Data Serialization

Data serialization is the process of storing a data structure or object state into a format that can be stored (or transmitted) and reconstructed later.

- Wikipedia

Data serialization is just about reading and writing files la

- Gerald Wong

…what is data?

Let’s think about data

What are we loading/saving?

Where are we loading/saving?

How are we loading/saving?

Why are we loading/saving?

Examples?

Non-exhaustive uses of data serialization

- Image files (jpeg, png, bmp)
- Game save files/custom data files (csv? json? txt?)
- Sound files (wav, mp3, ogg)
- Video files (mp4, avi, mkv, flv)
- Web pages (html, js, css)
- Web services (databases)
- Password storage

How about these?

- .exe files
- .sln files
- .c files
- .zip files
- directories? File systems?

DATA
DATA EVERYWHERE
imgflip.com

But why?

Data serialization is useful because you separate data away from the program for later consumption

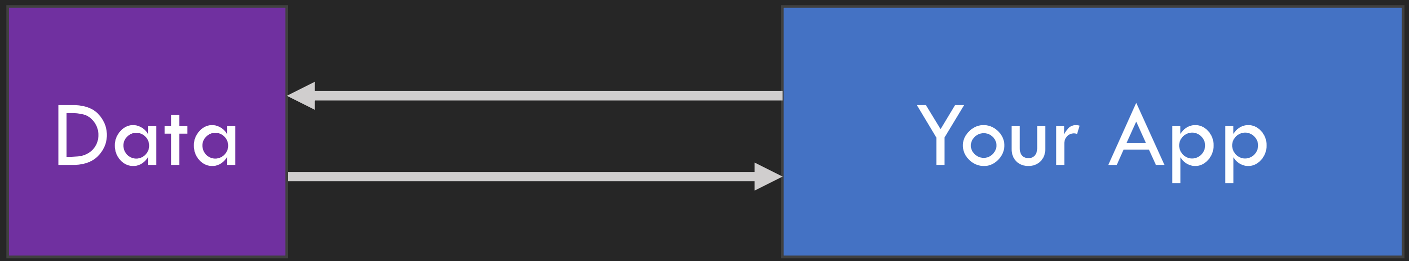
(by itself or even other apps!)

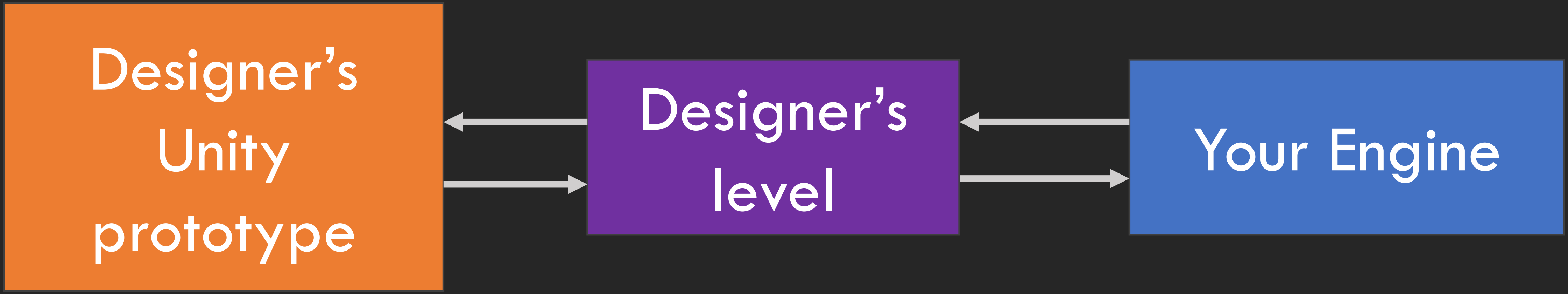
One advantage is that you don't have to recompile your program every time your data changes!

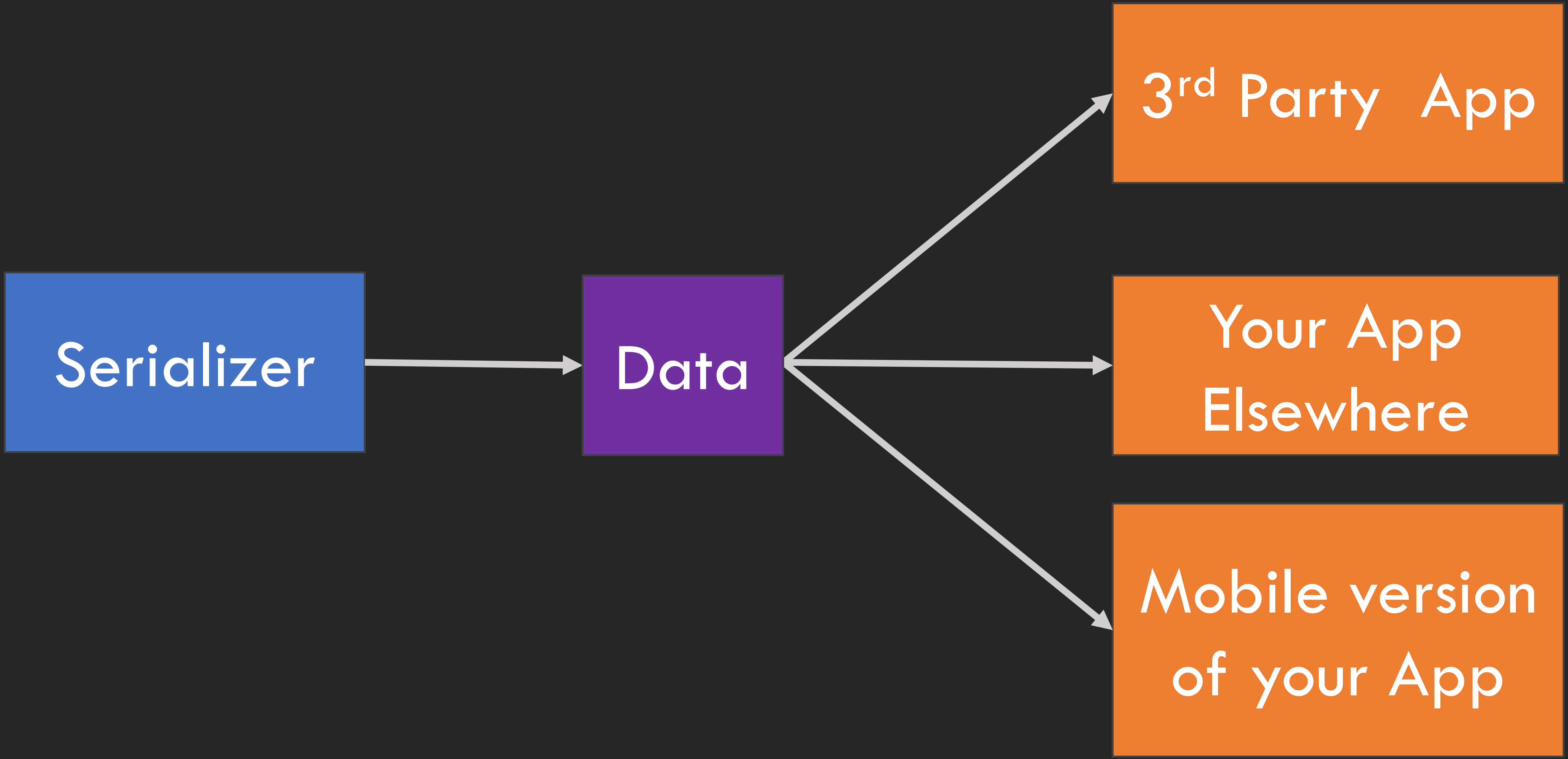
(e.g. position of objects in your game)

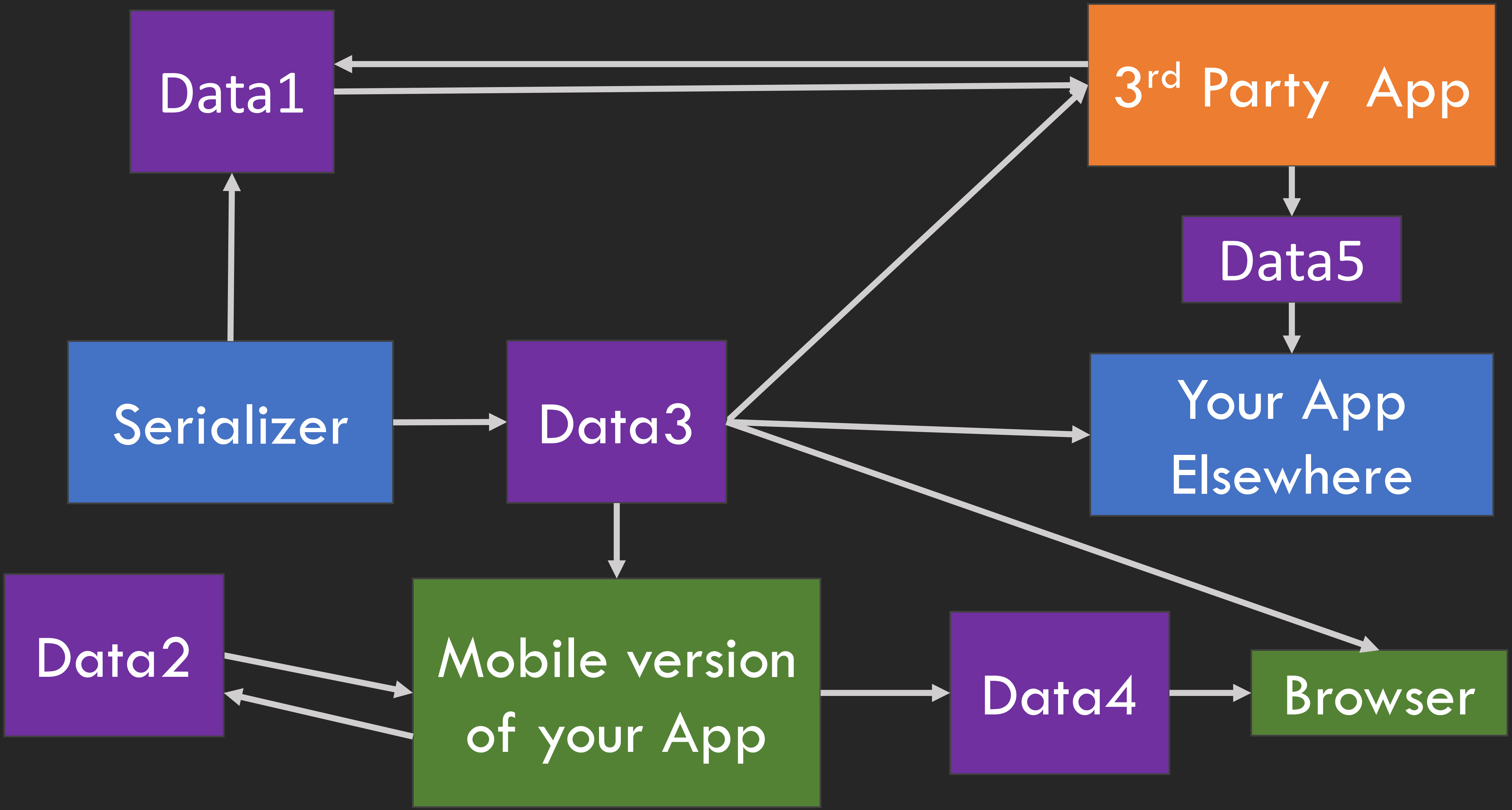
The hard part is figuring out what data should be serialized in your program, how to serialize that data, and how to structure your program to handle the serialization logic.

(Hint: it’s mostly case-by-case)


Data
Your App


Designer's Unity prototype
Designer's level
Your Engine


Serializer
Data
3rd Party App
Your App Elsewhere
Mobile version of your App


Data1
3rd Party App
Data5
Serializer
Data3
Your App Elsewhere
Data2
Mobile version of your App
Data4
Browser

DATA SERIALIZATION
LIGHTYEAR

Data serialization comes down to two main problems:

- How to read the data
- How to write the data

But what IS the data (concretely)?

Before we even talk about reading or writing the data, we must first KNOW the format of the data.

We have two main choices

- Roll your own custom file format.
  - Customized to your needs.
  - Write your own reader/writer.
- Choose an existing file format.
  - Find a format that fits your needs the best along with.
  - Write your own or use an existing reader/writer.

Rolling your own format

- Two main choices:
  - Binary. Easier to parse in code and for computer to read but harder for our human eyes. Not that portable.
  - Text. Easy to read with human eyes but harder to parse in code for computer to read. More portable.

Remember: at the end of the day,
it’s all bits and bytes to the computer ☺

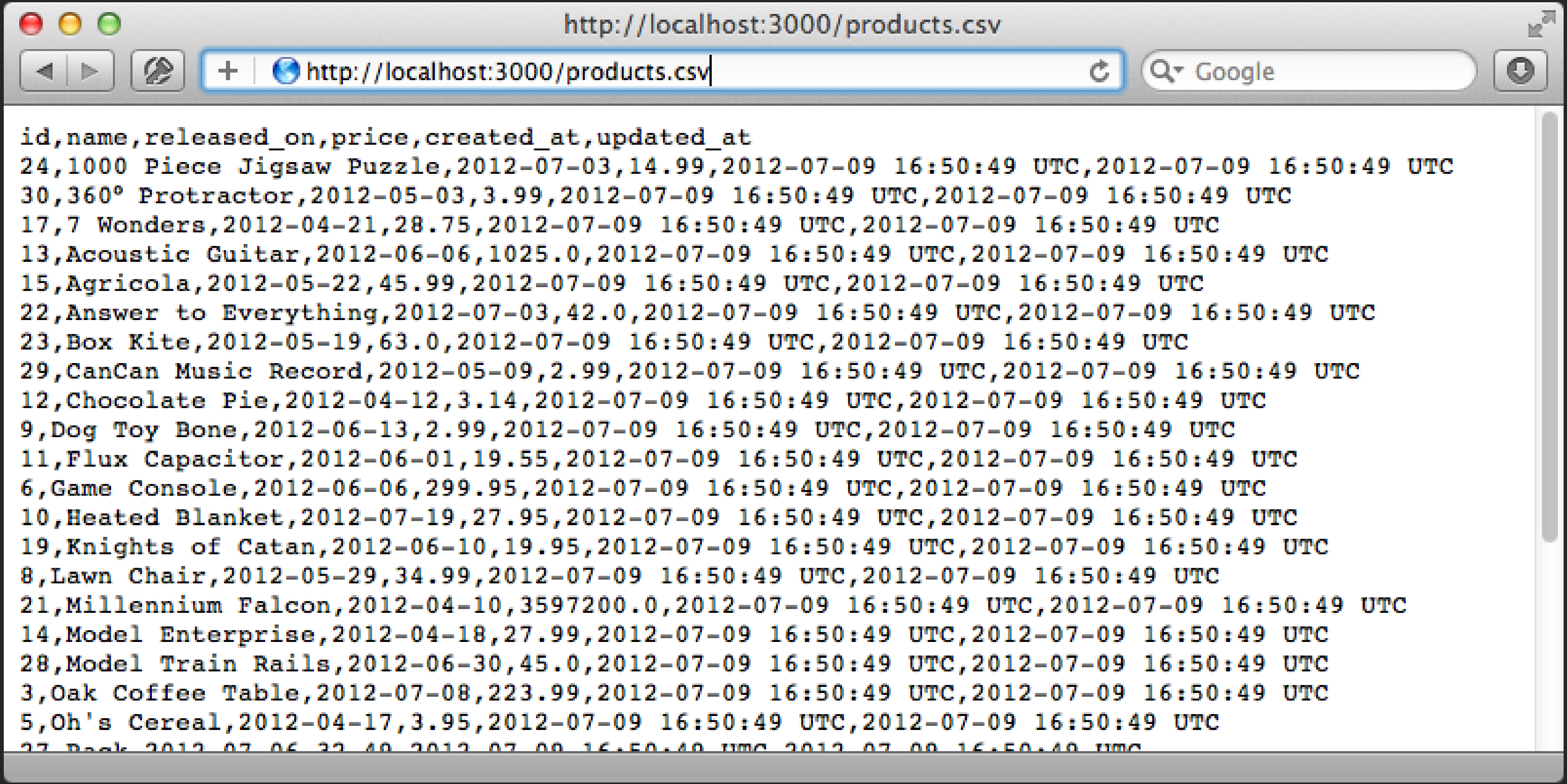
Showcase time!

Demo!

Choosing an existing format

Popular Data Formats

Case 1: CSV



http://localhost:3000/products.csv

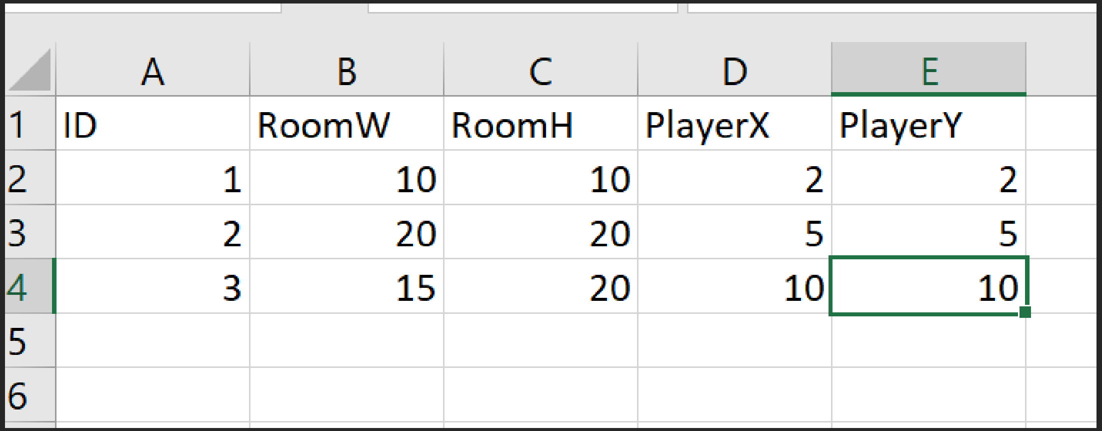
```
id,name,released_on,price,created_at,updated_at
24,1000 Piece Jigsaw Puzzle,2012-07-03,14.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
30,360° Protractor,2012-05-03,3.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
17,7 Wonders,2012-04-21,28.75,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
13,Acoustic Guitar,2012-06-06,1025.0,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
15,Agricola,2012-05-22,45.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
22,Answer to Everything,2012-07-03,42.0,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
23,Box Kite,2012-05-19,63.0,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
29,CanCan Music Record,2012-05-09,2.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
12,Chocolate Pie,2012-04-12,3.14,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
9,Dog Toy Bone,2012-06-13,2.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
11,Flux Capacitor,2012-06-01,19.55,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
6,Game Console,2012-06-06,299.95,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
10,Heated Blanket,2012-07-19,27.95,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
19,Knights of Catan,2012-06-10,19.95,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
8,Lawn Chair,2012-05-29,34.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
21,Millennium Falcon,2012-04-10,3597200.0,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
14,Model Enterprise,2012-04-18,27.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
28,Model Train Rails,2012-06-30,45.0,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
3,Oak Coffee Table,2012-07-08,223.99,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
5,Oh's Cereal,2012-04-17,3.95,2012-07-09 16:50:49 UTC,2012-07-09 16:50:49 UTC
```

Case 1: CSV

- Stands for "Comma-Separated Values"
- 2D table representing data.
- Spreadsheet tools can easily view it
  - Microsoft Excel
  - Google Sheets
- Limited on its own, but possible to represent complex objects via multiple related sheets.

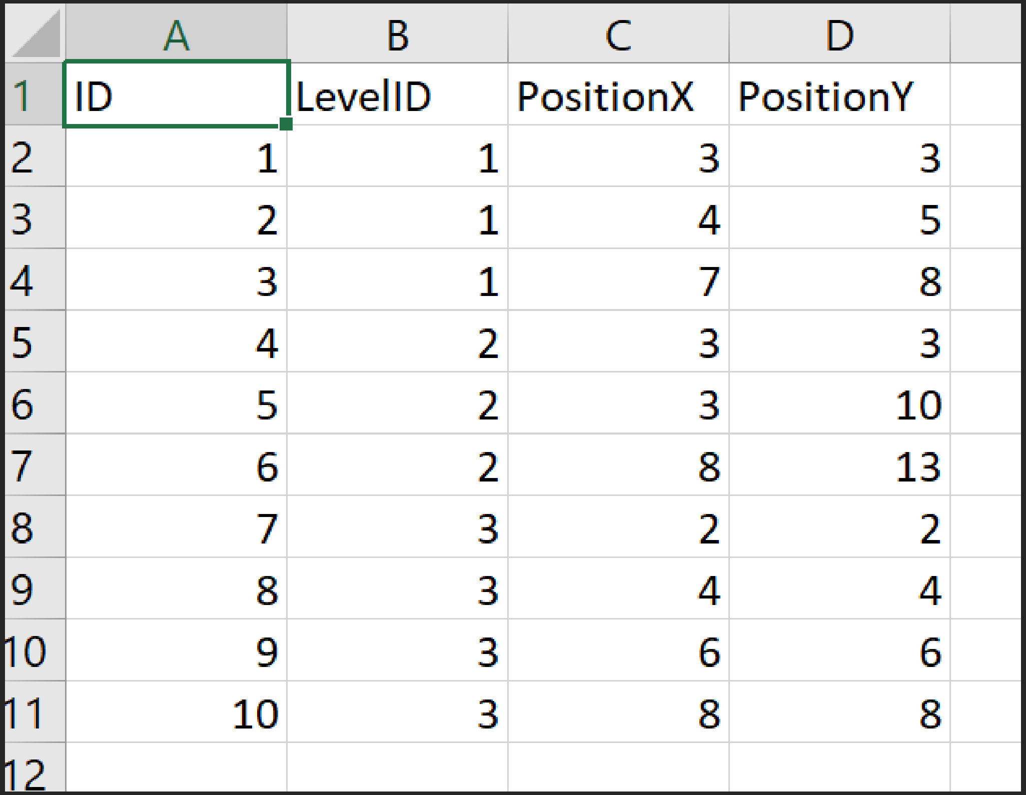
Case 1: CSV

Level.csv



	A	B	C	D	E
1	ID	RoomW	RoomH	PlayerX	PlayerY
2	1	10	10	2	2
3	2	20	20	5	5
4	3	15	20	10	10
5					
6					

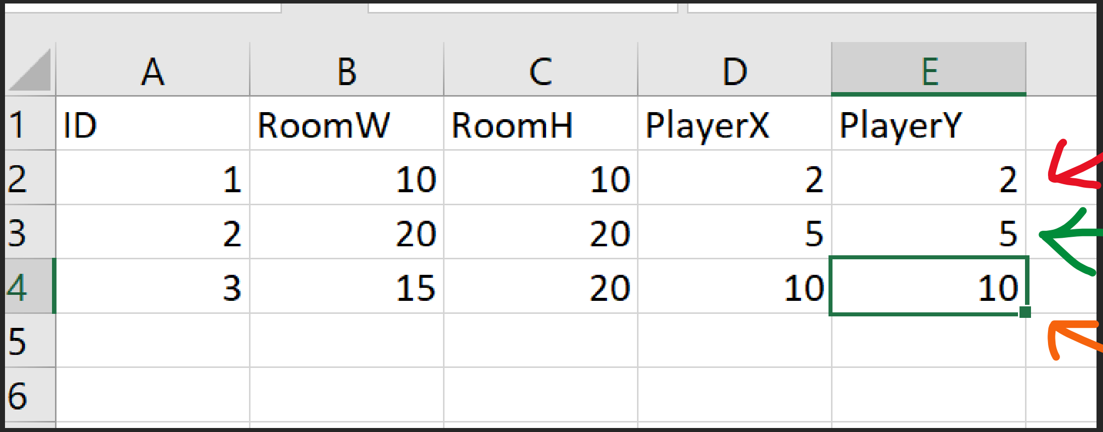
Npc.csv



	A	B	C	D
1	ID	LevelID	PositionX	PositionY
2	1	1	3	3
3	2	1	4	5
4	3	1	7	8
5	4	2	3	3
6	5	2	3	10
7	6	2	8	13
8	7	3	2	2
9	8	3	4	4
10	9	3	6	6
11	10	3	8	8
12				

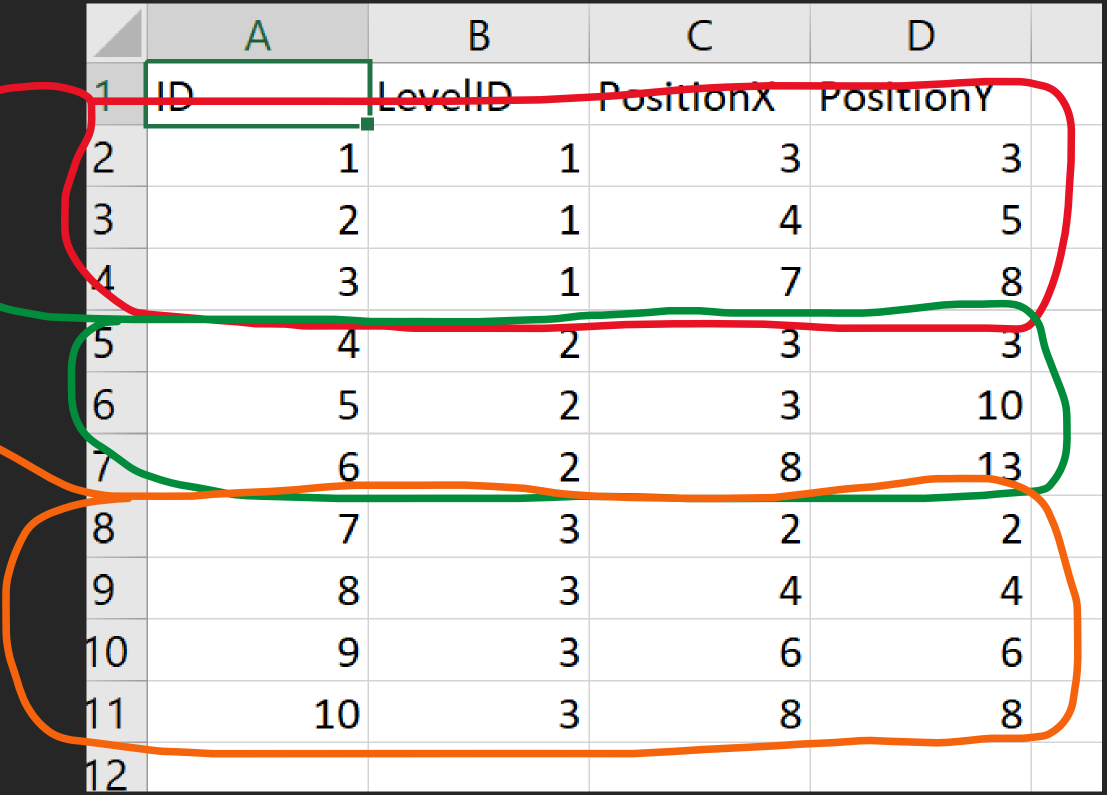
Case 1: CSV

Level.csv



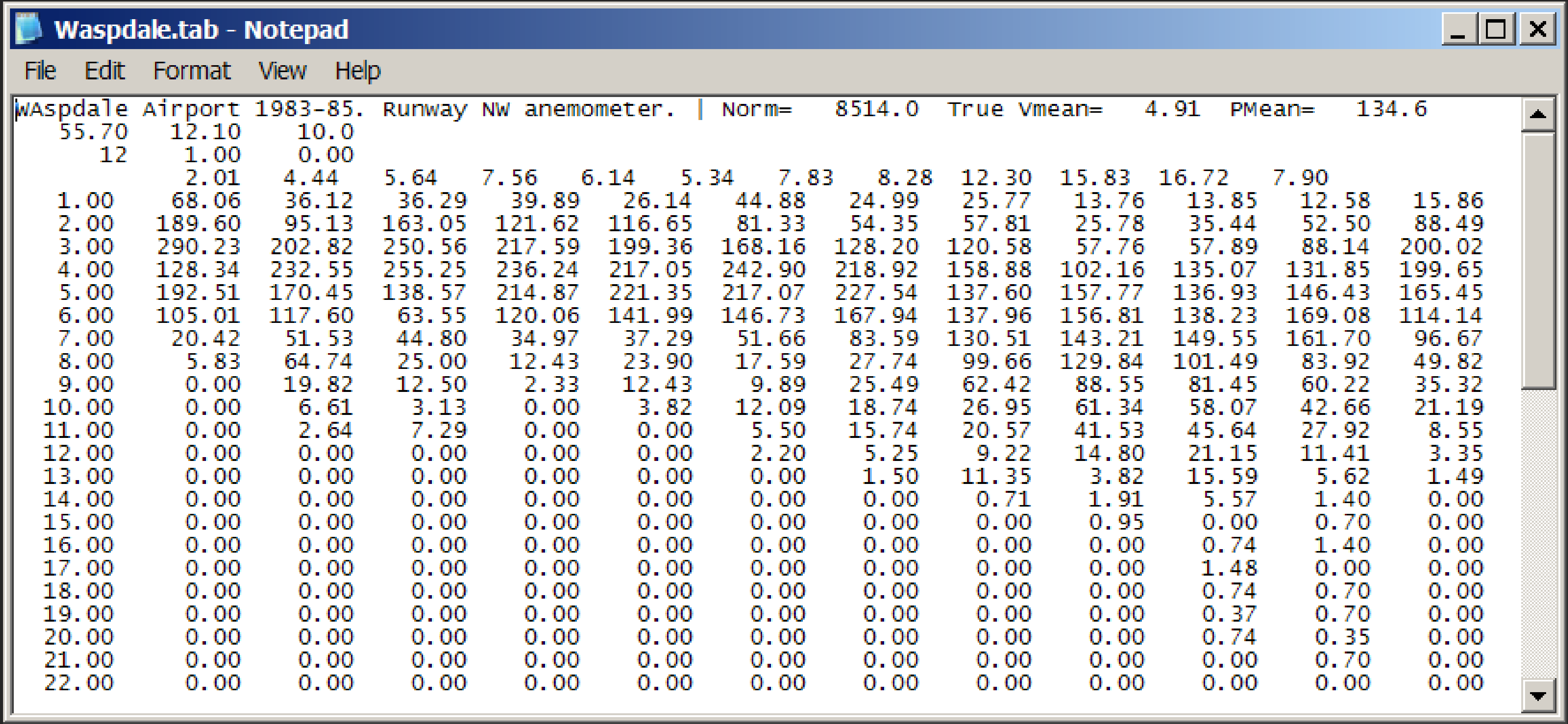
	A	B	C	D	E
1	ID	RoomW	RoomH	PlayerX	PlayerY
2	1	10	10	2	2
3	2	20	20	5	5
4	3	15	20	10	10
5					
6					

Npc.csv



	A	B	C	D
1	ID	LevelID	PositionX	PositionY
2	1	1	3	3
3	2	1	4	5
4	3	1	7	8
5	4	2	3	3
6	5	2	3	10
7	6	2	8	13
8	7	3	2	2
9	8	3	4	4
10	9	3	6	6
11	10	3	8	8
12				

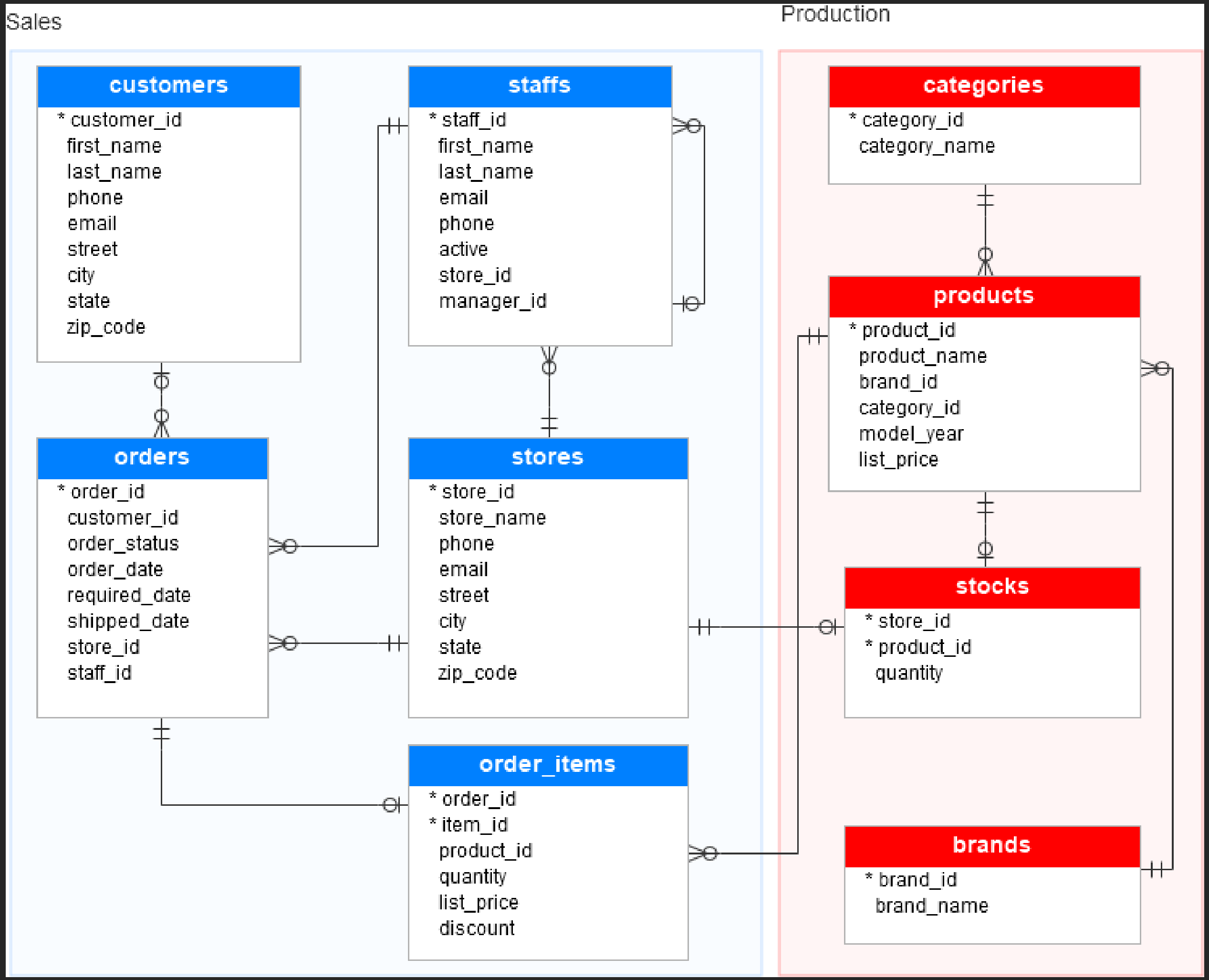
Similar table-style formats



```
WAspdale Airport 1983-85. Runway NW anemometer. | Norm=    8514.0  True Vmean=   4.91  PMean=   134.6
    55.70    12.10     10.0
       12     1.00     0.00
              2.01     4.44     5.64     7.56     6.14     5.34     7.83     8.28    12.30    15.83    16.72    7.90
     1.00    68.06    36.12    36.29    39.89    26.14    44.88    24.99    25.77    13.76    13.85    12.58    15.86
     2.00   189.60    95.13   163.05   121.62   116.65    81.33    54.35    57.81    25.78    35.44    52.50    88.49
     3.00   290.23   202.82   250.56   217.59   199.36   168.16   128.20   120.58    57.76    57.89    88.14   200.02
     4.00   128.34   232.55   255.25   236.24   217.05   242.90   218.92   158.88   102.16   135.07   131.85   199.65
     5.00   192.51   170.45   138.57   214.87   221.35   217.07   227.54   137.60   157.77   136.93   146.43   165.45
     6.00   105.01   117.60    63.55   120.06   141.99   146.73   167.94   137.96   156.81   138.23   169.08   114.14
     7.00    20.42    51.53    44.80    34.97    37.29    51.66    83.59   130.51   143.21   149.55   161.70    96.67
     8.00     5.83    64.74    25.00    12.43    23.90    17.59    27.74    99.66   129.84   101.49    83.92    49.82
     9.00     0.00    19.82    12.50     2.33    12.43     9.89    25.49    62.42    88.55    81.45    60.22    35.32
    10.00     0.00     6.61     3.13     0.00     3.82    12.09    18.74    26.95    61.34    58.07    42.66    21.19
    11.00     0.00     2.64     7.29     0.00     0.00     5.50    15.74    20.57    41.53    45.64    27.92     8.55
    12.00     0.00     0.00     0.00     0.00     0.00     2.20     5.25     9.22    14.80    21.15    11.41     3.35
    13.00     0.00     0.00     0.00     0.00     0.00     0.00     1.50    11.35     3.82    15.59     5.62     1.49
    14.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.71     1.91     5.57     1.40     0.00
    15.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.95     0.00     0.70     0.00
    16.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.74     1.40     0.00
    17.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     1.48     0.00     0.00
    18.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.74     0.70     0.00
    19.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.37     0.70     0.00
    20.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.74     0.35     0.00
    21.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.70     0.00
    22.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
```

TSV

Similar table-style formats



Relational databases (e.g. SQL)

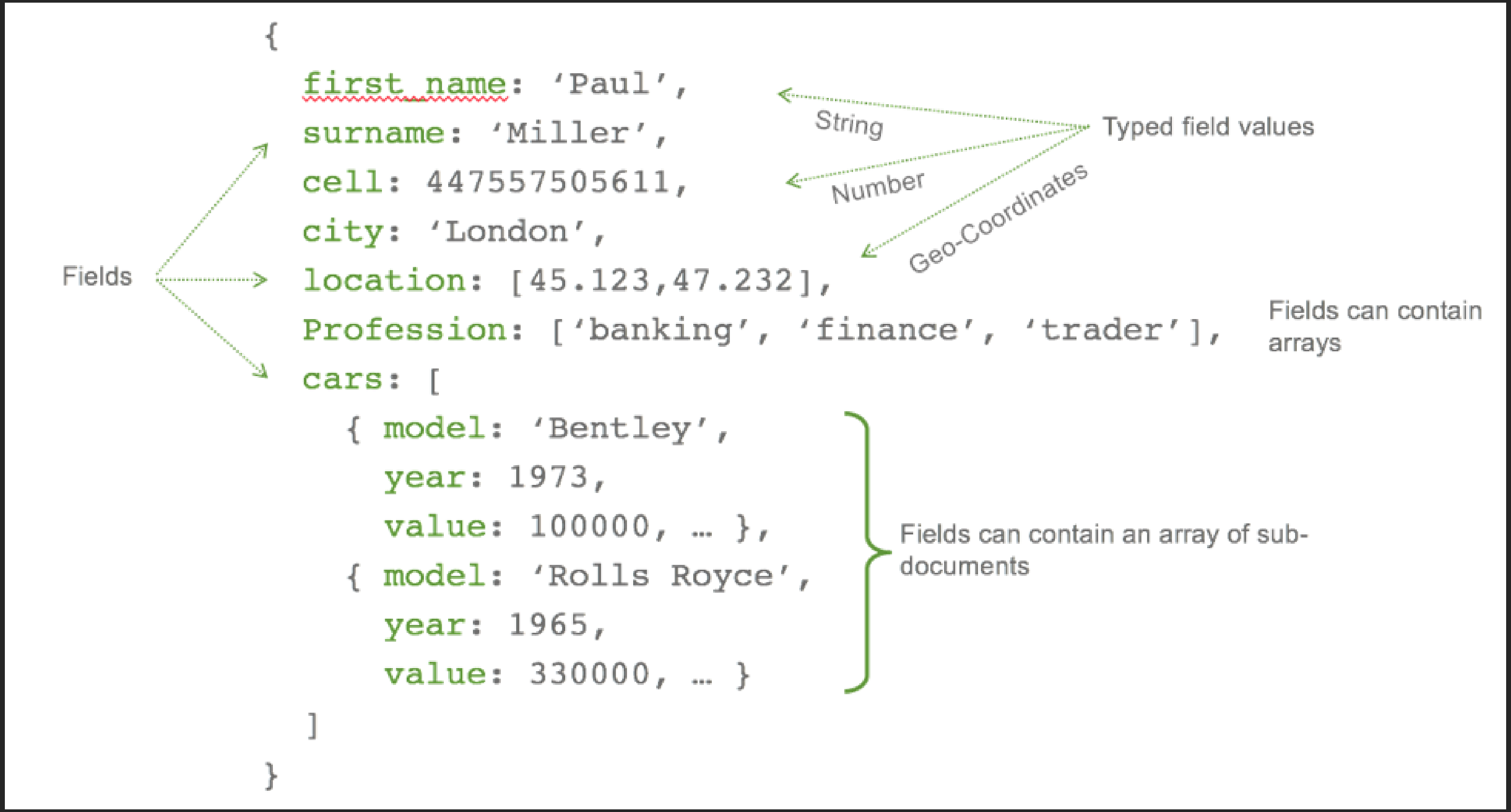
Think you are good with strings?

Are you bored and want more programming exercises?

Here's a challenge:

Write your own CSV reader/writer!

Case 2: JSON



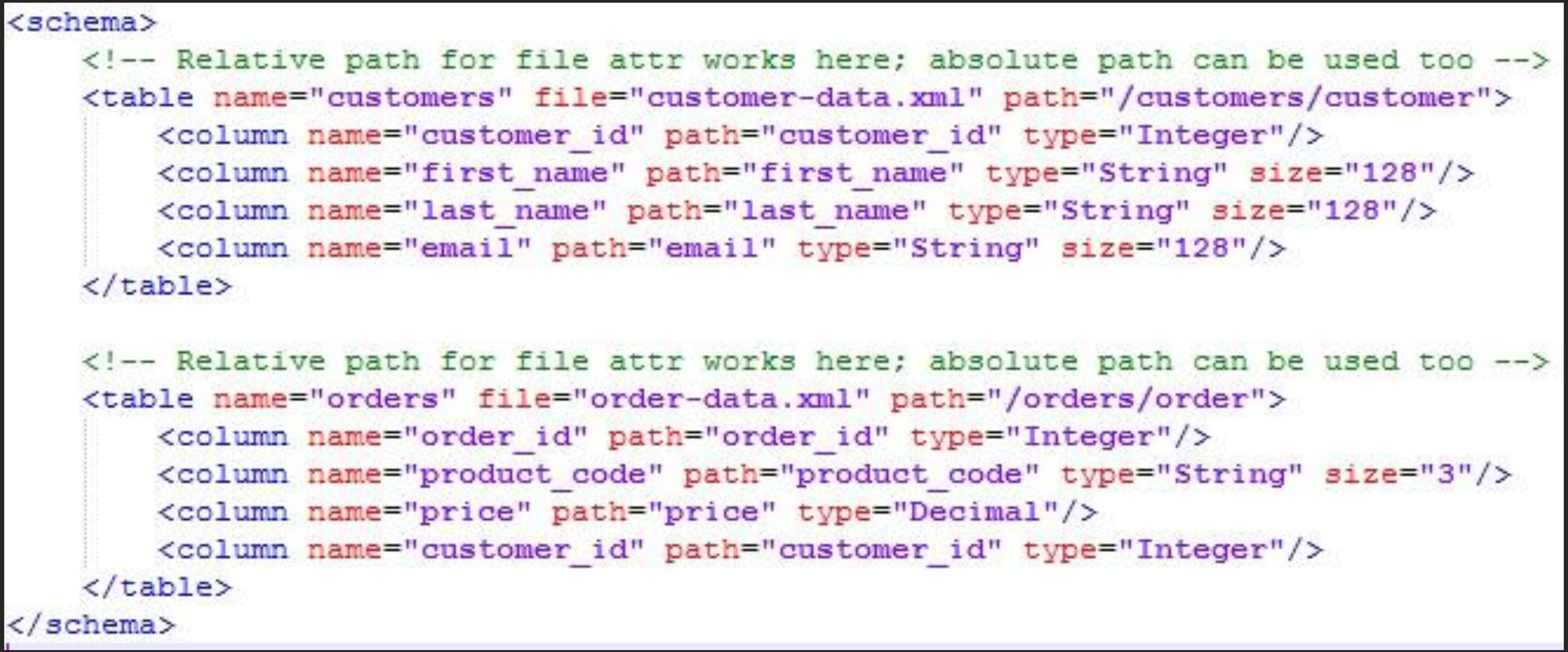
```
{
  first_name: ‘Paul’,
  surname: ‘Miller’,
  cell: 447557505611,
  city: ‘London’,
  location: [45.123,47.232],
  Profession: [‘banking’, ‘finance’, ‘trader’],
  cars: [
    { model: ‘Bentley’,
      year: 1973,
      value: 100000, … },
    { model: ‘Rolls Royce’,
      year: 1965,
      value: 330000, … }
  ]
}
```

Fields
String
Number
Geo-Coordinates
Typed field values
Fields can contain arrays
Fields can contain an array of sub-documents

Case 2: JSON

- Robust and Scalable
  - Can represent complex structures and be changed easily
  - The middle-ground between Human and Machine language.
    - We can read it and it represents typical programming data structures (objects, arrays, etc.)

Other similar object-style formats



```xml
<schema>
    <!-- Relative path for file attr works here; absolute path can be used too -->
    <table name="customers" file="customer-data.xml" path="/customers/customer">
        <column name="customer_id" path="customer_id" type="Integer"/>
        <column name="first_name" path="first_name" type="String" size="128"/>
        <column name="last_name" path="last_name" type="String" size="128"/>
        <column name="email" path="email" type="String" size="128"/>
    </table>

    <!-- Relative path for file attr works here; absolute path can be used too -->
    <table name="orders" file="order-data.xml" path="/orders/order">
        <column name="order_id" path="order_id" type="Integer"/>
        <column name="product_code" path="product_code" type="String" size="3"/>
        <column name="price" path="price" type="Decimal"/>
        <column name="customer_id" path="customer_id" type="Integer"/>
    </table>
</schema>
```

XML

Think you are REALLY GOOD with strings?

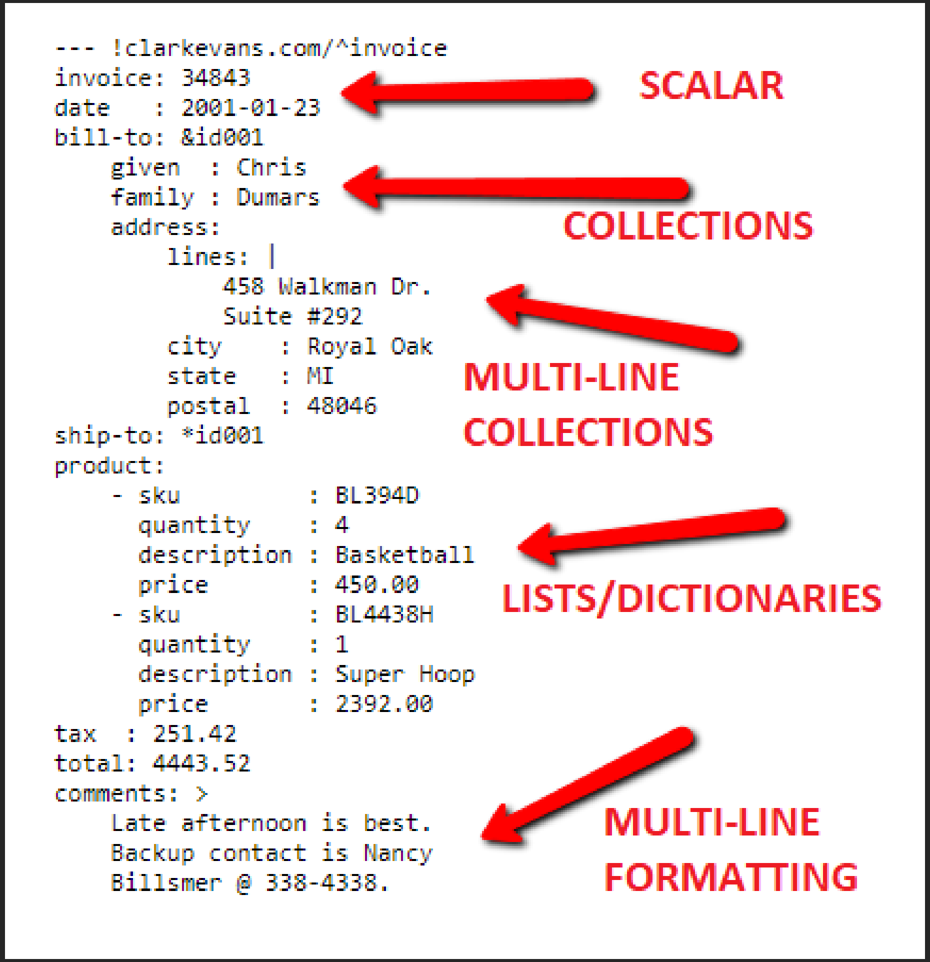
Are you bored and want more programming exercises?

Here's a challenge:

Write your own JSON reader/writer!

(err...)

Other similar object-style formats



```
--- !clarkevans.com/^invoice
invoice: 34843
date   : 2001-01-23
bill-to: &id001
    given  : Chris
    family : Dumars
    address:
        lines: |
            458 Walkman Dr.
            Suite #292
        city    : Royal Oak
        state   : MI
        postal  : 48046
ship-to: *id001
product:
    - sku         : BL394D
      quantity    : 4
      description : Basketball
      price       : 450.00
    - sku         : BL4438H
      quantity    : 1
      description : Super Hoop
      price       : 2392.00
tax  : 251.42
total: 4443.52
comments: >
    Late afternoon is best.
    Backup contact is Nancy
    Billsmer @ 338-4338.
```

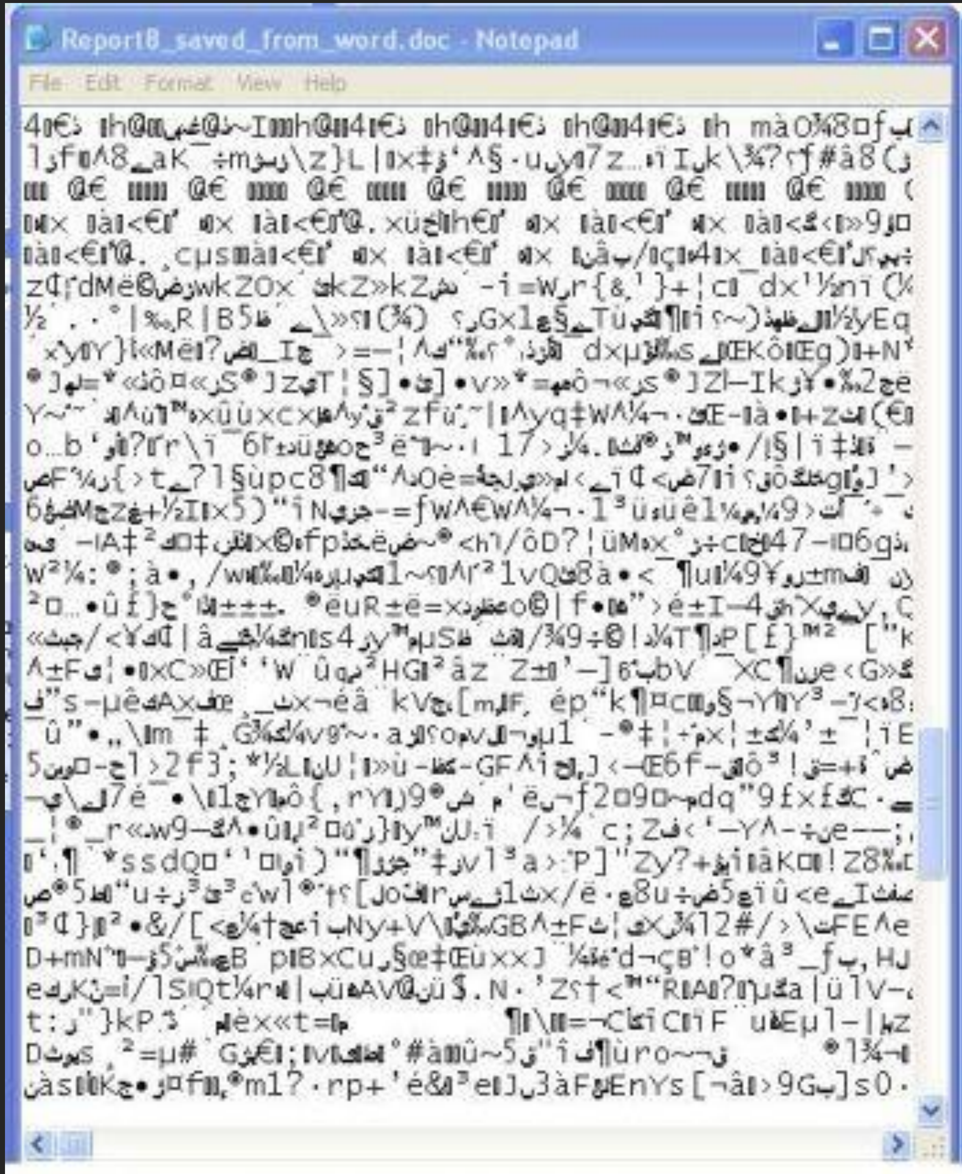
YAML

Case 3: Binary

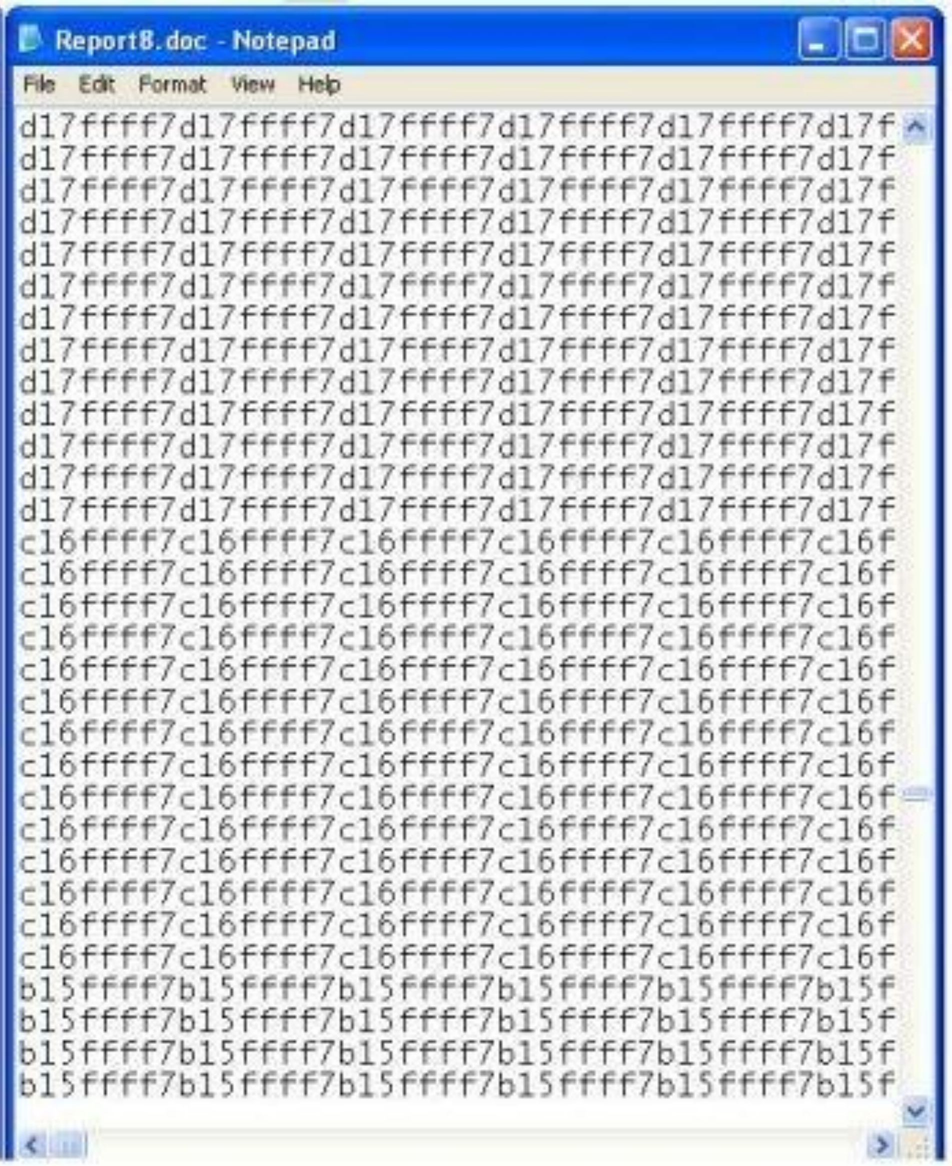
- Stores information in ‘raw’ format, typically matching what was in the memory.
- Idea: If data = memory, we should be able to dump the memory back to reload our scene right?

Case 3: Binary

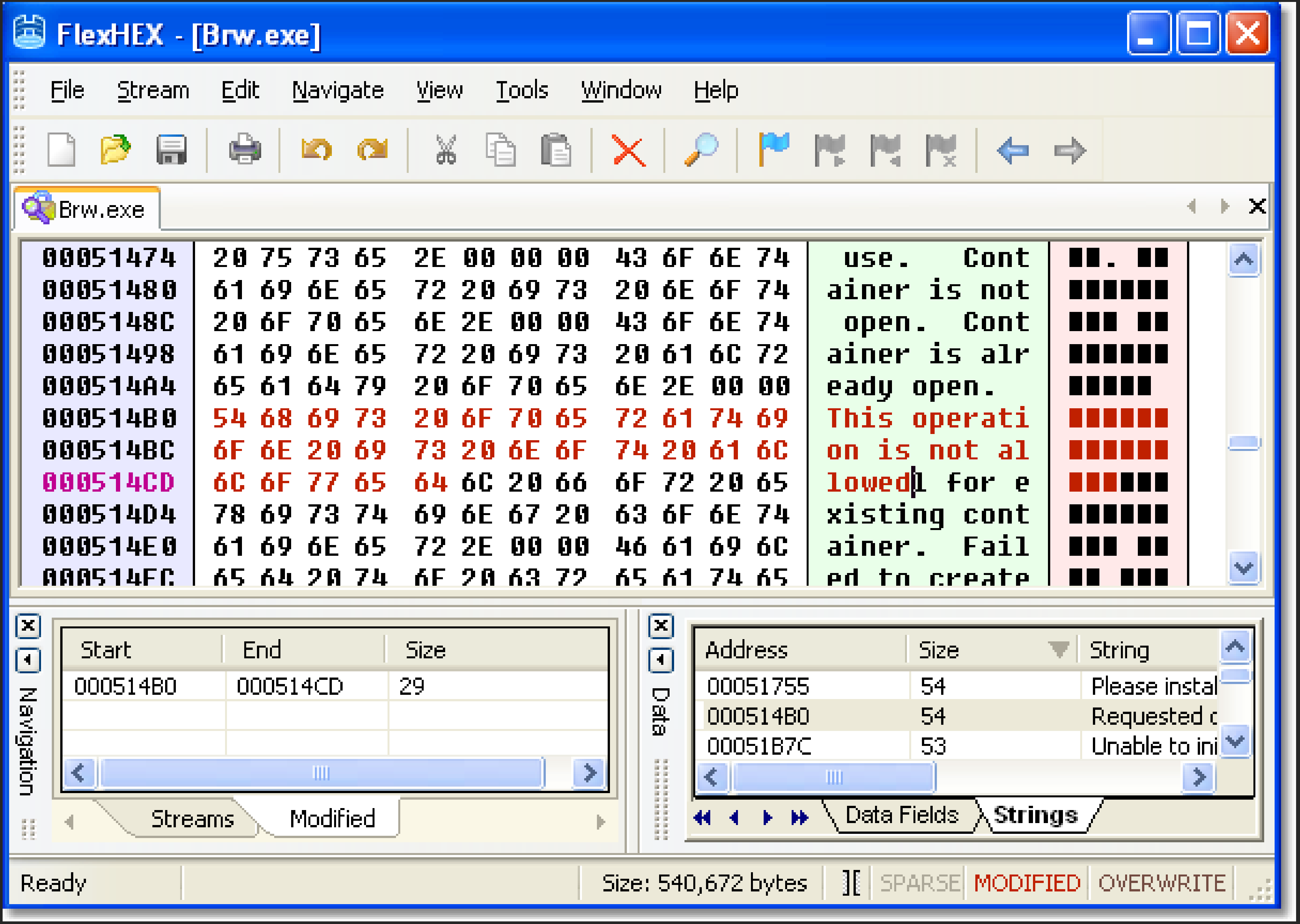
- Issues with platform specific encoding
  - Mac might read binary differently from Windows.
  - 32-bit systems might read differently from 64 bit systems.
- The leanest data format you can get (i.e. fast!)
- Hard to work with without a visual tool.
  - Have fun trying to edit in a hex editor!
  - Which some people (e.g, modders/hackers) do


Report8_saved_from_word.doc - Notepad
File Edit Format View Help

```
[illegible]
```



```
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
d17ffff7d17ffff7d17ffff7d17ffff7d17ffff7d17f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
c16ffff7c16ffff7c16ffff7c16ffff7c16ffff7c16f
b15ffff7b15ffff7b15ffff7b15ffff7b15ffff7b15f
b15ffff7b15ffff7b15ffff7b15ffff7b15ffff7b15f
b15ffff7b15ffff7b15ffff7b15ffff7b15ffff7b15f
b15ffff7b15ffff7b15ffff7b15ffff7b15ffff7b15f
```



```
00051474  20 75 73 65  2E 00 00 00  43 6F 6E 74   use.   Cont
00051480  61 69 6E 65  72 20 69 73  20 6E 6F 74  ainer is not
0005148C  20 6F 70 65  6E 2E 00 00  43 6F 6E 74   open.  Cont
00051498  61 69 6E 65  72 20 69 73  20 61 6C 72  ainer is alr
000514A4  65 61 64 79  20 6F 70 65  6E 2E 00 00  eady open.
000514B0  54 68 69 73  20 6F 70 65  72 61 74 69  This operati
000514BC  6F 6E 20 69  73 20 6E 6F  74 20 61 6C  on is not al
000514CD  6C 6F 77 65  64 6C 20 66  6F 72 20 65  lowedl for e
000514D4  78 69 73 74  69 6E 67 20  63 6F 6E 74  xisting cont
000514E0  61 69 6E 65  72 2E 00 00  46 61 69 6C  ainer.  Fail
000514EC  65 64 20 74  6F 20 63 72  65 61 74 65  ed to create
```

Start	End	Size
000514B0	000514CD	29

Address	Size	String
00051755	54	Please instal
000514B0	54	Requested c
00051B7C	53	Unable to ini

Ready | Size: 540,672 bytes | SPARSE | MODIFIED | OVERWRITE

Most of our files are binary!
(jpeg, png, pdf…)

Case 3: Binary

- Issues with platform specific encoding
  - Mac might read binary differently from Windows.
  - 32-bit systems read differently from 64 bit systems.
  - ENDIAN-NESS?!
- The leanest data format you can get (i.e. fast!)
- Hard to work with without a visual tool.
  - Have fun trying to edit in a hex editor!
  - Which some people (e.g, modders/hackers) do

Think you are REALLY GOOD with binary?

Are you bored and want more programming exercises?

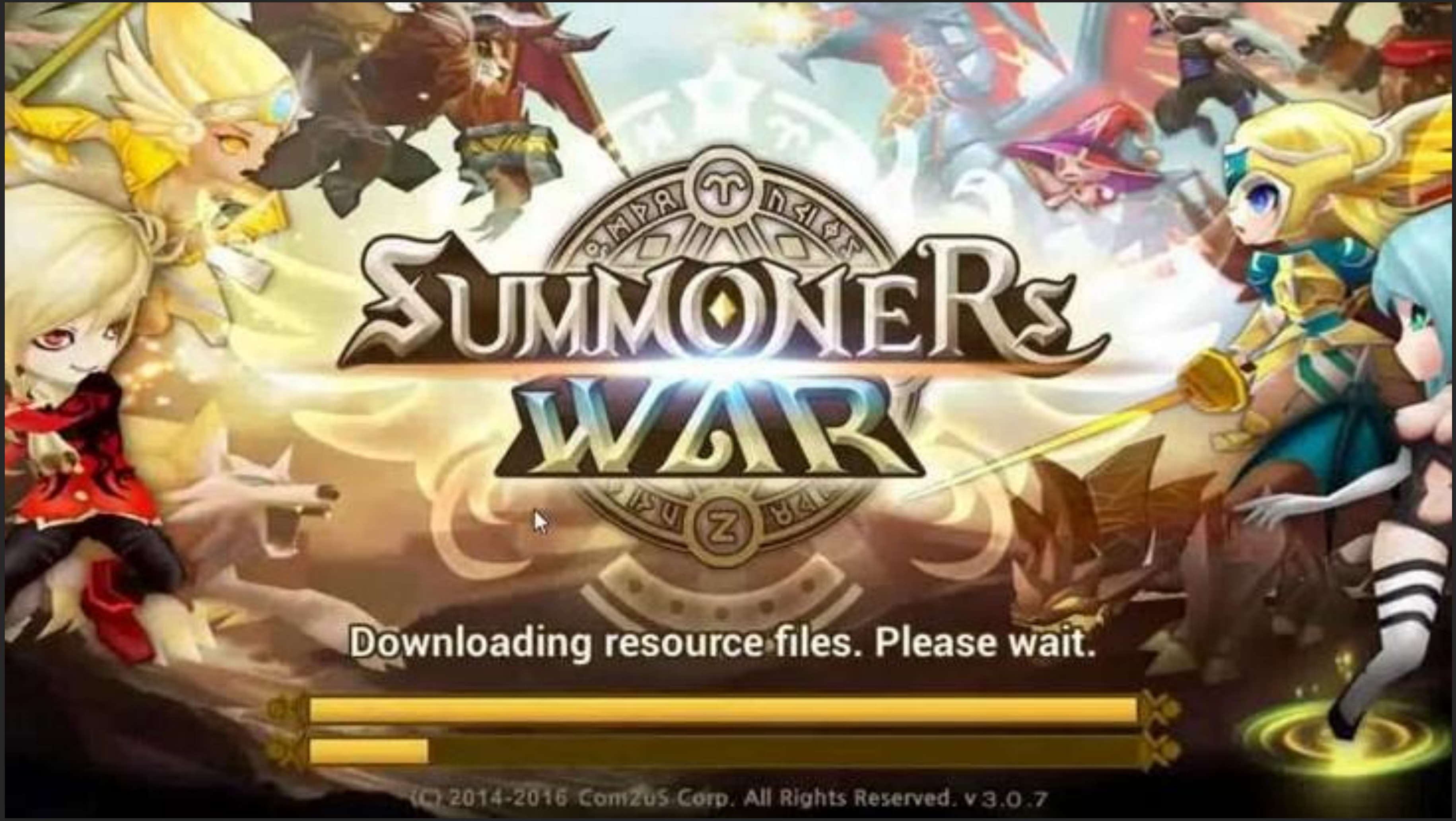
Here's a challenge:

Write your own BMP reader/writer!

(this is actually doable!)

Binary Format documentations

- Most binary formats usually comes with documentation because it is not obvious how to decode/encode them (unlike CSV)
  - PNG: https://datatracker.ietf.org/doc/html/rfc2083
  - BMP: https://www.fileformat.info/format/bmp/egff.htm


SUMMONERS WAR
Downloading resource files. Please wait.
(C) 2014-2016 Com2uS Corp. All Rights Reserved. v3.0.7

The Age of Patching

Data-Serialization is Key

Ending notes and advice

- If unsure, stick with CSV, JSON or a simple text format you can easily read, write and understand.
- There is no one size fit all solution (yet).
  - Yes, you can use JSON for many things, but please give it some thought.
- Write code that runs first, <u>then</u> think about what makes sense to export into a data format to read in.
  - Typically, things that frequency change which you don’t want to recompile a new binary for (e.g. levels)