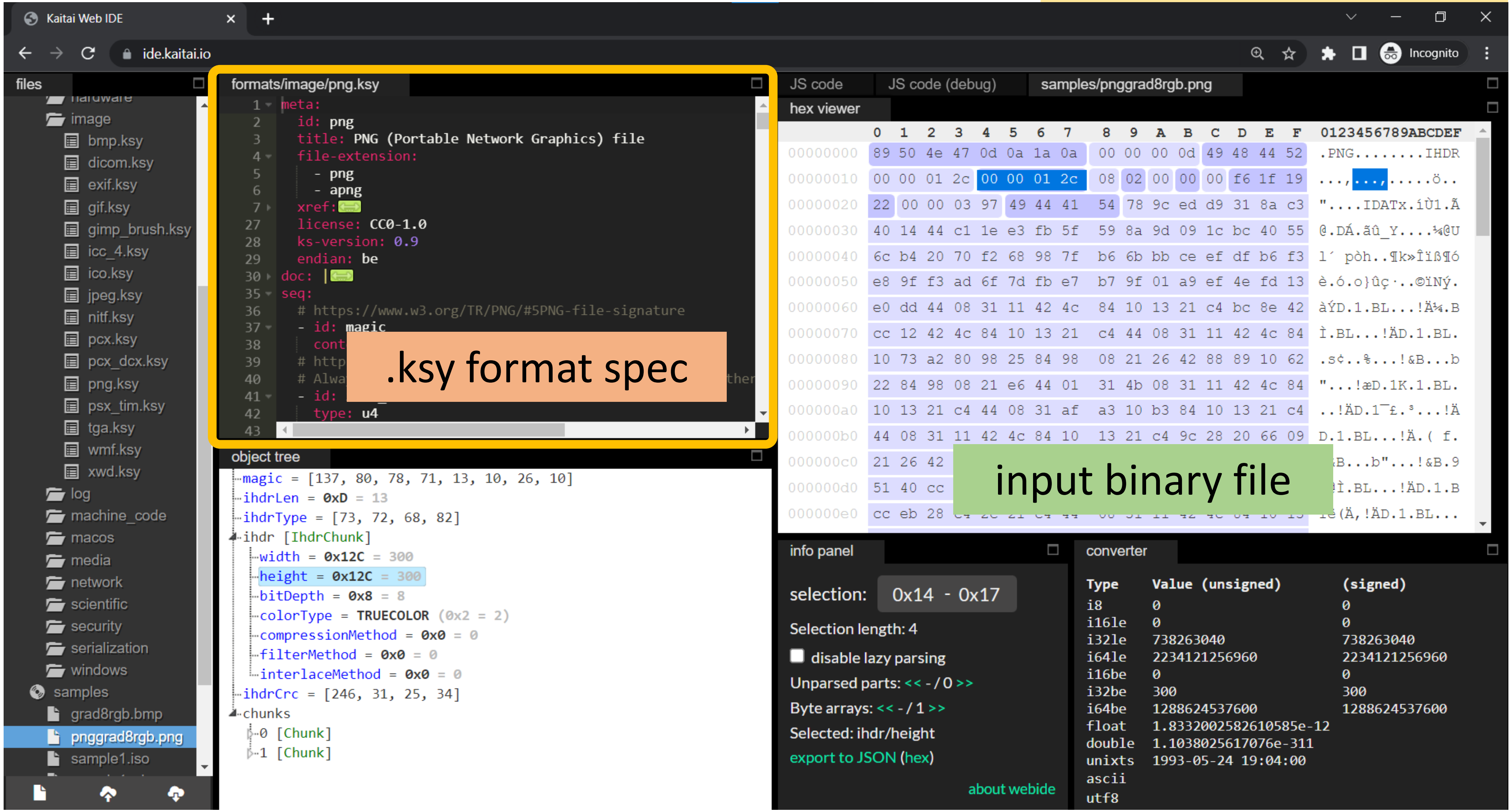
about webide

```
Type    Value (unsigned)         (signed)
i8      0                        0
i16le   0                        0
i32le   738263040                738263040
i64le   2234121256960            2234121256960
i16be   0                        0
i32be   300                      300
i64be   1288624537600            1288624537600
float   1.8332002582610585e-12
double  1.1038025617076e-311
unixts  1993-05-24 19:04:00
ascii
utf8
```

Why Kaitai

→ visualization and dumping tools

Web IDE (ide.kaitai.io)



formats/image/png.ksy

```
meta:
  id: png
  title: PNG (Portable Network Graphics) file
  file-extension:
    - png
    - apng
  xref:
  license: CC0-1.0
  ks-version: 0.9
  endian: be
doc: |
seq:
  # https://www.w3.org/TR/PNG/#5PNG-file-signature
  - id: magic
    cont
  # http
  # Alwa                                          ther
  - id:
    type: u4
```

.ksy format spec

object tree

```
magic = [137, 80, 78, 71, 13, 10, 26, 10]
ihdrLen = 0xD = 13
ihdrType = [73, 72, 68, 82]
ihdr [IhdrChunk]
  width = 0x12C = 300
  height = 0x12C = 300
  bitDepth = 0x8 = 8
  colorType = TRUECOLOR (0x2 = 2)
  compressionMethod = 0x0 = 0
  filterMethod = 0x0 = 0
  interlaceMethod = 0x0 = 0
ihdrCrc = [246, 31, 25, 34]
chunks
  0 [Chunk]
  1 [Chunk]
```

hex viewer

```
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   0123456789ABCDEF
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  .PNG........IHDR
00000010  00 00 01 2c 00 00 01 2c  08 02 00 00 00 f6 1f 19  ...,...,.....ö..
00000020  22 00 00 03 97 49 44 41  54 78 9c ed d9 31 8a c3  "....IDATx.íÙ1.Ã
00000030  40 14 44 c1 1e e3 fb 5f  59 8a 9d 09 1c bc 40 55  @.DÁ.ãû_Y....¼@U
00000040  6c b4 20 70 f2 68 98 7f  b6 6b bb ce ef df b6 f3  l´ pòh..¶k»Îïß¶ó
00000050  e8 9f f3 ad 6f 7d fb e7  b7 9f 01 a9 ef 4e fd 13  è.ó.o}ûç·..©ïNý.
00000060  e0 dd 44 08 31 11 42 4c  84 10 13 21 c4 bc 8e 42  àÝD.1.BL...!Ä¼.B
00000070  cc 12 42 4c 84 10 13 21  c4 44 08 31 11 42 4c 84  Ì.BL...!ÄD.1.BL.
00000080  10 73 a2 80 98 25 84 98  08 21 26 42 88 89 10 62  .s¢..%...!&B...b
00000090  22 84 98 08 21 e6 44 01  31 4b 08 31 11 42 4c 84  "...!æD.1K.1.BL.
000000a0  10 13 21 c4 44 08 31 af  a3 10 b3 84 10 13 21 c4  ..!ÄD.1¯£.³...!Ä
000000b0  44 08 31 11 42 4c 84 10  13 21 c4 9c 28 20 66 09  D.1.BL...!Ä.( f.
000000c0  21 26 42                                           B...b"...!&B.9
000000d0  51 40 cc                                           Ì.BL...!ÄD.1.B
000000e0  cc eb 28                                           (Ä,!ÄD.1.BL...
```

input binary file

info panel

selection: 0x14 - 0x17

Selection length: 4

□ disable lazy parsing

Unparsed parts: << - / 0 >>

Byte arrays: << - / 1 >>

Selected: ihdr/height

export to JSON (hex)

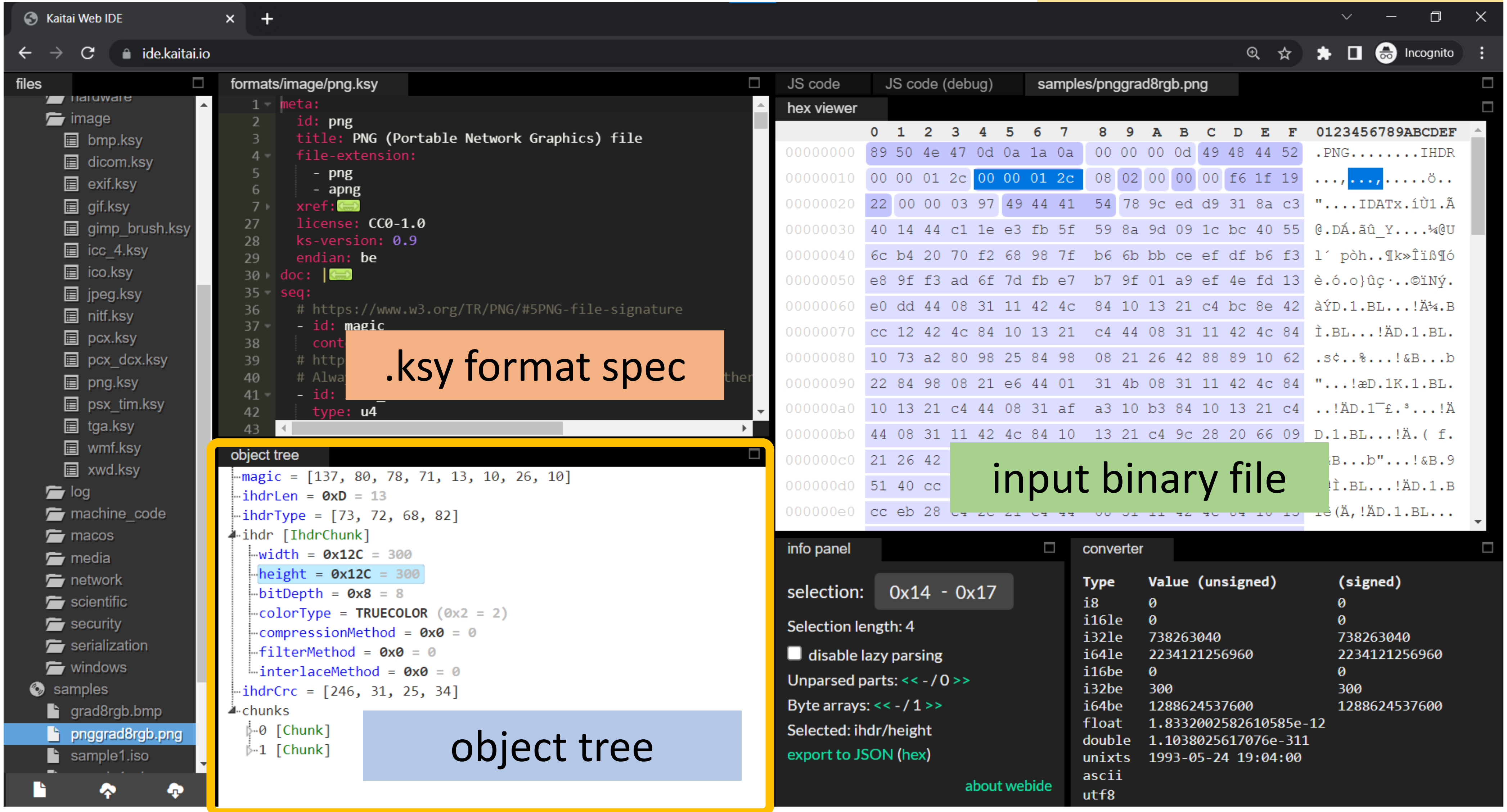
converter

```
Type    Value (unsigned)         (signed)
i8      0                        0
i16le   0                        0
i32le   738263040                738263040
i64le   2234121256960            2234121256960
i16be   0                        0
i32be   300                      300
i64be   1288624537600            1288624537600
float   1.8332002582610585e-12
double  1.1038025617076e-311
unixts  1993-05-24 19:04:00
ascii
utf8
```

Why Kaitai

→ visualization and dumping tools

Web IDE (ide.kaitai.io)



files:

- hardware
- image
  - bmp.ksy
  - dicom.ksy
  - exif.ksy
  - gif.ksy
  - gimp_brush.ksy
  - icc_4.ksy
  - ico.ksy
  - jpeg.ksy
  - nitf.ksy
  - pcx.ksy
  - pcx_dcx.ksy
  - png.ksy
  - psx_tim.ksy
  - tga.ksy
  - wmf.ksy
  - xwd.ksy
- log
- machine_code
- macos
- media
- network
- scientific
- security
- serialization
- windows
- samples
  - grad8rgb.bmp
  - pnggrad8rgb.png
  - sample1.iso

```
meta:
  id: png
  title: PNG (Portable Network Graphics) file
  file-extension:
    - png
    - apng
  xref: ⟺
  license: CC0-1.0
  ks-version: 0.9
  endian: be
doc: |⟺
seq:
  # https://www.w3.org/TR/PNG/#5PNG-file-signature
  - id: magic
    cont
  # http
  # Alwa                                          ther
  - id:
    type: u4
```

.ksy format spec

```
    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F  0123456789ABCDEF
00000000 89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  .PNG........IHDR
00000010 00 00 01 2c 00 00 01 2c  08 02 00 00 00 f6 1f 19  ...,...,.....ö..
00000020 22 00 00 03 97 49 44 41  54 78 9c ed d9 31 8a c3  "....IDATx.íÙ1.Ã
00000030 40 14 44 c1 1e e3 fb 5f  59 8a 9d 09 1c bc 40 55  @.DÁ.ãû_Y....¼@U
00000040 6c b4 20 70 f2 68 98 7f  b6 6b bb ce ef df b6 f3  l´ pòh..¶k»ÎïßΙó
00000050 e8 9f f3 ad 6f 7d fb e7  b7 9f 01 a9 ef 4e fd 13  è.ó.o}ûç·..©ïNý.
00000060 e0 dd 44 08 31 11 42 4c  84 10 13 21 c4 bc 8e 42  àÝD.1.BL...!Ä¼.B
00000070 cc 12 42 4c 84 10 13 21  c4 44 08 31 11 42 4c 84  Ì.BL...!ÄD.1.BL.
00000080 10 73 a2 80 98 25 84 98  08 21 26 42 88 89 10 62  .s¢..%...!&B...b
00000090 22 84 98 08 21 e6 44 01  31 4b 08 31 11 42 4c 84  "...!æD.1K.1.BL.
000000a0 10 13 21 c4 44 08 31 af  a3 10 b3 84 10 13 21 c4  ..!ÄD.1¯£.³...!Ä
000000b0 44 08 31 11 42 4c 84 10  13 21 c4 9c 28 20 66 09  D.1.BL...!Ä.( f.
000000c0 21 26 42                                          B...b"...!&B.9
000000d0 51 40 cc                                          Ì.BL...!ÄD.1.B
000000e0 cc eb 28                                          (Ä,!ÄD.1.BL...
```

input binary file

```
magic = [137, 80, 78, 71, 13, 10, 26, 10]
ihdrLen = 0xD = 13
ihdrType = [73, 72, 68, 82]
ihdr [IhdrChunk]
  width = 0x12C = 300
  height = 0x12C = 300
  bitDepth = 0x8 = 8
  colorType = TRUECOLOR (0x2 = 2)
  compressionMethod = 0x0 = 0
  filterMethod = 0x0 = 0
  interlaceMethod = 0x0 = 0
ihdrCrc = [246, 31, 25, 34]
chunks
  0 [Chunk]
  1 [Chunk]
```

object tree

selection: 0x14 - 0x17

Selection length: 4

□ disable lazy parsing

Unparsed parts: << - / 0 >>

Byte arrays: << - / 1 >>

Selected: ihdr/height

export to JSON (hex)

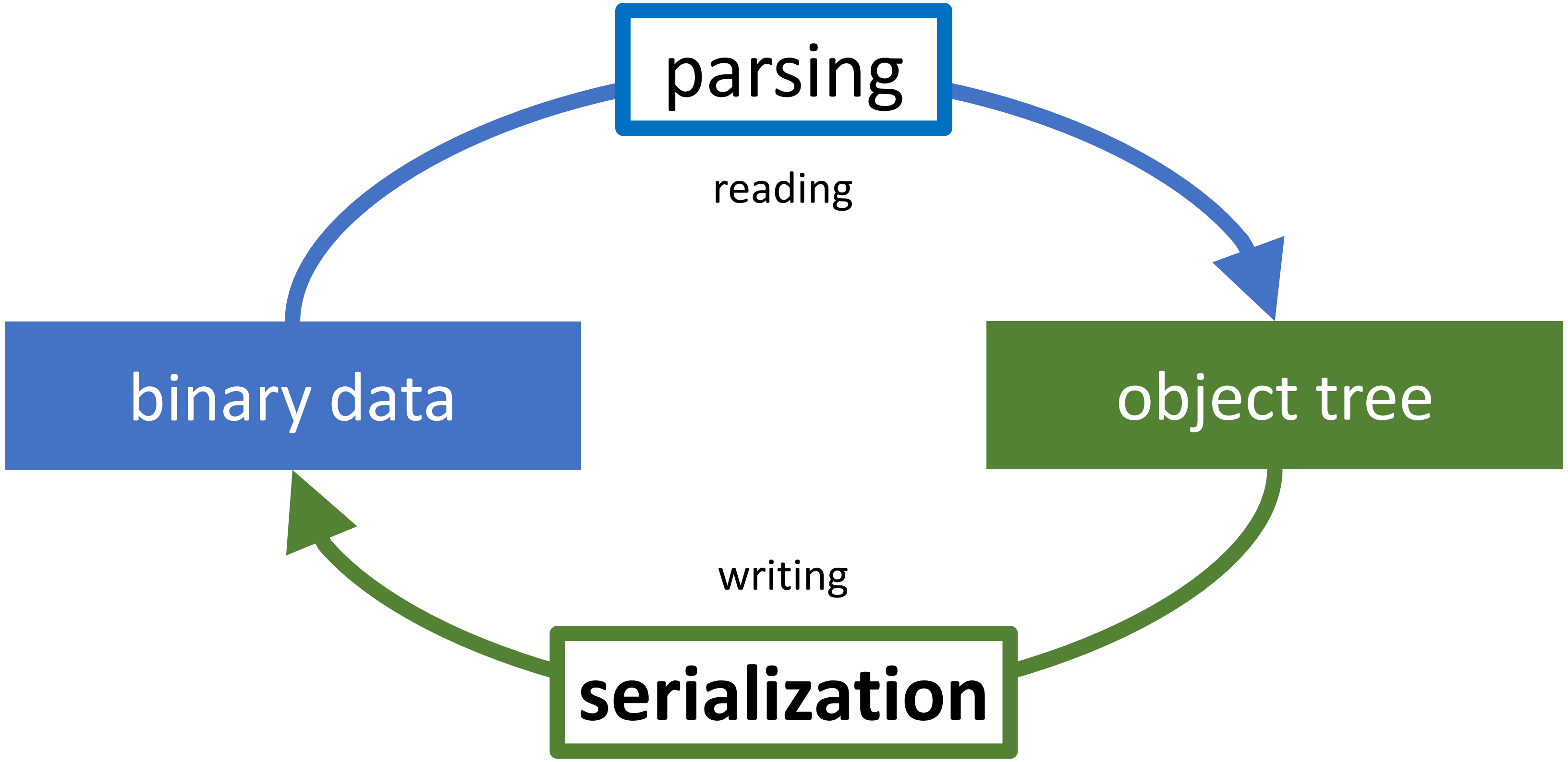
about webide

```
Type     Value (unsigned)          (signed)
i8       0                         0
i16le    0                         0
i32le    738263040                 738263040
i64le    2234121256960             2234121256960
i16be    0                         0
i32be    300                       300
i64be    1288624537600             1288624537600
float    1.8332002582610585e-12
double   1.1038025617076e-311
unixts   1993-05-24 19:04:00
ascii
utf8
```

Serialization

financially supported by NLnet Foundation

Serialization



fully working in Java,
C# and Python in development

Serialization

→ use cases

1. editing an existing file
2. creating a new file

→ areas of application

- format conversions (parse, transform, serialize to another format)
- fuzzing
- video games modding

Serialization

→ use cases

1. editing an existing file
2. creating a new file

→ areas of application

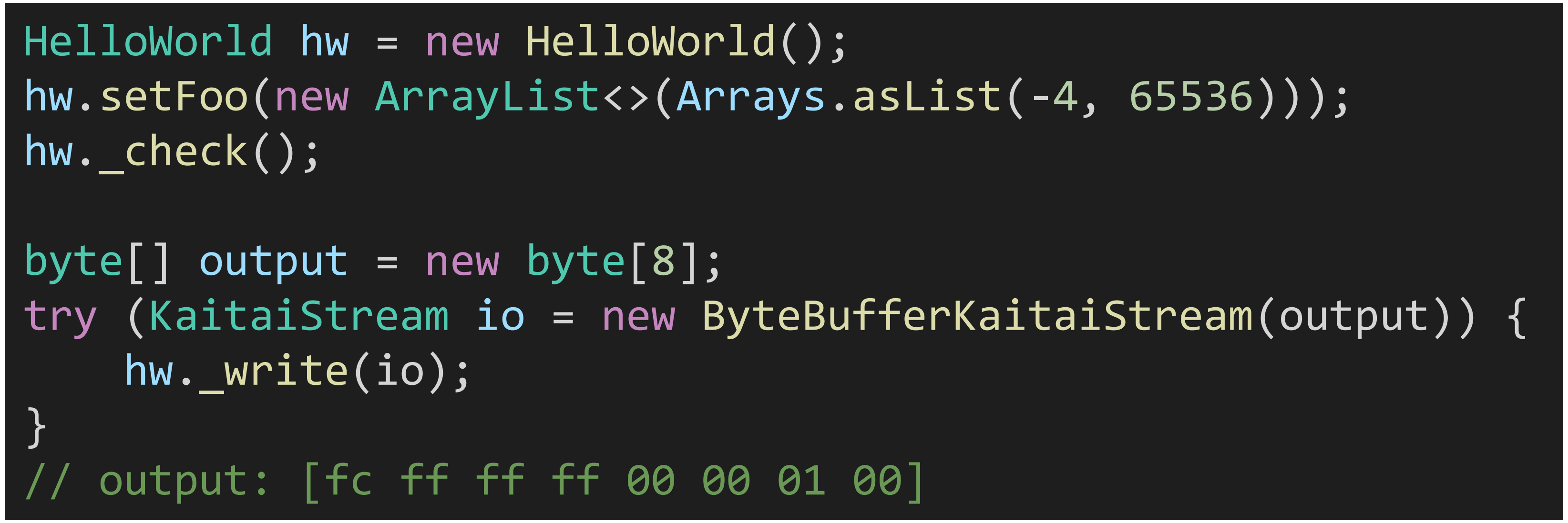
- format conversions (parse, transform, serialize to another format)
- fuzzing
- video games modding

Serialization

→ phases

More details at https://doc.kaitai.io/serialization.html

1. create a KS object



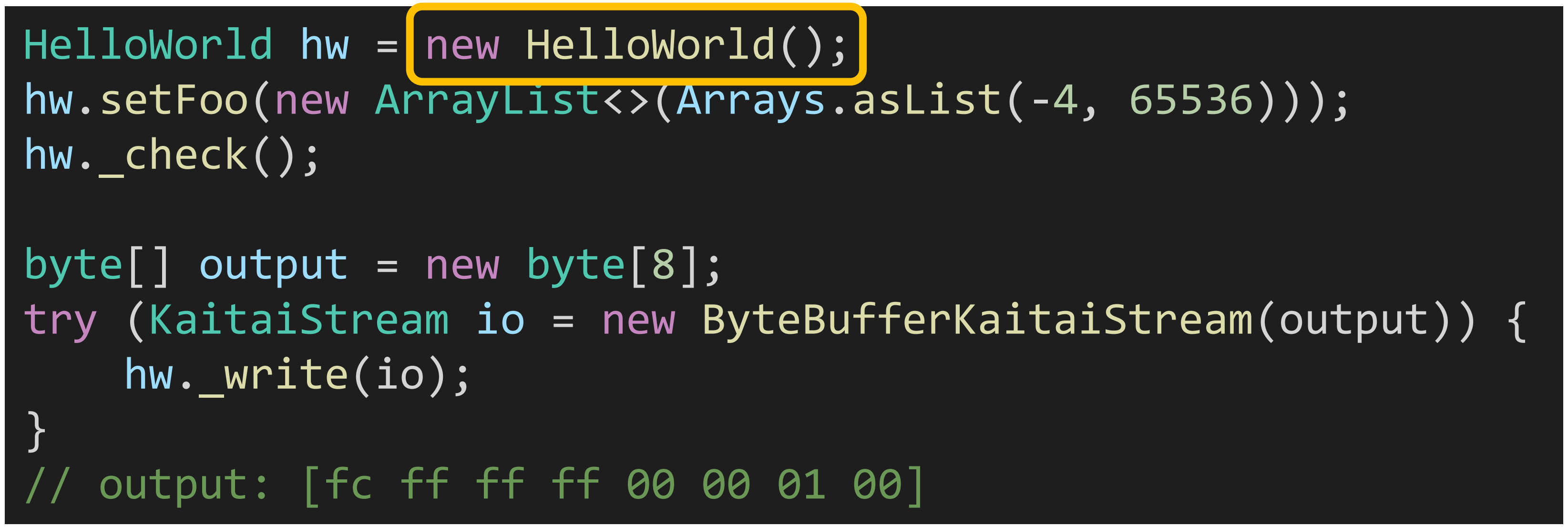
```
HelloWorld hw = new HelloWorld();
hw.setFoo(new ArrayList<>(Arrays.asList(-4, 65536)));
hw._check();

byte[] output = new byte[8];
try (KaitaiStream io = new ByteBufferKaitaiStream(output)) {
    hw._write(io);
}
// output: [fc ff ff ff 00 00 01 00]
```

Serialization

→ phases

More details at https://doc.kaitai.io/serialization.html

1. create a KS object



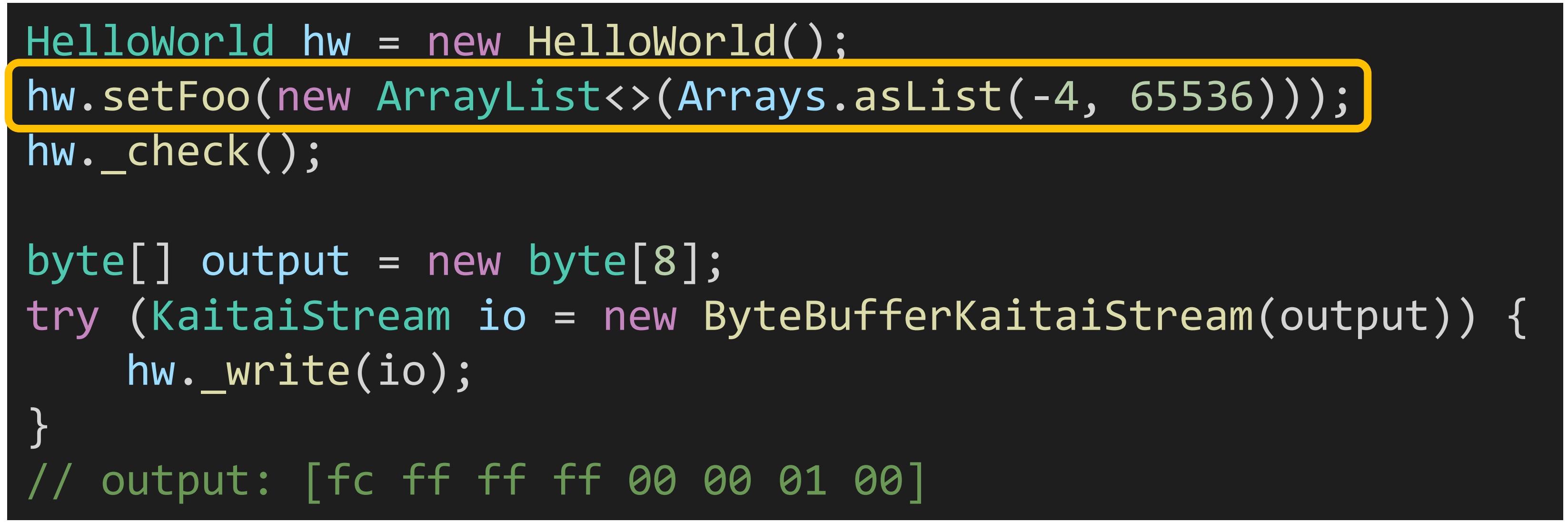
```java
HelloWorld hw = new HelloWorld();
hw.setFoo(new ArrayList<>(Arrays.asList(-4, 65536)));
hw._check();

byte[] output = new byte[8];
try (KaitaiStream io = new ByteBufferKaitaiStream(output)) {
    hw._write(io);
}
// output: [fc ff ff ff 00 00 01 00]
```

Serialization

→ phases

More details at https://doc.kaitai.io/serialization.html

1. create a KS object
2. set the object fields



```java
HelloWorld hw = new HelloWorld();
hw.setFoo(new ArrayList<>(Arrays.asList(-4, 65536)));
hw._check();

byte[] output = new byte[8];
try (KaitaiStream io = new ByteBufferKaitaiStream(output)) {
    hw._write(io);
}
// output: [fc ff ff ff 00 00 01 00]
```

Serialization

→ phases

More details at https://doc.kaitai.io/serialization.html

1. create a KS object
2. set the object fields
3. call _check on each KS object

```
HelloWorld hw = new HelloWorld();
hw.setFoo(new ArrayList<>(Arrays.asList(-4, 65536)));
hw._check();

byte[] output = new byte[8];
try (KaitaiStream io = new ByteBufferKaitaiStream(output)) {
    hw._write(io);
}
// output: [fc ff ff ff 00 00 01 00]
```

Serialization

→ phases

More details at https://doc.kaitai.io/serialization.html

1. create a KS object
2. set the object fields
3. call _check on each KS object
4. call _write on top-level object

```
HelloWorld hw = new HelloWorld();
hw.setFoo(new ArrayList<>(Arrays.asList(-4, 65536)));
hw._check();

byte[] output = new byte[8];
try (KaitaiStream io = new ByteBufferKaitaiStream(output)) {
    hw._write(io);
}
// output: [fc ff ff ff 00 00 01 00]
```

Serialization

→ Current scope of serialization support

- designed for the general case, not average
- user must set **everything** (including lengths, offsets, magic signatures), KS checks consistency
- only fixed-length streams
- value instances have no setters (instead, change inputs and **invalidate**)

Future plans

Future plans

→ serialization for C#, Python

→ add target languages: Rust, C, Julia

→ support Wireshark dissectors as target

Thanks!

https://kaitai.io/

https://github.com/kaitai-io

https://gitter.im/kaitai_struct/Lobby

@kaitai_io