

言語としての SHIORI

華和梨開発チーム
さとー
2010/9/18



この資料のライセンスは creative commons の「表示 2.1 日本」です。

Copyright S.Sato Some Rights Reserved.

Sep. 18th 2010

伺的ソフトウェア勉強会 大阪 #6

SHIORI って？

- ・ ゴースト記述用プログラミング言語の通称です。
- ・ 「言語ではない」と言われることもあります
- ・ が、なんだかんだいってプログラミング言語です。
- ・ 以下、「言語」 = 「プログラミング言語」として記述します



SHIORI の特徴？

ゴーストのしゃべりを記述するために特化しています。

- eruby、PHP の類似のテンプレートエンジン
- テキスト処理が得意
- ¥のエスケープが特殊（さくらスクリプトに合わせている）
- 使う人がプログラマじゃないことが多い

SHIORI の厄介な点

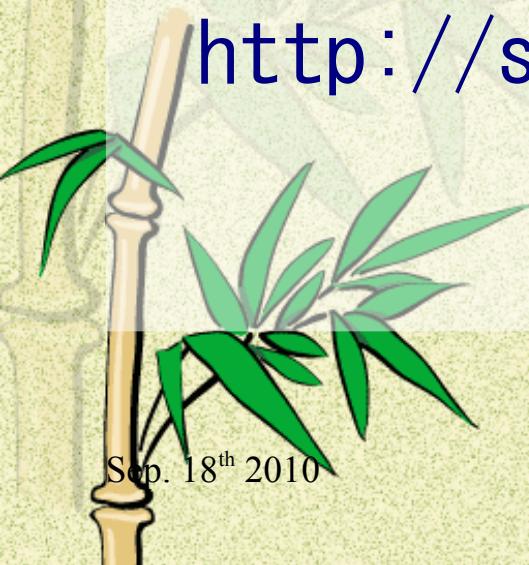
- nar (=zip) アーカイブを本体ソフトにDnDで、すぐに使用可能であることが求められる
- 作成者と実配布者（=ゴースト作者）が異なり再配布に当たるため、ライセンスが要注意

⇒Perl、Ruby、Python の既存実装は適用困難
(実装例は存在しますが、少数)

SHIORI もやる時はやります

本体と組んでやる気になれば、ゲームだってアプリだって組めます

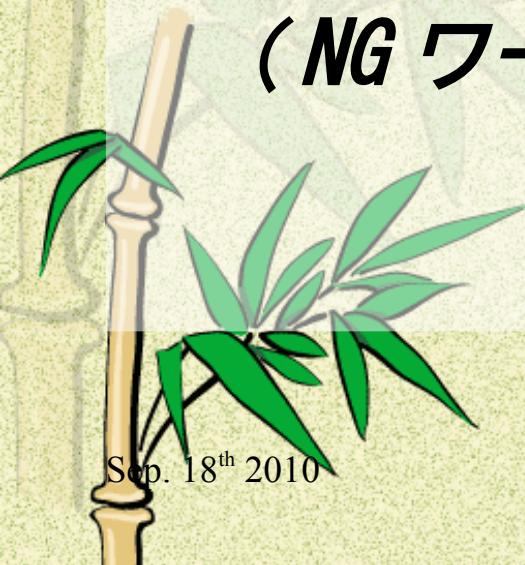
- 例：「となりの羽山さん」・・・ゲーム
<http://macapeng.web.fc2.com/>
- 例：「COLORS」・・・アプリ
<http://sites.google.com/site/colorsprj/>



でも、最近停滞気味ですね

- ここ5年ほど、メンテナンスぐらいしか行われていない
- 一般的の言語は、今でも日進月歩で進化してます
- 取り残されてない？

(NGワード：「おまえが言うな」)



・・・というわけで

- SHIORI の言語としての側面を点検しよう！
- ちゃんと「いまどき風のことが出来るか」見てみましょう
- 「いまどきのエンジニア」に見放されたら、表現の幅で他から取り残されるよ！（マジで）



点検対象

- 華和梨
- 里々（整備班力スタッフ）
- YAYA

他にもありますが、今回は割愛します。



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

そもそも「言語」とは？

- 根本は「人間のやりたいことを、機械に分かる形式で書いたもの」
- 人間の言葉は機械には難しかったり、機械で実行するには曖昧すぎるるので、専用の言語を作つて機械に伝えています



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

もうちょっと細かく言語について

- 記憶している論理に従い、入出力を処理する
 - 記憶・・・データ、論理の両方
 - 論理・・・「AならばB」等
 - I/O ・・・キーボード、ファイル、画面、音、etc...



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

わからん！ゴーストに例えて！

- 記憶 = トーク台本
- ロジック = ユーザの操作から動作を決定するスクリプト
- I/O = トークや画像を返す、ユーザの操作を受け付ける



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

いまどきの言語が備える機能

最低限必須

- 構造化
- ライブラリ

あって欲しい

- オブジェクト指向支援
- 関数プログラミング支援
- ガーベージコレクション

あると便利

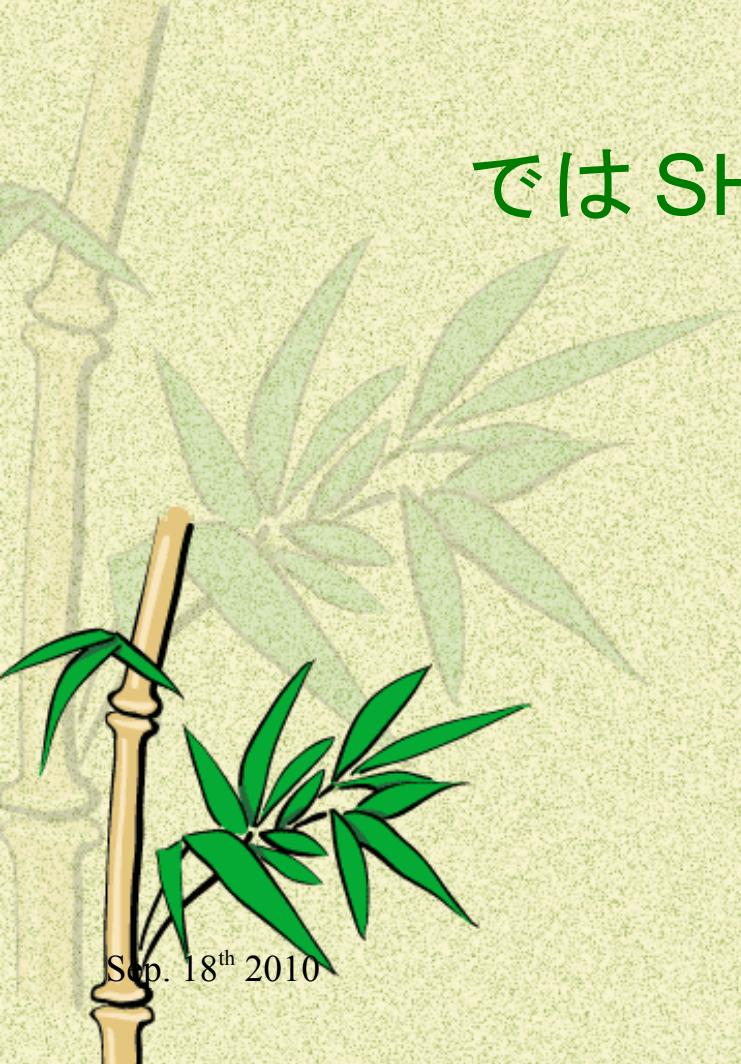
- DBへの接続
- メタプログラミング
- コルーチン
- クロージャ
- 型推論
- 並列実行



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6



では SHIORI を点検しましょう

Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

構造化

if-then、ループ、サブルーチン等のことです

- 華和梨 : ○
- 里々 : △
- YAYA : ○
- 一応すべてが備えています
- 里々は if が外部委託、自力ではジャンプ、サブルーチンのみ

条件分岐

- ・ 華和梨 : ○
 - if-then のみ（エントリコールで case 代用は可能）
- ・ 里々 : △
 - 条件付ジャンプはあるが、if-then は外部に丸投げ
- ・ YAYA : ◎
 - if-then、select-case(case-when) あり

YAYA が一番優等生です。

ループ

- 華和梨 : ○
 - 条件付きループ while、無条件回数ループ loop、foreach、break、continue あり
 - 里々 : △
 - loop あり (条件付きジャンプで代用は可能)
 - YAYA : ○
 - 条件付きループ while、for、foreach、break、continue あり
- 後置条件付きループ (do-loop) はどこも備えていませんね。

サブルーチン

- 華和梨：あり（引数渡し可能な関数が変数と別に存在する）
- 里々：あり（引数渡し・値戻し）
- YAYA：あり（引数渡し・値戻し）

メモ：

- YAYA はデータ構造までがサブルーチン。
- 里々はデータエントリを関数に出来る
- 華和梨は関数とデータ構造が概念上は別。
 - （実装で一致している）

オブジェクト指向

- 華和梨：○
 - 構造体支援あり、オブジェクトとして扱える
 - OpenKEEPS で実際にオブジェクトとして使用
- 里々：×
 - 支援機構なし
- YAYA：○
 - 構造体支援はある

データ保存形式

- ・華和梨：すべてがエントリ（ランダムアクセスと添字の使える配列）
- ・里々：トーク、単語、変数（トークは関数にもなる）
- ・YAYA：定数（マクロ）、変数、簡易配列、汎用配列、関数

メモ：

- YAYA は C 言語 + α- ポインタ
- 華和梨は「名前がすべてを司る」
- 里々は華和梨 + 変数…に近い

動的ロード

- ・ 華和梨 : ○
- ・ 里々 : ×
- ・ YAYA : ×

スクリプトファイルを動的に読み込めるのは華和梨だけ、他は起動時に指定したファイルを読む
(起動スクリプトに追記→再起動で回避は可能)



DB (永続変数の観点から)

- ・ 華和梨 : △ (自分で書かないとできない)
- ・ 里々 : ○
- ・ YAYA : ○

ゴースト起動をまたぐ変数管理は、唯一華和梨のみユーザサイドで実現しないとありません
(ミドルウェアに実装済みです)

DB (検索性の観点から)

- ・ 華和梨 : ○ (検索が一通り出来る、様々な単位でのコピー・移動・削除に支援あり)
- ・ 里々 : ×
- ・ YAYA : ○ (検索が一通り出来る)

SQLite の外部ライブラリを使うことで改善可能

正規表現対応

- ・ 華和梨 :△ (外部ライブラリが必要)
- ・ 里々 :△ (外部ライブラリが必要)
- ・ YAYA :○



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

多言語対応

- ・ 華和梨 : ×
- ・ 里々 : ×
- ・ YAYA : ○

日本語・英語以外でのゴースト作成は、実質的に
YAYA以外に選択肢がない



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

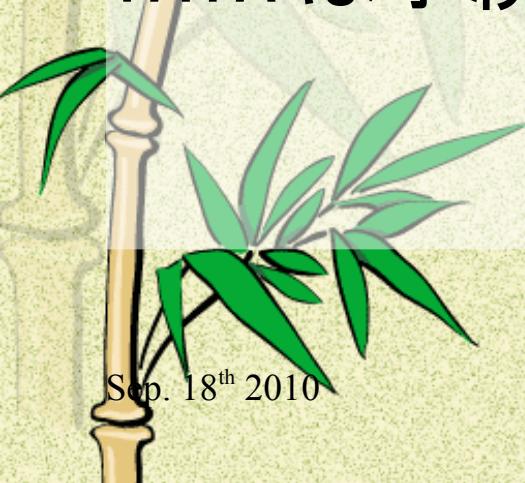
伺的ソフトウェア勉強会 大阪 #6

関数言語として

- ・ 華和梨 : ○
- ・ 里々 : △
- ・ YAYA : △

華和梨は戻り値として関数を返したり、引数に関数を与えることを意図しています。

YAYA は手続き型言語としての性格が強いです。



並列実行

- ・ 華和梨 : ×
- ・ 里々 : ×
- ・ YAYA : ×

マルチプロセス機能、マルチスレッド機能、コルーチン、トランザクションのいずれもありませ
ん



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

総評

- いまどきの言語と比べると、ちょっと機能が少ないです
- 機能整備度で見ると「YAYA \geq 華和梨 > 里々」
- perl・rubyとの類似度では華和梨、C言語への類似度ではYAYAが優れています
- システム最深部へのアクセス・多言語対応ではYAYA一択です

これから SHIORI に求められること

- ユーザの使いやすさは IDE・ミドルウェアで吸収できるので、機能の充実が必要です
- ゴーストだからって甘く見てはいけません！ UI 洗練させると AJAX 的動作が必要です。その上ではクロージャ、コレーチンはとても重要です。
- 多言語対応（パイは国内だけじゃないです）



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

道しるべは？

- ・ ゲーム向けに開発されたスクリプト言語 (Lua, Xtal) は SHIORI と似ています
- ・ 上記目的で使用されている言語の動向を観察することで、今後の参考になると考えられます。
- ・ IGDA のセミナーは特に有用です
- ・ 外の世界との関係を知った上で行動できればいいですね



御清聴ありがとうございました



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

Appendix



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

用語解説 (1)

- IDE

- 「統合開発環境」のこと。GUI の入力支援機能付きエディタ + デバッガ + α が多い。

- AJAX

- 「Asynchronous JAvascript + XML」。本来 一回のロードでしか通信できない HTML の上で JavaScript を使い、ロードとは非同期 (Asynchronous) に何回も XML を使った通信を行う技術。ページ切り替えを挟まないローカルアプリケーションのような UI を Web で実現できる。

- ミドルウェア

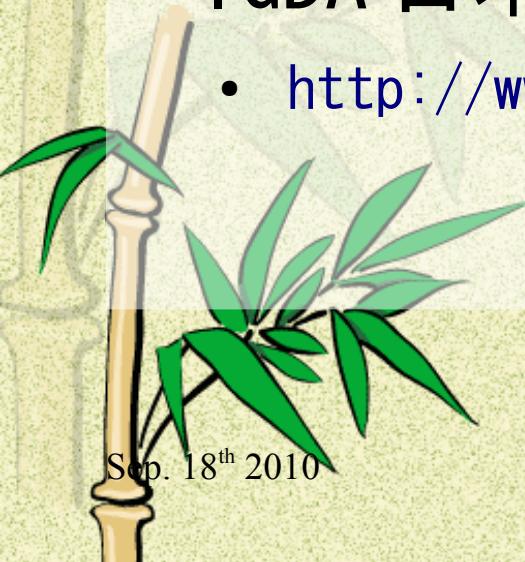
- 基底となるソフトとユーザーのアプリケーションの間にあって、アプリケーションの利便性を増してくれるソフトウェア

用語解説 (2)

- コルーチン
 - 一旦「サスPEND」し、もう一度呼ぶと「リジューム」できる関数のこと。
- クロージャ
 - 本来「同じ引数には必ず同じ値を返す」関数型言語において、副作用なしにローカルな値に基づいて異なる値を返させる手法。無名関数、高階関数と深い関係がある。
- 高階関数
 - 引数として関数を受け取る関数のこと。

参考文献 (1) IGDA SIG-GT

- ゲーム組み込みスクリプトは急速に進化中
 - IGDA 日本 SIG-GT12 レポート
 - <http://www.inside-games.jp/article/2009/01/18/33144.html>
- ゲームで使うためのスクリプト言語開発とは～
IGDA 日本 SIG-GT レポート
 - <http://www.inside-games.jp/article/2009/04/21/34870.html>



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

参考文献 (2) SHIORI

- 華和梨
 - <http://kawari.sourceforge.net/>
- 里々
 - <http://code.google.com/p/satoriya-shiori/>
- YAYA
 - <http://code.google.com/p/yaya-shiori/>



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6

参考文献 (3) プログラミング言語

- Perl : <http://www.perl.org/>
- Ruby : <http://www.ruby-lang.org/ja/>
- Python : <http://www.python.jp/Zope>
- PHP : <http://www.php.gr.jp/>
- Lua : <http://www.lua.org/>
- Xtal : <http://code.google.com/p/xtal-language/>



参考文献 (4) 読み物

- 「ほげ言語」のパラドックス
 - <http://practical-scheme.net/trans/beating-the-averages-j.html>
- 君のプログラミング言語で、これ、できる?
 - <http://local.joelonsoftware.com/wiki/%E5%90%9B%E3%81%AE%E3%83%97%E3%>



Sep. 18th 2010

Copyright S.Sato Some Rights Reserved.

伺的ソフトウェア勉強会 大阪 #6