

MARKDOWN-PDF

Guide d'installation de **markdown-pdf** pour générer les documents PDF depuis les fichier **Markdown** pour livraison.

Installation de GIT

En soit **GIT** n'est pas obligatoire, mais le fait que celui-ci intègre un shell linux, nous aurons ainsi tout ce qui nous faut pour travailler ! Si vous avez déjà un environnement Linux sur votre PC, passez cette étape.

Téléchargez GIT à l'adresse suivante, puis suivez les instructions d'installation. Les options à sélectionner lors de l'installation ne dépend que de vous.

Je vous conseille à minima d'intégrer au contexte Window l'élément : [Git Bash Here](#) .

GIT : <https://git-scm.com/downloads>

Installation de NodeJS

Pour disposer des outils de génération de PDF, nous avons aussi besoin de **NodeJs**. Si vous avez déjà NodeJS sur votre PC, passez cette étape.

Ici aussi, téléchargez l'installateur à l'adresse suivante puis suivez les instructions.

NodeJS : <https://nodejs.org/en/>

Installation de parkdown-pdf

Markdown-pdf est une application développée sous NodeJS. Pour l'installer, il suffit d'utiliser le gestionnaire de package de NodeJS : `npm`

Tapez la commande suivante pour installer markdown-pdf : `npm install --global markdown-pdf` .

Si tout se déroule correctement (Le nombre de packet installé est affiché), l'application est désormais fonctionnelle.

Installation des ressources

De base **Markdown-pdf** ne fait que rendre le texte dans un **PDF**. Pour l'esthétique, il faut venir ajouter un certain nombre d'élément qui vont servir à Markdown-pdf.

Création d'une fonction pour aliaser markdown-pdf

Pour éviter de taper à chaque fois une commande à rallonge, nous allons utiliser une fonction Bash qui elle même sera capable d'accepter des arguments.

Cela se passe dans le fichier `.bashrc` et éventuellement dans le fichier `.bash_profile` .

Si vous avez déjà de la personnalisation dans ces fichiers, il va falloir venir ajouter le code suivant dans le deux fichiers mentionnés, sinon copiez simplement les deux fichiers présent dans le dossier `src` dans le dossier utilisateur.

- Sous Windows, ce dossier est accessible via la variable `%USERPROFILE%` . Faites `Windows+R` , saisissez la variable puis validez.

- Sous Linux, ce dossier est accessible via la variable `$HOME` ou en faisant `cd ~`.

```
#
# Rend un document Markdown en document PDF
# $1 : Fichier Markdown à rendre
# $2 : Fichier JS à exécuter pour générer l'entête et le pied de page
# $3 : Feuille de style à utiliser : .markdown-pdf-stylesheet-$3.css
# $4 : Feuille de style pour Highlight.js : .markdown-pdf-hljs-$4.css
#
mpdf() {
    # Defaults:
    # Fichier Markdown à rendre
    mdfile=""
    js=""
    hljs="monokaiSublime"
    css=""
    preset=""

    run="--runnings-path $HOME/.markdown-pdf-default.js"
    sheet=".markdown-pdf-stylesheet.css"

    # Traitement des options
    # $1 = Fichier
    # $2 = runnings-path
    # $3 = Stylesheet
    # $4 = Highlight.js Stylesheet
    arg=0
    for i in "$@"; do
        ((++arg))

        case $i in
            --js=*)
                js="${i#*=}"
                ;;
            --theme=*|--css=*)
                css="${i#*=}"
                ;;
            --file=*)
                mdfile="${i#*=}"
                ;;
            --hl=*|--hljs=*)
                hljs="${i#*=}"
                ;;
            --preset=*)
                preset="${i#*=}"
                ;;
            *)
                if [ $arg = 1 ]; then
                    mdfile=$i
                elif [ $arg = 2 ]; then
                    js=$i
                elif [ $arg = 3 ]; then
                    css=$i
                elif [ $arg = 4 ]; then
                    hljs=$i
                fi
                ;;
        esac
    done

    if [ $preset ]; then
        case $preset in
            viseo)
                js="viseo"
                css="viseo"
                hljs="viseo"
                ;;
            uma)
                js="uma"
                css="uma"
                ;;
        esac
    fi
}
```

```

fi

if [ $js ]; then
    if [ $js = 'off' ] || [ $js = 'disable' ]; then
        echo "disable"
        run=''
    else
        run="--runnings-path $HOME/.markdown-pdf-$js.js"

        if [ ! -f ~/.markdown-pdf-$js.js ]; then
            echo "File .markdown-pdf-$js.js not found in $HOME"
            return 1
        fi
    fi
fi

if [ $css ]; then
    sheet=".markdown-pdf-stylesheet-$css.css"
    if [ ! -f ~/.$sheet ]; then
        echo "File $sheet not found in $HOME"
        return 1
    fi
fi

hljsheet=".markdown-pdf-hljs-$hljs.css"

if [ ! -f ~/.$hljsheet ]; then
    echo "File $hljsheet not found in $HOME"
    return 1
fi

markdown-pdf -m '{"breaks": false, "html": true, "linkyfy": true}' --paper-border "0cm" --css-
path ~/.$sheet --highlight-css-path ~/.$hljsheet $run $mdfile

return 0
}

```

Ajout des ressources

La commande `mpdf` précédemment requiert un minimum de fichier pour fonctionner :

- La feuille de style CSS appliquée au contenu du PDF : `.markdown-pdf-stylesheet-<theme>.css`
- La feuille de style CSS appliquée aux codes : `.markdown-pdf-hljs-<theme>.css`
- Le script de génération des entêtes et pieds de page : `.markdown-pdf-<theme>.js`

Les fichiers se trouvent dans le dossier `src` qui devront être déposés dans le dossier utilisateur :

- Sous Windows, ce dossier est accessible via la variable `%USERPROFILE%`. Faites `Windows+R`, saisissez la variable puis validez.
- Sous Linux, ce dossier est accessible via la variable `$HOME` ou en faisant `cd ~`.

Installation de Showdown

Si vous souhaitez générer les documents sous Word ou HTML, vous pouvez installer Showdown.

L'installation se fait également à l'aide de `npm` via la commande suivante `npm install --global showdown`

Ajouter un alias de commande

Si vous avez copié les fichiers `.bashrc` et `.bash_profile`, vous n'avez pas besoin d'effectuer cette étape. La commande `mhtml` est déjà disponible.

Sinon, ajouter le code suivant aux deux fichiers mentionnés :

```

#
# Rend un document Markdown en document HTML
# $1 : Fichier Markdown à rendre
# $2 : Classe à ajouter au body si nécessaire
# $3 : Feuille de style à utiliser : .markdown-pdf-stylesheet-$3.css
#
makehtml() {
    cat ~/.markdown-pdf-hljs-monokaiSublime.css > mhtml.css

    # Utiliser la feuille de style spécifiée
    if [ $3 ]; then
        sheet=".markdown-pdf-stylesheet-$3.css"

        if [ ! -f ~/.$sheet ]; then
            echo "File $sheet not found in $HOME"
            return 1
        fi

        cat ~/.$sheet >> mhtml.css
    else
        cat ~/.markdown-pdf-stylesheet.css >> mhtml.css
    fi

    showdown makehtml -i $1 -o $1.tmp --tables --tasklists

    sed -e "s/%TITLE%/$1/g" ~/.markdown-html-wrapper.html |
    sed -e "s/%CLASS%/$2/g" |
    sed -e '/%STYLE_CSS%/{r mhtml.css' -e 'd}' |
    sed -e '/%MARKDOWN_HTML%/{r $1.tmp' -e 'd}' > $1.html

    rm mhtml.css
    rm $1.tmp

    return 0
}

```

Installation de Highlight.js

Pour que les blocs de codes soient colorisés syntaxiquement, l'enveloppe HTML doit charger la librairie nommé **Highlight.js**. Pour "standardiser" l'installation, j'ai défini des emplacements qui sont toujours disponibles.

Sous Windows, il faut copier le fichier `src/.highlight.pack-min.js` et le déposer dans le dossier suivant : `C:\Windows\System\`

Sous Linux (...).