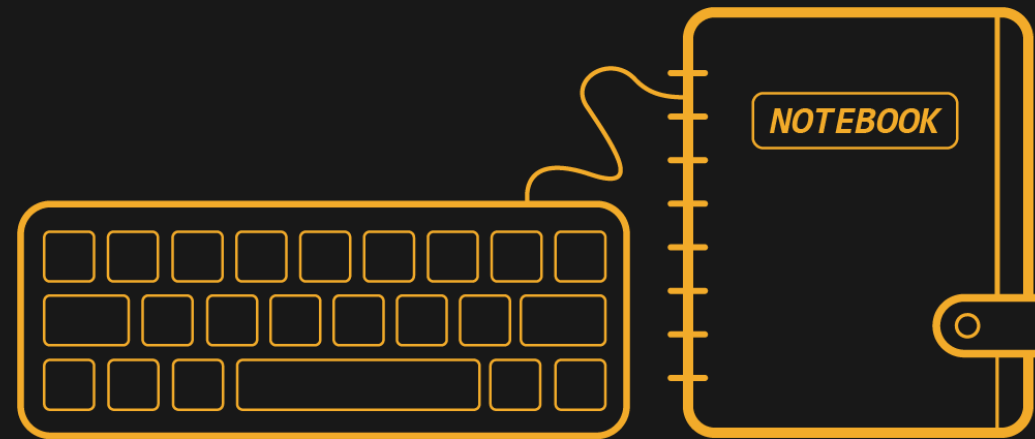


Ecrire son code dans un environnement de programmation lettrée



Hugues Pecout
UMR Géographie-Cités - CNRS

Bonjour à tout·te·s

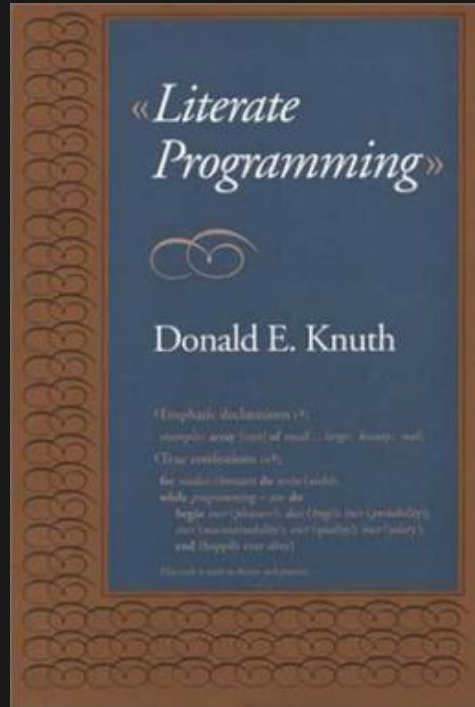
Ingénieur d'études CNRS en sciences de l'information géographique dans le domaine de la géographie humaine et quantitative (UMR Géographie-Cités).

Depuis 2016 :

- Évite l'utilisation de logiciels « clic-bouton » (QGIS, EXCEL, Philcarto...)
- **Analyse de données réalisée exclusivement avec le langage R**
- Création quasi hebdomadaire de script R.
- Gain de productivité et de polyvalence important, mais...

→ Implique de **nouvelles méthodes de travail** :

- Comment rendre mon travail intelligible pour mes collègues (et pour moi) ?
- Comment mettre à disposition du code ?
- Comment archiver et assurer une certaine reproductibilité de mes scripts ?



Literate programming

La **programmation lettrée** est une approche de la programmation qui se veut différente du paradigme de programmation structurée.

« Nous devons changer notre attitude traditionnelle envers la construction des programmes : au lieu de considérer que notre tâche principale est de dire à un ordinateur ce qu'il doit faire, appliquons-nous plutôt à expliquer à des êtres humains ce que nous voulons que l'ordinateur fasse. (...) »

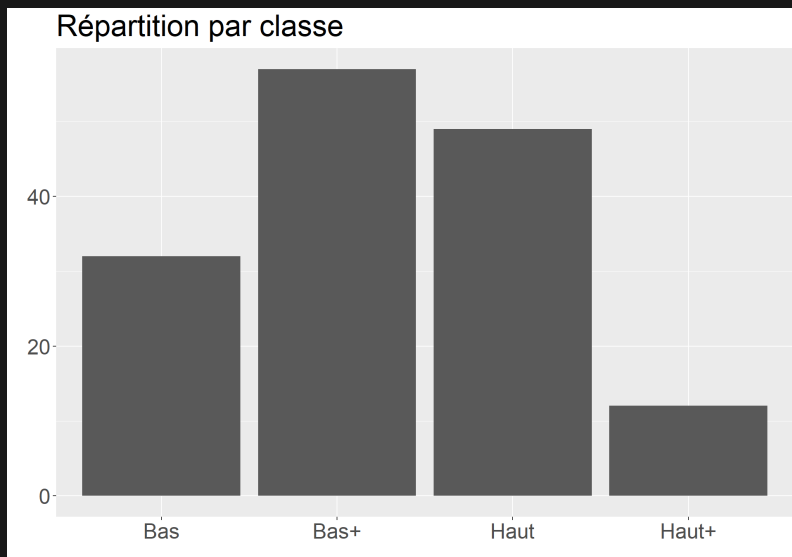
Celui qui pratique la programmation lettrée peut être vu comme un essayiste, qui s'attache principalement à exposer son sujet dans un style visant à l'excellence. Tel un auteur, il choisit, avec soin, le dictionnaire à la main, les noms de ses variables et en explique la signification pour chacune d'elles. Il cherche donc à obtenir un programme compréhensible parce que ses concepts sont présentés dans le meilleur ordre possible. Pour cela, il utilise un mélange de méthodes formelles et informelles qui se complètent. »

D. Knuth, « Literate Programming », *The Computer Journal*, British Computer Society, vol. 27, no 2, 1984, p. 97–111

Le paradigme de la programmation lettrée, **pensé et mis en oeuvre par Donald Knuth (1977-78)**, s'éloigne dans son écriture de l'ordonnancement imposé par l'ordinateur, et à la place **autorise les développeurs à utiliser un ordre imposé par la logique et le fil de leur pensée.**

Quelques lignes de code...

```
library(ggplot2)
ggplot(data = iris,
       aes(x = cut(Sepal.Length, breaks = c(min(Sepal.Length, 5),
                                             include.lowest = TRUE, labels = c("Bas", "Bas+", "Haut", "Haut+")),
           ylab("Nombre d'individus")) +
  geom_bar() +
  xlab("Classe") +
  ylab("Nombre d'individus") +
  ggtitle("Répartition par classe") +
  theme(plot.title = element_text(size=27), axis.title = element_text(size=18))
```



...En programmation lettrée

```
#####
#
#      Représentation graphique
#      de la répartition d'individus
#      discrétisés en quatre classes
#
#      H. Pecout
#      5 décembre 2022
#####

## Discrétisation la variable Sepal.Length (donnée)
# Statistiques sur la série statistique 'Sepal.Length'
summary(iris$Sepal.Length)

# Histogramme de la distribution
hist(iris$Sepal.Length)

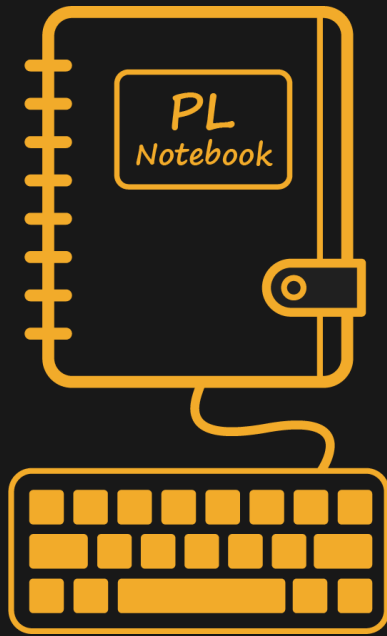
# Choix des bornes de classes (amplitude égale)
bornes_des_classes <- c(min(iris$Sepal.Length), 5,
                        max(iris$Sepal.Length))

# Choix des noms de classes
nom_des_classes <- c("Très petites", "petites", "grandes", "gigantes")

# Discrétisation de la variable 'Sepal.Length'
```

Quel outil pour la PL ?

Il est possible de faire de la programmation lettrée dans n'importe quel type de document... Ce paradigme n'est pas lié à format ou à un outil. Mais il en existe tout de même un parfait pour mettre en oeuvre ce concept.



Le notebook permet de combiner du langage naturel et du langage de programmation, et d'interpréter les deux :

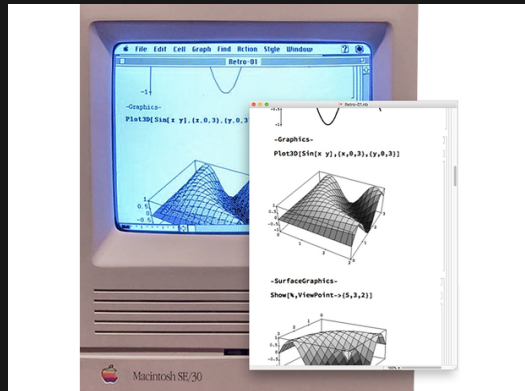
- Mise en forme du texte
- Exécution et/ou affichage du code

Les notebooks et la programmation lettrée sont apparus à la même période, celle de l'émergence du concept de recherche reproductible...

70's show

Les premiers notebooks

- 1970 : Scratchpad (Axiom)
- 1987 : MathCad
- 1988 : Mathematica
- 1989 : Maple
- 2011 : IPython



Émergence de la recherche reproductible

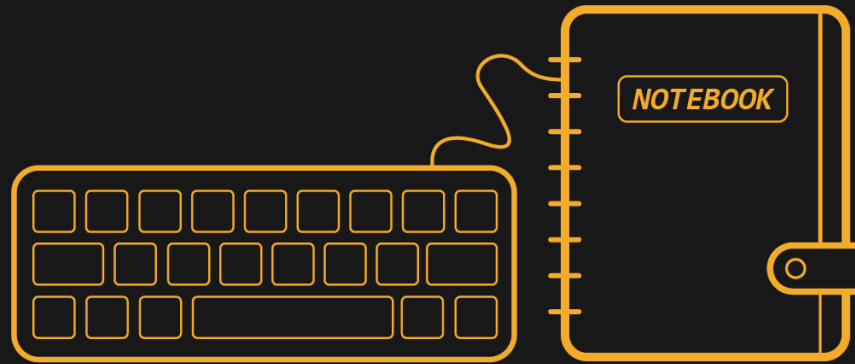
Jon Claerbout est l'un des premiers chercheurs à exprimer le fait que **les méthodes de calcul menacent la reproductibilité de la recherche** si l'on ne donne pas un accès libre aux données et aux logiciels qui sous-tendent.

En 1976, il publie un ouvrage référence en traitement de signal :
« *Fundamentals of Geophysical Data Processing* »

Cette 1ère version - non-reproductible - sera le début d'un long travail de recherche sur la reproductibilité... Via des notebooks !

Émergence du Literate Programming

Le concept de **programmation lettrée** est pensé et conçu par **Donald Knuth** (1977-78). Le notebook se présente comme un outil idéal pour mettre en œuvre ce paradigme.



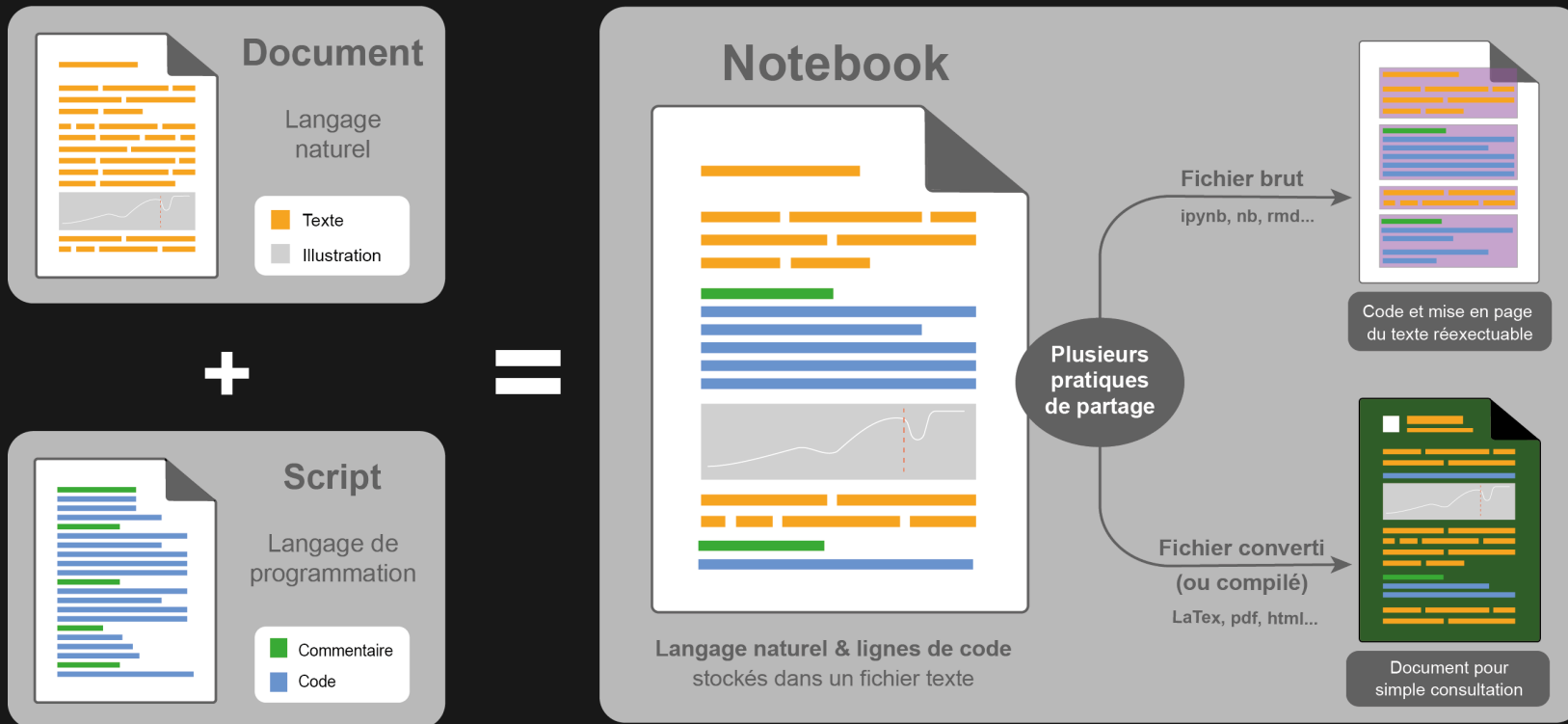
Définition

Interface de programmation qui permet de combiner des sections en langage naturel et des sections en langage informatique.

De nombreuses dénominations existent :

- *Article exécutable*
- *Document computationnel*
- *Document électronique interactif*
- *Cahiers de programmation*
- *Cahier électronique d'analyse*
- *Calepin électronique*
- *Carnet de code*
- *Manifeste algorithmique*
- **Notebook !**
- ...

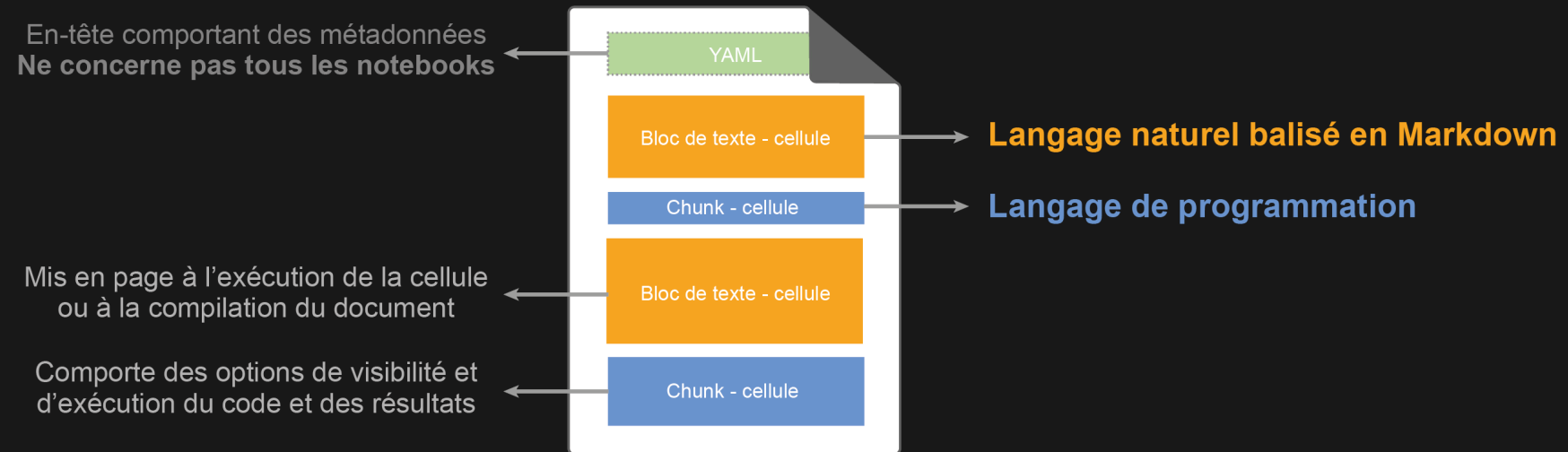
Un outils... des pratiques !



La mise en forme du texte se fait à l'aide d'un langage de balisage (Markdown, html...)

Anatomie d'un notebook

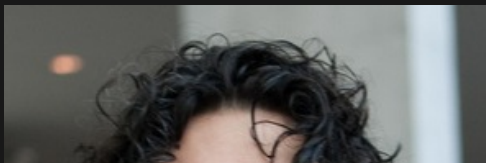
Un simple fichier texte (.rmd, .qmd, .ipynb...)



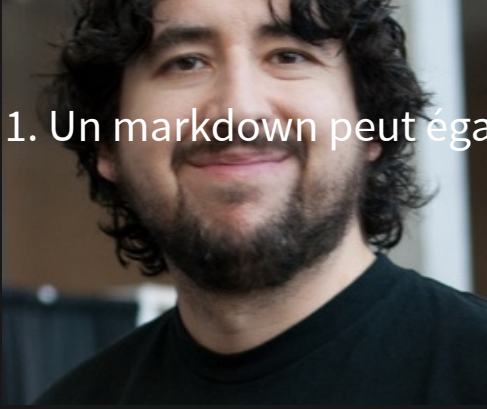

```
1 \
2
3 ### Le Markdown
4
5 Markdown est un langage de balisage léger créé en 2004 par
6 [John Gruber] (https://fr.wikipedia.org/wiki/John\_Gruber),
7 avec l'aide d'[Aaron Swartz] (https://fr.wikipedia.org/wiki/Aaron\_Swartz)
8 dans le but d'offrir une syntaxe facile à lire et à écrire.
9
10 ![] (https://upload.wikimedia.org/wikipedia/commons/6/64/John\_Gruber%20et%20Aaron\_Swartz.jpg)
11 John Gruber, 2004 :
12
13 > « Un document balisé par Markdown peut être lu en l'état sans donner
14 l'impression d'avoir été balisé ou formaté par des instructions particulières.
15
16 Un document balisé par Markdown peut être converti en :
17
18 >- HTML
19 >- PDF
```

Le Markdown

Markdown est un **langage de balisage léger** créé en 2004 par **John Gruber**, avec l'aide d'**Aaron Swartz**, dans le but d'offrir une **syntaxe facile à lire et à écrire**.




1. Un markdown peut également être converti en format word, LaTeX, ePub, odt...




John Gruber, 2004 :



« Un document balisé par Markdown peut être lu en l'état sans donner l'impression d'avoir été balisé ou


Les notebooks en SHS


 **Carnet Jupyter** 2014

 «Extension» du projet Ipython
Open Source
Plusieurs langages





Notebook réactif
Code réexécutable


IP[y]:  



 **Rmarkdown & co** 2015

  Studio comme acteur principal
Open Source
Plusieurs langages


Nombreux formats
Programmation lettrée


        

 **Notebook Observable** 2016

 De Mike Bostock, créateur de  D3.js
Opensource - Javascript

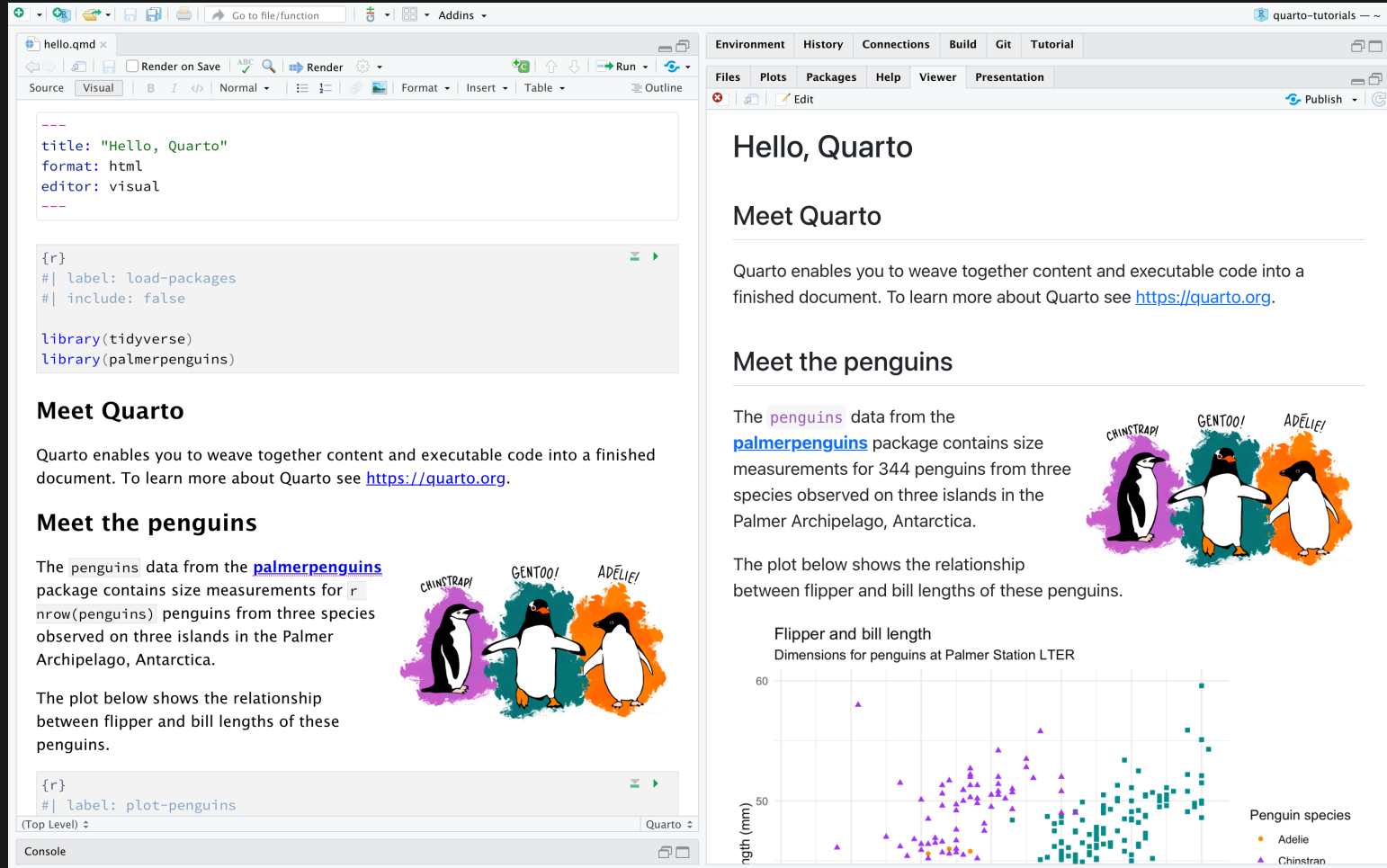
Notebook réactif **Représentation**
Code réexécutable **graphique interactive**

 **Notebook Pluto** 2020

 Open Source
Plusieurs langages

Notebook réactif
Code réexécutable

2022, l'année quarto®



The screenshot shows a Quarto document editor with two panels. The left panel displays the source code for a document titled "Hello, Quarto". The right panel shows the rendered output of the document.

Source Code (Left Panel):

```

---
title: "Hello, Quarto"
format: html
editor: visual
---

{r}
#| label: load-packages
#| include: false

library(tidyverse)
library(palmerpenguins)
  
```

Rendered Output (Right Panel):

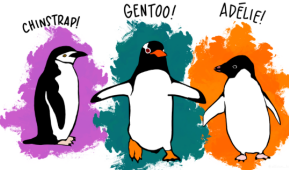
Hello, Quarto

Meet Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

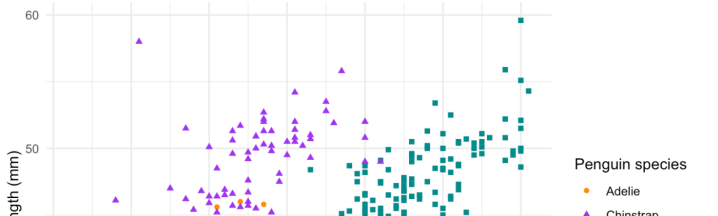
Meet the penguins

The `penguins` data from the [palmerpenguins](#) package contains size measurements for 344 penguins from three species observed on three islands in the Palmer Archipelago, Antarctica.



The plot below shows the relationship between flipper and bill lengths of these penguins.

Plot: Flipper and bill length
Dimensions for penguins at Palmer Station LTER



```

{r}
#| label: plot-penguins
  
```




Nombreux usages & formats

Nombreux templates,

multiples usages :

- *Article*
- *Manuel*
- *Rapports*
- *Livre*
- *Tutoriel*
- *Cours - Exercice*
- *Blog - Site web*
- *curriculum vitae*
- *Diaporama*
- ...

Equivalent LaTeX + **programmation :**

- Bibliographie, TOC, note de bas de page...
- Feuille de style CSS
- Et bien plus... **Quarto !**

Un **format de publication**
scientifique légitime !

- **Rzine**, pour la pratique de R en SHS
- **Computo** de la SFdS
- **Programming Historian**

Programmation lettrée, notebook et **reproductibilité**

Programmation lettrée + Notebook = 

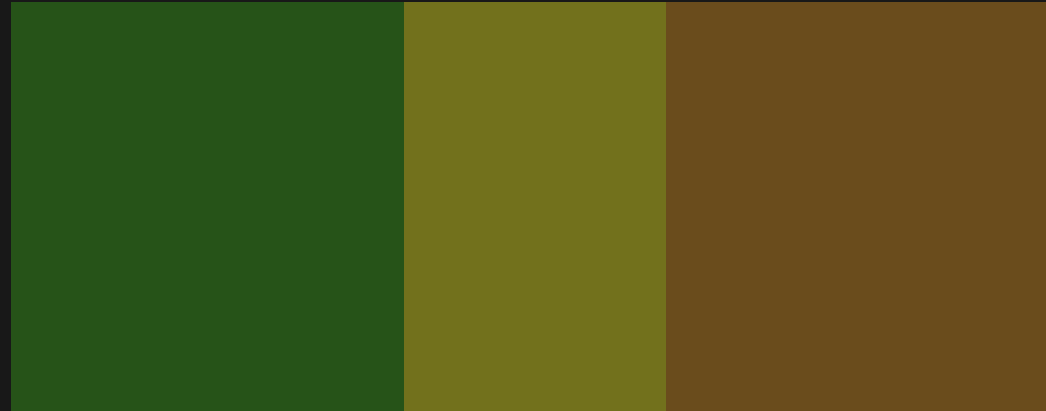


Programmation lettrée



Notebook

Programmation lettrée + Notebook = 🔥



Reproductibilité ?

PL + NB + ... ? = reproductibilité



Programmation lettrée



Notebook



Informations de session
et d'environnement



Mise à disposition
Conditions d'utilisation

PL + NB + ... ? = reproductibilité



Quel outil pour le partage ?

→ **GitLab** ou **GitHub (Git)**

Git est un système de gestion de version distribué pour :

- tracker les changements dans les fichiers texte
- gérer l'historique du code source
- partager le code dans des dépôts distants

GitLab et **GitHub** sont des services web d'hébergement et de gestion de version (issue tracker, collaboration, hébergement web, CI/CD), qui reposent sur le logiciel de version **Git**

Notebook + GitLab (ou GitHub)

- Les sources des **notebooks** sont de **simples fichiers texte**
 - Suivi des modifications (*commit*, branches, etc.)
 - **Partage** et mise à disposition des fichiers sources (dépôt public ou privé)
- GitLab et GitHub permettent de **compiler un notebook**
 - *Continuous Integration*
- Ils permettent de **déployer sur le web un notebook compilé en HTML**
 - *Continuous Deployment*

Exemple

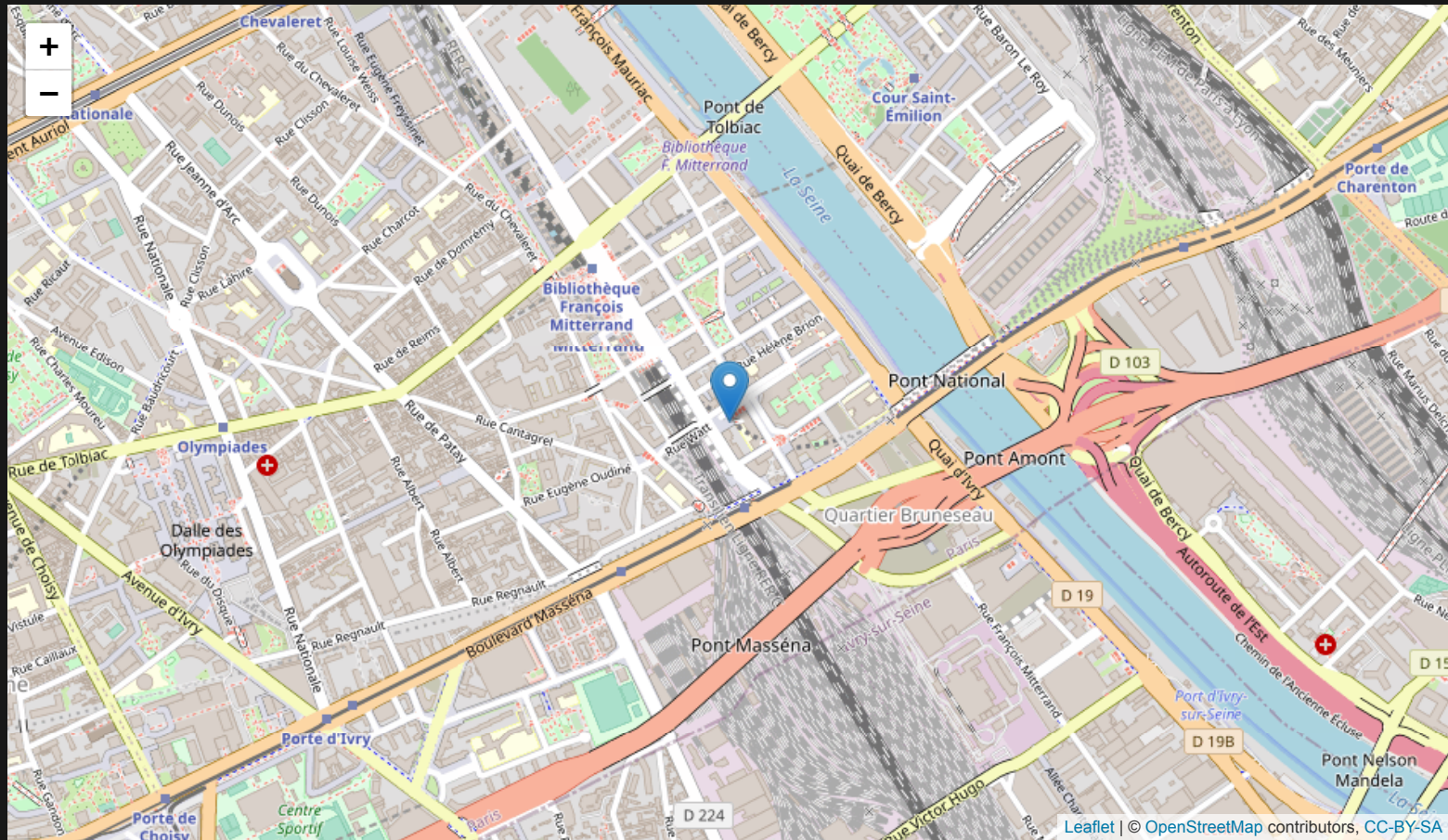
Ceci est un notebook !

Un peu de code :

```
1 # Chargement des librairies
2 library(sf)
3 library(leaflet)
4
5 # Création d'une point (localisation de l'amphi Turing - Université de Paris)
6 ptsfg <- st_point(c(2.3807381923614552, 48.827114532567876))
7
8 # On précise le système géographique de référence utilisé (WGS84)
9 ptsfc <- st_sfc(ptsfg, crs = (4326))
10
11 # Création d'un tableau contenant deux attributs
12 ptdf <- data.frame(id= 1, text = "Semaine Data-SHS 2022 ❤️")
13
14 # Construction d'un objet géographique (point + attributs)
15 Here <- st_sf(ptdf, geometry = ptsfc)
16
17 # Cartographie interactive (Leaflet) - affichage d'un marker (point créé)
18 mamap <- leaflet(Here, width = 950, height = 550) %>%
19   addTiles() %>%
```

Ceci est un notebook !

Affichage du résultat :



Ceci est un notebook !

Document libre :

Code source (GitHub) : https://github.com/HuguesPecout/DataSHS_2022

Slides en ligne via Github : https://huguespecout.github.io/DataSHS_2022/

Conditions d'utilisation :

Code source : Licence **MIT**

Diaporama : **CC BY-SA 4.0**

Informations session :

```
R version 4.2.1 (2022-06-23 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22000)
```

```
Matrix products: default
```

```
locale:
```

```
[1] LC_COLLATE=French_France.utf8 LC_CTYPE=French_France.utf8  
[3] LC_MONETARY=French_France.utf8 LC_NUMERIC=C  
[5] LC_TIME=French_France.utf8
```

attached base packages:

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

other attached packages:

Exemples divers

Communication

- **Le projet Rzine**
- **Analyse de données avec R**
- **Introduction à R**

Livre - manuel

- **Géomatique avec R**
- **Cartographie avec R**
- **Analyse de réseaux (R)**

Site web statique

- **Rzine.fr**
- **Groupe ElementR**
- **Ecole d'été CIST**

Collection Rzine

- **Analyse territoriale multiscalaire**
- **Le nouveau rideau de fer (carte 2.5D)**
- **Exploration spatio-temporelle**

Merci de votre attention

hugues.pecout@cns.fr

 [@rzine_shs](https://twitter.com/rzine_shs)

