

# Data Stores

## Section 4

In this Section, we are going to take a look at...

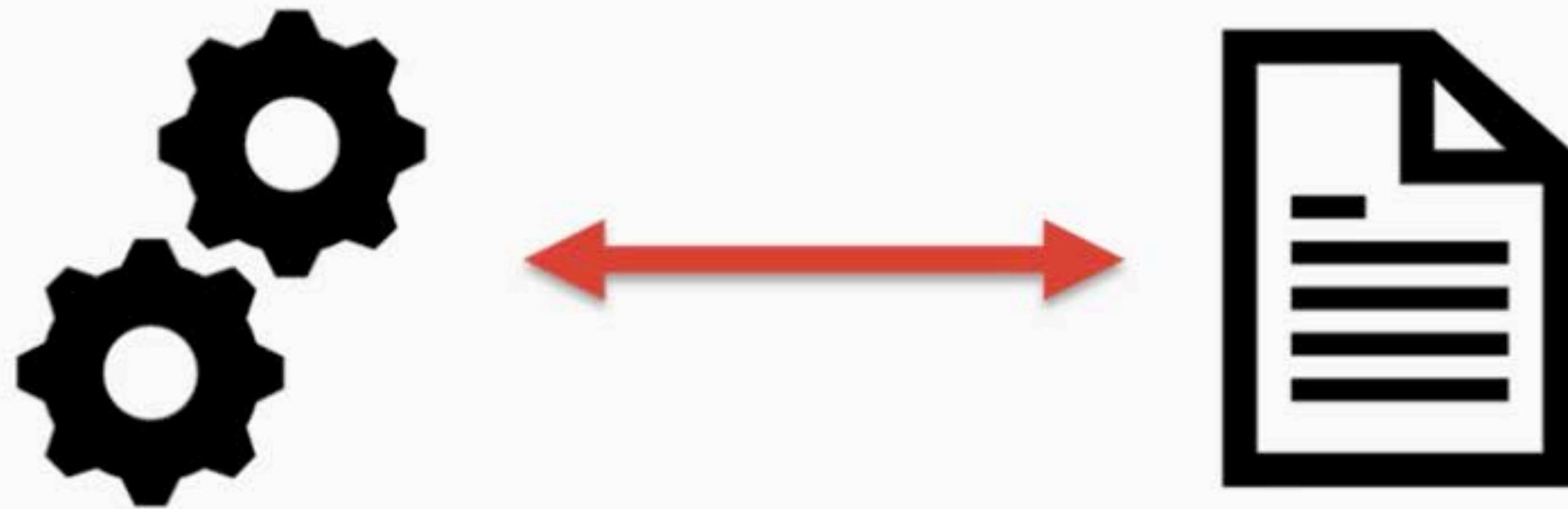
- Data stores
- ETS
- DETS and Mnesia

# ETS

In this Video, we are going to take a look at...

- Types of data stores
- ETS

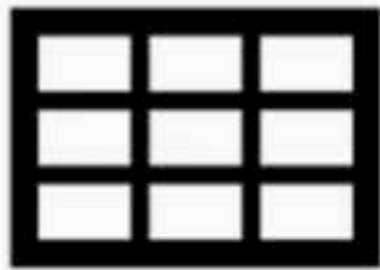
# Data Stores



# Data Stores



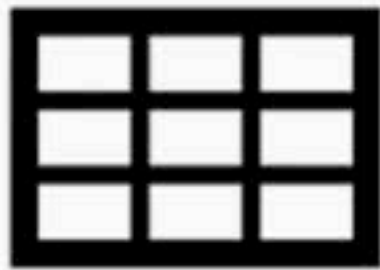
# Data Stores



**In-Memory  
Collections**



# Data Stores



**In-Memory  
Collections**

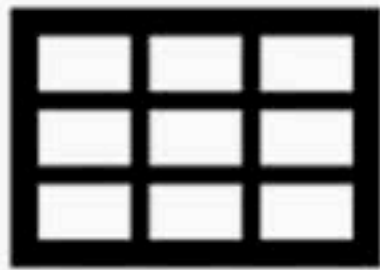


**Files**





# Data Stores



**In-Memory  
Collections**



**Files**



**Database  
Systems**

# ETS

Erlang Term Storage

# ETS

Erlang Term Storage

- In-Memory data store
- Owned by a process
- Allows storage of any Elixir data type
- Disappear once the owner process dies

# ETS – Book Catalog

Create a Book data-store  
using ETS

```
{:Title, 7, 320}
```

# ETS – Creating, Inserting, Deleting, and Looking-up

→ **Projects** iex

Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [kernel-poll:false] [dtrace]

Interactive Elixir (1.4.2) – press Ctrl+C to exit (type h() ENTER for help)

iex(1)> table = :ets.new(:books, [:set])

8211

iex(2)> table |> :ets.insert({:fairy\_tale, 10, 329})

true

iex(3)> table |> :ets.insert({:horror\_novel, 9, 128})

true

iex(4)> table |> :ets.lookup(:horror\_novel)

[{:horror\_novel, 9, 128}]

iex(5)> table |> :ets.lookup(:fairy\_tale)

[{:fairy\_tale, 10, 329}]

iex(6)> table |> :ets.delete(:horror\_novel)

true

iex(7)> table |> :ets.lookup(:horror\_novel)

[]

iex(8)> table |> :ets.delete()

true

iex(9)> table |> :ets.lookup(:horror\_novel)

\*\* (ArgumentError) argument error

(stdlib) :ets.lookup(8211, :horror\_novel)

iex(9)> █

# ETS - Types

The Type governs the way in which we can store and access terms.

|                      | Set | Ordered Set | Bag | Duplicate Bag |
|----------------------|-----|-------------|-----|---------------|
| Duplicate keys?      | ✗   | ✗           | ✓   | ✓             |
| Duplicate key-value? | ✗   | ✗           | ✗   | ✓             |
| Ordered?             | ✗   | ✓           | ✗   | ✗             |

## What about More Advanced Lookups?

match/2

select/2

fun2ms/1



## ETS – Advanced Look-up

```
iex(5)> table |> :ets.insert({:mediocre_story, 2, 48})
true
iex(6)> table |> :ets.insert({:fable, 5, 120})
true
iex(7)> table |> :ets.match({:"$1", 9, :"$2"})
[[:best_seller, 542], [:horror_novel, 128]]
iex(8)> table |> :ets.match({:"$2", 9, :"$1"})
[[542, :best_seller], [128, :horror_novel]]
iex(9)> table |> :ets.match({:"$1", 9, :"_"})
[[:best_seller], [:horror_novel]]
iex(10)> table |> :ets.select([{{:"$1", 9, :"$3"}, [], [:"$1"]}])
[:best_seller, :horror_novel]
iex(11)> table |> :ets.select([{{:"$1", 9, :"$3"}, [], [:"$$"]}])
[[:best_seller, 542], [:horror_novel, 128]]
iex(12)> table |> :ets.select([{{:"$1", 9, :"$3"}, [], [:"$_"]}])
[{:best_seller, 9, 542}, {:horror_novel, 9, 128}]
iex(13)> table |> :ets.select([{{:"$1", :"$2", :"$3"}, [{>=, :"$2", 9}], [:"$_"]}])
[{:fairy_tale, 10, 329}, {:best_seller, 9, 542}, {:horror_novel, 9, 128}]
iex(14)> expression = :ets.fun2ms(fn {title, rating, pages} = book when rating >
= 9 -> book end)
[{{:"$1", :"$2", :"$3"}, [{>=, :"$2", 9}], [:"$_"]}]
iex(15)> table |> :ets.select(expression)
[{:fairy_tale, 10, 329}, {:best_seller, 9, 542}, {:horror_novel, 9, 128}]
iex(16)> █
```