

Jl 2014



Session Offline - présentations 'éclair'

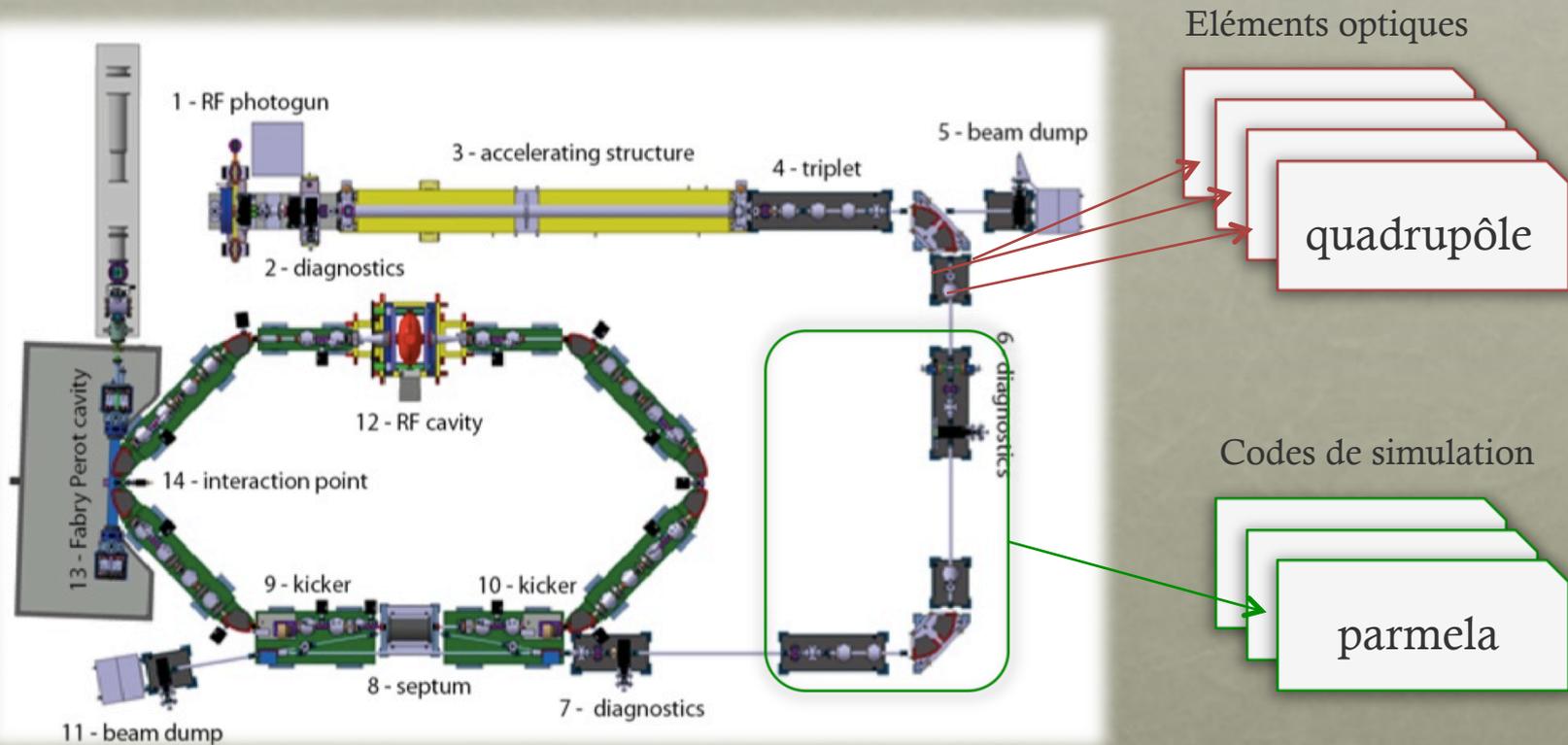
# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS

<http://groups.lal.in2p3.fr/pspa/>

Les journées informatique de l'IN2P3-IRFU  
13-16 octobre 2014



# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (I)



# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (II)

**Un ensemble important de logiciels de conception de machines :**

- ① dispersion des codes dans différents laboratoires
- ② multiplicité de versions d'un même code
- ③ dépendances des environnements informatiques
- ④ manque de publicité et de documentations

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (III)

**Un outil de simulation accessible à partir du WEB**

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (IV)

The image shows a web browser window titled "portail PSPA" with the address bar displaying "0.0.0.0:8077". The browser interface includes navigation buttons and a "Lecteur" (Reader) button. Below the browser window is a desktop environment with a dock on the left containing icons for "RF gun", a target, a red box, a white box, a grid, a red circle, a target, a hand, a green star, a grid, and a red "W".

The main application area is titled "my accelerator" and features a toolbar with icons for settings, a red triangle, a play button, and a bar chart. The interface is divided into several sections:

- Global view:** Contains a "sector 1" component represented by a white rounded rectangle.
- Component details:** Below "sector 1", there is a purple bar with a gear icon, the text "sector 1", and a dropdown menu set to "Linear". Below this is a dashed border area with the text "Drag elements here....".
- Computing view:** Contains a gear icon and a button labeled "init/expand machine".
- Computing blocks:** A grey bar with a white border.
- output:** A section with a "Search" input field and a large grey area below it.

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (V)

**Construire interactivement une machine**

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (VI)

The screenshot displays the PSPA (Particle Simulation Platform) interface within a web browser window titled "portail PSPA : philFit.aml". The browser address bar shows "0.0.0.0:8077". The interface is organized into several sections:

- Global view:** Shows a top-level view of the accelerator layout. A "sector 1" is highlighted, containing a sequence of components: RF gun, cell01, drift01, soleno01, drift02, bend01, and drift03.
- Computing view:** Provides a detailed view of the selected sector. It includes a gear icon for settings and a button labeled "init/expand machine".
- Computing blocks:** A configuration area for the simulation. It features a "from:" field set to "rfgun01" and a "to:" field set to "drift03". A "parmela" block is also visible, along with "+" and "-" buttons for adjusting parameters.

The interface includes a sidebar with various tool icons (e.g., RF gun, solenoid, drift, bend) and a top toolbar with icons for settings, zoom, and simulation control. The bottom of the screen shows an "output" window.

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (VII)

**Définir les traitements des processus physiques à  
appliquer sur les différentes sections de la machine**

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (VIII)

The screenshot displays the PSPA simulation platform interface. At the top, a browser window shows the URL "portail PSPA : philFit.aml" and the IP address "0.0.0.0:8077". Below the browser window is a toolbar with various icons, including a red arrow pointing to a blue play button icon. The main interface is divided into several sections:

- Global view:** Shows a "sector 1" label.
- Computing view:** Displays a configuration for "sector 1" with a "Type: Linear" dropdown. The configuration includes a sequence of components: rfgun01 (RF gun), cell01 (cell), drift01 (drift), soleno01 (solenoid), drift02 (drift), bend01 (bend), and drift03 (drift).
- Computing blocks:** A table showing the mapping of components to computing blocks.

| from     | to      | block     | + | - |
|----------|---------|-----------|---|---|
| rfgun01  | drift01 | parmela   | + | - |
| soleno01 | drift03 | transport | + | - |

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (IX)

**Lancer l'exécution**  
**Analyse des résultats produits**

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (X)

The screenshot displays the PSPA simulation platform interface. The main window shows a global view of an accelerator layout with components: rfgun01, cell01, drift01, soleno01, and drift03. Two data windows are open:

- kinetic energy** (Mon Oct 6 14:59:32 2014):  
case : pspa  
after element : drift01  
entries : 501  
mean : -0.00311566 MeV  
sigma rms : 18.2639 KeV
- phase space** (Mon Oct 6 14:59:06 2014):  
case : pspa  
after element : drift01  
emittance x, x' : 1.82085 pi. mm . mrad  
x max= 3.46679 mm  
x' max= 0.552101 mrad

The phase space plot shows a red ellipse representing the beam's distribution in the  $x$ - $x'$  plane. The  $x$  axis ranges from -4.000 to 4.000 mm, and the  $x'$  axis ranges from -0.6 to 0.6 mrad. The histogram shows the distribution of kinetic energy  $dE_{cin}/E_{cin}$  (%) from -0.200 to 0.200.

Computing blocks configuration:

| from     | to      | block     |
|----------|---------|-----------|
| rfgun01  | drift01 | parmela   |
| soleno01 | drift03 | transport |

# PSPA : UNE PLATEFORME DE SIMULATION EN PHYSIQUE DES ACCÉLÉRATEURS (XI)

- langage C++
- librairie Wt pour développer l'interface graphique web
- gestion des versions du code avec Mercurial
- description et gestion du code avec CMT
- un serveur Jenkins exécute de manière continue la construction du code
- site web réalisé avec le logiciel WordPress
- format XML pour décrire les données (AML)



# Intégration continue à l'IN2P3

Retour d'expérience: développement du framework Lavoisier

Foudil Brétel, Lionel Schwarz

8 octobre 2014

# Le besoin

L'équipe de développement de Lavoisier voulait :

- Etre alerté en (quasi) temps réel des tests (unitaires et intégration) en erreur

# Le besoin

L'équipe de développement de Lavoisier voulait :

- Etre alerté en (quasi) temps réel des tests (unitaires et intégration) en erreur
- Exporter les processus de build sur des environnements "neutres"

# Le besoin

L'équipe de développement de Lavoisier voulait :

- Etre alerté en (quasi) temps réel des tests (unitaires et intégration) en erreur
- Exporter les processus de build sur des environnements "neutres"
- Builder sur des environnements variés (JDK6, JDK7, Maven2, Maven3...)

# Le besoin

L'équipe de développement de Lavoisier voulait :

- Etre alerté en (quasi) temps réel des tests (unitaires et intégration) en erreur
- Exporter les processus de build sur des environnements "neutres"
- Builder sur des environnements variés (JDK6, JDK7, Maven2, Maven3...)
- Permettre à des utilisateurs de lancer des builds

# Présentation de la plateforme d'intégration continue

<http://ci.in2p3.fr>

- Proposée et hébergée par le CC-IN2P3

# Présentation de la plateforme d'intégration continue

<http://ci.in2p3.fr>

- Proposée et hébergée par le CC-IN2P3
- Basée sur l'infrastructure OpenStack du CC-IN2P3

# Présentation de la plateforme d'intégration continue

<http://ci.in2p3.fr>

- Proposée et hébergée par le CC-IN2P3
- Basée sur l'infrastructure OpenStack du CC-IN2P3
- Basée sur Jenkins <http://jenkins-ci.org/>

# Les avantages

- Délégation d'administration de machines grâce à l'infrastructure Cloud du CCIN2P3

# Les avantages

- Délégation d'administration de machines grâce à l'infrastructure Cloud du CCIN2P3
- Autonomie : mises à jour de Jenkins et installation de plugins personnalisés

# Les avantages

- Délégation d'administration de machines grâce à l'infrastructure Cloud du CCIN2P3
- Autonomie : mises à jour de Jenkins et installation de plugins personnalisés
- Sécurité : gestion avancée des utilisateurs et des privilèges

# Mais...

- L'empilement des couches (Jenkins, OpenStack, réseau/système) complexifie la configuration

# Mais...

- L'empilement des couches (Jenkins, OpenStack, réseau/système) complexifie la configuration
- L'investissement initial n'est pas négligeable

# Mais...

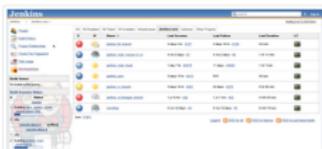
- L'empilement des couches (Jenkins, OpenStack, réseau/système) complexifie la configuration
- L'investissement initial n'est pas négligeable
- L'intégration continue est une tâche à part entière ("release manager")

# Bonne intégration continue !

Rendez-vous à <http://ci.in2p3.fr>



Continuous integration allows immediate feedback to developers on the quality.  
*build on multiple platforms, pass tests, check code quality, measure tests coverage, and do much more !*



## SaaS

Get a fresh, pre-configured, cloud-ready [Jenkins server](#) with full control in few minutes. We take care of boring hosting details like backups and monitoring.



## Build in the cloud

Benefit from the flexibility of our private [Openstack](#) infrastructure. Launch your tasks on any system, any OS, any hosts you need.



## Rich interface

Most of the configuration work is done through a single, user-friendly interface. Customize your instance with 800+ available plugins !



Google



Drive



File Manager



Adobe Reader



VNC Viewer



HP ePrint



AndFTP



Firefox



ES Explorateur..



Admin



L'Appli



JuiceSSH



Terminal Emu..



FitsActivity



*L'icône de mon appli*



L'émulateur



```
File Edit Options Buffers Tools Java Help

// The Android Workbench

package org.fonteny.android.workbench.fitsviewer;

import android.util.Log;

import nom.tam.fits.Fits;
import nom.tam.fits.BasicHDU;
import nom.tam.fits.ImageHDU;
import nom.tam.fits.Header;
import nom.tam.fits.FitsException;
import nom.tam.fits.PaddingException;
import nom.tam.util.Cursor;
import java.io.IOException;
import java.io.File;

/**
 * FitsExaminer : extracts content of a fits file.
 * It is meant to feed the android application.
 * See this
 * <a href="http://heasarc.gsfc.nasa.gov/docs/heasarc/fits/java/">doc</a>
 * @author Jean-Paul Le Fèvre
 */
//

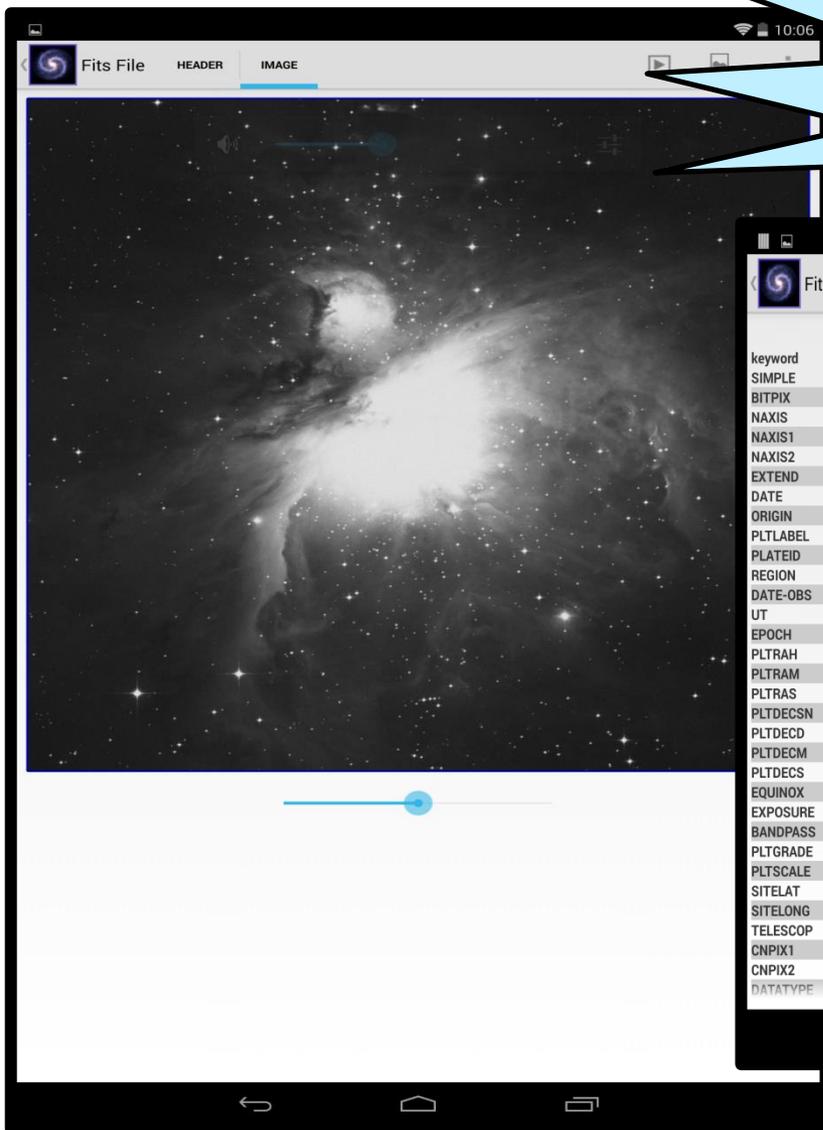
public class FitsExaminer {
    /**
     * The name of the file to examine.
     */
    private String filename;
    /**
     * The first HDU found in the file.
     */
    private BasicHDU hdu;
    /**
     * The number of bytes in a float (see Fits doc).
     */
}

}

</cs:checkstyle>
</target>
</project>
-

ls
AndroidManifest
ant.properties
bin
build.xml
-m custom_rules
<?xml version=
<project xmlns=
<target name=
<echo messa
<cs:checkst
U: FitsExaminer.java Top L1 SVN-3477 (Java/1 Abbrev)
<fileset
<right-fringe> <drag-mouse-1> is undefined
</cs:checkstyle>
</target>
</project>
-
Root
```

C'est moi qui l'ai fait !



| keyword   | value               | remark                         |
|-----------|---------------------|--------------------------------|
| SIMPLE    | T                   | FITS header                    |
| BITPIX    | 16                  | No.Bits per pixel              |
| NAXIS     | 2                   | No dimensions                  |
| NAXIS1    | 2376                | Length X axis                  |
| NAXIS2    | 2381                | Length Y axis                  |
| EXTEND    | T                   |                                |
| DATE      | 09/07/04            | Date of FITS file creation     |
| ORIGIN    | CASB - STScI        | Origin of FITS image           |
| PLTLABEL  | I16929              | Observatory plate label        |
| PLATEID   | A44C                | GSSS Plate ID                  |
| REGION    | IS768               | GSSS Region Name               |
| DATE-OBS  | 1996/01/27          | UT date of Observation         |
| UT        | #####               | UT time of observation         |
| EPOCH     | 1.9960710449219E+03 | Epoch of plate                 |
| PLTRAH    | 5                   | Plate center RA                |
| PLTRAM    | 42                  |                                |
| PLTRAS    | 1.9934887333337E+01 |                                |
| PLTDECSN  | -                   | Plate center Dec               |
| PLTDECD   | 4                   |                                |
| PLTDECM   | 58                  |                                |
| PLTDECS   | 4.0360059999999E+01 |                                |
| EQUINOX   | 2.0000000000000E+03 | Julian Reference frame equinox |
| EXPOSURE  | 9.0000000000000E+01 | Exposure time minutes          |
| BANDPASS  | 38                  | GSSS Bandpass code             |
| PLTGRADE  | 0                   | Plate grade                    |
| PLTSSCALE | 6.7199996948242E+01 | Plate Scale arcsec per mm      |
| SITELAT   | -31:16:24.00        | Latitude of Observatory        |
| SITELONG  | +149:03:42.00       | Longitude of Observatory       |
| TELESCOP  | UKSTU Schmidt       | Telescope where plate taken    |
| CNPIX1    | 16589               | X corner (pixels)              |
| CNPIX2    | 8851                | Y corner                       |
| DATATYPE  | INTEGER*2           | Type of Data                   |





# Table Ronde Nouveaux langages

Charles LOOMIS<sup>1</sup>, Vincent LAFAGE<sup>2</sup>

<sup>1</sup>Laboratoire de l'Accélérateur Linéaire

<sup>2</sup>Institut de Physique Nucléaire  
Université d'Orsay



UNIVERSITÉ  
PARIS  
SUD

FACULTÉ  
DES SCIENCES  
D'ORSAY

mercredi 15 octobre 2014



# Les présentations

- + Scala, la programmation fonctionnelle  
Foudil BRÉTEL (CC-IN2P3)
- + Le langage de programmation D, successeur de C++ ?  
Christophe MEESSEN (CPPM)
- + Go : concurrence et HEP  
Sébastien BINET (LAL)



# Discussion

paradigmes ↔ langages ↔ bibliothèques ↔ outils

Quelles caractéristiques des nouveaux langages améliorent ou facilitent la programmation scientifique ?

- ⇒ Quel type d'application scientifique ?
- ⇒ Calcul parallèle
- ⇒ Rapidité de programmation
- ⇒ Interopérabilité
- ⇒ Écosystème



# Adaptation

Le langage supporte bien quel type d'applications :

- online
- services
- analyses
- autres. . .

Le langage supporte quel(s) modèle(s) de programmation :

- procédural
- orientée objet
- fonctionnelle

Quid du fonctionnel objet ?



# Calcul parallèle

L'utilisation efficace des machines récentes avec leur puces multi-cœur nécessite une programmation parallèle.

Quelles caractéristiques de chaque langage facilitent ou empêchent la programmation parallèle / concurrente correcte ?

Comment le langage se compare-t-il aux plus anciens : Ada, Erlang, Fortran 2008 ?



# Rapidité de programmation

Il y a une tendance actuelle à utiliser des langages dynamiques qui rendent la programmation plus rapide :

- typage implicite
- évaluation interactive
- compilation dynamique

Est-ce que le langage a des caractéristiques qui rendront la programmation plus rapide par rapport aux « anciens » langages ?



# Interopérabilité

Il n'existe (presque) plus de systèmes programmés dans un seul langage et consistant en un service unique.

Est-ce que le langage a des caractéristiques qui permettent l'interopérabilité avec les codes écrits dans les autres langages et/ou les APIs web (REST, SOAP, ...)?



# Ecosystème

L'écosystème autour d'un langage a un fort impact sur l'utilité de ce langage pour les applications scientifiques.

L'outillage pour le langage est complet et mature ?

- IDE
- tests unitaires
- analyse statique

Y a-t-il une bonne gamme de bibliothèques disponibles ?

- mathématiques
- bases de données
- XML/JSON
- web



# Go Workshop

Sébastien Binet

LAL/IN2P3

2014-10-14

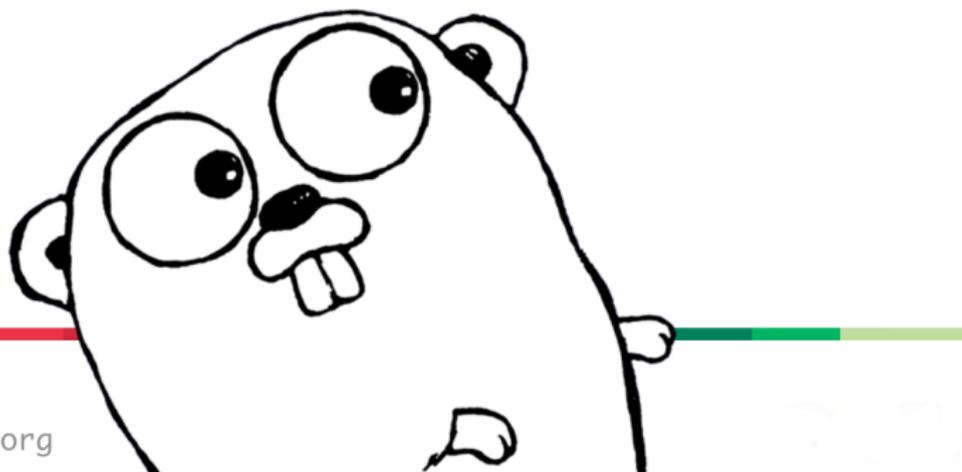


- obligatory hello world example...

```
package main
```

```
import "fmt"
```

```
func main() {  
    fmt.Println("Hello JI-2014")  
}
```



<http://golang.org>

- founding fathers:
  - ▶ Russ Cox, Robert Griesemer, Ian Lance Taylor
  - ▶ Rob Pike, Ken Thomson
- **concurrent**, compiled
- **garbage collected**
- an open-source general programming language
- best of both "worlds":
  - ▶ feel of a **dynamic language**
    - ★ limited verbosity thanks to a **type inference system**, map, slices
  - ▶ safety of a **static type system**
  - ▶ compiled down to machine language (so it is fast)
    - ★ goal is within **10% of C**
- **object-oriented** (w/o classes), **builtin reflection**
- first-class functions with **closures**
- **duck-typing** à la `python` (but better) thanks to its **interfaces**

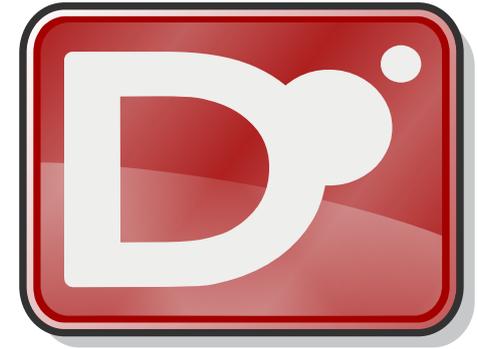
- simple exercises (create a command, handle arguments)
- discover a bit of the surrounding tooling ecosystem:
  - ▶ doc system,
  - ▶ build system,
  - ▶ CPAN/PyPI/. . .-like market-store;
- discover a bit of the standard library (`json`, `io`, `os`, . . .)
- touch upon `interfaces` and concurrency:
  - ▶ `channels`
  - ▶ `goroutines`

[github.com/sbinet/ji-2014-go](https://github.com/sbinet/ji-2014-go)



# Preview Atelier D

Christophe Meessen



## Le langage de programmation D

<http://dlang.org/>

- Auteur : Walter Bright  
compilateurs C & C++ DigitalMars
- Simple et facile à apprendre
- Évite les pièges et défauts du C++
- Améliorations : tests unitaires, programmation par contrat, programmation parallèle, programmation fonctionnelle, ...



# Preview Atelier D

Christophe Meessen

## Objectifs de l'atelier

- Découverte du langage D
- Écrire vos premiers programmes en D
- Apprentissage par l'exemple
- Exercices simples et courts
- Notions de base pour développer en D

# Preview Atelier D

Christophe Meessen

## **Au menu ...**

- Affichage et manipulation des arguments du programme
- Affichage des noms et tailles des fichiers d'un répertoire
- Écriture et lecture dans un fichier texte
- Écriture et lecture dans un fichier binaire

# Preview Atelier D

Christophe Meessen

## Programme Hello World !

```
import std.stdio;  
  
void main() {  
    writeln("Hello world!");  
}
```

Jl 2014



Session Offline - présentations 'éclair'