

BE A LIBRARY DEVELOPER!

Remember than Imagine

Kouhei Sutou

ClearCode Inc.

2013/06/01

Table of contents

- ✓ Share this talk's goal
- ✓ Describe key idea
- ✓ Apply the key idea
- ✓ Wrap up & The next step

Share the goal

- ✓ Share this talk's goal
- ✓ Describe key idea
- ✓ Apply the key idea
- ✓ Wrap up & The next step

This talk's goal

You know
how to develop better softwares

Better?

Show by example

API

```
context.save  
context.circle(50, 50, 10)  
context.stroke  
context.restore
```

from cairo gem

API: Point

```
context.save      # <- Point  
context.circle(50, 50, 10)  
context.stroke  
context.restore  # <- Point
```

from cairo gem

API: Better

```
context.save do
  context.circle(50, 50, 10)
  context.stroke
end
```

from cairo gem

API: Why

Why is it better?

API: Because

Because it's more Ruby-ish

Ruby-ish?

-ish ==

Similar to others

Is "similar" better?

- ✓ Similarity makes consistency
- ✓ Consistency makes readability
- ✓ Readability is important to develop better softwares

Is "readability" important?

- ✓ Readability makes maintainability
- ✓ Maintainability is needed for improving your softwares repeatedly

So

Ruby-ish is
better

Let's validate
the example

Remember

File

Remember: File

```
# Setup  
file = File.open(path)  
file.read  
# Teardown  
file.close
```

Remember: File

```
# Setup  
File.open(path) do |file|  
  file.read  
end # Teardown
```

API (reprise)

```
context.save      # Setup  
context.circle(50, 50, 10)  
context.stroke  
context.restore  # Teardown
```

from cairo gem

API: Better (reprise)

```
context.save do # Setup
  context.circle(50, 50, 10)
  context.stroke
end # Teardown
```

from cairo gem

Wrap up

Better

Ruby-ish

In other words

Similar to others

The goal (reprise)

You know
how to develop better softwares

In other words

You know
what is "similar"
and do similarly

Describe key idea

- ✓ Share this talk's goal
- ✓ Describe key idea
- ✓ Apply the key idea
- ✓ Wrap up & The next step

Key idea

Remember
than
Imagine

Remember than Imagine

Easy! 

Knowledge



Remember



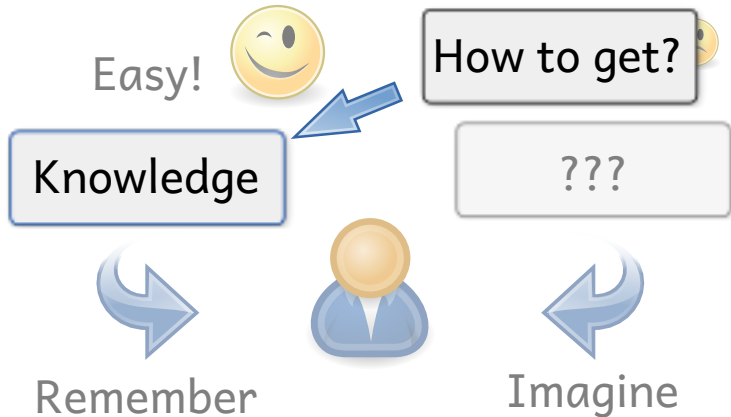
Difficult... 

???



Imagine

To remember,



How to get knowledge?

- ✓ Experience
- ✓ Ask
- ✓ Observe

Experience!

- ✓ Experience ← Do this first!
- ✓ Ask
- ✓ Observe

Key idea

Remember
than
Imagine

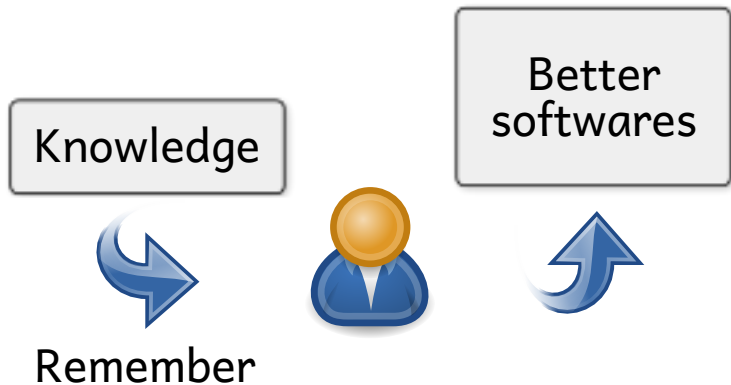
Apply the key idea

- ✓ Share this talk's goal
- ✓ Describe key idea
- ✓ Apply the key idea
- ✓ Wrap up & The next step

The goal (reprise)

You know
how to develop better softwares

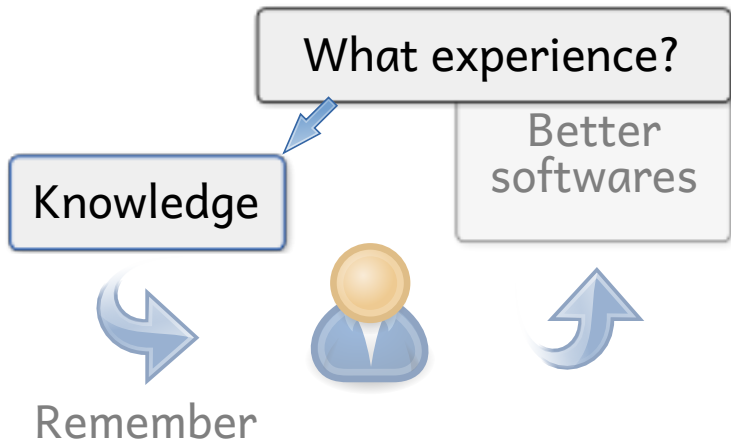
Apply the key idea



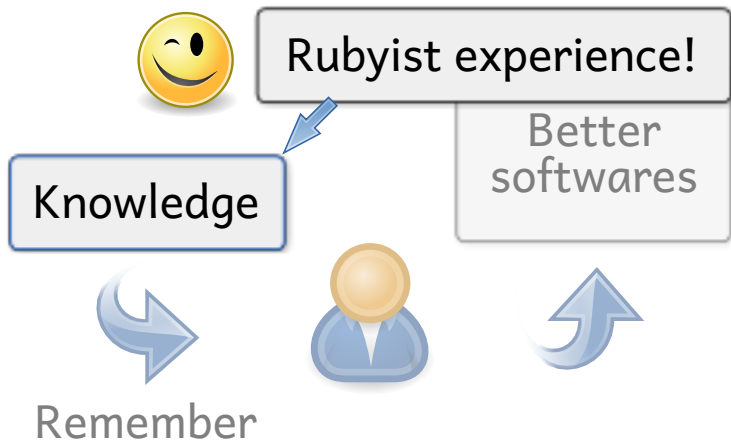
To achieve the goal,

What should
you experience?

What experience?



Rubyist



Rubyist?

You experienced!

Try

API: Getter

```
window.get_property("opacity")  
# What is better API???
```

from gtk2 gem

API: Getter (better)

```
window.get_property("opacity")  
window.opacity # Better
```

from gtk2 gem

Now,

You found
"remember" is
"difficult"

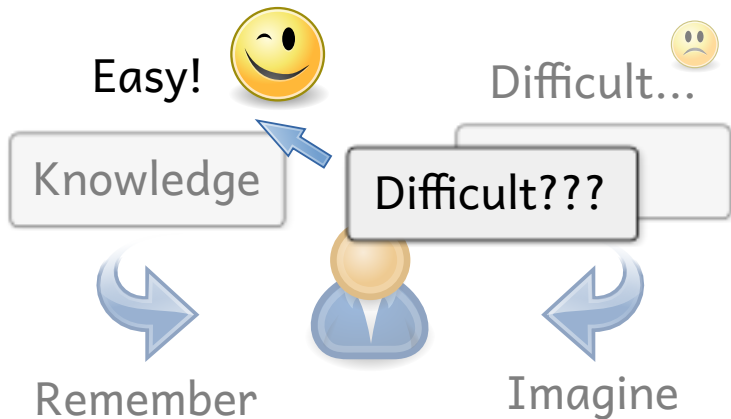
Difficult!

Remember!?

What is

"Ruby-ish"!?

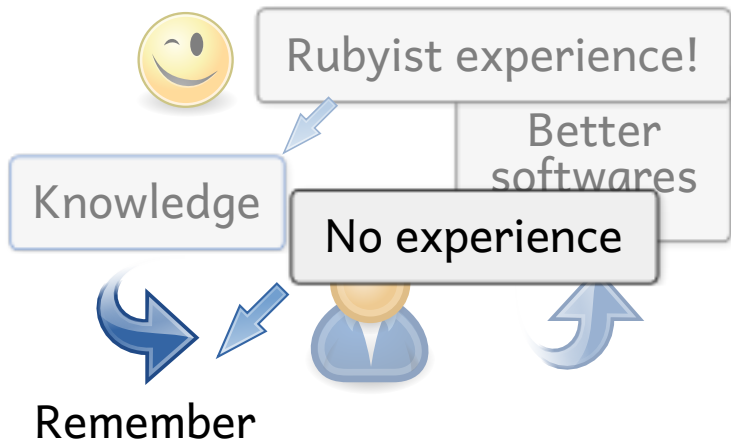
Difficult?



Because

You don't have
experience about
"Remember than
Imagine"

No "remember" experience



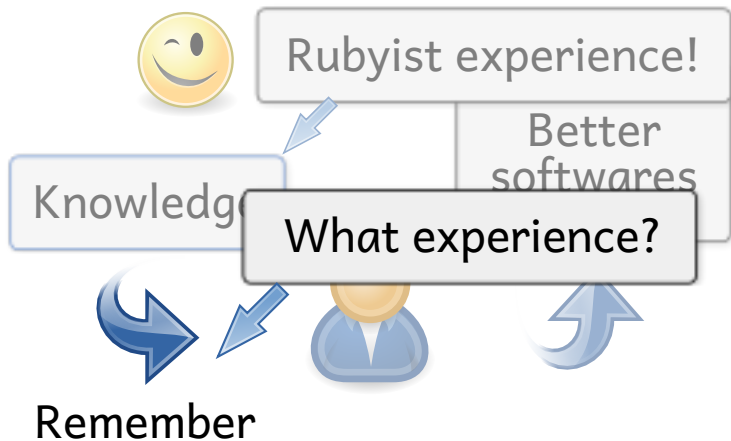
Your status

- ✓ Experience
- ✓ Ask (Heard) ← You are here!
- ✓ Observe

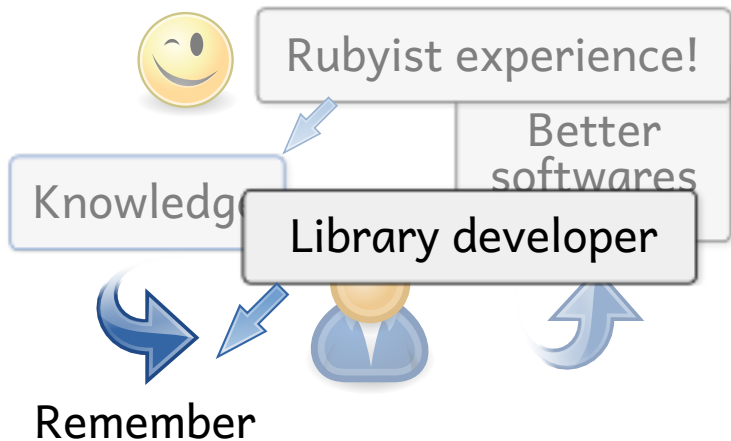
To achieve the goal,

What should
you experience?

What experience?



Library developer



Library developer

Considers about

- ✓ easy to use API
 - ✓ as a Rubyist
- ✓ easy to understand document
 - ✓ as a library user

Library developer

Considers them
many times

"Many times"

It's very good
exercise

Exercises

✓ API

✓ Document

API

API: Getter (reprise)

```
# Low level  
window.get_property("opacity")  
# Better  
window.opacity
```

from gtk2 gem

API: Better?

```
# Low level  
window.get_property("visible")  
# Better  
# ???: hint: "visible" is bool
```

from gtk2 gem

API: Better

```
# Low level  
window.get_property("visible")  
# Better  
window.visible?
```

from gtk2 gem

API: Getter

```
# Better for record as collection  
record["name"]  
# Better for record as object  
record.name
```

from rroonga gem

API: Init

```
require "gst"  
# ???  
Gst.init # <- What?  
Gst::ElementFactory.make("playbin")
```

from gstreamer gem

API: Init (usage)

```
require "gst"  
# For advanced use  
Gst.init("--gst-debug=*:9")  
Gst::ElementFactory.make("playbin")
```

from gstreamer gem

API: Init (better)

```
require "gst"  
# Make optional  
# Gst.init  
Gst::ElementFactory.make("playbin")
```

from gstreamer gem

API: Init (impl)

```
class << Gst
  def const_missing(name)
    init; const_get(name); end
  def init(*argv)
    # ...initialize library...
    class << self
      remove_method(:const_missing)
    end; end; end
```

from gstreamer gem

Document

Document: Install

Install:

For Debian GNU/Linux:

```
% sudo apt-get install libgtk2.0-dev
```

```
% gem install gtk2
```

For OS X:

...

Document: Better

Install:

```
% gem install gtk2
```

libgtk2-0-dev is installed automatically

"gem install GEM_NAME" is popular install way

Exercises end

- ✓ We remembered about "what is similar?"
- ✓ We done "similarly"

Do you understand

about

"Remember than

Imagine"?

Wrap up

- ✓ Share this talk's goal
- ✓ Describe key idea
- ✓ Apply the key idea
- ✓ **Wrap up** & The next step

This talk's goal

You know
how to develop better softwares

Better

Ruby-ish

Similar to others

Goal: In other words

You know
what is "similar"
and do similarly

Key idea

Remember
than
Imagine

Remember than Imagine

Easy! 

Knowledge



Remember



Difficult... 

???

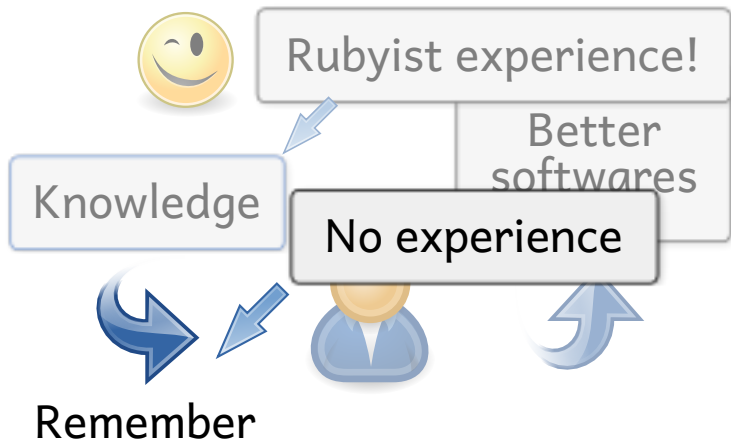


Imagine

For development

- ✓ Experience Rubyist for knowledge
- ✓ Then **remember** the knowledge
 - ✓ But it is difficult because you don't have experience about "**remember**"

No "remember" experience



Experience lib developer

It's very good exercise

The next step

- ✓ Share this talk's goal
- ✓ Describe key idea
- ✓ Apply the key idea
- ✓ Wrap up & **The next step**

The next step

Use "library developer"
experience to other
things

For example,
Use it to
develop other
softwares

Develop other softwares

You can remember about

- ✓ a better bug report
 - ✓ How to reproduce?
- ✓ a better patch
 - ✓ How to commit? (size? message?)

Conclusion

Be a library
developer!

FYI

ClearCode

- ✓ We accept applications about our internship
- ✓ You can develop a library with us

Bonus tracks

- ✓ Similar vs. Innovation
- ✓ No "imagine" communication

Similar

vs.

Innovation

Innovation

- ✓ Doing "similarly" will not achieve "innovation"
- ✓ Should all we aspire innovation?

No

- ✓ We have regular life
- ✓ It's also important that we improve our regular life

No

"imagine"

communication

No "imagine"

- ✓ Show all u need from the start
 - ✓ Don't omit anything
 - ✓ If you emit something, other people need "imagine"
- ✓ Use your experience for it!

Key idea

Remember
than
Imagine