



B-INN-000

WORKSHOP

Generate CV with R Markdown language

2.0



WORKSHOP

language: R, RMarkdown



INSTALLATION

Etape 1, R :

Dans un premier temps il faut installer l'outil *R*.

```
Terminal
~/B-INN-000> sudo apt-get install r-base r-base-dev
```

Pour plus d'informations sur le logiciel *R* [cliquez ici](#).

Etape 2, RMarkdown :

A présent, il faut installer les packages *knitr*, *markdown* et *rmarkdown* qui seront utiles pour la génération du pdf.

Pour cela, créez votre premier script *R*.

```
Terminal
~/B-INN-000> touch install.R
```

Saisissez vous de votre IDE pour éditer le script *install.R*.

Ecrivez la ligne suivante dans le nouveau fichier :

```
install.packages("knitr")
install.packages("markdown")
install.packages("rmarkdown")
```

Puis lancez le script à l'aide de la commande suivante :

```
Terminal
~/B-INN-000> sudo Rscript install.R
```

Pour plus d'informations sur le package *RMarkdown* [cliquez ici](#).

Etape 3, LaTeX :

Pour l'installation de latex, il faudra exécuter la commande suivante :

```
Terminal
~/B-INN-000> sudo apt-get install texlive-latex-extra lmodern
texlive-lang-french texlive-latex-base texlive-fonts-extra texlive-extra-utils
texlive-latex-recommended pandoc
```



UTILISATION BASIQUE DE RMARKDOWN

Mon premier script RMarkdown :

Comme fait précédemment, créez deux nouveaux fichiers `createPdf.R` et `firstPdf.Rmd`.

Le premier fichier `createPdf.R` lancera le second script `firstPdf.Rmd` à l'aide de la ligne suivante :

```
rmarkdown::render(input = "firstPdf.Rmd", output_format = "pdf_document")
```

Le second script `firstPdf.Rmd` contiendra un simple message pour commencer :

```
Hello world!
```

Pour lancer la génération du pdf, cela se passera comme pour l'installation des packages :

```
Terminal
~/B-INN-000> Rscript createPdf.R
```

Effets basiques :

Maintenant que vous savez générer un pdf à l'aide de R, essayez de reproduire l'affichage suivant :

Header N 1

Header N 2

Header N 3

Header N 4

Header N 5

Header N 6

Italic

Bold

~~Strikethrough~~

Pour avoir des renseignements sur la mise en page Markdown ainsi que son affichage, [cliquez ici](#).



GÉNÉRATION DE SON PREMIER CV

Les templates :

Pour cette partie, nous allons utiliser un template en latex qui nous facilitera la tâche !

Celui-ci effectuera une belle mise en page car comme vous avez pu le remarquer, le markdown est vite limité par son esthétisme...

Pour se faire, il va falloir choisir un template qui répond à notre demande sur internet. Voici un GitHub regroupant de nombreux templates : [cliquez ici](#).

Pour ce Workshop nous allons utiliser un template que j'ai modifié, disponible sur mon GitHub : [cliquez ici](#).

L'en tête :

Nous allons ajouter cette fois-ci dans le `.Rmd` une en-tête contenant différents paramètres utiles pour le template. L'en-tête est délimité par trois tirets (---) avant et après:

```
---
output:
  pdf_document:
    latex_engine: pdflatex
    template: ./TemplateCV.tex

title: "Recherche de stage"
firstname: "Nathan"
lastname: "Fain"
jobtitle: ""
address: ""
city: ""
country: ""
phone: ""
web: ""
email: ""
domaine: ""
profil: "moi.png"
---
```

Maintenant que vous savez utiliser des templates, essayez de reproduire l'affichage suivant à l'aide des paramètres et des effets basiques de RMarkdown:



Nathan Fain

Développeur freelance

Recherche de stage

Formation

2015-2018 - Baccalauréat Série S, Lycée Philippe de Girard, Arignon.
Option Sciences de l'Ingénieur, Spécialité I2S.

2018-2020 - Expert en technologies de l'information, Epitech, Montpellier.

2020-2021 - IUT-STM, école de double compétence.
Formation permettant l'apprentissage du tuteur, Paris.

Expérience professionnelle

2019-2020 - Institut national de la recherche agronomique, Arignon.
Stage de 6 mois au centre de recherche INRA.

2020-2021 - Assistant pédagogique à Epitech

Langues

Anglais, Niveau B1

Italien, Niveau A2

Compétences informatiques

C. Niveau avancé. J'ai une année d'expérience assez intensive avec ce langage de programmation qui m'a permis de réaliser de nombreux projets graphiques, mathématiques et logiciels.

Java. Niveau intermédiaire. Réalisation d'un projet graphique pour téléphone. Travail avec ce langage en utilisant le framework Spring.

R. Niveau avancé. Utilisation de ce langage durant 6 mois de stage. Conception de différents scripts de tests et génération de pdf.

Résidence des Arts 20 rue Cassini de la Salle appartement 62 - Montpellier

☎ +33 7 82 89 26 86 ✉ nathan.fain@epitech.eu
🌐 github.com/NathanFain



INTERFACE DE SAISIE

Pour se faire, nous allons utiliser le packages `shiny`.
Comme précédemment, il faudra l'installer via la ligne :

```
install.packages("shiny")
```

Ce nouveau package nous permettra de créer une simple page web pour saisir les données à rentrer dans le CV (nom, prénom, photo, ...).

Pour plus d'information sur le package shiny [cliquez ici](#).

Voici une page web basique, vous pouvez utiliser ce code comme base :

```
library(shiny)

options(shiny.port = 8100)

ui <- fluidPage(
  titlePanel("Générateur_de_CV"),
  textInput("firstname", "Prénom:", ""),
  actionButton("do", "Lancer la génération du CV")
)

server <- function(input, output) {
  observeEvent(input$do, {
    rmarkdown::render(input = "firstPdf.Rmd", output_format = "pdf_document",
      params = list(
        firstname = input$firstname
      ))
  })
}

shinyApp(ui, server)
```

L'objectif sera par la suite de passer en paramètres toutes les informations repris par l'utilisateur sur la page web et de générer le CV en fonction de celle-ci.

Exemple

`createPdf.R`:

```
...

rmarkdown::render(input = "firstPdf.Rmd", output_format = "pdf_document",
  params = list(
    title = input$title,
    firstname = input$firstname,
    lastname = input$lastname,
  )
...

```



ScriptCV.Rmd:

. . .

```
params:
  title: ""
  firstname: ""
  lastname: ""
```

. . .

```
title: "r params$title"
firstname: "r params$firstname"
lastname: "r params$lastname"
```

. . .

A vous de jouer!