

# Intro

---

## Storing data

- memory
  - memory will be cleared on every boot/program start
- disk
  - for persistence

## disk storage

- binary file
  - data in binary format
  - images
  - audio
  - video
  - compiled programs
  - [en.wikipedia.org/wiki/List\\_of\\_file\\_signatures](https://en.wikipedia.org/wiki/List_of_file_signatures)
  - [en.wikipedia.org/wiki/Magic\\_number\\_\(programming\)](https://en.wikipedia.org/wiki/Magic_number_(programming))
- plain text file

## bits 0 and 1

- 1 bit can represent 2 different states 0 or 1
- 2 bits you have 4 states: 00, 01, 10, 11
- 3 bits 8 states  $2^3 = 8$
- binary system is base 2 positional system
- octadecimal system use 3 bits 421
  - $000=0 \Rightarrow 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 0$
  - $001=1$
  - $010=2$
  - $011=3 \Rightarrow 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 0 + 2 + 1 = 3$
  - $100=4$
  - $101=5$
  - $110=6$
  - $111=7$

text file containing "unstructured" free text

Matt River  
2024

Memo

This year 2024 was great start in the development of GTX series. This new product was made by Vera River who also was attending the launch....

Memo

This year 2024 was great start in the development of GTX series. This new product was made by Vera River who also was attending the launch....

Matt River  
2024

Problem: hard to process programmatically

made by Matt River  
year 2024

Memo

This year 2024 was great start in the development of GTX series. This new product was made by Vera River who also was attending the launch....

Better but still not good

Solution: structured text format

## XML

```
<document>
  <writer>
    <firstname>Matt</firstname>
    <lastname>River</lastname>
  </writer>
  <year>2024</year>
  <title>Memo</title>
  <text>
    This year 2024 was great start in the development
    of GTX series. This new product was made by Vera River
    who also was attending the launch....
  </text>
</document>
```

## JSON

```
{
  "writer":{
    "firstname":"Matt",
    "lastname":"River"
  },
  "year":2024,
  "title":"Memo",
  "text":[
    "This year 2024 was great start in the development",
    "of GTX series. This new product was made by Vera River",
    "who also was attending the launch...."
  ]
}
```

## structures text files

- xml ([www.w3.org](http://www.w3.org))
- json ([www.json.org](http://www.json.org))
- csv/tsv (Comma Separated Value/ Tab Separated Value)
  - handy format for transferring tabular data between spreadsheet and your program.
- ini
- html/css (<https://html.spec.whatwg.org/multipage/>)
- markdown
- svg (for images, xml file)

## Examples

### XML

```
<?xml version="1.1" encoding="UTF-8">
<persons>
  <person>
    <firstname>Matt</firstname>
    <lastname>River</lastname>
    <age>25</age>
  <person>
  <person>
    <firstname>Vera</firstname>
    <lastname>River</lastname>
    <age>27</age>
  <person>
</persons>
```

### JSON

```
[
  {
    "firstname": "Matt",
    "lastname": "River",
    "age": 25
  },
  {
    "firstname": "Vera",
    "lastname": "River",
    "age": 27
  }
]
```

**csv**

```
firstname,lastname,age  
Matt,River,25  
Vera,River,27
```

```
"firstname","lastname","age"  
"Matt","River, Jr","25"  
"Vera","River","27"
```

**ini**

```
;this is a comment in ini file  
[database]  
driver=mysql  
host=localhost  
port=3306  
database=persondb  
[user]  
username=zeke
```

**Html/css**

look at markdown.md

**svg**