

# Des exemples

---

## Ligne horizontale

Paragraphe 1 ... sauter une ligne

---

Paragraphe 2 ... Pour aller à la ligne, soit on change de paragraphe en sautant une ligne, soit, moins recommandé on met deux espaces en fin de ligne.

## Liste

Une liste, comme un nouveau paragraphe, doit être précédée d'une ligne vide.

Placer chaque item sur une ligne commençant par -, ou par 1 pour une liste numérotée.

Placer une espace avant chaque item

Markdown est **très** utilisé dans de *nombreuses* situations

- Jupyter
- MkDocs
- Stack Exchange
- ...

## URL et mail

Voici [un lien](#) vers un moteur de recherche.

Ou avec une info bulle en lien : Un autre moteur de recherche est [DuckDuckGo](#).

Ou simplement :

- Mon site web [https://mon\\_joli\\_site.fr](https://mon_joli_site.fr)
- Mon mail <prénom.nom@chez.moi>

## Italique et gras

- Pour l'emphase faible, *un* tiret-bas.
- Pour l'emphase forte, **deux** astérisques.

## Citations

Texte décalé et chaque ligne est précédée d'un chevron >.

### Cookie :

- Anciennement petit gâteau sucré, qu'on acceptait avec plaisir.
- Aujourd'hui : petit fichier informatique drôlement salé, qu'il faut refuser avec véhémence.

De Luc Fayard, *Dictionnaire impertinent des branchés*

## Tableaux

Objectif	Markdown	Rendu
Créer un lien	<code>[texte cliquable](https://mon_lien.fr)</code>	texte cliquable
Emphase faible	<code>Un _mot_ discret</code>	Un <i>mot</i> discret
Emphase forte	<code>Un <b>**mot**</b> visible</code>	Un <b>mot</b> visible
Du code en ligne	<code>Une boucle `for`</code>	Une boucle <code>for</code>

## Code en ligne

La définition de la fonction `premier` commence avec le mot clé `def`

Ou encore si notre code contient des accents graves : **du code** avec accent grave : `un entier `n``

## Code en bloc

Pour qu'un extrait ne soit pas formaté, décaler le bloc de 4 espaces ou plus. On saute également une ligne avant et après, comme pour un paragraphe.

Voici un code Python

```
n = 47**2 + 31**2
```

Pour davantage de code : écrire avant le bloc ````<nom_du_langage>` et après le bloc `````

```
for i in range(10):  
    print("Salut à tous !")
```

```
#include<stdio.h>  
int main(){  
    int i;  
    for(i = 0; i < 10; i++){  
        printf("Salut à tous !\n");  
    }  
    return 0;  
}
```

Avec numérotation de ligne

```
for i in range(10):  
    print("Salut à tous !")
```

```
$ python --version
Python 3.8.5
$ pip --version
pip 21.0.1 from .../lib/python3.8/site-packages/pip (python 3.8)
```

<https://ens-fr.gitlab.io/mkdocs/markdown-mkdocs/#options-sur-les-blocs-de-code>

## Les images

La syntaxe est avec ou sans l'info-bulle

```
! [<texte alternatif>](<url> "<info-bulle>")
! [<texte alternatif>](<url>)
```

Il faut faire attention à bien écrire son url.

Si c'est en local, vérifier le chemin d'accès.  
Si c'est sur le web, bien préfixer avec le protocole http

## Afficher des touches

```
++ctrl+alt+del++ ++"↑ Maj."+"Entrée ←"++
```

## Cases à cocher



Une liste de tâches 📌

- ☐ à faire
- ☒ fini
- ☐ presque
- ☒ fait depuis longtemps

!!! done "Exemple 2" Cocher le meilleur identifiant pour une variable Python.

```
!!! faq "Propositions"
- [ ] `pv`
- [ ] `p_vie`
- [ ] `points_vie`
- [ ] `les_points_de_vie`

??? done "Réponse"
- ❌ `pv`
- ❌ `p_vie`
```

-  `points\_vie`
-  `les\_points\_de\_vie`

## Les boites (admonitions)

!!! info "Pourquoi ?" Elles permettent de délimiter des blocs de contenu avec une touche de couleur, sans créer de nouvelles entrées dans la table des matières.

Elles structurent donc sans alourdir les onglets de navigation.

!!! note "Pour une note" **note** ou **seealso**

!!! tldr "Pour un résumé" **tldr**, **summary** ou **abstract**

!!! info "Pour une information" **info** ou **todo**

!!! tip "Pour une astuce" **tip**, **hint** ou **important**

!!! done "Pour une réussite" **done**, **check** ou **success**

!!! faq "Pour une question" **faq**, **help** ou **question**

!!! warning "Pour une difficulté" **warning**, **caution** ou **attention**

!!! fail "Pour un échec" **fail**, **failure** ou **missing**

!!! danger "Pour un danger" **danger** ou **error**

!!! bug "Pour un bogue" **bug**

!!! example "Pour des exemples" **example**

!!! cite "Pour une citation" **cite**

## Les boutons

Une console Basthon{ .md-button }

## Changer les couleurs du thème

<https://squidfunk.github.io/mkdocs-material/setup/changing-the-colors/>

## Liste de définitions

**Lorem ipsum dolor sit amet** : Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis.

**Cras arcu libero** : Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante.

Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis.  
Nam vulputate tincidunt fringilla.  
Nullam dignissim ultrices urna non auctor.

## Mettre en ligne le site sur github

---

### Méthode 1

- Modifier un fichier source.
- Vérifier le résultat avec mkdocs serve.
- Re-construire le site avec mkdocs build.
- Avec Git, faire un commit, puis un push.

### Méthode 2

- Modifier un fichier source.
- Vérifier le résultat avec mkdocs serve.
- Avec Git, faire un commit, puis un push.
- Utiliser l'outil de déploiement mkdocs gh-deploy --force

Dans tous les cas, sur GitHub, il y a des risques de confusion avec une branche gh-pages qui pourrait être créée. On compte aussi 4 étapes.

ici