



MINISTÈRE  
DE LA TRANSITION  
ÉCOLOGIQUE  
ET DE LA COHÉSION  
DES TERRITOIRES

*Liberté  
Égalité  
Fraternité*



Feuille de route ministérielle de la donnée,  
des algorithmes et des codes sources

Facilitateur de vos transitions  
#CMVRH



# "Qu'entend-on par « coder » quand on parle de données ?"

6ème webinaire d'un cycle sur la culture de la donnée

Vendredi 15 mars 2024 de 9h30 à 12h00

## Introduction générale

*Hélène BEGON - CGDD/SRI/Ecolab*

*Auteure de : La transformation numérique des administrations, La documentation française, août 2021*

*Support préparé avec les collègues de l'équipe data-IA de l'Ecolab du Commissariat général au développement durable*



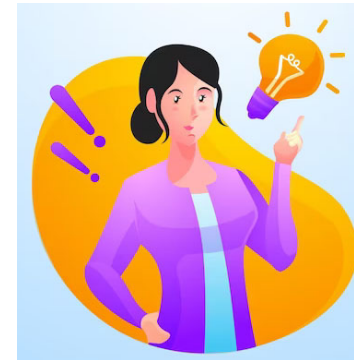
Qui ne s'est pas déjà retrouvé dans l'une de ses situations :

Devoir analyser des données très volumineuses

Devoir collecter et traiter des données de formats très différents

Devoir mettre à disposition des données au plus grand nombre

Devoir reproduire d'années en années des indicateurs sur des données actualisées



Dans ces situations, il peut être nécessaire de recourir à du codage

# Vocabulaire et notions

« Coder » et « programmer » sont deux mots courants pour exprimer la même chose

On code dans un langage que l'ordinateur pourra comprendre et exécuter

Sans utilisation d'un langage compréhensible par l'ordinateur, celui-ci ne fournit rien

On peut voir le codage comme une opération de traduction humain → machine

C'est là qu'il y a une ambiguïté dans le langage courant :

Tout ce que l'ordinateur exécute a été codé

**MAIS**

Nous, utilisateurs, tirons de l'ordinateur de nombreuses opérations que nous n'avons pas eu besoin de coder nous-mêmes : d'autres ont pu le faire pour nous avant (logiciels, lignes de codes dans des bibliothèques...)

En conséquence, quand on va être amené à coder, c'est pour créer des opérations qui n'existent pas déjà, pour répondre à un besoin particulier

# Vocabulaire et notions

## Représentation simplifiée du processus quand on code :

Problème à résoudre

Ecriture de **l'algorithme** / des algorithmes permettant de résoudre le problème

**Développement** : traduction intelligente (minimisation du nombre de calculs, des risques d'erreurs...) de l'algorithme en suites de procédures à coder : le futur **programme**

**Codage** du **programme** (ou **programmation**) avec un **langage de programmation** dit « de **haut niveau** ».

Le programme est une suite d'instructions. Il peut s'agir de quelques lignes.

Le langage de programmation est encore compréhensible par l'humain car il utilise des mots usuels des langues naturelles (très souvent de l'anglais) et des symboles mathématiques familiers. Il permet d'écrire le **code source** du programme.

Selon l'environnement, le langage de programmation sera imposé ou non mais en général les programmes en langage de haut niveau sont plus facilement portables du fait de leur niveau d'abstraction.

Il existe au moins des centaines de langages de programmation de haut niveau (Python, Java, Fortran, SQL, R ...)

Il peut être nécessaire parfois de coder en langage de bas niveau, plus polyvalent

**Compilateur** ou **interpréteur** de langage de programmation

Ils traduisent automatiquement le code écrit en langage de programmation en code écrit en langage machine dit « de **bas niveau** », **exécutable** par les **processeurs** de l'ordinateur (*au début de l'informatique il fallait écrire ce langage machine*). Ce langage machine, dit **binaire**, est une suite de **bits** (à partir de 0 et de 1).



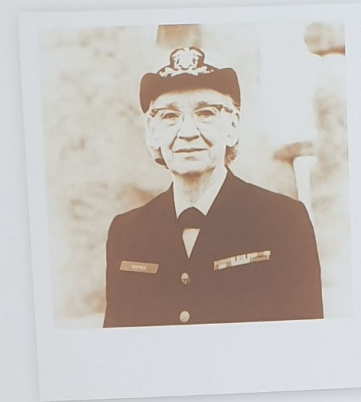
## AUGUSTA ADA LOVELACE

MATHÉMATICIENNE INFORMATICIENNE

1815-12-10 - 1852-11-27

**F**ille d'un poète britannique (lord Byron) et d'une amatrice de mathématiques (Anne Isabella Milbanke), Augusta Ada King naît le 10 décembre 1815 à Londres et témoigne, comme sa mère, d'un grand intérêt pour les mathématiques. Devenue femme du comte de Lovelace, elle rencontre Charles Babbage, inventeur de la « machine à différences », une **calculatrice** mécanique. Le **mathématicien** travaille alors sur la « machine analytique », système mécanique capable de réaliser une série de calculs établis à l'avance et inscrits sur des cartes perforées, considéré comme le **précurseur des ordinateurs**. La machine ne fut jamais construite entièrement mais elle était **fonctionnelle, comme l'a démontré une réalisation effectuée en 1991.**

La collaboration de Lady Ada Lovelace n'est pas connue précisément mais on considère qu'elle a réalisé les premières ébauches d'une écriture formelle des instructions à employer avec cette machine analytique pour réaliser des calculs donnés. En clair, elle a travaillé sur ce que l'on appelle aujourd'hui un langage informatique. En 1978, le nom Ada fut donné, en son hommage, à l'un de ces langages informatiques élaborés aux États-unis entre 1977 et 1983 chez CII-Honeywell Bull sous la direction de Jean Ichbiah.



## GRACE HOPPER

1906 → 1992

Informaticienne et docteure en mathématiques, Grace Hopper s'engage dans la marine américaine en 1943, avant d'y devenir amirale.

Elle commence à travailler sur le Harvard Mark I, le premier ordinateur, et devient ainsi la première programmeuse de l'Histoire. Elle continuera à travailler sur le développement des versions suivantes d'ordinateurs et inventera le premier compilateur. Mais elle ne s'arrêtera pas là ! Considérant que les programmes devraient s'écrire dans un langage plus proche de l'anglais que du langage machine, elle créa le COBOL, langage de programmation très utilisé encore aujourd'hui pour le développement des logiciels de gestion.



**\*Animafac**  
Le réseau des associations étudiantes

# Vocabulaire et notions

## Précisions (1)

**Algorithme** : ce n'est pas le code, c'est l'idée qui sera ensuite codée. La distinction entre un algorithme et un programme est un peu comme la différence entre un plan et un bâtiment.

- un algorithme correspond à une suite d'instructions et d'opérations, qui peuvent être ou non exprimées en langage mathématique, permettant de résoudre un problème
- un logiciel est un code informatique exécutable par un ordinateur, qui décrit comment appliquer un ou plusieurs algorithmes à certaines données
- un **traitement algorithmique** correspond à la mise en œuvre d'un tel logiciel pour une finalité particulière

• On **code** pour différents usages :

- Créer ou modifier un outil numérique : logiciel, application, site web...
- Automatiser, reproduire des actions répétitives et manuelles et minimiser les erreurs
- Trouver, extraire, ajouter un élément surtout quand on traite des données volumineuses
- Mettre en place une interface / un bouton / une interopérabilité entre données, tableaux, bases de données, sites
- Faire une analyse, créer une visualisation, corriger une erreur, paramétrer une fonction
- Réaliser toute sorte de produits sans changer d'outil
- Collaborer (un code peut être relu et complété), ouvrir son code à d'autres usagers ou au citoyen, etc.  
= Automatiser, répliquer, simplifier, accélérer, massifier, personnaliser, partager... → ne plus être un « simple utilisateur »

**Langage de programmation** : comme tout langage, il a un vocabulaire (lexique = la liste des mots admis), une grammaire (syntaxe = comment écrire et structurer correctement les instructions à partir des mots), un sens (sémantique = le sens des instructions)

Il en existe au moins des centaines car c'est le grand plaisir des informaticiens : à chacun il manque à un moment des structures de phrases permettant de bien exprimer sa pensée. La plupart du temps les langages nouveaux sont dérivés des précédents auxquels on ajoute quelque chose.

Lorsque deux programmeurs traitent le même problème, il peut arriver que les deux codes sources soient très différents, même en travaillant avec le même langage.

# Vocabulaire et notions

## Précisions (2)

**Développement (sprint)** : il organise la façon dont le code va être écrit et s'il est bien fait, les parties de codes écrites par plusieurs personnes vont bien se coordonner.

Si le produit numérique à développer est complexe (un logiciel, une plateforme, un site web...), il y aura une phase avant le développement proprement dit : la rédaction des **spécifications (spec)** du produit (fonctionnelles : ce que le produit doit savoir faire, le besoin ; techniques : avec quelles technologies, quel matériel, quelle architecture). Les **specs** ont pour objectif de partir sur des bases claires et de mettre toutes les parties prenantes d'accord.

Avec la méthode du **cycle en V**, les spécifications fonctionnelles sont détaillées dès le début du projet, de manière exhaustive. Avec une **méthode agile** type Scrum, elles sont rédigées tout au long du projet, avant le cycle de développement (sprint), par échanges très réguliers entre le **chef de produit (product owner)** et le développeur.

**Low-code, no-code** : méthodes facilitant la programmation, plus graphiques, plus ludiques, à base de glisser-déposer et autres raccourcis. Il s'agit toujours de coder in fine, mais le langage de programmation est « derrière » l'interface utilisateur.

**Intelligence artificielle générative (IAG)** : utilisation de l'IA pour créer de nouveaux contenus, comme du texte, des images, de la musique, de l'audio et des vidéos. Certaines IAG commencent à proposer du code à partir d'une commande passée en langage naturel (ou à chercher des bugs dans du code, etc.).

Selon l'étude *The economic potential of generative AI: The next productivity frontier – Mac Kinsey, juin 2023*, l'impact direct de l'IA sur la productivité de l'ingénierie logicielle pourrait représenter de 20 à 45 % des dépenses annuelles actuelles dans ce domaine.

Il faut toutefois détenir des compétences en programmation pour vérifier la fiabilité de la réponse de l'IAG, et tenir compte du fait que le développement de logiciels est une opération bien plus complexe que la simple production de code.

# Vocabulaire et notions

## Règles d'hygiène d'un bon code source

Un code source doit éviter le code inutile.

En effet, cela rend le programme plus confus, plus lent, plus sujet aux erreurs et plus gourmand en énergie.

Un code source doit avoir un enregistrement de ses différentes versions (versioning) pour comprendre ses évolutions, permettre de revenir à la version antérieure...

Un code source doit être testé pour détecter les bugs et vérifier le bon fonctionnement du développement.

Un code source doit être le plus portable possible, c'est-à-dire exécutable dans des environnements logiciels ou techniques différents avec peu ou pas de modifications.

Un code source doit être maintenable, c'est-à-dire facile à modifier et à étendre, résistant aux bugs, adaptables aux évolutions numériques.

Un code source doit être réversible c'est-à-dire ne pas être trop attaché à un code « propriétaire » au point d'en être dépendant.

Un code source doit être compréhensible pour faciliter son utilisation par d'autres, ainsi que son contrôle dans le cadre des codes sources publics.

Un code source doit être assorti de commentaires et d'une documentation.

Un code source peut être open source.

En open source, les développeurs mettent gratuitement le code source à disposition d'autres programmeurs.

Le but est de travailler ensemble sur des projets et de développer ou d'améliorer des logiciels à long terme.

Il existe une communauté open source très active, par exemple sur les plateformes (ou forges) GitHub ou GitLab.

Certaines conditions doivent être respectées pour utiliser le code, comme nommer l'auteur et faire apparaître la licence open source (il en existe de nombreux types : GPL, CeCILL, MIT, BSD...).

Se pose déjà la question de l'éthique des codes développés par des IA génératives (propriété intellectuelle, contrôle des effets du code...)



# Vocabulaire et notions

## Règles d'hygiène d'un code source public

Tout code source obtenu ou développé par un organisme remplissant une mission de service public est considéré comme un document administratif, relevant des obligations de publication en open data.

Depuis 2019, la DINUM recense les codes sources ouverts par les administrations et identifie parmi eux ceux qui ont le plus fort potentiel de réutilisation (<https://code.gouv.fr/fr/>)

L'utilisation de codes sources ouverts et de Communs numériques par les administrations est encouragée.

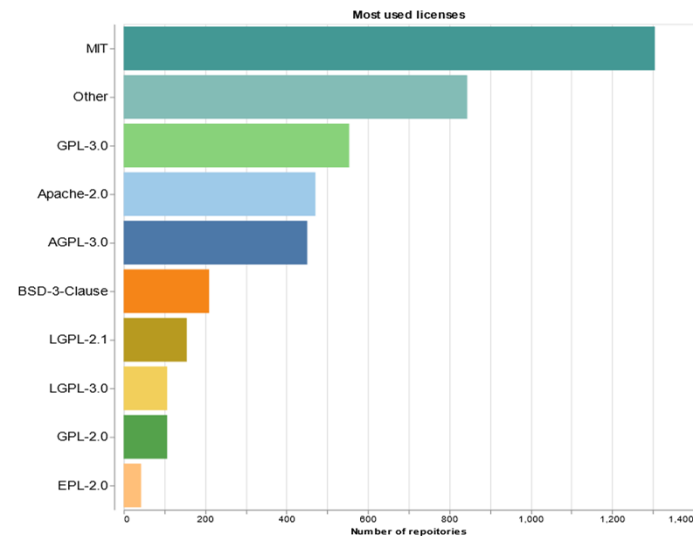
➔ Plan d'action logiciels libres et communs numériques (2021) : <https://code.gouv.fr/fr/plan-action-logiciels-libres-et-communs-numeriques/>  
Le socle interministériel de logiciels libres (SILL) est la liste des logiciels libres recommandés par la DINUM pour toutes les administrations (<https://code.gouv.fr/sill>). Il y a en 444 référencés au 29 février 2024.

**29 février 2024 sur code.gouv.fr**

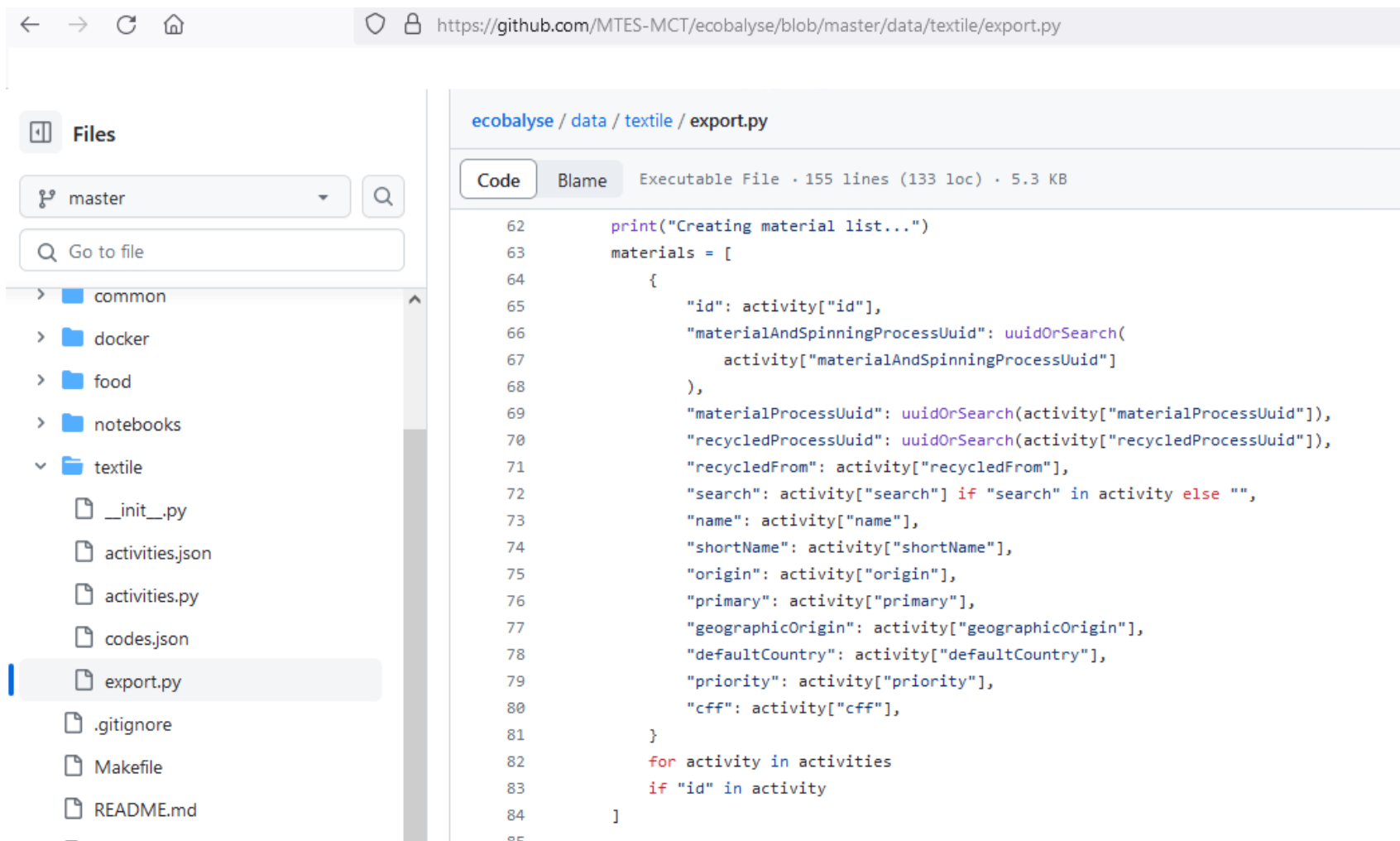
2994 organisations publiques ont déposé 19033 codes source

Langages de programmation les plus utilisés

- JavaScript 22.39 %
- Java 20.65 %
- Python 20.19 %



## Extrait du code d'Ecobalyse, site internet porté par le ministère (CGDD/SEVS) pour la mesure de l'empreinte environnementale des produits distribués en France



The screenshot shows a web browser displaying a GitHub repository. The address bar shows the URL: `https://github.com/MTES-MCT/ecobalyse/blob/master/data/textile/export.py`. On the left, a file explorer shows the repository structure with folders like `common`, `docker`, `food`, `notebooks`, and `textile`. The `textile` folder is expanded, showing files like `__init__.py`, `activities.json`, `activities.py`, `codes.json`, `export.py` (highlighted), `.gitignore`, `Makefile`, and `README.md`. On the right, the code editor shows the content of `export.py`, which is a Python script defining a list of materials for export.

```
ecobalyse / data / textile / export.py
Code Blame Executable File · 155 lines (133 loc) · 5.3 KB
62     print("Creating material list...")
63     materials = [
64         {
65             "id": activity["id"],
66             "materialAndSpinningProcessUuid": uuidOrSearch(
67                 activity["materialAndSpinningProcessUuid"]
68             ),
69             "materialProcessUuid": uuidOrSearch(activity["materialProcessUuid"]),
70             "recycledProcessUuid": uuidOrSearch(activity["recycledProcessUuid"]),
71             "recycledFrom": activity["recycledFrom"],
72             "search": activity["search"] if "search" in activity else "",
73             "name": activity["name"],
74             "shortName": activity["shortName"],
75             "origin": activity["origin"],
76             "primary": activity["primary"],
77             "geographicOrigin": activity["geographicOrigin"],
78             "defaultCountry": activity["defaultCountry"],
79             "priority": activity["priority"],
80             "cff": activity["cff"],
81         }
82     for activity in activities
83     if "id" in activity
84 ]
85
```

## Je vous remercie de me l'avoir posée...

A quel endroit code-t-on, et comment lance-t-on son code ?

Par exemple Visual Studio

R Studio (pour écrire en langage R) est un IDE

On code dans un fichier avec un éditeur de texte ou un environnement de développement intégré (*IDE - Integrated Development Environment*), outils qui permettent de debugger, de synchroniser le code avec l'outil de contrôle de version, etc.

Puis on utilise ce fichier avec des commandes pour lancer son code

- depuis l'IDE,
- depuis la ligne de commande d'un terminal (ou PowerShell pour Windows),
- depuis son navigateur pour des applications web et notebooks,
- ou en (double-)cliquant sur une icône quand dans certains cas le code devient une application sur le poste de travail

⇒ ces outils sont les interpréteurs / compilateurs du code (ex : en Python on appelle l'interpréteur Python).

Quand on lance une commande SQL, on peut l'écrire directement dans l'interpréteur interactif SQL.

Toutes les applications numériques peuvent-elles accueillir du nouveau code ?

Oui, mais pas de tout le monde.

Derrière toute application il y a un code mais si le code de l'application est fermé (ex : Windows, Word, et de très nombreuses autres applications), seuls les développeurs de ces applications peuvent le faire.

## Je vous remercie de me l'avoir posée...

Low code, no code, faire écrire du code par une IA générative... Est-il plus accessible de coder aujourd'hui ?

Tout le monde devrait-il savoir un peu coder ?

Plutôt oui (disons c'est plus facile de s'y mettre), mais :

- avant même de (low-no-)coder, il faut la compétence pour réfléchir à la meilleure façon de résoudre son problème, comprendre la logique derrière un code et un traitement de données
  - avec le le low-code et no-code, le code est moins « subtil » , il s'adapte moins au besoin précis
  - il faut détenir des compétences en programmation pour vérifier la fiabilité de la réponse de l'IA. Actuellement l'IA générative sert surtout aux codeurs confirmés
- Et plutôt non si on considère qu'aujourd'hui il faut utiliser plusieurs outils qui requièrent tous des habiletés différentes (« il faut apprendre à se servir de l'outil qu'on a en face de soi »).

Pour les personnes qui travaillent régulièrement avec la donnée, il est à peu près impossible de travailler sans coder à un moment ou un autre.

Pour les autres,

- dès qu'on a une action numérique répétitive, il est intéressant de la coder
- apprendre à coder prend du temps. Il faut être motivé.e et s'entraîner régulièrement, mais sans forcément besoin de formation longue pour débiter
- il semble utile que tout le monde comprenne la logique qui sous-tend le codage et en ait pratiqué un peu pour cela. On progresse ainsi dans sa maîtrise globale du numérique ... y compris pour savoir efficacement solliciter l'aide d'un.e collègue qui sait coder.

## ILLUSTRATION

Un des innombrables exemples que l'on trouve sur le net quand on demande à ChatGPT de proposer des lignes de code

La demande (« prompt ») :

Code le squelette d'un site en HTML, CSS et JavaScript. Le titre de mon site est "ChatGPT", le site est responsive et contient une barre de texte. Quand l'utilisateur entre du texte dans cette barre de texte, le texte s'ajoute en bas de la page.

La réponse de ChatGPT (extrait)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ChatGPT</title>
  <style>
    body {
      margin: 0;
      padding: 0;
      font-family: Arial, sans-serif;
    }
    header {
      background-color: #333;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
    main {
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
      display: flex;
      flex-direction: column;
      align-items: center;
    }
    textarea {
      width: 100%;
      height: 100px;
      padding: 10px;
      margin-bottom: 20px;
      box-sizing: border-box;
      resize: none;
    }
    #messages {
```

L'analyse du résultat par l'utilisateur :

Ce code HTML crée une page avec un titre "ChatGPT", un en-tête contenant ce titre, et un corps avec une zone de texte où les utilisateurs peuvent entrer des messages. Le code CSS applique un style minimal pour rendre la page responsive, avec une largeur maximale de 800 pixels pour le contenu principal et une mise en page centrée. Le code JavaScript permet d'ajouter les messages entrés par l'utilisateur en bas de la page lorsqu'il appuie sur la touche Entrée dans la zone de texte. Les messages s'affichent dans une liste avec un style de fond différent pour les distinguer.

Donc : ça correspond à la demande !!! (*pas bien complexe ici*)

## Je vous remercie de me l'avoir posée (2)...

Y a-t-il des langages particuliers pour les données ?

←  
Système de gestion de bases de données  
**PostgreSQL** au ministère

*NB : Il est à la mode de déstructurer les bases de données pour permettre leur répartition sur plusieurs serveurs. On nomme ces bases **NoSQL** pour « not only SQL »*

On peut citer **SQL** (*Structured Query Language / langage de requêtes structurées*), qui est un langage spécifique pour définir, interroger et modifier des bases de données, structurées ou non.

- La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles
- Sa partie langage de définition des données permet de créer et de modifier l'organisation des données dans la base de données
- Sa partie langage de contrôle de transaction permet de commencer et de terminer des transactions (une réservation, un achat, un paiement...)
- Sa partie langage de contrôle des données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.

On peut citer aussi le semi-langage **JSON** (format dérivé du langage javascript) pour les échanges de données, c'est aujourd'hui le standard des API

Y a-t-il des particularités de programmation dans un système d'information géographique ?

Plutôt pas, les langages sont assez standards (R, Python). Mais il y a une abstraction particulière nécessaire à la compréhension du type de donnée qu'est la « donnée géographique » : projection, tuilage, formats de données... Cela a suppose aussi d'acquérir un langage spécifique (intersection, union, soustraction, calcul de distance ...).

Y a-t-il des niveaux de programmation inaccessibles sans une formation longue ?

Oui, dès que l'on programme pour que ce programme soit utilisé par d'autres. C'est un métier.

Aujourd'hui 15 mars 2024 :

Par Amal YEFERNI et Nicolas ARIAS, de l'Ecolab du CGDD

*Une proposition de points de repère sur la notion de programmation pour la donnée*

➔ **Démonstrations dans le cadre du projet « Indicateurs territoriaux pour la transition écologique » et autres projets**

Par Olivier CHANTREL, du service des données et des études statistiques (SDES) du CGDD

*Une proposition de points de repère sur la notion de programmation pour la donnée*

➔ **Démonstrations dans le cadre de l'outil « R »**

A méditer...

Avec un compagnonnage durant la formation, on peut transformer 30 à 50% des agents en codeurs / codeuses.

Les plus gros freins sont dans l'adaptation au réseau ministériel : proxy, droits admin, etc...

Thomas GRANDJEAN

Chargé du profil environnemental régional et études transversales-DREAL Hauts-de-France



A méditer...

« Code sans conscience n'est que bug assuré »

Jean-Yves Chatelier

Correspondant Intelligence artificielle et données métier · Ineris

BONUS

Accueil / Les ressources

## Le 03 Juin 2022 L'art de traiter la donnée

Publié sur le 03/06/2022



Ce webinaire explique les différents langages informatiques pour permettre d'analyser et de traiter ses données.

### Intervenants

- Julien Guerard (SGARE Grand Est)
- Thierry Zorn (DREAL Grand Est)
- Christophe Heinkelé (CEREMA)

**A noter:** les présentations ci-dessous sont diffusées à partir de la chaîne **YouTube DataGrandEst**. Si vous rencontrez des difficultés pour les visionner, nous vous invitons à utiliser les liens mentionnés dans la rubrique "Document (lien) plus bas sur cette page"

**Regarder le webinaire complet**

### LES OUTILS

- Rechercher des données
- Construire sa carte
- Consulter des cartes
- Réutiliser des données

### ACCÈS RAPIDES

[Annuaire](#) | [FAQ](#) | [Agenda](#)

[https://intra.dreal-grand-est.e2.rie.gouv.fr/langages-python-r-restitution-de-l-etude-a62240.html?id\\_rub=977&TARGET=https%3A%2F%2Fintra.dreal-grand-est.e2.rie.gouv.fr%2Flangages-python-r-restitution-de-l-etude-a62240.html%3Fid\\_rub%3D977](https://intra.dreal-grand-est.e2.rie.gouv.fr/langages-python-r-restitution-de-l-etude-a62240.html?id_rub=977&TARGET=https%3A%2F%2Fintra.dreal-grand-est.e2.rie.gouv.fr%2Flangages-python-r-restitution-de-l-etude-a62240.html%3Fid_rub%3D977)

Actualités SCDD

# Langages Python/R : restitution de l'étude comparative

Publié le 27 janvier 2023 | Mis à jour le 1er février 2023

Langages R et Python : sont-ils adaptés pour répondre à tous nos besoins ?  
Si vous souhaitez approfondir le sujet

Dans le cadre du projet Datahub de la DREAL Grand Est, l'équipe projet avait besoin d'une expertise sur des choix techniques des langages à utiliser selon les cas d'usages du Datahub : collecte de données, traitement et analyse de données, Data visualisation et application Web.

Cette étude a donc été commandée au prestataire qui accompagne le SCDD sur le projet. L'événement s'est tenu le jeudi 12 janvier 2023.

## Langages R et Python : sont-ils adaptés pour répondre à tous nos besoins ?

Nous mettons à votre disposition la [présentation](#) (format pdf - 214.4 ko - 19/01/2023)

Vous pouvez également regarder la restitution de cette étude en [replay](#).

La vidéo comporte 30 minutes de présentation et 1h00 d'échanges entre les participants et les intervenants experts en la matière, missionnés par notre prestataire.

## Si vous souhaitez approfondir le sujet

Vous pourrez revoir

- le webinaire DataGrandEst : [L'art de traiter la donnée](#)
- le webinaire du Funathon : [Pourquoi faire du Python quand on connaît R](#)

Vous pouvez également assister à des formations

- pour le langage R, Sachez que des [formations](#) sont proposées aux agents du MTE. Le SCDD dispose d'un formateur expérimenté en langage R.
- pour le langage Python : [Python : des fondamentaux aux concepts avancés du langage](#). Des [sessions de formation](#) sont organisées par le ministère de l'Agriculture.

<https://open.dataactivist.coop/docs/r-ou-python>

**DATAACTIVIST**

```
505 self._headers_buffer = []
506 self._headers_buffer.append((" %s %d %s\r\n" %
507     (self.protocol_version, code, message)).encode(
508         'latin-1', 'strict'))
509
510 def send_header(self, keyword, value):
511     """Send a MIME header to the headers buffer."""
512     if self.protocol_version == HTTP/1.0:
513         self._headers_buffer.append((" %s: %s\r\n" %
514             (keyword, value)).encode('latin-1', 'strict'))
515     else:
516         self._headers_buffer.append((" %s: %s\r\n" % (keyword, value)).encode('latin-1', 'strict'))
517
518 if keyword.lower() == 'connection':
519     if value.lower() == 'close':
520         self.close_connection = True
521     elif value.lower() == 'keep-alive':
```

## R ou Python - quel outil pour la Data Science ?

📅 27 mars 2023    ✍️ Blog

data science   R   python   R Studio   langage   librairies

👤 Anthony Gigerich





Comment choisir entre R et Python, deux langages de data science incontournables.

## Encore un peu de ... vocabulaire et notions

**Plug-in** : également appelé module d'extension, add-in ou add-on, c'est un programme additionnel pour une application internet ou un logiciel. Développé en cohérence avec un programme principal, il lui offre de nouvelles fonctionnalités sans modifier son code source. *L'équivalent d'une option pour une voiture*  
L'ajout de plug-in est très à la mode et les éditeurs de logiciels en prévoient nativement la possibilité (en général). On peut les choisir « sur étagère » et les « activer », mais aussi en programmer.

**Fork** : nouveau logiciel créé à partir du code source d'un logiciel existant. Les forks sont courants dans le domaine des logiciels libres et permettent de proposer d'autres possibilités (*et le meilleur gagne*). À l'inverse des forks, certains projets de logiciels libres décident de **fusionner** leur base de code car ils partagent des objectifs similaires

**Script** : un (souvent petit) programme écrit dans un langage interprété pour automatiser certaines tâches d'une application (ex : mise à jour d'informations dans une base de données)

**Prompt** : art de questionner une intelligence artificielle générative (IAG) en langage naturel pour lui faire émettre la réponse la plus proche de celle souhaitée. C'est une forme originale et nouvelle de programmation, qui se nourrit de l'expérience des utilisateurs faute de savoir comment l'IAG a « appris » (probabilités, réseaux de neurones...)

**Devops** : approche combinant équipes de développeurs et équipes d'exploitation informatique pour mener des projets de développement numérique (logiciels, etc.) plus agiles (facilitation des ajustements en continu)

## Encore un peu de ... vocabulaire et notions

**Package** : une application logicielle est livrée sous forme d'ensemble de fichiers et de répertoires appelés packages. La création sous forme d'un ou plusieurs packages facilite le transfert du logiciel sur un support de distribution. Certains packages sont utilisables pour plusieurs usages différents et permettent de gagner du temps de codage.

**Framework** (ou infrastructure logicielle) : ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application.

Le framework fonctionne comme un cadre ou un patron : il établit les fondations d'un logiciel ou son squelette applicatif, que les développeurs peuvent enrichir. L'objectif du framework est de simplifier et d'uniformiser le travail des développeurs. Un framework est souvent construit autour d'un langage de programmation.

Les développeurs ont tendance à se spécialiser sur un ou plusieurs frameworks dont ils affectionnent la logique et les performances.

**Tech stack** : ensemble de technologies, de langages de programmation, de **frameworks**, de bibliothèques et d'outils (serveurs, etc.) utilisés pour créer et exécuter une application logicielle ou un système particulier.

**Usine logicielle** : approche organisée du développement logiciel qui fournit aux équipes de conception et de développement une infrastructure, des outils et une méthode reproductible et bien définie pour créer et mettre à jour des logiciels.

Les outils majeurs dans une usine logicielle comprennent notamment ceux qui permettent la gestion du cycle de vie d'un projet. Les projets deviennent managés avec par exemple des tableaux de bord.

## Encore un peu de ... vocabulaire et notions

**Pseudo-code** : un pseudo-code (également appelé LDA pour Langage de Description d'Algorithmes) fait référence à une syntaxe ressemblant à un code, généralement utilisée pour indiquer aux humains le fonctionnement d'une syntaxe de code ou pour illustrer la conception d'un élément d'architecture de code. Il n'est pas exécutable comme un véritable code.

**Notebook** : en programmation, un notebook, aussi fréquemment appelé calepin électronique, voire simplement calepin, est une interface de programmation interactive permettant de combiner des sections en langage naturel et des sections en langage informatique. Les notebooks sont notamment utilisés en science des données pour explorer et analyser des données. Les notebooks peuvent ensuite être enregistrés sous forme de document. C'est un exemple de programmation lettrée.

**Mashup** (ou application composite) : application qui combine du contenu ou du service provenant de plusieurs logiciels ou services de données plus ou moins hétérogènes et les affiche sur une seule interface. Il offre une expérience visuelle et interactive aux utilisateurs. Le développement est simple et rapide, car les développeurs ont besoin de moins de code.

**Linting** : vérification automatisée du code source pour détecter les erreurs de programmation et de style.



## Textes et notions liées au codage

Code des relations entre le public et l'administration  
Article L300-2

Sont considérés comme documents administratifs, au sens des titres Ier, III et IV du présent livre, quels que soient leur date, leur lieu de conservation, leur forme et leur support, les documents produits ou reçus, dans le cadre de leur mission de service public, par l'Etat, les collectivités territoriales ainsi que par les autres personnes de droit public ou les personnes de droit privé chargées d'une telle mission. Constituent de tels documents notamment les dossiers, rapports, études, comptes rendus, procès-verbaux, statistiques, instructions, circulaires, notes et réponses ministérielles, correspondances, avis, prévisions, **codes sources** et décisions.

Loi n° 2016-1321 du 7 octobre 2016 pour une République numérique  
Article 16

Les administrations mentionnées au premier alinéa de l'article L. 300-2 du code des relations entre le public et l'administration veillent à préserver la maîtrise, la pérennité et l'indépendance de leurs systèmes d'information.

**Elles encouragent l'utilisation des logiciels libres et des formats ouverts lors du développement, de l'achat ou de l'utilisation, de tout ou partie, de ces systèmes d'information.** Elles encouragent la migration de l'ensemble des composants de ces systèmes d'information vers le protocole IPV6, sous réserve de leur compatibilité, à compter du 1er janvier 2018.

## Textes et notion liées au codage

### Code des relations entre le public et l'administration

#### •Article L311-3-1

Sous réserve de l'application du 2° de l'article [L. 311-5](#) (non communicables), une décision individuelle prise sur le fondement d'un **traitement algorithmique** comporte une mention explicite informant l'intéressé. Les règles définissant ce **traitement** ainsi que les principales caractéristiques de sa mise en œuvre sont communiquées par l'administration à l'intéressé s'il en fait la demande. Les conditions d'application du présent article sont fixées par décret en Conseil d'Etat.

#### •Article R311-3-1-1

La mention explicite prévue à l'article [L. 311-3-1](#) indique la finalité poursuivie par le **traitement algorithmique**. Elle rappelle le droit, garanti par cet article, d'obtenir la communication des règles définissant ce **traitement** et des principales caractéristiques de sa mise en œuvre, ainsi que les modalités d'exercice de ce droit à communication et de saisine, le cas échéant, de la commission d'accès aux documents administratifs, définies par le présent livre.

#### •Article R311-3-1-2

L'administration communique à la personne faisant l'objet d'une décision individuelle prise sur le fondement d'un **traitement algorithmique**, à la demande de celle-ci, sous une forme intelligible et sous réserve de ne pas porter atteinte à des secrets protégés par la loi, les informations suivantes :

- 1° Le degré et le mode de contribution du **traitement algorithmique** à la prise de décision ;
- 2° Les données traitées et leurs sources ;
- 3° Les paramètres de **traitement** et, le cas échéant, leur pondération, appliqués à la situation de l'intéressé ;
- 4° Les opérations effectuées par le **traitement**.

#### Article L312-1-3

Sous réserve des secrets protégés en application du 2° de l'article L. 311-5, les administrations mentionnées au premier alinéa de l'article L. 300-2, à l'exception des personnes morales dont le nombre d'agents ou de salariés est inférieur à un seuil fixé par décret, publient en ligne les règles définissant les principaux **traitements algorithmiques** utilisés dans l'accomplissement de leurs missions lorsqu'ils fondent des décisions individuelles.

### Loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés

#### Article 47

(...)

Aucune décision produisant des effets juridiques à l'égard d'une personne ou l'affectant de manière significative ne peut être prise sur le seul fondement d'un **traitement automatisé** de données à caractère personnel, y compris le profilage, à l'exception : (...)

2° Des décisions administratives individuelles prises dans le respect de l'article L. 311-3-1 et du chapitre Ier du titre Ier du livre IV du code des relations entre le public et l'administration, à condition que le traitement ne porte pas sur des données mentionnées au I de l'article 6 de la présente loi (*traitements interdits*). Ces décisions comportent, à peine de nullité, la mention explicite prévue à l'article L. 311-3-1 du code des relations entre le public et l'administration. Pour ces décisions, le responsable de traitement s'assure de la maîtrise du **traitement algorithmique** et de ses évolutions afin de pouvoir expliquer, en détail et sous une forme intelligible, à la personne concernée la manière dont le traitement a été mis en œuvre à son égard.

Voir : Etalab, guide des algorithmes publics : <https://guides.etalab.gouv.fr/algorithmes/guide/>

## Petit glossaire de la donnée

- La **donnée** est l'élément de base de la connaissance. On parle de **jeu de données** ou de **collection de données** pour un recueil de données.
- On parle de **données brutes** pour caractériser des données tout juste acquises (via des appareils de mesure par exemple). On peut **croiser** des données, les **enrichir**, les **agréger**, les **dégrader**, ...
- Une **base de données** est un recueil **structuré** de données.
- On structure une base de données à l'aide d'une **modélisation conceptuelle de données** ou d'un **schéma de données** : pour cela il faut collaborer entre experts de la donnée et experts métiers. Les schémas de données permettent de décrire des modèles de données : quels sont les différents champs, comment sont représentées les données, quelles sont les valeurs possibles etc.
- Un **standard de données** est un schéma de données associé à des règles de gouvernance, l'ensemble étant consensuel et partagé par une communauté ou un écosystème
- Quand on connaît le schéma de données, on peut émettre des **requêtes** sur la base de données pour n'obtenir que les données qui obéissent à nos critères. Les **API** (Application programming interface) sont des applications informatiques qui permettent à d'autres programmes d'obtenir des données selon des requêtes.
- On peut croiser des bases de données si on dispose d'une information identique et discriminante dans chaque base, c'est la **jointure**.
- Les informations qui décrivent les données ou les bases de données s'appellent les **métadonnées** (titre, date d'acquisition, date de mise en ligne, propriétaire, licence, informations sur la qualité, URL de téléchargement, ...)
- Les métadonnées permettent de constituer des **catalogues de données** : les données y sont **exposées**. Ainsi un bureau d'études va consulter les catalogues pour connaître l'état des connaissances sur un domaine qu'il va analyser. Il pourra alors **consommer** les données dont il a besoin. On dit que les données **circulent**, elles sont **partagées**.
- Certaines données sont **ouvertes** (on parle d'**opendata**), d'autres qui portent des informations sensibles ne peuvent être consommées que par des utilisateurs habilités : le partage, ce n'est pas que de l'opendata.
- Les catalogues peuvent communiquer entre eux automatiquement et se mettre à jour des dernières évolutions d'autres catalogue, c'est le **moissonnage**.
- Les données sont **hébergées** au sein **d'infrastructures de données** (portails, systèmes d'information, ...).

## On parle de la donnée au sein du pôle ministériel parce que :

Notre pôle ministériel, vu sa taille et ses missions, génère et utilise (et doit ouvrir et protéger) **une grande masse de données** très utiles au fonctionnement quotidien de la Nation comme à la réussite de la transition écologique et énergétique (NB : lutter aussi contre les contre-vérités). C'est aussi le cas des collectivités, avec lesquelles notre pôle ministériel, et notamment les services déconcentrés, est en relation pour gérer le territoire. La **planification écologique** du projet France Nation Verte va avoir besoin de s'appuyer sur ces données existantes et à créer, notamment au niveau territorial

Il existe des dispositifs juridiques spécifiques sur le droit de toute personne d'**accéder aux informations relatives à l'environnement**. Il existe aussi des dispositifs juridiques spécifiques imposant la production et le partage des données des transports, de l'énergie, des valeurs foncières, ...

Certains opérateurs ont **pour mission de produire et diffuser de la donnée** : Météo France, Institut national de l'information géographique et forestière (IGN), Office français de la biodiversité (OFB), notamment

Le pôle ministériel compte en son sein un important **réseau scientifique et technique** qui crée et publie des données de recherche

Le pôle ministériel doit former à la donnée dans ses **écoles** (ENTE, ENTPE, ENPC, ENSG, ENSM, ENSAM, ENAC, INSTN)

Plusieurs grandes politiques du pôle ministériel s'appuient sur des délégataires de service public, dont les données peuvent être qualifiés **d'intérêt général**

La directive du 14 mars 2007 établissant une infrastructure d'information géographique européenne (« **Inspire** ») a eu une très grande importance au sein de nos services pour structurer et ouvrir les jeux de données géographiques et environnementales. Dans les 228 données de références au sens d'Inspire, au moins 220 sont issues du pôle ministériel. Ceci explique que les géomaticiens fassent partie *de facto* du champ des métiers de la donnée (même si une montée en compétences est demandée)

Une grande partie des « **Données de forte valeur** » qui devront être partagées au niveau européen au 10 avril 2024 émaneront aussi de nos politiques publiques ministérielles : géospatiales ; observation de la Terre et environnement ; météorologiques ; mobilité (+ statistiques sur les entreprises et la population)

Plusieurs entreprises appartenant aux secteurs économiques régulés par le pôle ministériel font partie des « opérateurs de service essentiels » et des futurs « secteurs hautement critiques » des directives européennes sur la **cybersécurité** de 2016 et 2022 : transports, énergie, eaux potables et usées

Le pôle ministériel a en charge des enjeux de **régulation** d'usages experts de la donnée : drones, véhicules autonomes, ville « intelligente » (*smart city*), ...

Le pôle ministériel est fortement présent sur des **nouveaux types et usages de données** : satellitaires, lidar 3D, jumeaux numériques, ...

Le saviez-vous ?

- L'agence nationale de la cohésion des territoires (**ANCT**), opérateur de notre pôle ministériel depuis juillet 2022, porte les politiques d'accès des territoires et des personnes au numérique et donc à de multiples usages de la donnée
- **Météo France**, opérateur de notre pôle ministériel, possède deux des plus gros supercalculateurs français pour faire tourner ses modèles : Belenos et Taranis
- L'**IGN** cartographie l'anthropocène par des cartes thématiques sur un nombre limité d'enjeux écologiques majeurs qui rendent compte des changements rapides du territoire et des conséquences sur l'environnement

## Principes généraux de qualité de la donnée

Document web facile à lire et librement réutilisable

Ministère de la Transition écologique et de la Cohésion des territoires  
Ministère de la Transition énergétique

[https://www.ecologie.gouv.fr/sites/default/files/Principes\\_generaux\\_qualite\\_des\\_donnees\\_MTE\\_MTECT\\_1-1.pdf](https://www.ecologie.gouv.fr/sites/default/files/Principes_generaux_qualite_des_donnees_MTE_MTECT_1-1.pdf)



Pour aller plus loin sur la donnée et la transformation numérique des administrations

2/4

**SAUVEZ LA DATE !**

**11 juin 2024**

**Journée de la donnée  
du pôle ministériel**

Présentiel (Sequoia-La Défense) et distanciel


# Pour aller plus loin sur la donnée et la transformation numérique des administrations

3/4

Cour des comptes 

## COMPARAISON DE 10 STRATÉGIES NATIONALES SUR L'INTELLIGENCE ARTIFICIELLE

Avril 2023

 **CONSEIL D'ÉTAT** Conseil du Premier ministre

### Intelligence artificielle et action publique : construire la confiance, servir la performance

Etude adoptée en assemblée générale plénière du 31/03/2022




## DATAculturation

Vidéos courtes en ligne

## Feuille de route de la donnée, des algorithmes et des codes sources

Le Mardi 17 janvier 2023

Pilotée par le Commissariat général au développement durable, la feuille de route propose un cap pour améliorer l'usage de la donnée publique et accélérer la transition écologique sur un horizon de trois à quatre ans. Elle constitue une réponse à la nouvelle politique publique de la donnée inscrite dans la circulaire du Premier ministre du 27 avril 2021 relative à une politique de la donnée, des algorithmes et des codes sources.






## Maîtriser l'IA au service de l'action publique

Une responsabilité individuelle et collective

Sous la direction de Christian Paul et Daniel Le Métayer

Avec les contributions de Joffrey Becker, Hugues Berry, Bilal Benbouzid, Vincent Boustitou, Marc Clément, Antoine Coumou, Nicolas Delfieux, Clément Hérin, Laurent Inard, Daniel Kaplan, Clément Maki, Yannick Meheucour, Benjamin Nguyen, Bertrand Ralhès, Fabienne Perez, Patrick Perrot, Juan-David Pinzon, Hervé Rivano, Isabelle Sayn, Kitty Steward, Charles Thibout, Félicien Vallet, Renaud Vedel, Henri Verdier

### Feuille de route Intelligence artificielle et transition écologique du pôle ministériel 2021-2024



## Pour une politique publique de la donnée


### Rapport

Eric Bothorel - Stéphanie Combes – Renaud Vedel, décembre 2020



HÉLÈNE BÉGON-TAVERA

## LA TRANSFORMATION NUMÉRIQUE DES ADMINISTRATIONS




 **Etalab**  
Politique publique de la donnée

## Ouvrir, partager et valoriser les données publiques

 **AVIS DU CONSEIL NATIONAL DU NUMÉRIQUE**  
JUILLET 2020

## FAIRE DES DONNÉES ENVIRONNEMENTALES DES DONNÉES D'INTÉRÊT GÉNÉRAL



## Cours en ligne gratuits et en libre accès sur la plateforme Openclassrooms

L'Ecolab invite tout agent du pôle ministériel à suivre les cours de la plateforme en ligne OpenClassRooms

On trouve 61 cours dédiés à la donnée (et assimilés) sur la plateforme OpenClassRooms

(non compris les cours généraux sur le langage Python)

Objectif IA : initiez-vous à l'intelligence artificielle	Sécurisez vos données avec la cryptographie	Maîtrisez les bases de données NoSQL
Explorez vos données avec des algorithmes non supervisés	Réalisez des calculs distribués sur des données massives	Modélisez vos données avec les méthodes ensemblistes
Classez et segmentez des données visuelles	Analysez vos données textuelles	Initiez-vous au Deep Learning
Réalisez une analyse exploratoire de données	Découvrez le fonctionnement des algorithmes	Découvrez le monde des Systèmes d'Information
Utilisez Spring Data MongoDB pour interagir avec des bases de données NoSQL	Implémentez vos bases de données relationnelles avec SQL	Initiez-vous au langage R pour analyser vos données
Concevez des architectures Big Data	Initiez-vous à la statistique inférentielle	Initiez-vous au Machine Learning
Manipuler l'information	Comprendre le Web	Découvrez l'univers de la cybersécurité
Créez votre Data Lake	Récupérez et affichez des données distantes	Initiez-vous au Design Thinking
Appropriiez-vous la démarche UX en pratique	Réalisez un dashboard avec Tableau	Maîtrisez les bases des probabilités
Analysez et modélisez des séries temporelles	Maîtrisez les fondamentaux d'Excel	Perfectionnez-vous sur Excel
Mettez en place un système de veille informationnelle	Découvrez le métier de chef de projet SI	UX design : découvrez les fondamentaux !
Evaluez les performances d'un modèle de machine learning	Mettez en place un plan de gouvernance de votre SI	Initiez-vous aux traitements de base des images numériques
Maîtrisez les risques juridiques liés au numérique	Entraînez un modèle prédictif linéaire	Utilisez des modèles supervisés non linéaires
Réalisez des dashboards avec Power BI	Modélisez vos bases de données	Initiez-vous à Python pour l'analyse de données
Gérez vos données localement pour avoir une application 100 % hors-ligne	Découvrez les bibliothèques Python pour la Data Science	Comprendre le big data à travers les films de cinéma
Développez votre culture des données	Initiez-vous à la gouvernance des données	Nettoyez et analysez votre jeu de données
Requêtez une base de données avec SQL	Protégez les données personnelles	Gérez des flux de données temps réel
Réalisez des modélisations de données performantes	Réalisez des rapports statistiques clairs et impactants	Appliquez l'apprentissage statistique aux objets connectés
Analysez des données pour prendre des décisions de design	Utilisez ChatGPT pour améliorer votre productivité	Implémentez votre base de données relationnelle avec ASP.NET Core
Utilisez ChatGPT pour améliorer votre productivité	Trouvez votre premier emploi en data	Découvrez les fondamentaux VBA
Analysez vos données avec VBA		



## Pour revoir les précédents séminaires du cycle sur la culture de la donnée

*Soit s'inscrire, soit déjà inscrit.e et demander le renvoi du lien d'accès.*

*Si vous souhaitez les diaporamas projetés pendant ces webinaires, le plus simple est de les demander à [helene.begon@developpement-durable.gouv.fr](mailto:helene.begon@developpement-durable.gouv.fr)*

### Séminaire n°1 du 26 janvier 2023

**Pourquoi parle-t-on de la donnée au sein du pôle ministériel ?**

<https://app.livestorm.co/cvrh-de-clermont-ferrand/les-donnees-pourquoi-en-parle-t-on>

### Séminaire n°2 du 14 avril 2023

**Explorer des jeux de données du pôle ministériel**

<https://app.livestorm.co/cvrh-de-clermont-ferrand/decouvrir-les-donnees-du-pole-ministeriel>

### Séminaire n°3 du 9 juin 2023

**L'intelligence artificielle et le pôle ministériel**

<https://app.livestorm.co/cvrh-de-clermont-ferrand/lintelligence-artificielle-et-le-pole-ministeriel>

### Séminaire n°4 du 29 août 2023

**La datavisualisation**

<https://app.livestorm.co/cvrh-de-clermont-ferrand/la-datavisualisation>

### Séminaire n°5 du 6 octobre 2023

**« Que font concrètement les collègues qui travaillent sur la donnée ? »**

<https://app.livestorm.co/cvrh-de-clermont-ferrand/que-font-concretement-nos-collegues-qui-travaillent-sur-la-donnee>

## L'Ecolab du CGDD\* en quelques mots

### Administration ministérielle des données Gouvernance de la donnée

pour le Commissaire général au développement durable, administrateur ministériel des données, des algorithmes et des codes sources

**Feuille de route ministérielle du numérique et de la donnée** (avec la Dnum)  
**Feuille de route ministérielle de l'intelligence artificielle**  
**Ecosphères : futur guichet d'accès** à la donnée de la transition écologique  
**Information géographique** : CNIG \*, directive Inspire, partenariat avec l'Afigeo \*  
**France Nation Verte** : Indicateurs territoriaux transition écologique, financements  
Montée de la culture et des **compétences** sur la donnée  
Structuration de **l'offre de service juridique** sur la donnée

### Innovation ouverte autour de la donnée pour la transition écologique

**Offre d'un bouquet de services** sur la donnée : Services Données  
**Communauté données en santé-environnement** : Green data for health (GD4H)  
**Démonstrateurs d'intelligence artificielle** dans les territoires  
**Mission pour les services « Connaissance »** du pôle ministériel

### GreenTech Innovation

Labellisation et accompagnement de **start-up et PME** de la GreenTech  
Animation du réseau national des **incubateurs** de la GreenTech  
Accès à la **commande publique et privée** pour les entreprises GreenTech  
Animation du **Living-Lab** de Saint Mandé



Etc.

- *Commissariat général au développement durable*
- *Conseil national de l'information géolocalisée*
- *Association française pour l'information géographique*