

# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



James Edward Gray, II  
Bruce A. Tate  
*edited by Jacquelyn Carter*



# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



@redrapids

James Edward Gray, II  
Bruce A. Tate  
*edited by Jacquelyn Carter*

growx.io

Learning

Career Rocket Fuel  
for Curious Coders

bring on the  
**FREE BEEES**

20% off PP books:

Code **BEAM\_SF\_2020**



# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



"Do Fun Things  
with  
Big, Loud  
Worker-Bees"

by James E Gray, II  
and Bruce A. Tate



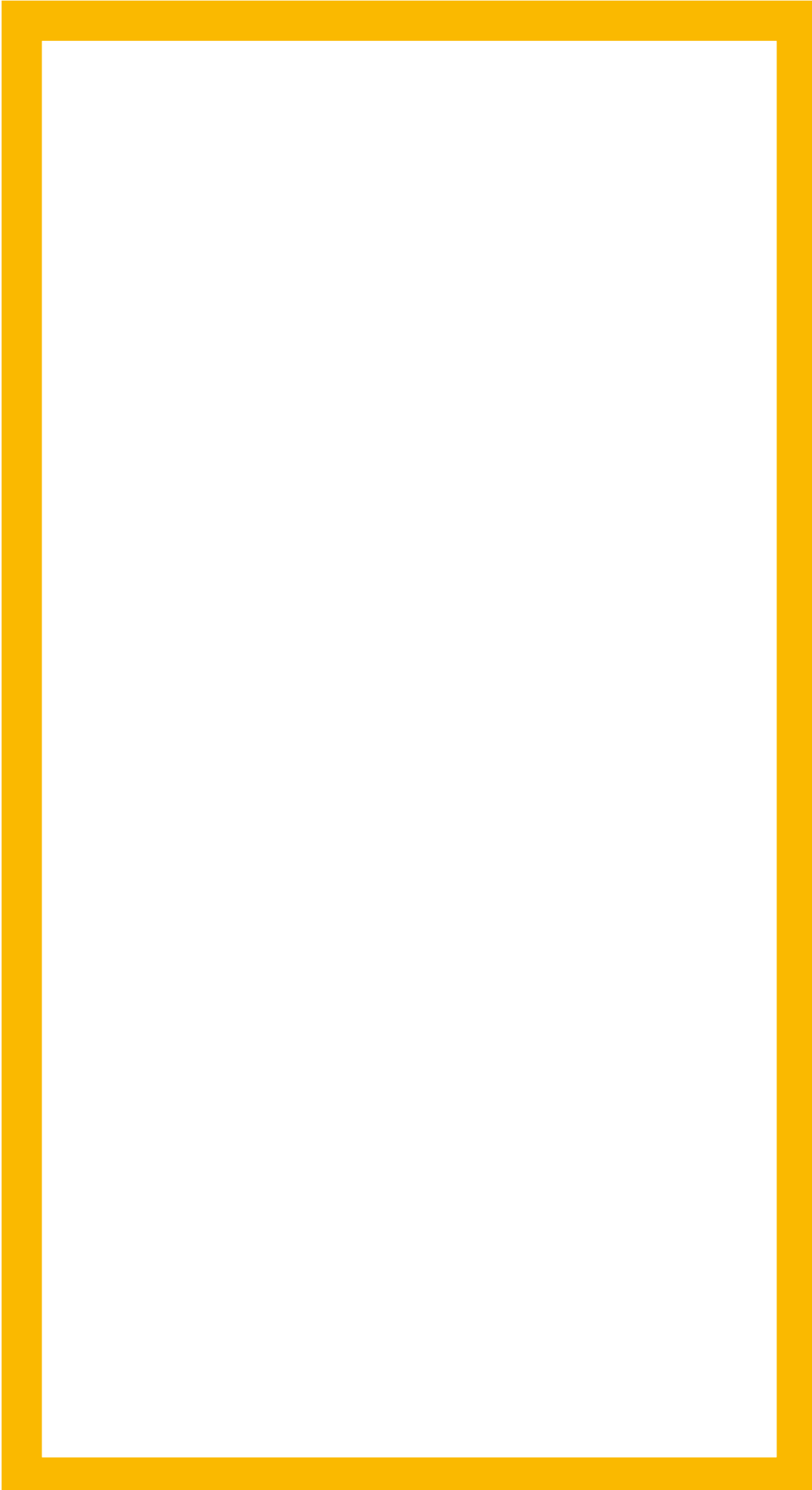
# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



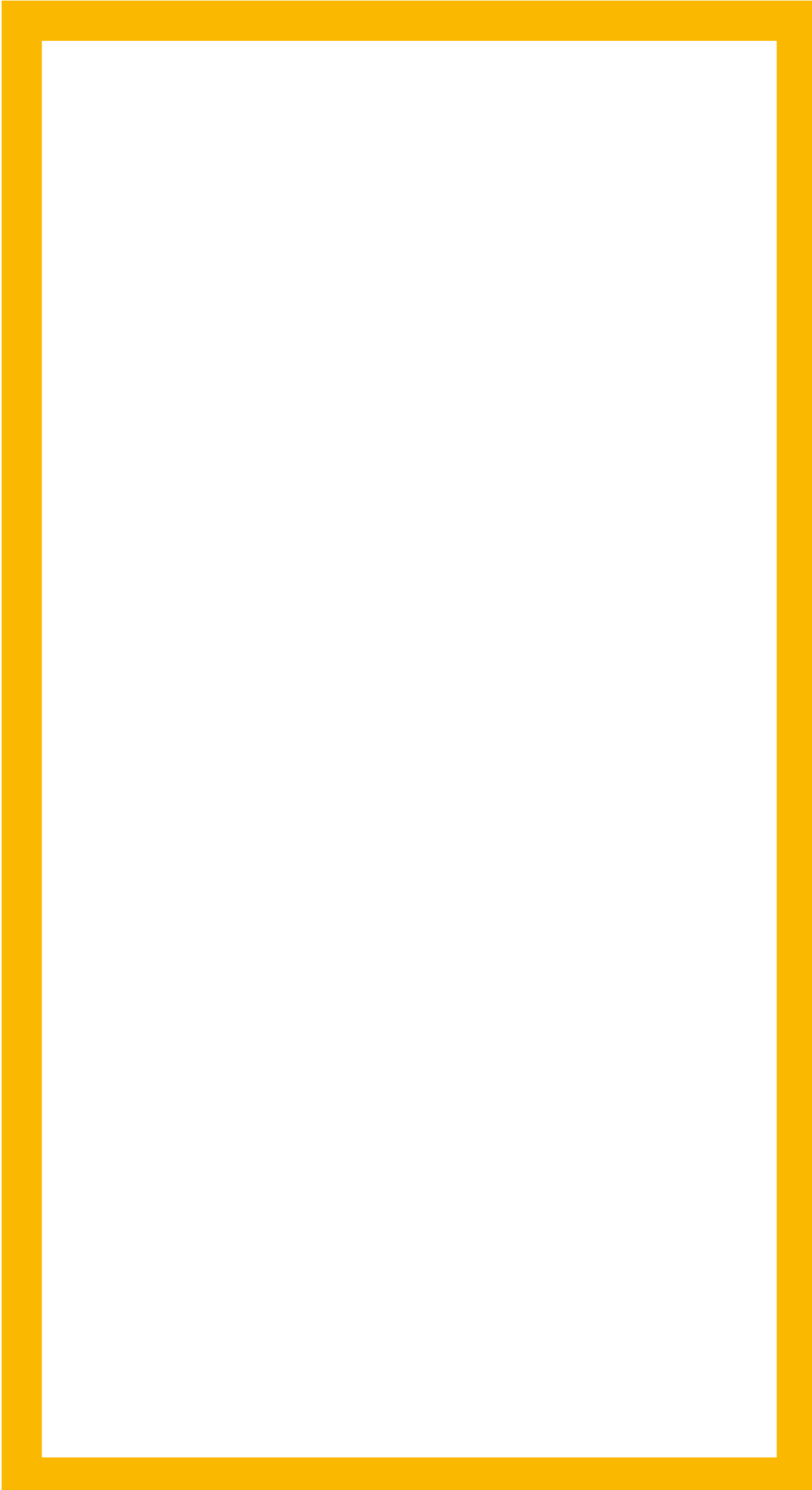
"Do Fun Things  
with  
Big, Loud  
Worker-Bees"

by James E Gray, II  
and Bruce A. Tate



Do





Fun



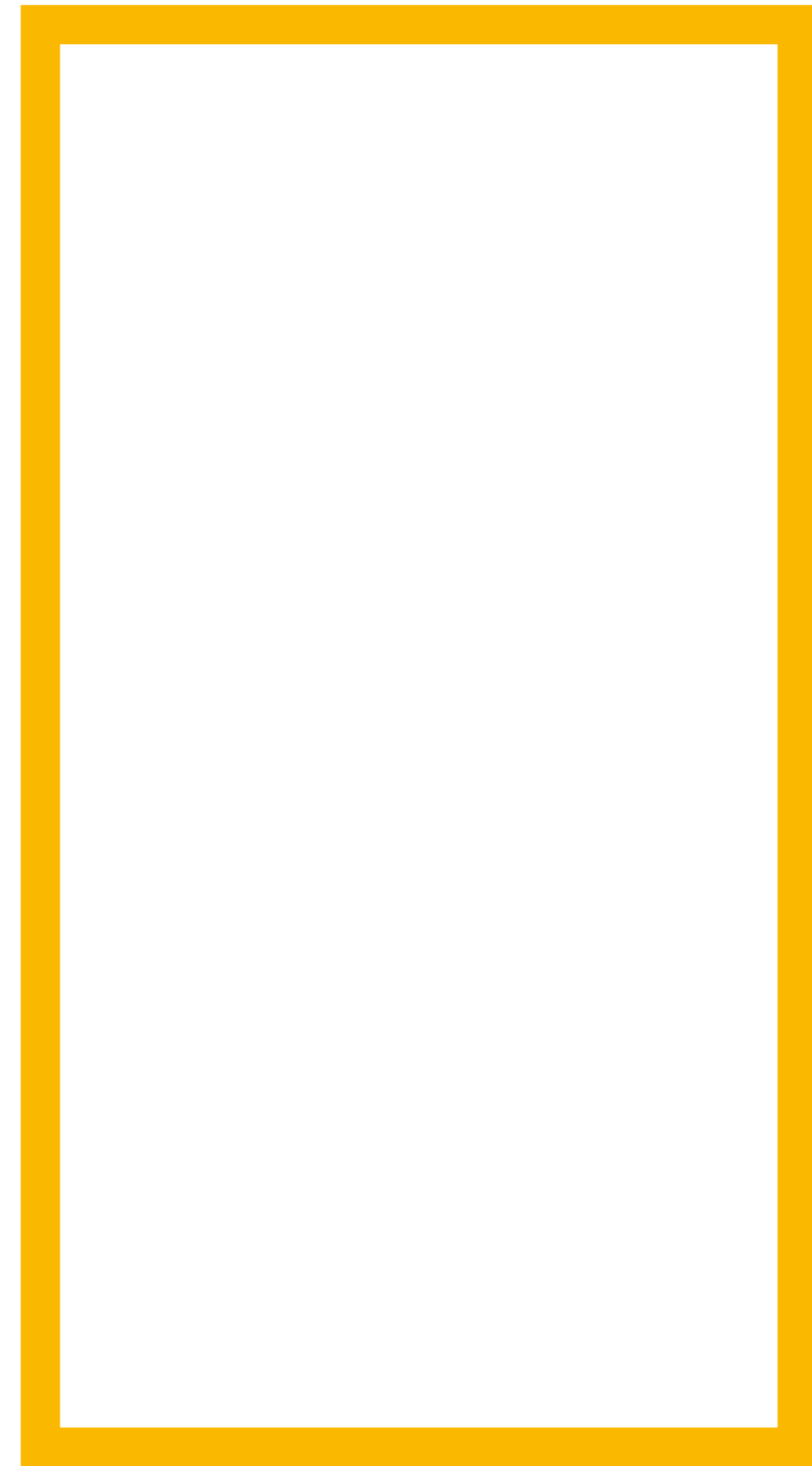
# Things

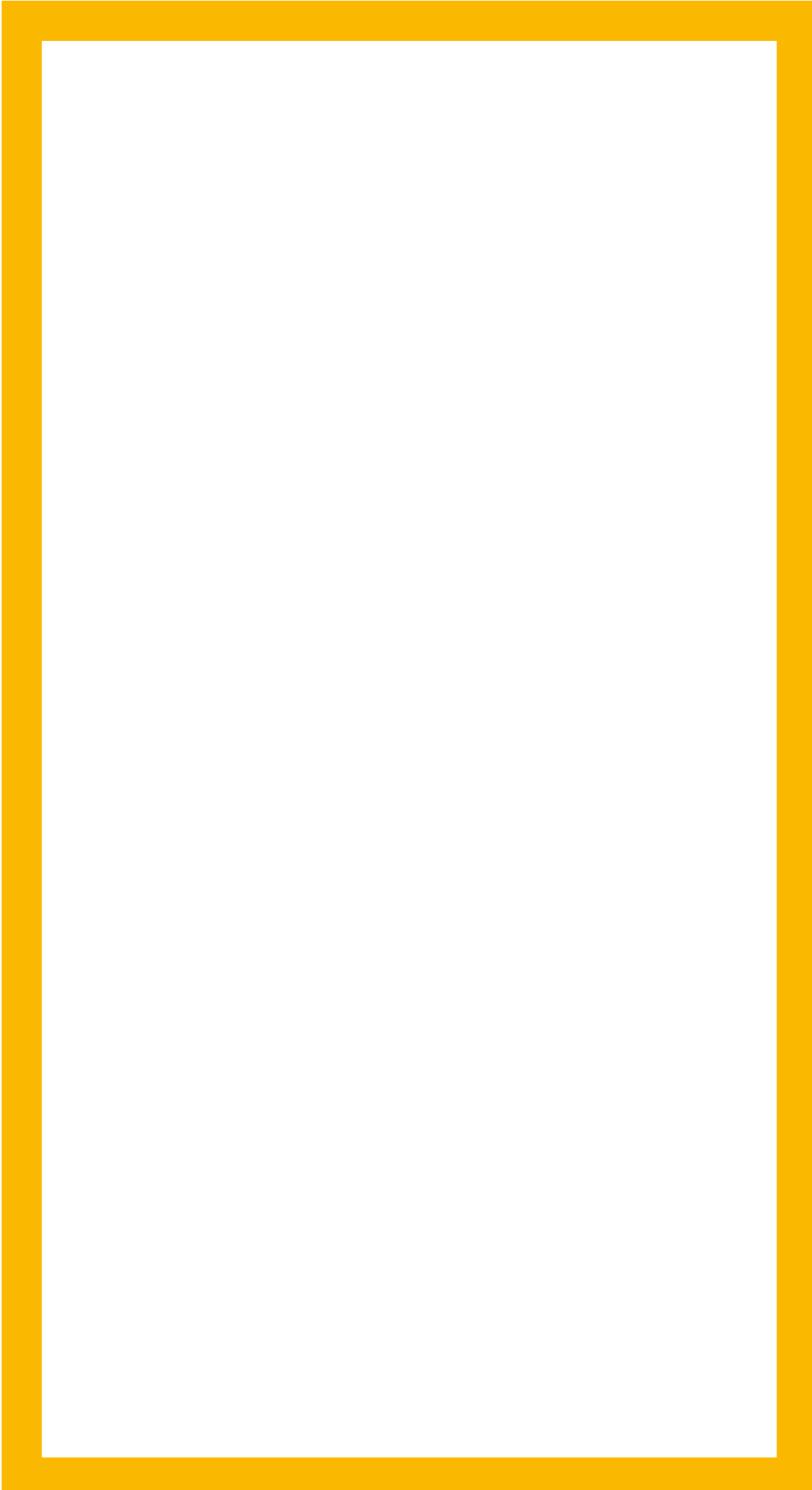
Big



Loud

# WBees

An empty rectangular box with a thick yellow border, positioned in the top row of the layout.An empty rectangular box with a thick yellow border, positioned in the top row of the layout.A large, tall empty rectangular box with a thick yellow border, positioned on the right side of the layout.A wide, horizontal empty rectangular box with a thick yellow border, positioned in the middle row of the layout.A wide, horizontal empty rectangular box with a thick yellow border, positioned in the bottom row of the layout.



**Data**

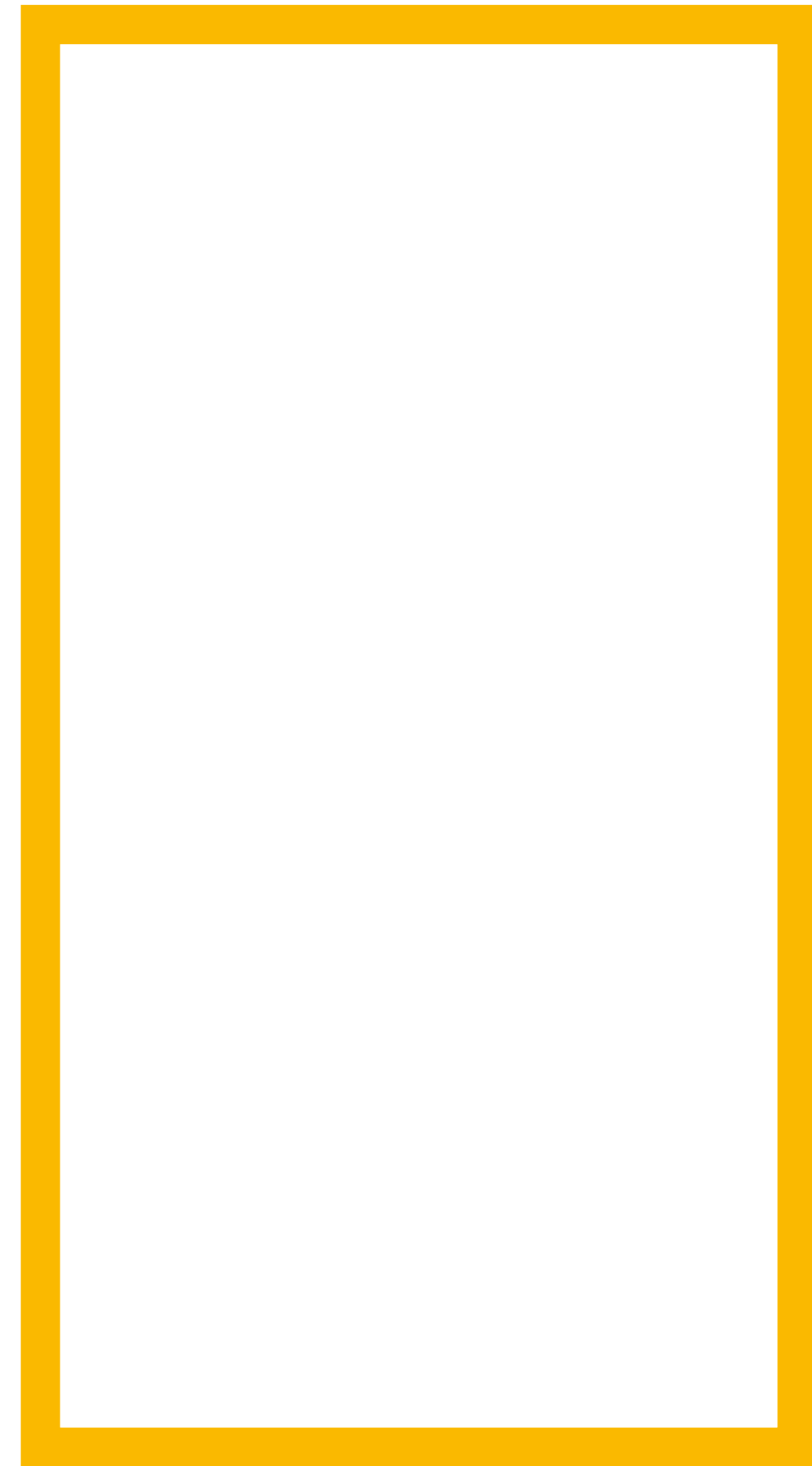




# Functions

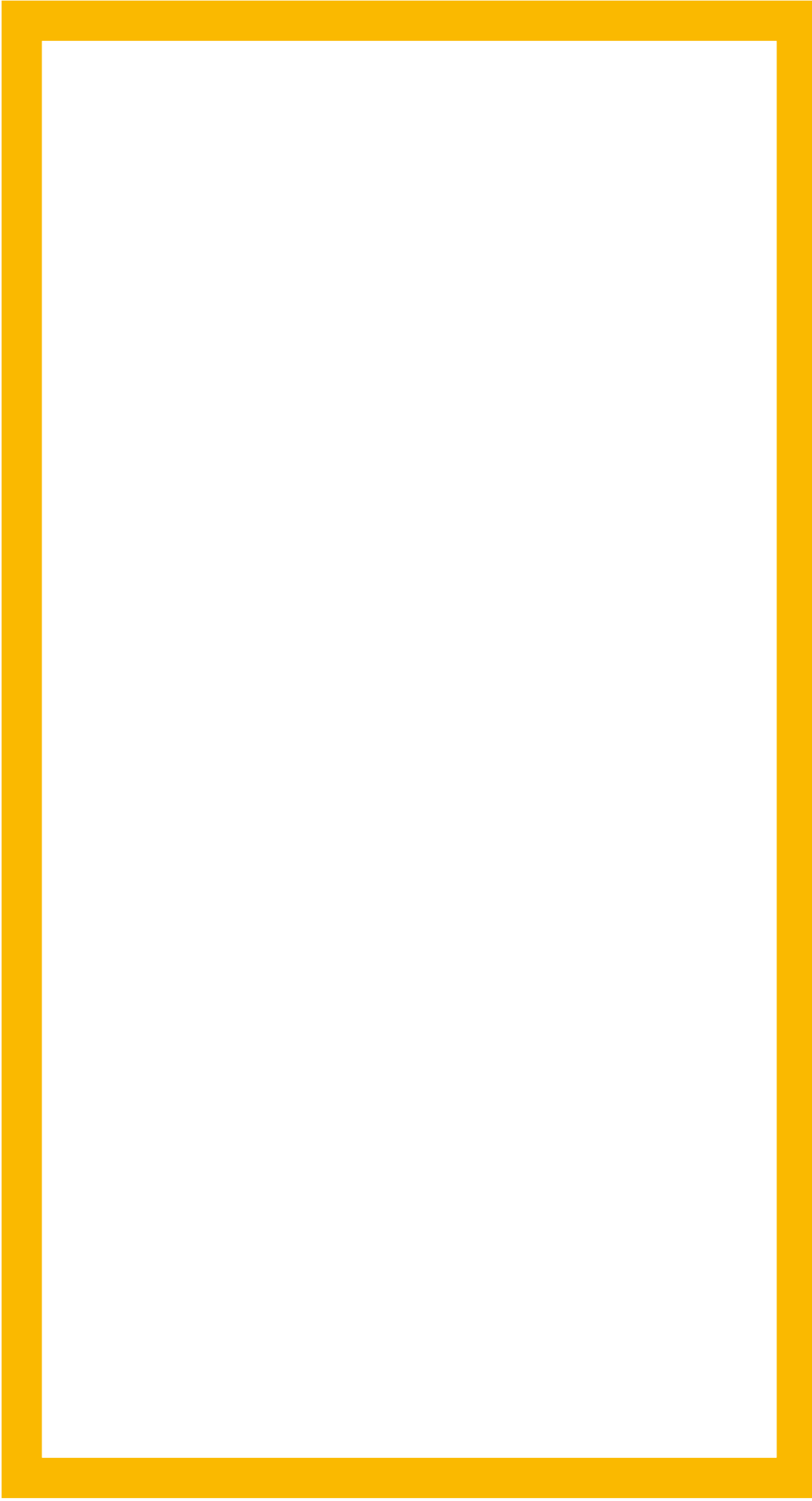
# Tests

# Boundaries

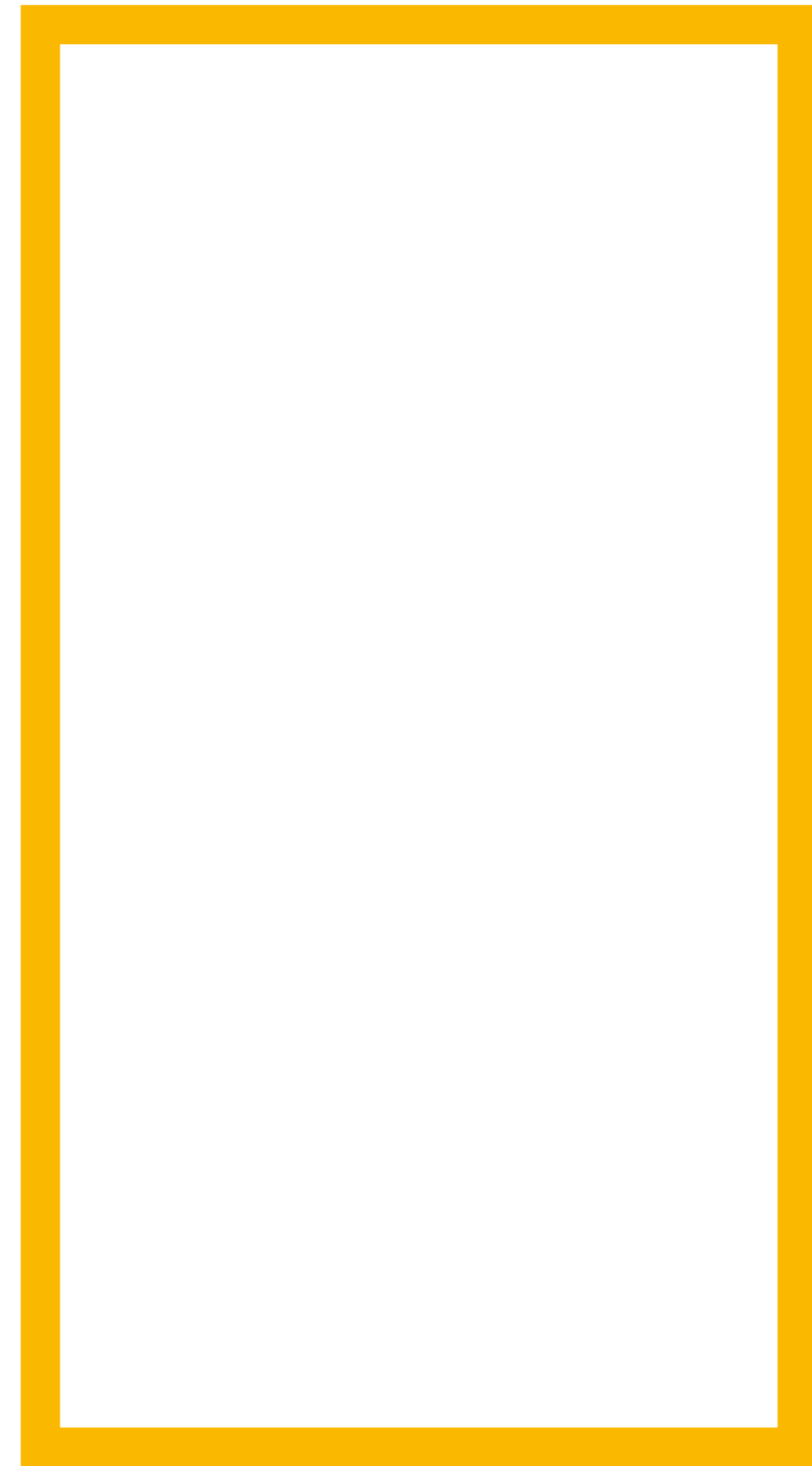











# Lifecycles



# Workers

An empty rectangular box with a thick yellow border, positioned in the top row of the diagram.An empty rectangular box with a thick yellow border, positioned in the top row of the diagram.A large empty rectangular box with a thick yellow border, positioned on the right side of the diagram.A wide empty rectangular box with a thick yellow border, positioned in the middle row of the diagram.A wide empty rectangular box with a thick yellow border, positioned in the bottom row of the diagram.

- ▼  lib
  - ▼  mastery
    - >  boundary
    - >  core
    - >  examples
    -  application.ex
  -  mastery.ex



- lib
  - mastery
    - boundary
    - core
    - examples
    - application.ex
    - mastery.ex



Do  
Fun  
Things

- lib
  - mastery
    - boundary
    - core
    - examples
    - application.ex
  - mastery.ex



**Big  
Loud  
WBees**

- lib
  - mastery
    - boundary
    - core
    - examples
    - application.ex
  - mastery.ex



API

# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



"Do Fun Things  
with  
Big, Loud  
Worker-Bees"

by James E Gray, II  
and Bruce A. Tate





Do





**Data**



A wooden honey dipper with a spiral-shaped head is resting on a white plate. The dipper is coated with honey, and a small amount of honey is visible on the plate. The background is a soft, out-of-focus white.

**Harmony**



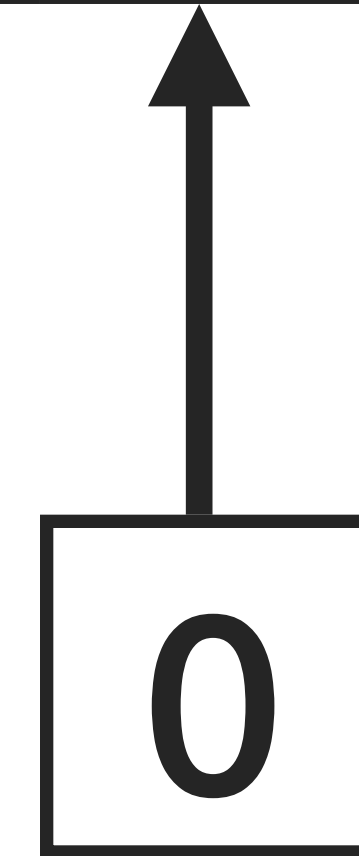


Cost of

Immutability



**O O P**



$$\mathbf{a[3] = 0}$$



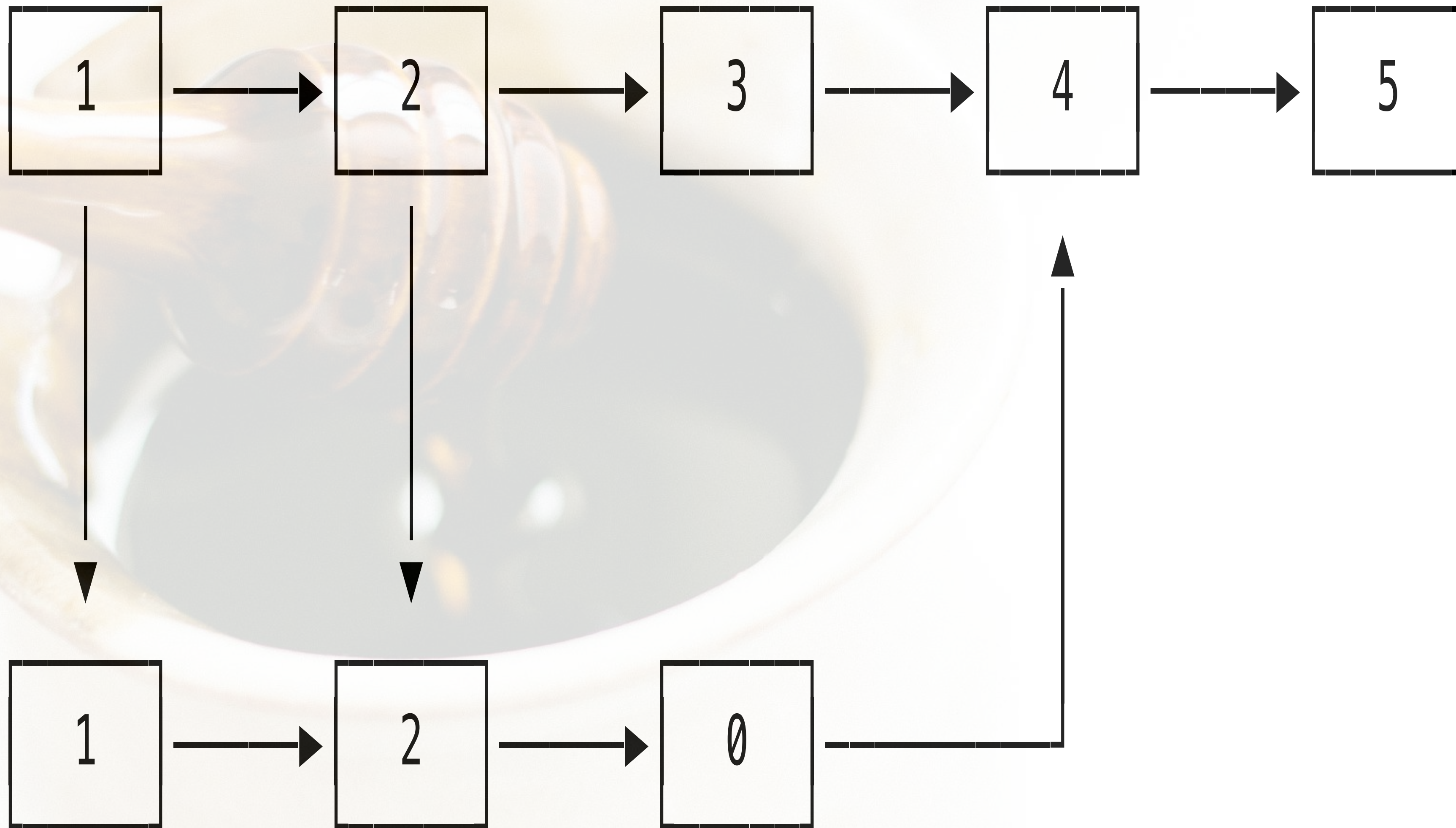


1	2	3	0	5
---	---	---	---	---

FFP

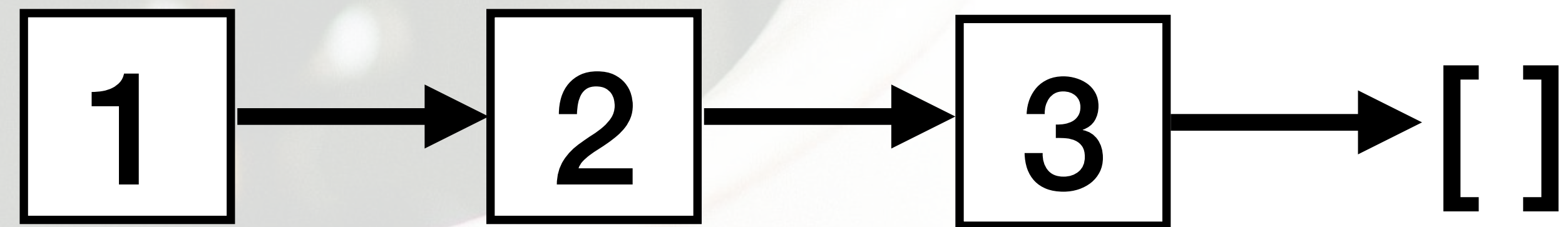
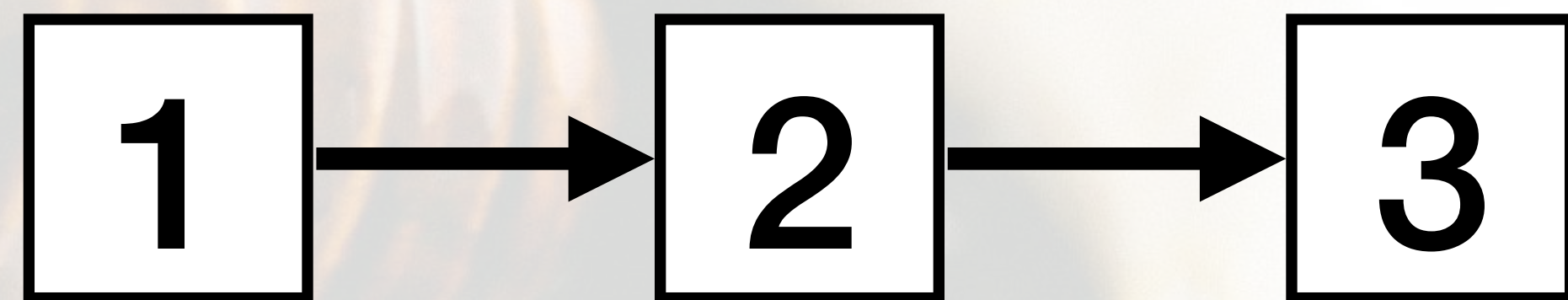
$a[3] = 0$



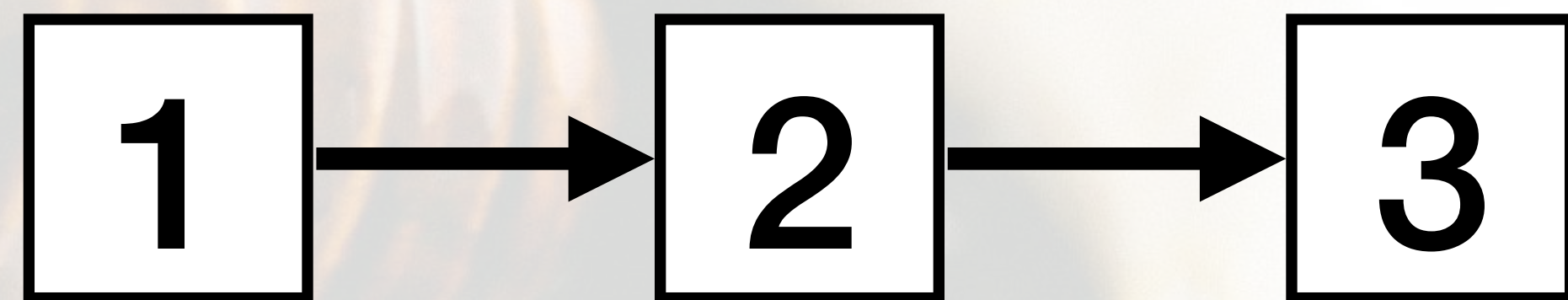


```
List.replace_at([1, 2, 3, 4, 5], 2, 0)
```

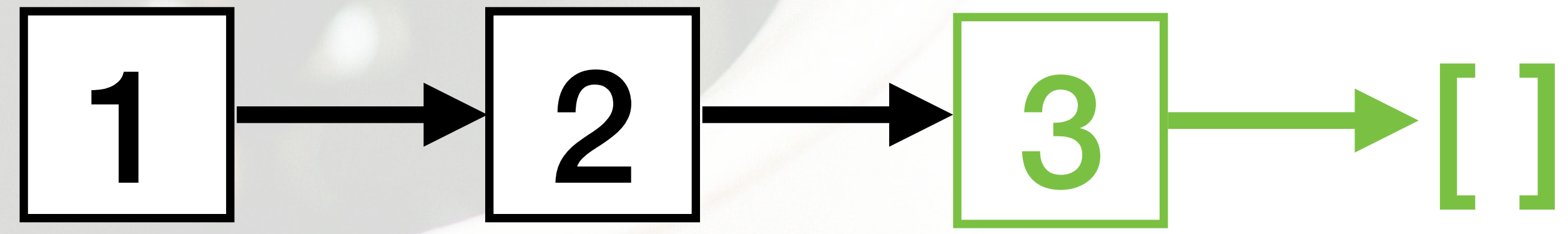
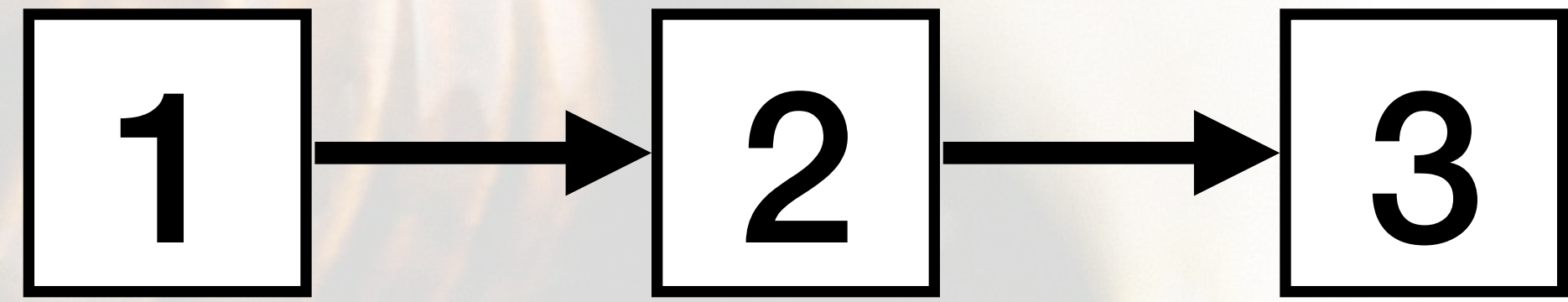




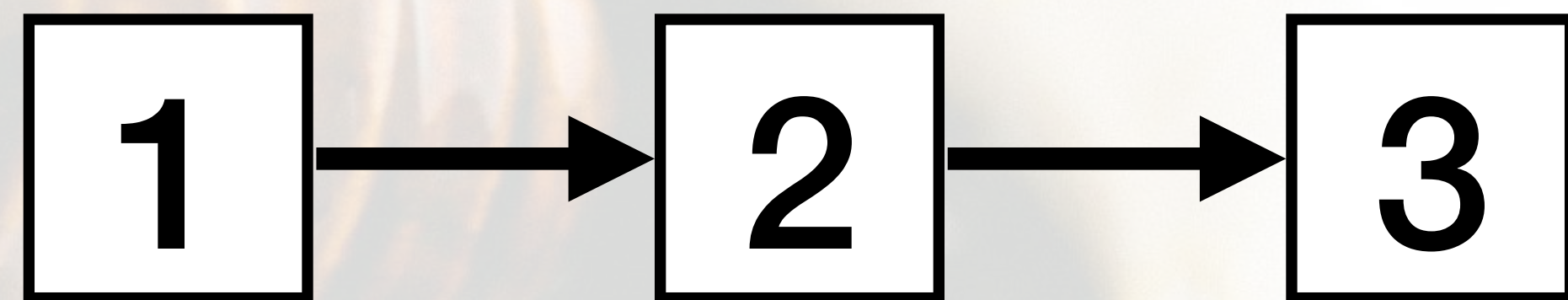




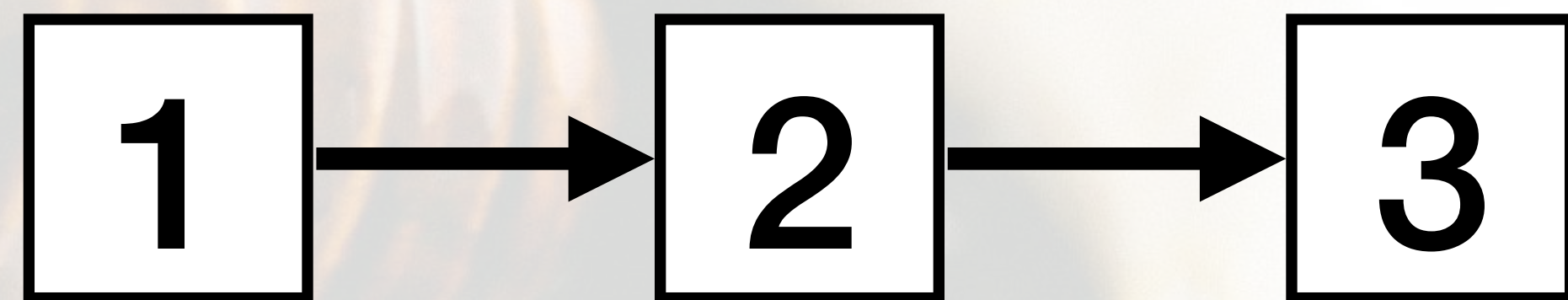




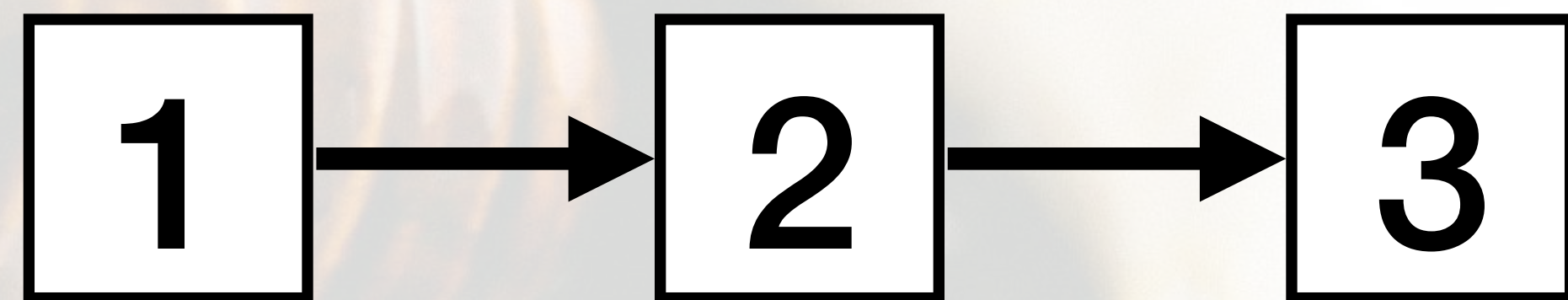




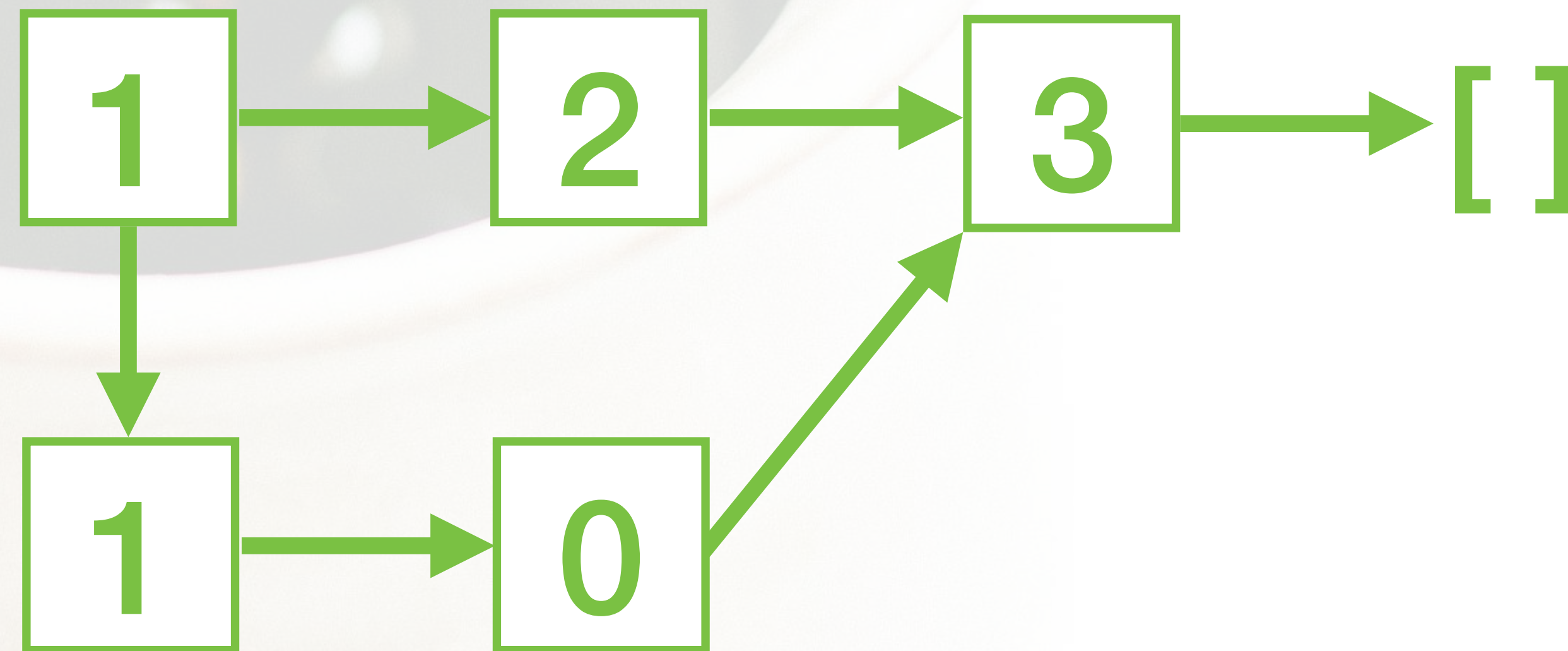
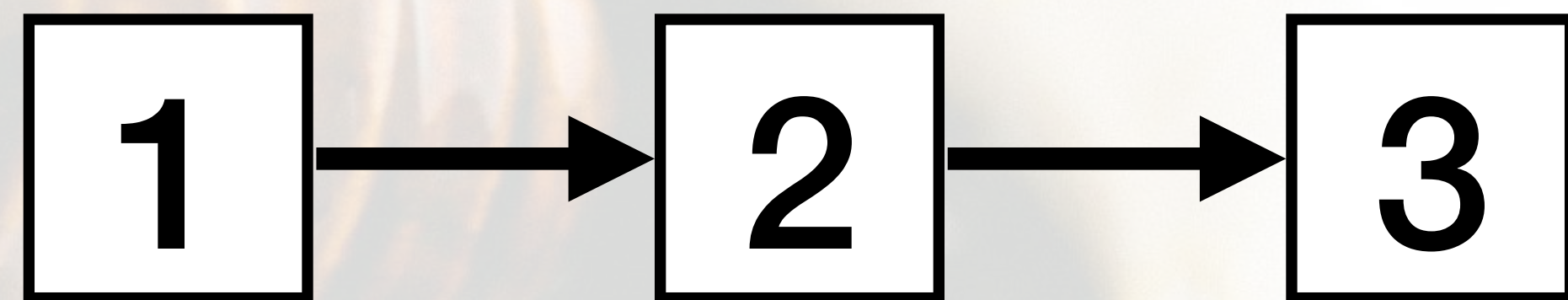




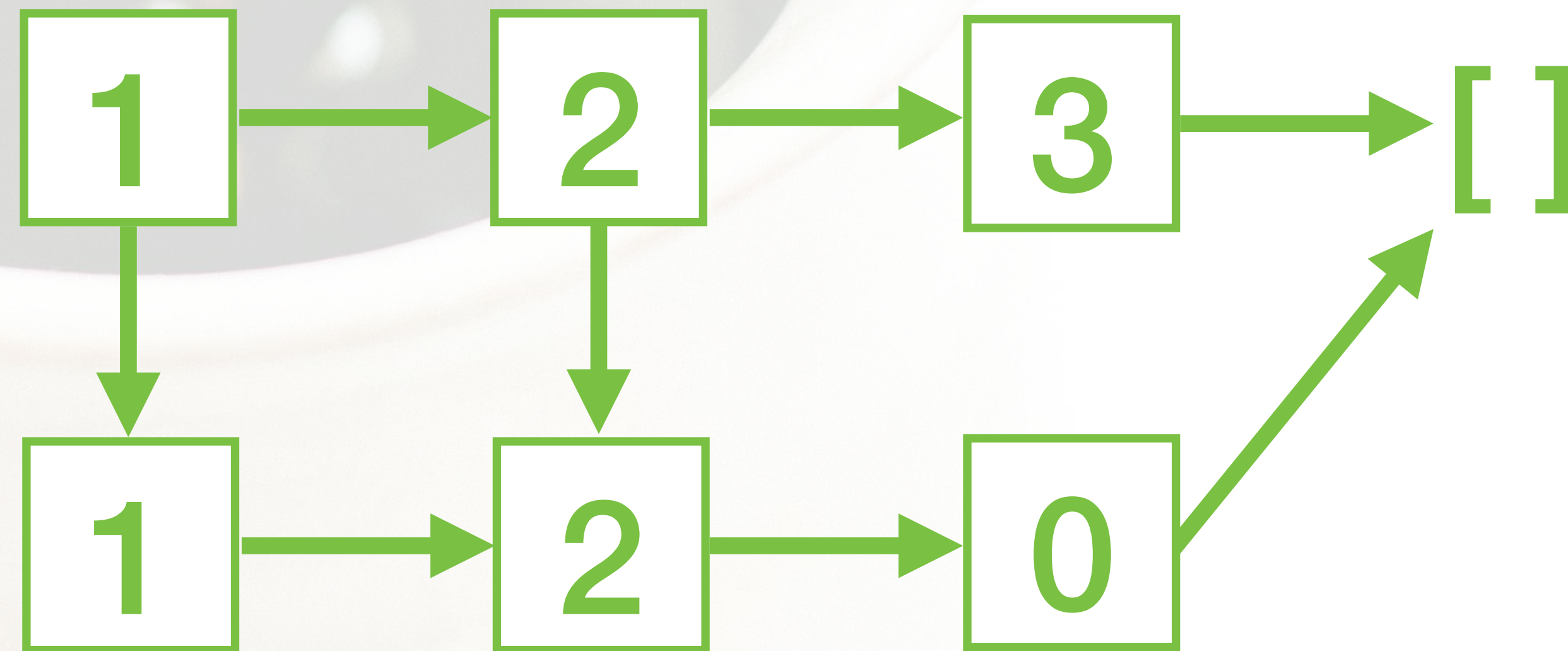
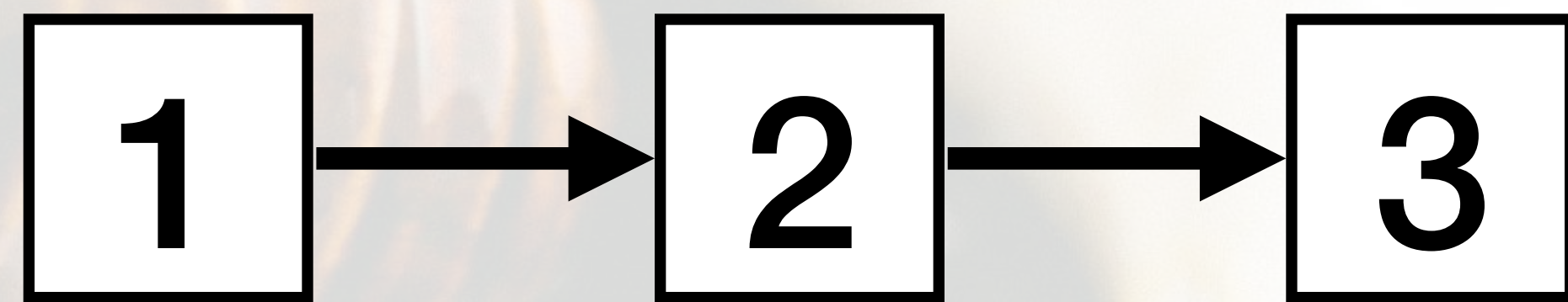
















Embrace

Immutability



A hand holding a wooden honey dipper over a bowl of honey. The background is a soft, warm yellow, and the honey is a rich golden color. The text is overlaid on the image in a bold, golden font.

**Bank:**

**- Balance**



A hand holding a wooden honey dipper over a bowl of honey. The background is a soft, warm yellow, and the honey is a rich golden color. The text is overlaid on the image in a large, bold, golden font.

**Bank:**  
**- Balance**  
**+ Transactions**



A wooden honey dipper with a spiral-shaped head is resting on a white plate. The dipper is coated with a thick layer of golden honey, which is dripping down its handle. The background is a soft, out-of-focus white surface.

# CRUD

create read update delete



A wooden honey dipper with a spiral-shaped head is resting on a white plate. The dipper is coated with honey, and some honey is visible on the plate. The background is a soft, out-of-focus white.

# CRUD

create read ~~update~~ ~~delete~~



A wooden honey dipper with a spiral-shaped head is resting on a white plate. The dipper is coated with honey, and a small amount of honey is visible on the plate. The background is a soft, out-of-focus white.

**Harmony**





Fun





Func-  
tions



A person wearing a full white protective suit, including a hood and a clear face shield, is working on a wooden structure. The person is leaning over, and their hands are visible near the wood. The background is a blurred wooden wall. The text 'Happy Place' is overlaid on the image in a large, golden-yellow, serif font.

# Happy Place





# The Core





**Pipe-land**

**The Core**





**Pipe-land**

**The Core**

**Certainty**





**Pipe-land**

**Sanitized**

**The Core**

**Certainty**





**Pipe-land**

**Sanitized**

**The Core**

**Certainty**

**Valid**





**Pipe-land**

**Sanitized**

**The Core**

**Certainty**

**Library**

**Valid**



A person wearing a full white protective suit, including a hood with a clear face shield and gloves, is kneeling on a wooden deck. They are holding a small, clear plastic cup in their hands. The background shows a wooden fence and some dry grass. The image is semi-transparent, serving as a background for the text.

Purity

is not the

Goal





Purity  
:random.uniform  
is not the  
Goal



A person wearing a full white protective suit, including a hood and a clear face shield, is working in a field. The person is leaning forward, possibly examining something on the ground. The background shows a wooden fence and some vegetation.

Purity

**:random.uniform**

is not the

**:DateTime.utcnow()**

Goal



A person wearing a full white protective suit, including a hood and mask, is shown in a field. They are leaning forward, possibly working with a tool or equipment. The background consists of wooden planks and some dry grass. The image is semi-transparent, serving as a background for the text.

Abstractions

are the

Goal



A person wearing a full white protective suit, including a hood with a clear face shield and gloves, is bent over in a field. They appear to be working with a small container or tool on the ground. The background shows a wooden fence and some dry grass. The image is semi-transparent, serving as a background for the text.

# Reducers



A person wearing a full white protective suit, including a hood and mask, is working on a wooden structure. They are holding a yellow container and appear to be applying something to the wood. The background is a wooden wall and some dry grass in the foreground.

**list = [1, 2, 3]**



A person wearing a full white protective suit, including a hood and mask, is bent over in a field, possibly working with plants or soil. The background shows a wooden fence and some dry grass.

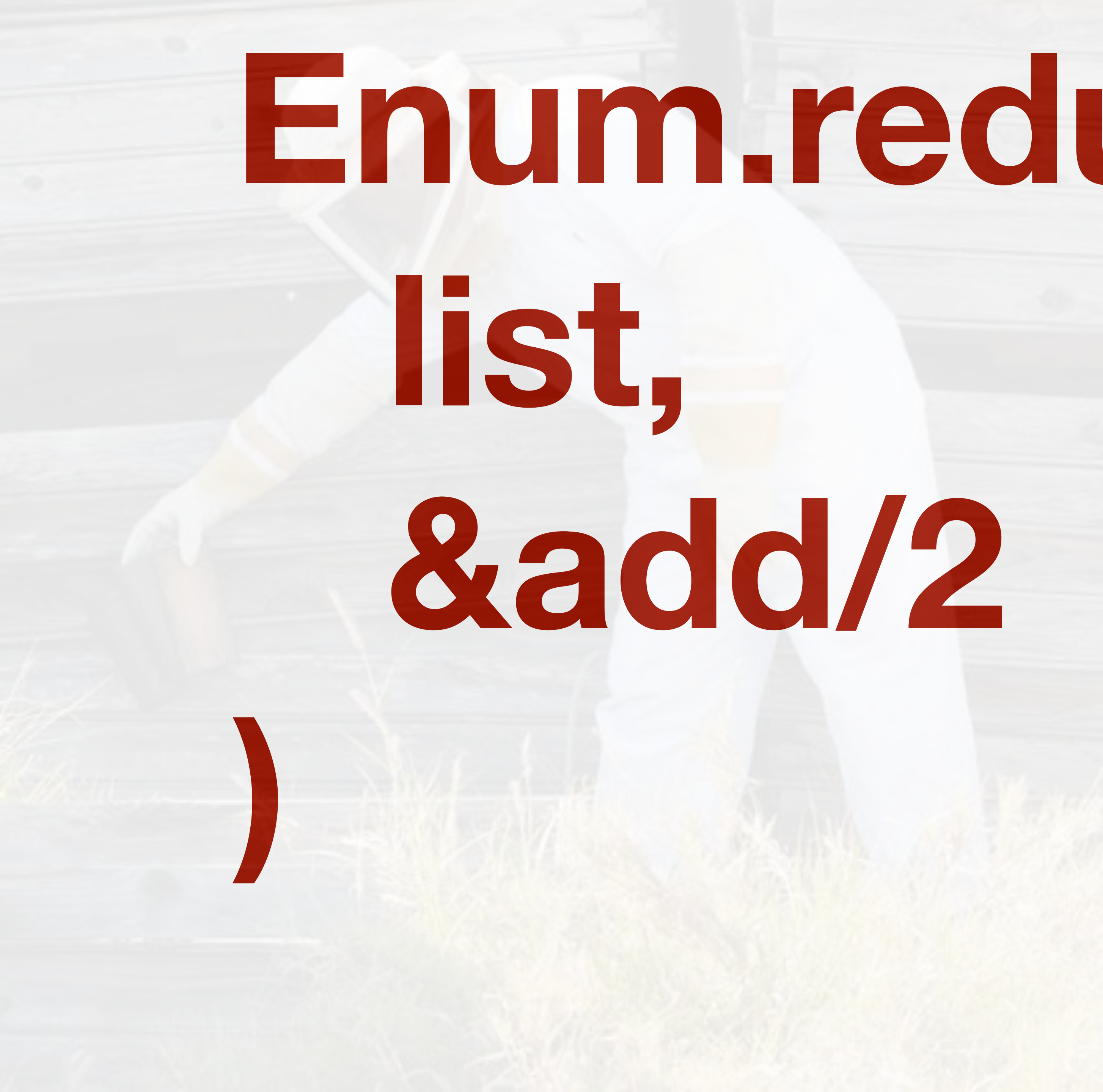
```
list = [1, 2, 3]
```

```
def add(x, y) do
```

```
  x + y
```

```
end
```



A person wearing a full-body white protective suit, including a hood and gloves, is kneeling in a field of tall grass. They appear to be working with a piece of equipment or a sample. The background is a blurred field of grass under a bright sky.

**Enum.reduce(  
list,  
&add/2  
)**



**Enum.reduce(**

**list,**

**&add/2**

**)**

**-> 6**



1





A person wearing a full white protective suit, including a hood and a clear face shield, is bent over and working on a wooden structure. The person is holding a tool, possibly a hammer, and is focused on their task. The background shows horizontal wooden planks and some dry grass in the foreground.

**1**

**|> add(2)**



A person wearing a white protective suit, a white hard hat, and a clear face shield is working on a wooden structure. The person is leaning forward, and their hands are near the wood. The background is a wooden wall with horizontal planks. The foreground shows some dry grass.

**1**

**|> add(2)**

**|> add(3)**



A person wearing a white protective suit, a white hard hat, and a clear face shield is working in a field. They are leaning forward, possibly examining something on the ground. The background shows a wooden fence and some dry grass.

**1**

**| > add(2)**

**| > add(3)**

**-> 6**

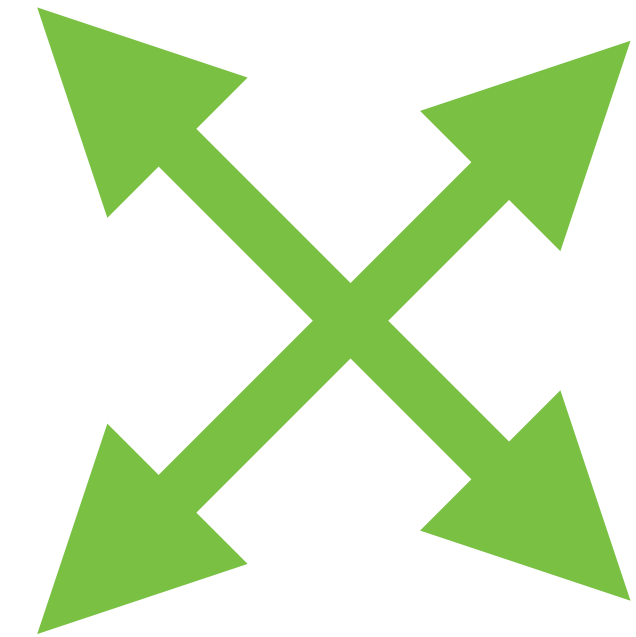




I cheated....



**def reducer(item, acc)**



**def piped(acc, item)**



A person wearing a full white protective suit, including a hood and a clear face shield, is working on a wooden structure. The person is leaning over, and their hands are visible near the wood. The background is a blurred wooden wall. The text 'Happy Place' is overlaid on the image in a large, golden-yellow, serif font.

# Happy Place





Things





Tests





Predictable



Test the Core



# Property Based Test



```
non_empty( list integer() )
```

```
▷ forall( fn list →
```

```
    biggest(list) = list ▷ sort ▷ last
```

```
end)
```



# Generator

```
non_empty( list integer() )
```

```
▷ forall( fn list →
```

```
  biggest(list) = list ▷ sort ▷ last
```

```
end)
```



# Property

```
non_empty( list integer() )
```

```
▷ forall( fn list →
```

```
  biggest(list) = list ▷ sort ▷ last
```

```
end)
```



```
non_empty( list integer() )
```

```
▷ forall( fn list →
```

```
    biggest(list) = list ▷ sort ▷ last
```

```
end)
```



# Props to...

<https://jeffkreeftmeijer.com/mix-proper/>

Jeff Kreeftmeijer



# Props to...

John Hughes



Props to...

Fred Hebert



# Pipes in Tests



```
[{1, 1}]
```

```
▷ Points.mirror
```

```
▷ assert_point({4, 1})
```

```
▷ Points.flip
```

```
▷ assert_point({4, 4})
```

```
▷ Points.rotate_90
```

```
▷ assert_point({1, 4})
```

```
▷ Points.rotate_90
```

```
▷ assert_point({1, 1})
```



```
[{1, 1}]
```

```
▷ Points.mirror
```

```
▷ assert_point({4, 1})
```

```
▷ Points.flip
```

```
▷ assert_point({4, 4})
```

```
▷ Points.rotate_90
```

```
▷ assert_point({1, 4})
```

```
▷ Points.rotate_90
```

```
▷ assert_point({1, 1})
```

Reducer  
with  
side  
effect



```
def assert_point([actual], expected) do
  assert actual == expected
  [actual]
end
```



```
def assert_point([actual], expected) do
  assert actual == expected
  [actual]
end
```



```
def assert_point([actual], expected) do
  assert actual == expected
  [actual]
end
```



```
[{1, 1}]
```

```
▷ Points.mirror
```

```
▷ assert_point({4, 1})
```

```
▷ Points.flip
```

```
▷ assert_point({4, 4})
```

```
▷ Points.rotate_90
```

```
▷ assert_point({1, 4})
```

```
▷ Points.rotate_90
```

```
▷ assert_point({1, 1})
```



# Streams for Impure



```
def eventually_match(generators, number) do
  Stream.repeatedly(fn →
    build_question(generators: generators)
      .substitutions
  end)
  ▷ Enum.find(fn substitution →
    Keyword.fetch!(substitution, :left) == number
  end)
end
```



```
def eventually_match(generators, number) do
  Stream.repeatedly(fn →
    build_question(generators: generators)
    .substitutions
  end)
  ▷ Enum.find(fn substitution →
    Keyword.fetch!(substitution, :left) == number
  end)
end
```



```
def eventually_match(generators, number) do
  Stream.repeatedly(fn →
    build_question(generators: generators)
      .substitutions
  end)
  ▷ Enum.find(fn substitution →
    Keyword.fetch!(substitution, :left) == number
  end)
end
```





Predictable



# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



"Do Fun Things  
with  
Big, Loud  
Worker-Bees"

by James E Gray, II  
and Bruce A. Tate





Big





# Boundaries



A person wearing a white protective suit, including a helmet and gloves, is working in a field. They are holding a clear plastic container and standing next to an open cardboard box. The background shows a field with some structures in the distance. The word "Maybe" is overlaid in a large, golden, serif font.

Maybe



# Boundaries



**Not Just OTP**

**Boundaries**



**Not Just OTP**

**Boundaries.**  
**Complexity**

**Uncertainty**



Not Just OTP

Unsanitized

Process

Boundaries.

Complexity

Uncertainty



With Land



```
def build_quiz(fields) do
  with :ok ← QuizValidator.errors(fields),
       :ok ← GenServer.call(..) do
    :ok
  else
    error →
      error
  end
end
```



But I want...



# More Pipes



# More Reducers



More

Happiness







Skinny

Handlers



```
def handle_event("keydown", %{"key" => "Arro  
  {:noreply, move(:right, socket)}  
end
```

```
def handle_event("start", _, socket) do  
  {:noreply, new_game(socket)}  
end
```



Represent

Data as Errors



```
def work(n) do
  if :rand.uniform(10) == 1 do
    raise "Oops!"
  else
    {:result, :rand.uniform(n * 100)}
  end
end
end
```



```
def make_work_safe(dangerous_work, arg) do
  try do
    apply(dangerous_work, [arg])
  rescue
    error →
      {:error, error, arg}
  end
end
```



```
def stream_work do
  Stream.iterate(1, &(&1 + 1))
  ▷ Stream.map(fn i →
    make_work_safe(&work/1, i)
  end)
end
```

```
end
```



Error Data

Composes



A person wearing a white protective suit, including a helmet and gloves, is working in a field. They are holding a small container and standing next to a large, open cardboard box. The background shows a field with some structures in the distance. The word "Maybe" is overlaid in a large, golden, serif font across the center of the image.

Maybe



A close-up photograph of a bee on a yellow flower. The bee is covered in bright yellow pollen, particularly on its head and thorax. Its wings are partially spread, and its legs are visible as it moves on the flower. The background is a soft, out-of-focus yellow, suggesting other parts of the flower or a similar environment. The word "Loud" is written in a large, white, serif font in the upper right corner of the image.

Loud



# Lifecycles







# Paradigm



```
def start(_type, _args) do  
    children = [
```



QuizManager.lookup\_quiz\_by\_title



```
Supervisor.start_link(children, opts)
```



*strategy: :one\_for\_one*



# Clean Startup



Clean Startup

+ Clean Shutdown



Clean Startup

+ Clean Shutdown

+ Supervision



OTP goodness





# Paradigm





# Worker Bees





workers



The background of the image is a close-up, slightly blurred view of a honeycomb. The hexagonal cells are filled with a golden-yellow substance, likely honey. Numerous bees are scattered across the surface, some appearing to be in motion. The overall color palette is warm, dominated by yellows and oranges.

# Balance



Imbalance

→ boundary

Imbalance

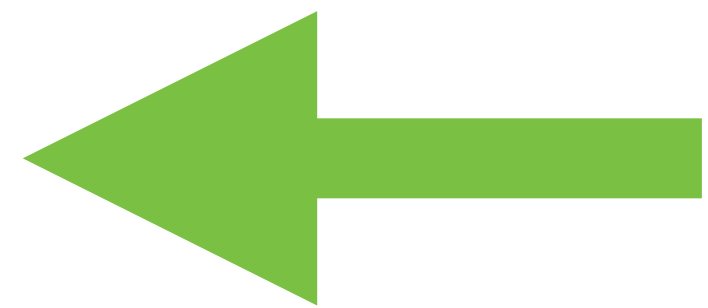
~~XX~~

→ boundary



# Imbalance

core



Delay

Persistence

Decision



```
def answer_question(name, answer,  
persistence_fn \\ @persistence_fn)
```



```
def handle_call({:answer_question, answer, fun})
  fun = fun || fn r, f -> f.(r) end
  response = Response.new(quiz, email, answer)
  fun.(response, fn r ->
    quiz
    |> Quiz.answer_question(r)
    |> Quiz.select_question
  end)
  |> maybe_finish(email)
end
```



The background of the image is a close-up, slightly blurred view of a honeycomb. The hexagonal cells are filled with a golden-yellow substance, likely honey. Numerous bees are scattered across the surface, some appearing to be in motion. The overall color palette is warm, dominated by yellows and oranges.

# Balance



# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



# "Do Fun Things with Big, Loud Worker-Bees"

by James E Gray, II  
and Bruce A. Tate



# Designing Elixir Systems with OTP

Write Highly Scalable,  
Self-Healing Software with Layers



"Do Fun Things  
with  
Big, Loud  
Worker-Bees"

by James E Gray, II  
and Bruce A. Tate



The background features a repeating pattern of honeycomb cells in shades of yellow and orange. Numerous bees are scattered across the scene, some in flight and others on the honeycomb, creating a busy, naturalistic setting.

James E.

Gray II



