

Material per a Mkdocs

Taula de continguts

- 1 1. El tema Material
- 2 2. Personalització
- 3 3. Búsqueda i navegació
- 4 4. Admonitions-Alertes
- 5 5. Formatació de codi
- 6 6. Diagrames amb Mermaid
- 7 7. Creació de PDF

11. El tema Material

1.1 Què és Material?

Material per Mkdocs és un tema basat en el disseny Material Design creat per Google, per a llocs web generats amb **Mkdocs**. Material Design és un conjunt de directrius de disseny que se centra en l'ús de l'espai, les animacions suaus i els colors vius per crear una experiència d'usuari coherent i atractiva.

El tema Material per Mkdocs adapta aquestes directrius per a la creació de documentació tècnica, facilitant la personalització i una gran flexibilitat visual. Les seves característiques inclouen:

- Disseny net i modern.
- Suport per a temes clars i foscos.
- Taules de contingut dinàmiques.
- Integració fàcil amb extensions com **Mermaid**, **Admonitions**, i més.
- Compatibilitat amb diferents dispositius, ja que és completament responsive.

1.2 Com funciona la plantilla?

El tema **Material** funciona mitjançant un conjunt de plantilles i fulls d'estil que s'integren en el sistema de **Mkdocs**. Quan es configura en el fitxer `mkdocs.yml`, **Mkdocs** utilitza aquests fulls d'estil per renderitzar el lloc web amb el disseny Material.

Per a personalitzar el tema, només cal modificar el fitxer `mkdocs.yml` per configurar opcions com els colors, el disseny, la tipografia o les imatges de fons, entre d'altres. Això fa que el procés de creació de llocs web de documentació siga ràpid i flexible.

1.3 Requisits previs i instal·lació

Per treballar amb Material per a Mkdocs partirem de la instal·lació que ja tenim realitzada prèviament en el nostre entorn virtual de Python.

Així doncs, partint d'aquesta instal·lació, en primer lloc activem el nostre entorn virtual que vam anomenar `mkdocsenv` executant:

```
source mkdocsenv/bin/activate
```

O bé per als usuaris de Windows:

```
mkdocsenv\Scripts\activate
```



Entorn virtual

Canvieu el nom `mkdocsenv` pel nom del vostre entorn virtual en cas que el nom del vostre entorn siga un altre.

Una vegada amb l'entorn activat, instal·lem el tema *Material*:

```
pip install mkdocs-material
```

I finalment, configurem el tema al fitxer `mkdocs.yml`. Recordem que aquest fitxer es fa servir per definir les opcions generals del lloc web, inclòs el tema que s'utilitza. Un exemple bàsic de configuració seria:

```
site_name: Nom del projecte
theme:
    name: 'material'
```

1.4 Plugins i extensions: ampliant les possibilitats de MkDocs

El tema **Material per a MkDocs** ofereix moltes funcionalitats per defecte, però la seu autèntica potència apareix quan hi afegim **plugins i extensions Markdown**.

Aquests components permeten ampliar les capacitats bàsiques de MkDocs i afegir-li característiques més avançades, com ara:

- **Generar documents PDF** de tot el lloc o d'una pàgina concreta.
- **Afegir diagrames Mermaid**, per representar fluxos, processos o esquemes amb text.
- **Crear blocs informatius (admonitions)** per remarcar notes, advertències o exemples.
- **Millorar la visualització del codi font de programes informàtics** amb ressaltat de sintaxi i numeració de línies.
- **Afegir búsqueda interna** per facilitar la localització de contingut concret.
- **Mostrar imatges en finestres emergents (lightbox)**.
- **Personalitzar la interfície** amb configuracions de tema, colors i icones addicionals.

Els plugins i extensions s'instal·len mitjançant **pip** dins del nostre entorn virtual i s'activen al fitxer `mkdocs.yml`. La seua configuració sol seguir un esquema semblant a aquest:

```
plugins:
  - search
  - glightbox
  - pdf-export

markdown_extensions:
  - admonition
  - pymdownx.highlight
  - pymdownx.superfences
```

Als següents apartats ens dedicarem a conéixer algunes d'aquestes funcionalitats de manera més detallada.

2. Personalització

Tematització de Material per Mkdocs

Una de les grans potències del tema **Material per Mkdocs** és la seva **capacitat de personalització**. Ens permet modificar diversos aspectes de l'aparença de la nostra documentació: colors, tipografies, imatges de fons, icones, etc. Podem fer-ho mitjançant la configuració al fitxer `mkdocs.yml` i la personalització amb CSS per obtenir un disseny únic que s'adapte als nostres gustos o a la identitat del nostre projecte.

1. Selecció de colors i temes

El tema **Material per Mkdocs** inclou opcions per configurar tant el tema clar com el fosc, així com altres elements de color. La manera més senzilla de modificar els colors és mitjançant el fitxer de configuració `mkdocs.yml`.

Canviar entre tema clar i fosc

Mkdocs Material té un mode clar i un mode fosc integrats. Per defecte, el tema segueix el mode del sistema operatiu de l'usuari (per exemple, si el sistema té configurat el mode fosc, el tema ho adaptarà automàticament). Però també podem forçar un mode específic.

1. Activar el mode clar:

```
theme:  
  name: 'material'  
  palette:  
    scheme: default
```

1. Activar el mode fosc:

```
theme:  
  name: 'material'  
  palette:  
    scheme: slate
```

Aquesta configuració permet triar entre diversos esquemes de colors per ajustar el disseny visual del lloc web.



Alternar entre tema clar i oscur

Oferir una paleta de colors clars i foscos fa que la documentació siga agradable de llegir a diferents hores del dia, de manera que l'usuari pot triar en conseqüència. Afegiu les línies següents a `mkdocs.yml` per mostrar un component de canvi d'oscur a clar:

```
theme:
  palette:
    - scheme: default
      toggle:
        icon: material/brightness-7
        name: Canvia a mode oscur

    - scheme: slate
      toggle:
        icon: material/brightness-4
        name: Canvia a mode clar
```

2. Personalització amb CSS

Per a una personalització més profunda, podem utilitzar **CSS personalitzat**. Mkdocs permet afegir un fitxer CSS extern que es carregarà en cada pàgina generada.

Afegir un fitxer CSS personalitzat

1. **Crea un fitxer CSS** amb els estils personalitzats. Per exemple, crea un fitxer anomenat `styles.css` dins la carpeta `docs/css/`:

```
/* Exemple de CSS personalitzat */
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
}
```

1. **Configura el fitxer al `mkdocs.yml`** per carregar-lo:

```
extra_css:
  - css/styles.css
```

Aquesta configuració carrega el fitxer CSS personalitzat i el aplica a totes les pàgines generades, permetent modificar aspectes com fonts, colors de fons, i altres estils.

3. Canviar tipografia

El tema **Material** permet canviar la tipografia utilitzada a la documentació. Pots utilitzar fonts de Google Fonts o altres fonts personalitzades.

Canviar la tipografia

Per canviar la tipografia del teu lloc, pots afegir les següents configuracions al teu fitxer `mkdocs.yml`:

1. Exemple per utilitzar Google Fonts:

```
theme:
  name: 'material'
  font:
    text: 'Roboto, sans-serif'
    code: 'Courier New, monospace'
```

Això canviarà la font principal de text a **Roboto** i la font del codi a **Courier New**. Pots ajustar les fonts a les teves preferències.

4. Creació o modificació de plantilles per defecte

Mkdocs Material utilitza plantilles per defecte per generar les pàgines del lloc web. Però podem **modificar o crear plantilles personalitzades** si volem canviar el disseny de pàgines específiques, com la pàgina d'índex o altres.

Crear una plantilla personalitzada per la pàgina d'índex

1. **Crea una plantilla personalitzada** a la carpeta `docs/` anomenada `index.html`. Aquest fitxer substituirà el comportament per defecte de la pàgina d'inici.
2. **Modifica el contingut de la plantilla** per adaptar-ho a les teves necessitats, com afegir seccions especials, estil personalitzat, o elements extra.

Exemple d'una plantilla bàsica de `index.html`:

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{{ site_name }}</title>
  </head>
  <body>
    <header>
      <h1>Benvinguts a la documentació de {{ site_name }}</h1>
    </header>
    <main>
      <p>Aquesta és la pàgina d'inici personalitzada per al vostre projecte.</p>
    </main>
  </body>
</html>
```

1. **Configura el fitxer** `mkdocs.yml` per utilitzar aquesta plantilla personalitzada:

```
extra_templates:
  - docs/index.html
```

5. Modificació d'ícones i altres elements visuals

El tema **Material** utilitza ícones per a enllaços, botons i altres elements visuals. Pots canviar les ícones utilitzades al lloc mitjançant el fitxer de configuració `mkdocs.yml`.

Canviar les ícones

1. **Afegir o modificar ícones** (per exemple, utilitzant la biblioteca de **FontAwesome**):

```
theme:  
  name: 'material'  
  icon:  
    repo: 'fontawesome/brands/github'
```

Aquesta configuració canvia la icona associada als enllaços de repositori a **GitHub** (utilitzant FontAwesome).

6. Consells i bones pràctiques per a la tematització

- **Bones pràctiques:** utilitza la personalització sols en cas de necessitat i solament quan no es puga fer a través de la configuració de l'mkdoc.yml.
- **Consistència visual:** Quan personalitzis el tema, mantí una paleta de colors consistent per a tot el lloc. Això ajudarà a crear una aparença coherent i professional.
- **Ús de fonts legibles:** Tria tipografies que siguen fàcils de llegir, especialment per a textos llargs com la documentació.
- **Prova a diferents dispositius:** Assegura't que les personalitzacions funcionin bé en diversos dispositius i mides de pantalla, ja que **Mkdocs Material** és totalment responsive.

3. Búsqueda i navegació

Búsqueda i Navegació

La búsqueda i la navegació són components clau per a la usabilitat de qualsevol lloc web, especialment en documentació tècnica on els usuaris necessiten trobar informació de manera ràpida i eficient. El tema **Material per Mkdocs** inclou un sistema de búsqueda interna i funcionalitats de navegació que poden ser configurades de manera senzilla.

1. búsqueda interna amb el plugin `search`

Una de les funcions més bàsiques però imprescindibles per a la navegació és la **búsqueda interna**. Gràcies al plugin `search` de **Mkdocs**, els usuaris poden buscar contingut dins del lloc web generat, buscant per paraules clau, títols o altres termes definits en el contingut de les pàgines.

Configuració del plugin `search`

El plugin `search` ja ve integrat per defecte a **Mkdocs**, així que no cal instal·lar res addicional. Només hem de configurar-lo al fitxer `mkdocs.yml`.

1. Configuració bàsica del plugin de búsqueda:

```
plugins:  
  - search
```

2. Navegació a través de seccions i sub-seccions

La **navegació** del lloc web és un altre aspecte important. **Mkdocs Material** ofereix diverses funcionalitats per crear una **navegació per seccions i sub-seccions**, ajudant als usuaris a navegar fàcilment pel contingut.

Seccions i sub-seccions

Una manera de millorar la navegació dins del lloc és afegir la funcionalitat de **seccions i sub-seccions** al menú de navegació lateral. Això es fa de manera automàtica si utilitzem l'estructura jeràrquica a l'arbre de fitxers del nostre projecte, però també podem configurar-lo manualment al fitxer `mkdocs.yml`.

Per exemple, la configuració bàsica per activar la visualització de seccions i sub-seccions seria:

```
theme:  
  name: 'material'  
  features:  
    - navigation.sections
```

Aquesta opció afegeix una navegació més detallada amb seccions desplegables i enllaços a sub-seccions.

3. Barra de navegació superior

Mkdocs Material permet afegir una **barra de navegació superior**, la qual és una altra manera d'ajudar a l'usuari a desplaçar-se ràpidament pel lloc.

Activar la barra de navegació superior

Per activar la barra de navegació superior, afegim la següent configuració al fitxer `mkdocs.yml`:

```
theme:  
  name: 'material'  
  features:  
    - navigation.top
```

Aquesta configuració afegeix una barra de navegació fixa a la part superior de la pàgina amb els enllaços als títols i seccions més importants del lloc.

4. Exemple de configuració de navegació

Una configuració completament funcional podria ser la següent:

```
theme:  
  name: 'material'  
  features:  
    - navigation.sections  
    - navigation.top
```

Amb això, tindrem:

- **Navegació lateral** amb seccions i sub-seccions desplegables.
- **Barra de navegació superior** que permet l'accés ràpid a les parts més importants del lloc.

5. Consells i bones pràctiques per a la búsqueda i navegació

- **Organitza bé les seccions:** Mantingues una estructura clara amb una jerarquia lògica de contingut. Això ajudarà tant en la cerca com en la navegació.
- **Facilita la cerca:** Si tens una gran quantitat de contingut, et recomanem fer servir la cerca per ajudar als usuaris a trobar ràpidament el que busquen.
- **No sobrecarregues la barra de navegació:** Mantí la barra de navegació superior senzilla i només inclou els enllaços més rellevants.

4. Admonitions-Alertes

Creació d'Admonitions

Les **admonicions** o **avisos** són blocs d'informació destacada dins de la documentació. Aquestes són útils per cridar l'atenció sobre informació important, advertències, consells, exemples o notes. En **Mkdocs Material**, les admonicions es gestionen mitjançant l'extensió **pymdownx.admonition**, que permet afegir diversos tipus de blocs visualment destacats, que ajuden a millorar l'estructura i la claredat de la documentació.

1. Instal·lació de l'extensió Admonition

Per utilitzar les admonicions, cal instal·lar l'extensió **pymdownx.admonition**, que forma part de **Pymdown Extensions**.

1. Instal·lació de Pymdown Extensions:

Si no tens **Pymdown Extensions** instal·lades, pots fer-ho amb el següent comandament:

```
pip install pymdown-extensions
```

1. Configuració al fitxer `mkdocs.yml`:

Un cop instal·lada l'extensió, l'hem de configurar dins del fitxer de configuració `mkdocs.yml` a la secció `markdown_extensions`.

```
markdown_extensions:  
  - admonition
```

2. Sintaxi i tipus

Amb l'extensió **admonition** activada, podem afegir diversos tipus de blocs d'informació dins dels nostres documents. La sintaxi per crear una admonició és la següent:

```
!!! type "Títol"  
    Contingut de l'admonició.
```

On `type` pot ser un dels següents:

- **note**



note

Aquesta és una **nota** que ofereix informació addicional.

- **abstract**

**abstract**

Aquest és un **resum** del contingut que es presenta a continuació.

- **info**

**info**

Aquesta és una **informació** rellevant sobre el tema que estem tractant.

- **tip**

**tip**

Un **consell** útil per facilitar la tasca.

- **success**

**success**

Aquesta és una **notícia d'èxit** per destacar un resultat positiu.

- **question**

**question**

Què passaria si **fossis** el responsable d'aquesta situació?

- **warning**

**warning**

Atenció: Això podria causar problemes si no es fa amb cura.

- **failure**

**failure**

Aquesta acció ha estat un **fracàs**, no va funcionar com s'esperava.

- **danger**



Perill: Pot ser perillós si no es pren la precaució necessària.

- **bug**



Aquest és un **error** que cal resoldre.

- **example**



A continuació, un **exemple** d'una implementació funcional.

- **quote**



Aquesta és una **cita** d'una font destacada que il·lustra el punt que estem discutint.

Aquesta sintaxi crea blocs que es mostren amb fons colorit i un títol destacat, facilitant la visualització de la informació rellevant.

3. Altres configuracions

1. Eliminem el títol

Podem eliminar el títol d'una admonició simplement passant com a títol una cadena buida.

```
!!! note ""
```

Està anotació apareix sense el títol.

Està anotació apareix sense el títol.

2. Imbriquem les admonicions.

```
!!! note "Nota de fora"
```

Esta és la admonició de més a fora.

```
!!! note "Nota de dins"
```

Esta és la de més a dins.

Nota de fora

Esta és la admonició de més a fora.

Nota de dins

Esta és la de més a dins.

3. Admonicions plegables.

Per a que les admonicions tinguen la possibilitat de ser plegades, canviem la sintaxi de !!! per ????. Si adicionalment volem que apareguen desplegades (per defecte apareixen plegades), hem d'afegir el símbol + després dels interrogants.

Configuració per a Admonicions plegades

Necessitem afegir `pymdownx.details` al `mkdocs.yml` per a que funcione.

Exemple dús d'admonicions plegades

```
```text
???:+ note "Pregunta"
 Quina és l'arrel quadrada de 9?
 ??? question "Veure la resposta"
 3
 ...
```

### Pregunta

Quina és l'arrel quadrada de 9?

### Veure la resposta

3

#### 4. Blocs en línia

Les advertències també es poden representar com a blocs en línia (per exemple, per a barres laterals), col·locant-les a la dreta utilitzant els modificadors `inline + end` o a l'esquerra utilitzant només el modificador `inline`:

##### Exemple dús d'admonicions plegades

```
!!! note inline "Admonition en bloc"

Este admonition es situarà a l'esquerra

!!! note inline end "Admonition en bloc"

Este admonition es situarà a la dreta
```



##### Admonition en bloc

Este admonition es situarà a l'esquerra



##### Admonition en bloc

Este admonition es situarà a la dreta

#### 5. Consells i bones pràctiques

- **Ús moderat:** Tot i que les admonicions són molt útils per destacar informació important, és millor utilitzar-les amb moderació. Un excés de blocs pot fer que el contingut siga difícil de llegir i que deixen de captar l'atenció del lector.
- **Claredat:** Assegura't que el títol i el contingut de l'admonició siguin clars i directes.

# 5. Formatació de codi

## Formatació de Codi

Un dels aspectes més importants en la documentació tècnica és la presentació correcta del **codi font**. El tema **Material per Mkdocs** permet integrar fàcilment **formatació de codi** per a diversos llenguatges de programació, utilitzant extensions com **Pymdown Extensions** i el suport per a resaltar la sintaxi del codi i afegir numeració de línies.

A continuació, veurem com configurar-ho i utilitzar-ho al nostre lloc de documentació.

### 1. Instal·lació de les extensions per formatar el codi

Per habilitar la formatació de codi amb resalta de sintaxi i numeració de línies, necessitem instal·lar l'extensió **Pymdown Extensions**, que inclou la funcionalitat de resaltar sintaxi amb **highlight** i altres eines útils per a blocs de codi.

#### 1. Instal·lació de Pymdown Extensions:

```
pip install pymdown-extensions
```

#### 1. Configuració al fitxer `mkdocs.yml`:

Un cop instal·lada l'extensió, afegim la configuració al fitxer `mkdocs.yml` per activar les extensions de **highlight** i **superfences**. La configuració bàsica seria la següent:

```
markdown_extensions:
 - pymdownx.highlight:
 linenums: true
 - pymdownx.superfences
```

- `linenums: true` activa la numeració de línies per als blocs de codi.
- **superfences** afegeix una sintaxi més flexible i potent per incloure fragments de codi en diferents formats.

### 2. Sintaxi per afegir blocs de codi

Un cop configurat, podem afegir blocs de codi dins dels nostres fitxers Markdown. La sintaxi per crear blocs de codi és la següent:

```
```language
# Aci el codi font
```

```

On `language` és el llenguatge de programació que vols resaltar. Per exemple, per mostrar codi Python, utilitzem:

```
```python
def salutacio():
    print("Hola món!")
```

```

```
def salutacio():
 print("Hola món!")

```

### 3. Configuracions sobre blocs de codi

A més de la sintaxi bàsica per a blocs de codi, també podem personalitzar l'aspecte del codi a través de l'estil CSS o altres configuracions avançades. Algunes opcions addicionals inclouen:

1. **Afegir títols als fragments de codi:** Per tal de proporcionar context addicional, es pot afegir un títol personalitzat a un bloc de codi utilitzant l'opció `title="<custom title>"` després del tipus de llenguatge, per exemple per mostrar el nom d'un fitxer.

#### Exemple de títol

```
``` py title="bubble_sort.py"
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

```

#### bubble\_sort.py

```
def bubble_sort(items):
 for i in range(len(items)):
 for j in range(len(items) - 1 - i):
 if items[j] > items[j + 1]:
 items[j], items[j + 1] = items[j + 1], items[j]

```

2. **Afegir numeració de línies:** Si tens habilitada l'opció `linenums`, es mostrarà la numeració de línies al costat esquerre del codi.

Exemple de configuració al fitxer `mkdocs.yml`:

#### mkdocs.yml

```
markdown_extensions:
 - pymdownx.highlight:
 linenums: true

```

```
```python
def salutacio():
    print("Hola món!")
```

```

```

1 | def salutacio():
2 | print("Hola món!")

```

- 3. Botó de còpia de codi:** Els blocs de codi poden mostrar automàticament un botó a la dreta per permetre a l'usuari copiar el contingut d'un bloc de codi al porta-retalls.

```

theme:
features:
- content.code.copy

```

```

```python
def salutacio():
    print("Hola món!")
```

```

```

def salutacio():
 print("Hola món!")

```

- 4. Anotacions de codi:** Les anotacions de codi ofereixen una manera còmoda i amigable d'adjuntar contingut arbitrari a seccions específiques de blocs de codi afegint marcadors numèrics al bloc i comentaris en línia en l'idioma del bloc de codi.

```

theme:
features:
- content.code.copy

```

#### Anotacions al codi

```

```python
def salutacio():
    print("Hola món!") # (1)!
    pass # (2)!
```

```

1. Sóc una anotació de codi!
2. Puc contindre codi, text formatat, imatges... bàsicament qualsevol cosa que es puga escriure en Markdown.

```

def salutacio():
 print("Hola món!") ①
 pass ②

```

- ① Sóc una anotació de codi!
- ② Puc contindre codi, text formatat, imatges... bàsicament qualsevol cosa que es puga escriure en Markdown.

- 5. Incrustació de fitxers externs:** Quan els fragments de codi estan habilitats, es pot incrustar contingut d'altres fitxers (inclosos els fitxers font) utilitzant la --8--notació directament des d'un bloc de codi:

```
markdown_extensions:
 - pymdownx.snippets
```

#### Code block with external content

```
``` title="docs/src_example/hello_world.py"  
--8<-- "docs/src_example/hello_world.py"  
```
```

[docs/src\\_example/hello\\_world.py](#)

```
def function():
 print("Hello world!")
```

## 6. Consells i bones pràctiques

- **Ús moderat:** Tot i que la formatació de codi és una part crucial de la documentació, intenta evitar posar massa codi dins d'una mateixa pàgina. Si tens fragments molt grans de codi, considera posar-los en fitxers externs i posar els enllaços per a que se'ls descarreguen i els analitzen en local.
- **Claritat en els exemples:** Assegura't que el codi siga clar i fàcil de llegir. Afegeix comentaris o anotacions als fragments per explicar què fa cada part del codi.
- **Tipus de codi:** Si el codi prové d'exemples que no són tan habituals, afegeix el llenguatge correctament a la sintaxi per assegurar que el resultat es fa adequadament.

# 6. Diagrames amb Mermaid

## Creació de Diagrames Mermaid

Una de les funcionalitats de **Mkdocs Material** és la capacitat d'integrar **diagrames Mermaid** en la nostra documentació. **Mermaid** és un llenguatge basat en text que permet crear diagrames i gràfics de manera senzilla. Aquests diagrames poden ser de diversos tipus: fluxos, gràfics de xarxa, de Gantt, de seqüència, i més. El millor és que els podem crear directament en Markdown, utilitzant una sintaxi molt senzilla.

### 1. Instal·lació de l'extensió Mermaid

Per començar a utilitzar **Mermaid** amb **Mkdocs**, cal instal·lar l'extensió **Pymdown Extensions**, que inclou el suport per a **Mermaid**. Això permetrà que els diagrames Mermaid es renderitzen correctament dins del lloc web generat per **Mkdocs**.

#### 1. Instal·lació de Pymdown Extensions:

```
pip install pymdown-extensions
```

#### 1. Configuració al fitxer `mkdocs.yml`:

Una vegada instal·lada l'extensió, hem d'afegir-la al fitxer de configuració `mkdocs.yml` per activar-la. Afegeix la següent configuració a la secció de `markdown_extensions`:

```
- pymdwnx.superfences:
 custom_fences:
 - name: mermaid
 class: mermaid
 format: !!python/name:pymdwnx.superfences.fence_code_format
```

L'extensió **superfences** és la que permet a **Mkdocs** identificar i renderitzar correctament els diagrames Mermaid.

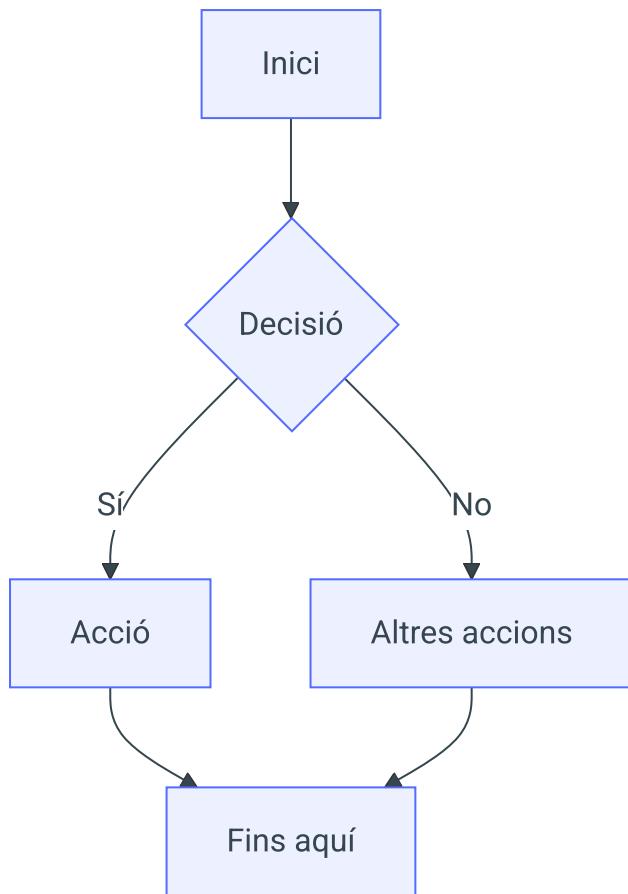
### 2. Sintaxi de Mermaid

Després de configurar, podem començar a utilitzar **Mermaid** a la nostra documentació amb la sintaxi següent:

#### Diagrama de flux

```
```mermaid
graph TD;
A[Inici] --> B{Decisió}
B -- Sí --> C[Acció]
B -- No --> D[Altres accions]
C --> E[Fins aquí]
D --> E[Fins aquí]
```

```



En aquest exemple, estem creant un **diagrama de flux** senzill amb les següents característiques:

- **A** és el punt inicial.
- **B** és una decisió (en forma de diamant).
- **C i D** són accions dependent de la decisió.
- **E** és el final del flux.

### 3. Tipus de diagrames disponibles

Amb **Mermaid**, podem crear diversos tipus de diagrames. Alguns dels més comuns inclouen:

#### 1. Diagrama de flux (Flowchart):

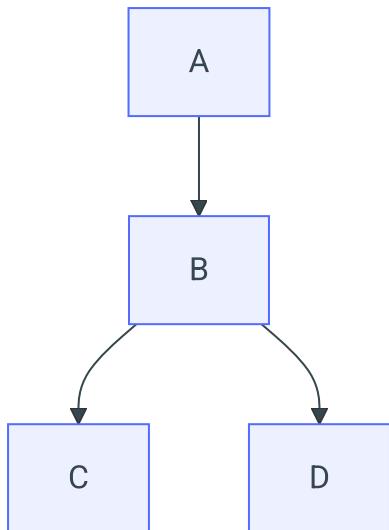
Utilitzant la sintaxi `graph TD` (top-down):

##### Diagrama de flux

```

```mermaid
graph TD;
A --> B --> C;
B --> D;
```

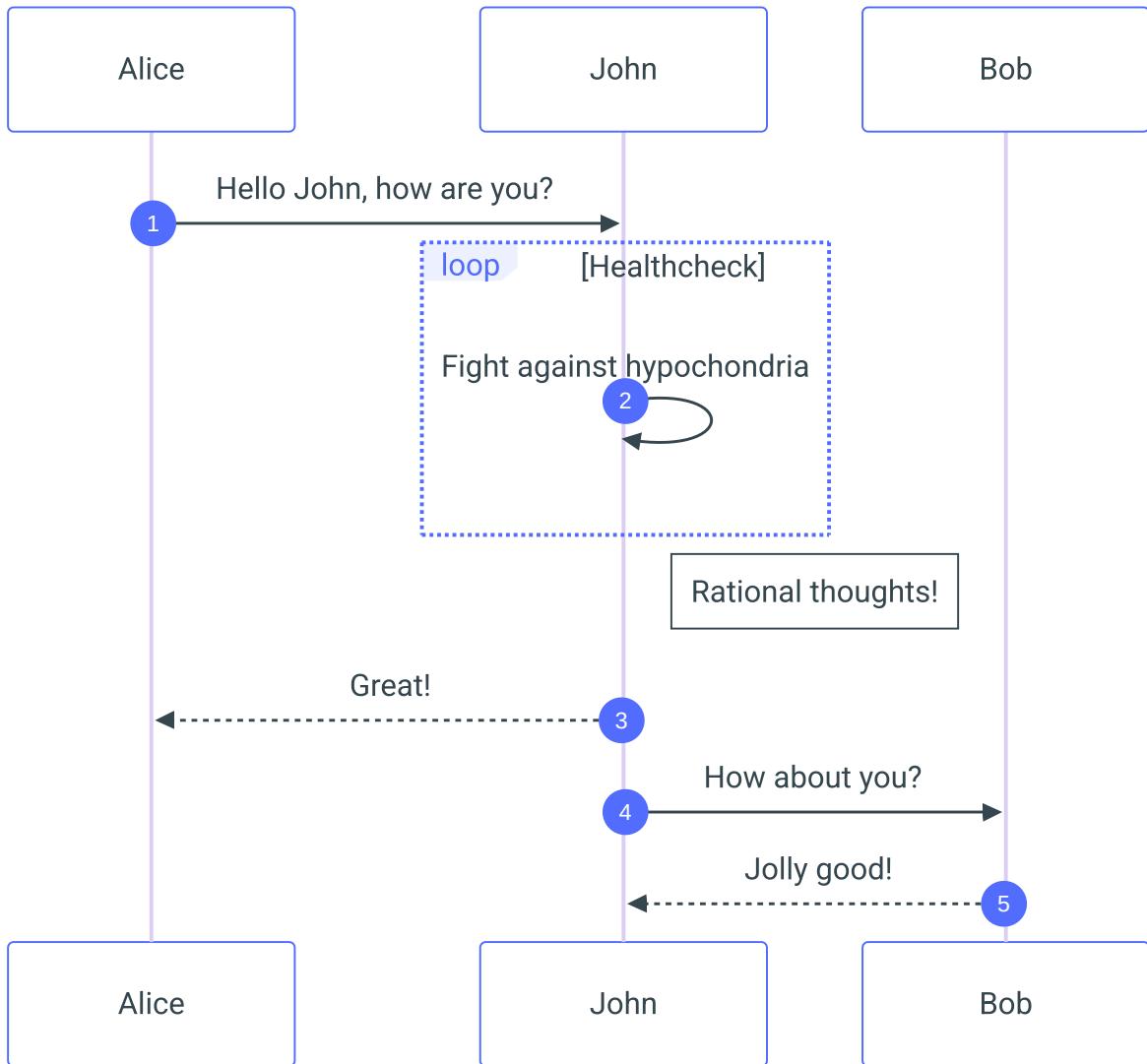
```



2. **Diagrama de seqüència (Sequence Diagram):** Per descriure escenaris amb múltiples actors que interaccions entre ells:

#### Diagrama de seqüència

```
```mermaid
sequenceDiagram
autonumber
Alice->>John: Hello John, how are you?
loop Healthcheck
    John->>John: Fight against hypochondria
end
Note right of John: Rational thoughts!
John-->>Alice: Great!
John->>Bob: How about you?
Bob-->>John: Jolly good!
```
```



### 3. Diagrames d'Estat (State Diagram): Per a descriure el comportament de sistemes finits.

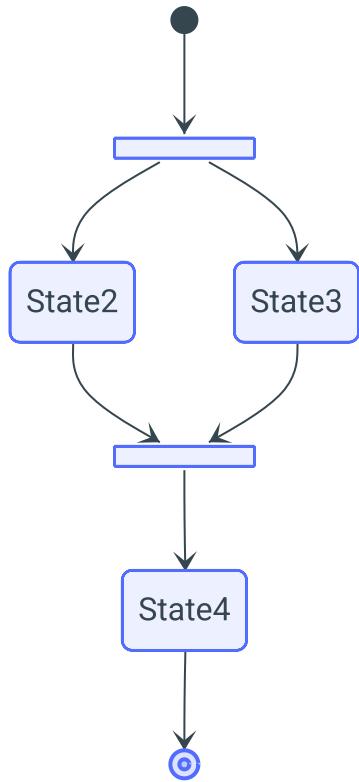
#### Diagrames d'estat

```

```mermaid
stateDiagram-v2
state fork_state <> fork_state
[*] --> fork_state
fork_state --> State2
fork_state --> State3

state join_state <> join_state
State2 --> join_state
State3 --> join_state
join_state --> State4
State4 --> [*]
```
```

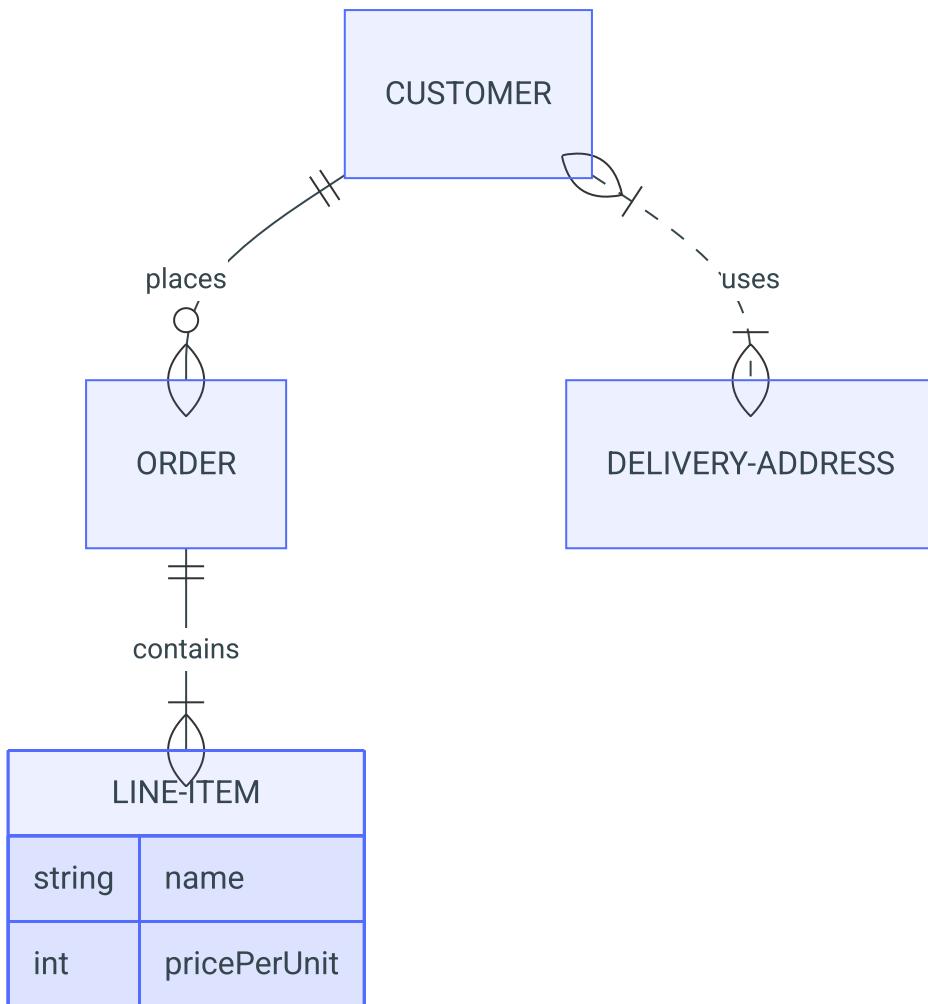
```



4. **Diagrames entitat-relació (E-R Diagram):** Per representar coses interrelacionades en un domini de coneixement específic.

Diagrames entitat-relació

```
```mermaid
erDiagram
 CUSTOMER ||--o{ ORDER : places
 ORDER ||--|{ LINE-ITEM : contains
 LINE-ITEM {
 string name
 int pricePerUnit
 }
 CUSTOMER }|..|{ DELIVERY-ADDRESS : uses
````
```

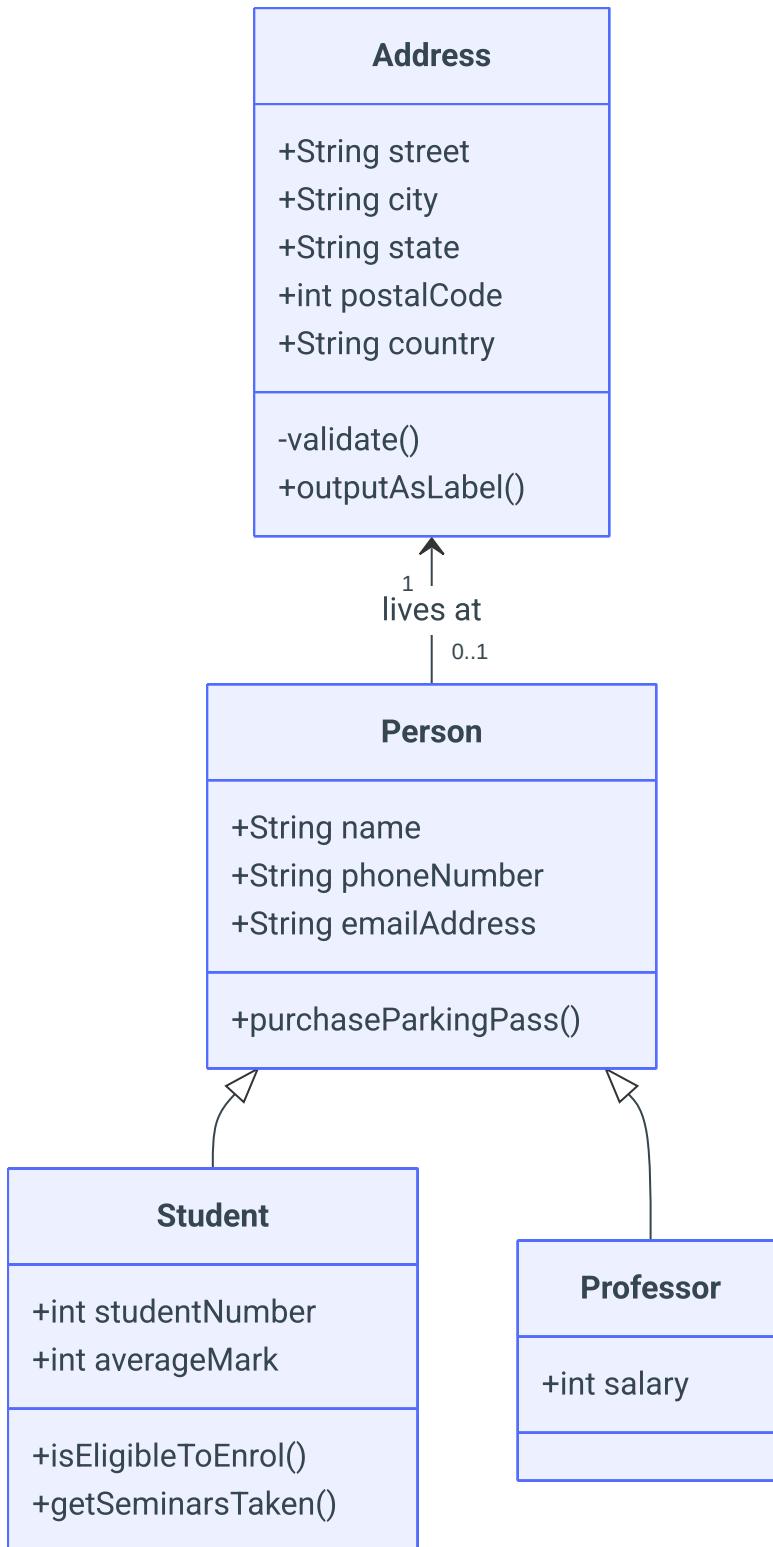


5. **Diagrama de classe (Class Diagram)**: Per mostrar la relació entre classes en un sistema orientat a objectes.

Diagrames de classe

```
``` mermaid
classDiagram
Person <|-- Student
Person <|-- Professor
Person : +String name
Person : +String phoneNumber
Person : +String emailAddress
Person: +purchaseParkingPass()
Address "1" <-- "0..1" Person:lives at
class Student{
 +int studentNumber
 +int averageMark
 +isEligibleToEnrol()
 +getSeminarsTaken()
}
class Professor{
 +int salary
}
class Address{
 +String street
 +String city
 +String state
 +int postalCode
 +String country
 -validate()
 +outputAsLabel()
}
```
```

```



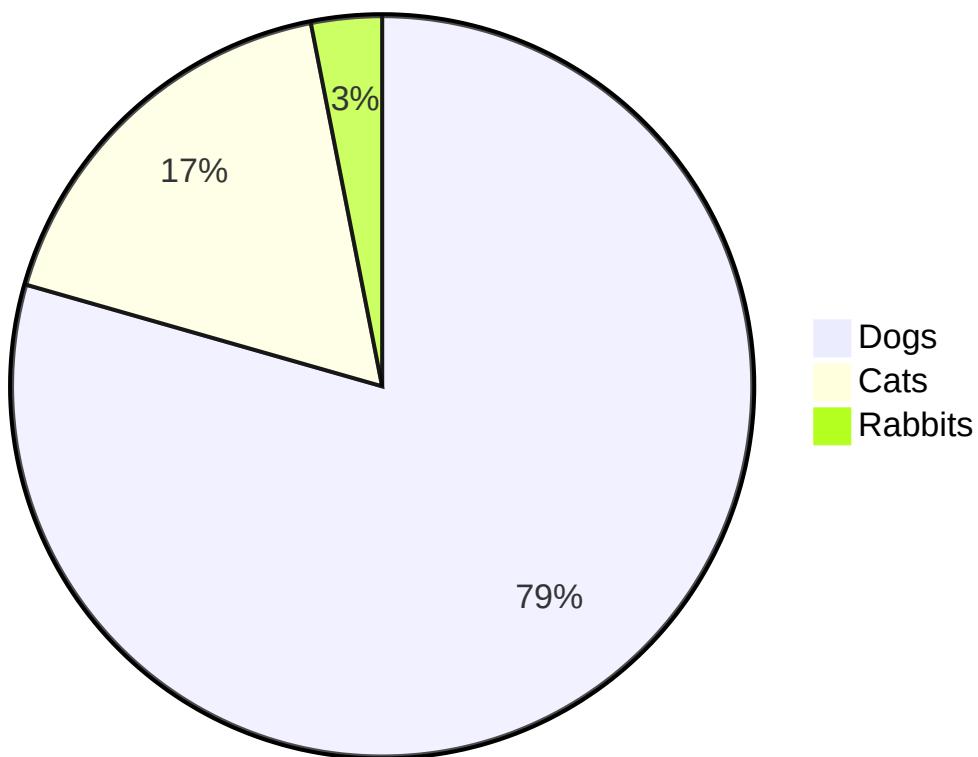
## 6. Diagrames circulars (Pie Chart): Per crear gràfics circulars.

Diagrames circulars

```
```mermaid
pie title Pets adopted by volunteers
    "Dogs" : 386
    "Cats" : 85
    "Rabbits" : 15
...```

```

Pets adopted by volunteers



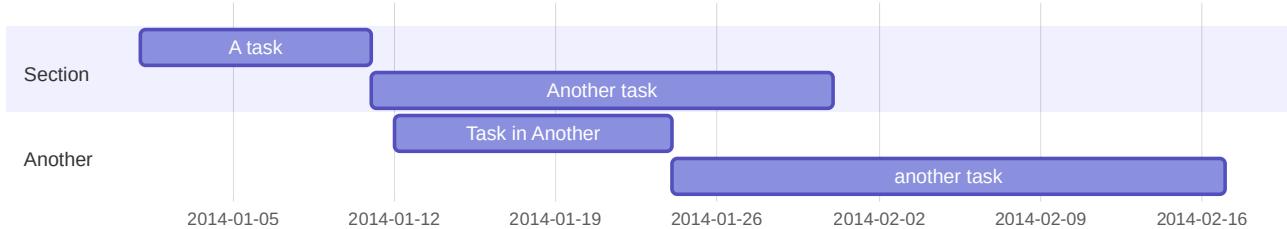
7. Diagrames de Gantt (Gantt Diagrams): Per a mostrar cronologies de projectes.

Diagrames de Gantt

```
```mermaid
gantt
title A Gantt Diagram
dateFormat YYYY-MM-DD
section Section
 A task :a1, 2014-01-01, 10d
 Another task :after a1, 20d
section Another
 Task in Another :2014-01-12, 12d
 another task :24d
...```

```

## A Gantt Diagram



### Altres diagrames

Existeixen altres diagrames disponibles. Pots consultar més informació als següents enllaços:

- [Diagrames MkDocs Material](#)
- [Mermaid](#)

## 4. Consells i bones pràctiques

- **Tipus de diagrames:** És important saber quin tipus de diagrama s'adapta millor a la informació que vols presentar. Els diagrames de flux són ideals per a processos, mentre que els de Gantt són més útils per a cronologies.
- **Claritat:** Intenta mantenir els diagrames simples i nets. Un diagrama excessivament complex pot ser difícil d'entendre.
- **Proves i ajustos:** Els diagrames poden tindre xicotets errors de visualització. Si no es veu bé, prova de simplificar-lo o ajustar la sintaxi.

# 7. Creació de PDF

## Generació de Documents PDF

Un dels avantatges més destacats del tema **Material per Mkdocs** és la capacitat de generar un **document PDF** de tot el lloc web creat amb Mkdocs o de pàgines independents. Això pot ser útil per distribuir la documentació de manera més accessible, o per generar una versió imprimible de la documentació en format PDF.

Per aconseguir-ho, utilitzarem diversos plugins que ens permetran crear un PDF de tot el lloc web o d'una pàgina concreta, mantenint el disseny original i la jerarquia de contingut.

### 1. Llista de plugins

Al següent enllaç podeu trobar informació sobre alguns dels plugins més utilitzats i valorats per la comunitat:

Plugins per a conversió de formats des d'mkdocs (<https://github.com/mkdocs/catalog?tab=readme-ov-file#site-conversion-pdfepubetc>)

Comentarem alguns d'ells que ens semblen interessants.

### 2. pdf-export

El plugin pdf-export exportarà totes les pàgines de markdown del vostre repositori MkDocs com a fitxers PDF mitjançant WeasyPrint.

Per a utilitzar este plugin utilitzarem la següent ordre:

```
pip install mkdocs-pdf-export-plugin
```

Posteriorment activarem el plugin a l nostre arxiu de configuració `mkdocs.yml`.

#### `mkdocs.yml`

```
plugins:
 - search
 - pdf-export
```



#### Error en la generació de pdfs

Alguns usuaris han reportat alguns errors encara no resolts pels desenvolupadors del plugin indicant que sols genera una pàgina o que els apareix una pàgina en blanc inicialment. Si és el vostre cas utilitzeu un altre mètode per genrar els vostres pdfs.

Podeu personalitzar el plugin passant opcions a `mkdocs.yml`:

**mkdocs.yml**

```
plugins:
 - pdf-export:
 combined: true
```

L'opció més interessants és:

- **combined**: si li posem el valor a true ens traure un sol pdf amb tot el nostre projecte, del contrari generarà un pdf per pàgina.

Més informació sobre opcions a aquest [link](https://github.com/zhaoterryy/mkdocs-pdf-export-plugin) (<https://github.com/zhaoterryy/mkdocs-pdf-export-plugin>)

### 3. mkdocs-page-pdf

Este plugin és molt semblant a l'anterior però afegí un botó de descàrrega del pdf per cada pàgina de la nostra web.

Per a instal·lar-lo utilitzem l'ordre:

```
pip install mkdocs-page-pdf
```

Ara simplement afegim la configuració per a utilitzar-lo al `mkdocs.yml`:

**mkdocs.yml**

```
plugins:
- ...
- page-to-pdf
```

En cas de voler posar opcions el fitxer anterior quedaria així:

**mkdocs.yml**

```
plugins:
- ...
- page-to-pdf :
 opcio: valor
```

Algunes opcions:

- **disable** (bool): Desabilita el renderitzat de pdf, útil per desactivar-lo ràpidament sense eliminar la configuració del plugin. Per defecte és *False*.
- **disableOnServe** (bool): Desabilita el renderitzat de pdf quan s'utilitza mkdocs serve. Per defecte és *False*.

Per a més opcions consulteu la [documentació oficial](https://github.com/brospars/mkdocs-page-pdf) (<https://github.com/brospars/mkdocs-page-pdf>) .

### 4. mkdocs-with-pdf

Aquest plugin generarà un únic fitxer PDF des del teu repositori MkDocs.

## CARACTERÍSTIQUES

- Portada i Taula de Continguts integrades al PDF.
- Números automàtics a les capçaleres (h1-h3).
- Desplaça cap avall el nivell de les subcapçaleres.

## Exemples de pdfs generats amb este plugin

- PDF de la documentació de 'MkDocs' (<https://github.com/orzih/mkdocs-with-pdf/blob/master/samples/mkdocs/document.pdf>)
- PDF de la documentació de 'Material for MkDocs' (<https://github.com/orzih/mkdocs-with-pdf/blob/master/samples/mkdocs-material/document.pdf>)

## Instal·lació

1. Instal·la el paquet amb pip:

```
pip install mkdocs-with-pdf
```

2. Activa el plugin al teu `mkdocs.yml`:

```
plugins:
 - with-pdf
```

## Configuració

Pots personalitzar el plugin passant opcions al teu `mkdocs.yml`:

```

plugins:
 - with-pdf:
 #author: QUI
 #copyright: ANY TEXT
 #
 #cover: false
 #back_cover: true
 #cover_title: TITLE TEXT
 #cover_subtitle: SUBTITLE TEXT
 #custom_template_path: TEMPLATES PATH
 #
 #toc_title: TOC TITLE TEXT
 #heading_shift: false
 #toc_level: 3
 #ordered_chapter_level: 2
 #excludes_children:
 # - 'release-notes/:upgrading'
 # - 'release-notes/:changelog'
 #
 #exclude_pages:
 # - 'bugs/'
 # - 'appendix/contribute/'
 #convert_iframe:
 # - src: IFRAME SRC
 # img: POSTER IMAGE URL
 # text: ALTERNATE TEXT
 # - src: ...
 #two_columns_level: 3
 #
 #render_js: true
 #headless_chrome_path: headless-chromium
 #
 #output_path: any-place/document.pdf
 #enabled_if_env: ENABLE_PDF_EXPORT
 #
 #debug_html: true
 #show_anchors: true
 #verbose: true

```

Sols comentarem algunes de les opcions:

- **author:** autor del pdf
- **copyright:** text de drets de còpia
- **cover:** true si vols portada i false si no en vols.
- **back\_cover:** true si vols contraportada i false si no en vols.
- **cover\_title:** Estableix el títol a la pàgina de portada. Per defecte utilitza el valor del site\_name del teu mkdocs.yml del projecte.
- **cover\_subtitle:** Estableix el subtítol a la pàgina de portada. Per defecte: None
- **cover\_logo:** Estableix la imatge del logotip a la pàgina de portada. Aquest valor pot ser una URL o simplement especificar la ruta relativa al directori de docs.
- **toc\_title:** Estableix el títol de la Taula de Continguts. Per defecte Table of Content.

- **toc\_level**: Estableix el nivell de la Taula de Continguts. Aquest valor pot ser entre 1 i 3. Per defecte: 3.
- **output\_path**: Aquesta opció et permet utilitzar una destinació diferent per al fitxer PDF. Per defecte: pdf/document.pdf.

## 5. mkdocs-print-site-plugin

L'últim dels plugins que veurem és mkdocs-print-site-plugin, que ens permet generar un arxiu html de tot el lloc i utilitzar l'opció d'imprimir a pdf tot el lloc.

La instal·lació la fem executant el següent comandament:

```
pip3 install mkdocs-print-site-plugin
```

Configuurem l'mkdocs.yml per a que faça ús del plugin:

```
plugins:
 - search
 - print-site
```

En fer la contrucció del nostre lloc amb `mkdocs build` observareu a l'explorador d'arxius que s'ha creat un arxiu `/print_page/index.html`. Este arxiu conté tot el lloc web en una sola pàgina. Exporta a pdf amb l'opció d'imprimir a pdf i obtindràs un arxiu pdf tal qual es renderitza al navegador.

### Opcions

Pots personalitzar mkdocs-print-site-plugin modificant la configuració de l' `mkdocs.yml`:

```
plugins:
 - print-site:
 add_to_navigation: false
 print_page_title: 'Print Site'
 print_page_basename: 'print_page'
 add_print_site_banner: false
 # Table of contents
 add_table_of_contents: true
 toc_title: 'Table of Contents'
 toc_depth: 6
 # Content-related
 add_full_urls: false
 enumerate_headings: true
 enumerate_figures: true
 add_cover_page: true
 cover_page_template: ""
 path_to_pdf: ""
 include_css: true
 enabled: true
```

- **add\_to\_navigation** El valor per defecte és false. Afegeix un enllaç «Imprimeix el lloc» a la navegació del lloc.
- **print\_page\_title** El valor per defecte és 'Print Site'. Quan `add_to_navigation` s'estableix en true aquesta opció, controla el nom de la pàgina d'impressió a la navegació del lloc. Aquesta opció s'ignora quan `add_to_navigation`

s'estableix en false.

- **add\_table\_of\_contents** El valor per defecte és true. Afegeix una secció de taula de continguts al principi de la pàgina per imprimir (a la versió per imprimir, la versió HTML té una taula de continguts diferent a la barra lateral).
- **toc\_title** El valor per defecte és 'Table of Contents'. Quan add\_table\_of\_contents s'estableix en truea questa opció, controla el nom de la taula de continguts de la versió impresa de la pàgina d'impressió. Aquesta opció s'ignora quan add\_table\_of\_contentss'estableix en false.
- **toc\_depth** El valor per defecte és 3. Quan add\_table\_of\_contents s'estableix en true aquesta opció, es controla la profunditat de la taula de continguts a la versió impresa de la pàgina impresa. Aquesta opció s'ignora quan add\_table\_of\_contentss'estableix en false.
- **add\_full\_urls** El valor per defecte és false. Quan s'imprimeix una pàgina, no es pot veure la destinació d'un enllaç. Aquesta opció afegeix l'URL de destinació entre parèntesis darrere d'un enllaç.
- **enumerate\_headings** Per defecte true. Això afegirà numeració (enumeració) a tots els encapçalaments i seccions, així com a la taula de continguts. Tingueu en compte que això només enumerarà la pàgina del lloc web per imprimir; si voleu enumerar tot el lloc web, podeu utilitzar mkdocs-enumerate-headings-plugin .
- **enumerate\_headings\_depth** Per defecte 6. Si enumerate\_headings, la profunditat fins a la qual s'ennumeren els encapçalaments i les seccions.
- **enumerate\_figures** Per defecte true. Això afegirà numeració a tots els peus de foto de les figures (per exemple, "Figura 1:"). Funciona especialment bé amb mkdocs-img2fig-plugin .
- **add\_cover\_page** Per defecte false. Quan està activat, s'afegeix una portada a la pàgina d'impressió, que mostra la informació site\_title i altra informació del fitxer mkdocs.yml. Vegeu també Personalització de la portada

Per a més informació consulta la [documentació del projecte](https://timvink.github.io/mkdocs-print-site-plugin/options.html) (<https://timvink.github.io/mkdocs-print-site-plugin/options.html>) .