Calling Java from Ruby

http://bit.ly/
java-from-ruby

REPL

```
$ jruby -S irb
irb(main):001:0> █
```

Static Fields vs. Constants

Java

```
Locale.US
```

Ruby

```
Locale::US
```

CamelCase vs. snake_case

Java

```
System.currentTimeMillis()
```

Ruby

```
System.current_time_millis
```

Properties vs. Accessors

Java

```
locale.getLanguage()
date.getTime()
date.setTime(0)
file.isDirectory()
```

Ruby

```
locale.language
date.time
date.time = 0
file.directory?
```

Conversions

- primitives: numbers, strings, booleans, nil
- collections:
  - arrays -> lists
  - hashes -> maps
- attempt "principle of least surprise"

Conversions

```
ruby_array = [10, 5, 1]
java.util.Collections.sort(ruby_array)
ruby_array
# => [1, 5, 10]
```

Extending Java

```
h = java.util.HashMap.new
h["key"] = "value"
h["key"]
  # => "value"
h.get("key")
  # => "value"
h.each {|k,v| puts k + ' => ' + v}
  # key => value
h.to_a
  # => [["key", "value"]]
```

Extending Java

```
module java::util::Map
  include Enumerable

  def each(&block)
    entrySet.each { |pair| block.call([pair.key, pair.value]) }
  end

  def [](key)
    get(key)
  end

  def []=(key,val)
    put(key,val)
    val
  end
end
```

Conversion Helpers

```
["a", "b", "c"].to_java :string
  # => new String[] {"a", "b", "c"}
```

```
import java.io.FileOutputStream
stream = FileOutputStream.new("/tmp/hello.txt")
stream.write("Hello".to_java_bytes)
stream.close
```

Interface Conversion

- **what happens if...**

```
package java.util.concurrent;
public class Executors {
    // ...
    public static Callable callable(Runnable r) {
        // ...
    }
}
```

Interface Conversion

```
class SimpleRubyObject
end

import java.util.concurrent.Executors
callable = Executors.callable(SimpleRubyObject.new)
callable.call

# => undefined method `run' for #<SimpleRubyObject:0xfd5428>
(NoMethodError)

class SimpleRubyObject
  def run
    puts "hi"
  end
end
callable.call
# => hi
```

Closure Conversion

- taking that last example further...

```
import java.util.concurrent.Executors
callable = Executors.callable { puts "hi" }
callable.call
# => hi
```

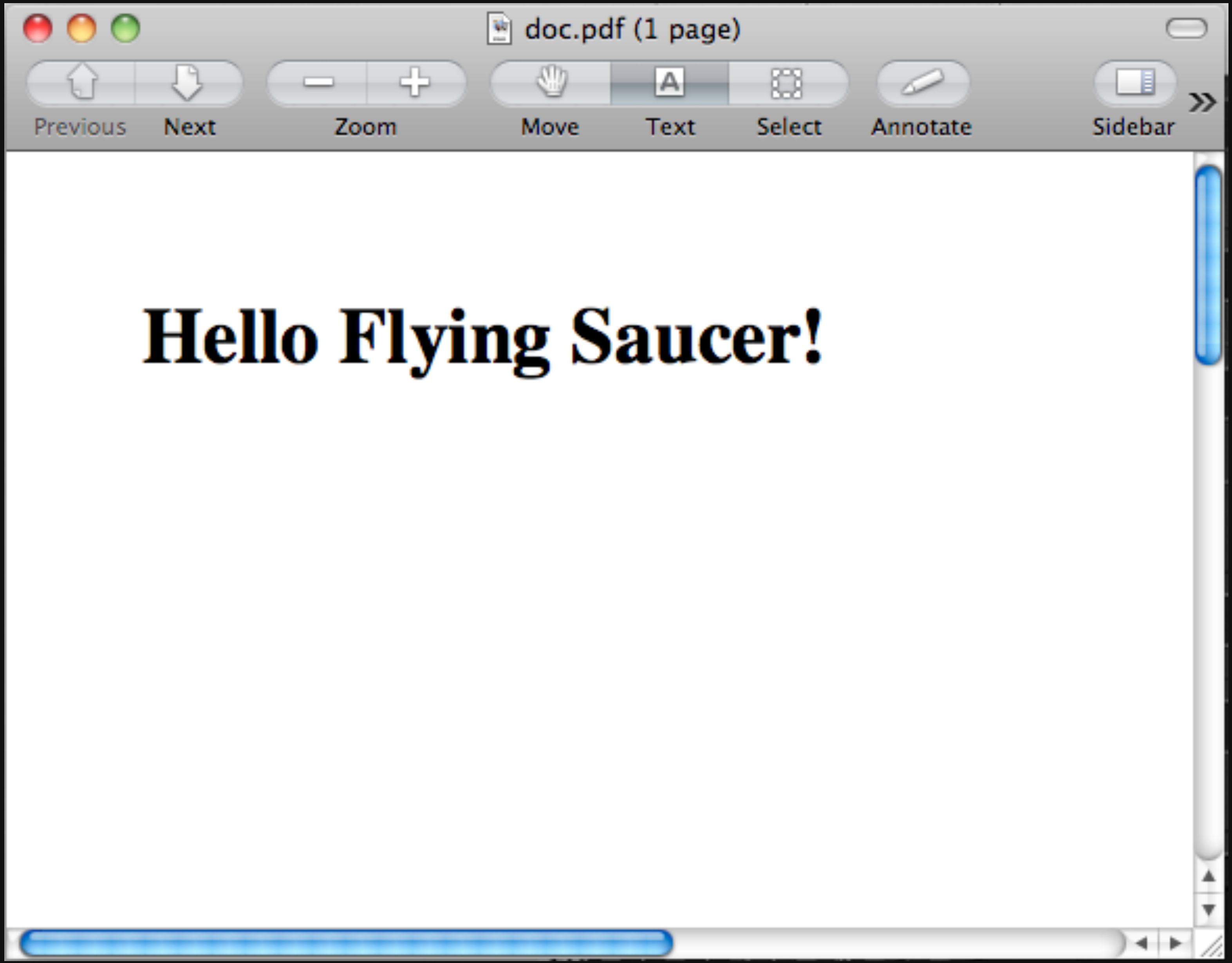
```
button.add_action_listener do |event|
  event.source.text = "Pressed!"
end
```

```ruby
require 'java'
require 'flying_saucer'

java_import org.xhtmlrenderer.pdf.ITextRenderer

document = <<-HTML
<html><body><h1>Hello Flying Saucer!</h1></body></html>
HTML

File.open("doc.pdf", "wb") do |out|
  renderer = ITextRenderer.new
  renderer.set_document_from_string document
  renderer.layout
  renderer.create_pdf out.to_outputstream
end
```



Hello Flying Saucer!

```
require 'java'
```

```
java_import org.xhtmlrenderer.pdf.ITextRenderer
```

Ruby

```ruby
renderer = ITextRenderer.new
```

```java
ITextRenderer renderer = new ITextRenderer();
```

Java

Ruby

```
renderer.set_document_from_string document
```

```
renderer.setDocumentFromString(document);
```

Java

```
File.open("doc.pdf", "wb") do |out|
  ...
  renderer.create_pdf out.to_outputstream
end
```

explicit conversion

Embed

```java
import org.jruby.embed.ScriptingContainer;

public class EmbedJRuby {
    public static void main(String[] args) {
        ScriptingContainer container =
            new ScriptingContainer();
        container.runScriptlet(
            "puts 'Hello from Ruby'");
    }
}
```

```java
// import org.jruby.embed.EmbedEvalUnit;

EmbedEvalUnit unit =
    container.parse("'Ruby' * @times");

container.put("@times", 2);
System.out.println(unit.run()); // RubyRuby

container.put("@times", 4);
System.out.println(unit.run()); // RubyRubyRubyRuby
```

http://wiki.jruby.org/RedBridge

Compile

```ruby
require 'java'
java_package 'demo'

class Engine
  java_implements 'java.lang.Runnable'

  java_signature 'void run()'
  def run
    puts "The #{self.inspect} is running."
  end
end
```

```java
package demo;

public class Engine implements Runnable {
    public void run() { ... }
}
```

(generated) demo/Engine.java

```java
import demo.Engine;

public class Starter {
    public static void main(String[] args) {
        Engine engine = new Engine();
        engine.run();
    }
}
```

Starter.java

```
$ jrubyc --javac compile.rb Starter.java
Generating Java class Engine to demo/Engine.java
javac -cp jruby.jar:. demo/Engine.java Starter.java

$ java -cp jruby.jar:. Starter
The #<Engine:0x59c958af> is running.
```