

Bilingual 2-Column PDF Generator

Converts a Markdown document into a side-by-side bilingual PDF with the source language in the left column and its translation in the right column. Supports any language pair available through Google Translate. Defaults to French→Spanish. Corresponding paragraphs are vertically aligned.

Quick start

```
# French → Spanish (default)
bilingual_pdf my_doc.md

# Spanish → French
bilingual_pdf mi_doc.md \
    --source es --target fr

# English → German
bilingual_pdf my_doc.md \
    --source en --target de
```

Installation

1. download the zip file appropriate for your platform from the [Releases](#) page
2. unzip and move the `bilingual_pdf` executable into a directory which is on your system PATH
3. if needed, handle the security settings for the executable `bilingual_pdf`

On a Mac computer, you can allow running the app by removing the quarantine attribute:

```
xattr -d com.apple.quarantine \
/path/to/bilingual_pdf
```

Usage

Generador de PDF bilingüe de 2 columnas

Convierte un documento Markdown en un PDF bilingüe uno al lado del otro con el idioma de origen en la columna de la izquierda y su traducción en la columna de la derecha. Admite cualquier par de idiomas disponible a través de Google Translate. El valor predeterminado es francés → español. Los párrafos correspondientes están alineados verticalmente.

Inicio rápido

```
# French → Spanish (default)
bilingual_pdf my_doc.md

# Spanish → French
bilingual_pdf mi_doc.md \
    --source es --target fr

# English → German
bilingual_pdf my_doc.md \
    --source en --target de
```

Instalación

1. descargue el archivo zip apropiado para su plataforma desde la página [Versiones](#)
2. descomprima y mueva el ejecutable `bilingual_pdf` a un directorio que esté en la RUTA de su sistema
3. si es necesario, maneje la configuración de seguridad para el ejecutable `bilingüe_pdf`

En una computadora Mac, puedes permitir la ejecución de la aplicación eliminando el atributo de cuarentena:

```
xattr -d com.apple.quarantine \
/path/to/bilingual_pdf
```

Uso

```

# French → Spanish (default)
bilingual_pdf document.md

# Any language pair
bilingual_pdf document.md \
    --source en --target de

# Use a pre-translated markdown file
bilingual_pdf document.md \
    --translation document_es.md

# Specify output filename
bilingual_pdf document.md \
    -o bilingual.pdf

# Choose font size:
# small, medium (default), or large
bilingual_pdf document.md \
    --font-size small

# Also save the intermediate HTML
# (useful for debugging)
bilingual_pdf document.md --html

# Get full help
bilingual_pdf --help

# Also save the translation markdown
bilingual_pdf document.md \
    --save-translation

# Append attribution line to output
bilingual_pdf document.md -a

# List of supported language codes
# (for --source and --target)
bilingual_pdf --list-languages

```

Default output filename: <stem>.<source>.<target>.pdf (or .html with --html). If the input already ends with .<source>.md, the source suffix is not repeated (e.g. doc.fr.md → doc.fr.es.pdf, not doc.fr.fr.es.pdf).

Input format

The input Markdown should contain simple text, optionally formatted with headings, paragraphs, lists, code blocks, blockquotes, horizontal rules and web links. For example:

```

# French → Spanish (default)
bilingual_pdf document.md

# Any language pair
bilingual_pdf document.md \
    --source en --target de

# Use a pre-translated markdown file
bilingual_pdf document.md \
    --translation document_es.md

# Specify output filename
bilingual_pdf document.md \
    -o bilingual.pdf

# Choose font size:
# small, medium (default), or large
bilingual_pdf document.md \
    --font-size small

# Also save the intermediate HTML
# (useful for debugging)
bilingual_pdf document.md --html

# Get full help
bilingual_pdf --help

# Also save the translation markdown
bilingual_pdf document.md \
    --save-translation

# Append attribution line to output
bilingual_pdf document.md -a

# List of supported language codes
# (for --source and --target)
bilingual_pdf --list-languages

```

Nombre de archivo de salida predeterminado: <fuente>.<fuente>.<destino>.pdf (o .html con --html). Si la entrada ya termina en .<fuente>.md, el sufijo fuente no se repite (por ejemplo, doc.fr.md → doc.fr.es.pdf, no doc.fr.fr.es.pdf).

Formato de entrada

El Markdown de entrada debe contener texto simple, opcionalmente formateado con encabezados, párrafos, listas, bloques de código, citas en bloque, reglas horizontales y enlaces web. Por ejemplo:

```
# Main Title
```

```
## Section
```

A paragraph of text. Multiple lines in the source are joined into a single paragraph.

Another paragraph, separated by a blank line

A web link:
[OpenAI](<https://www.openai.com>)

The app does not support more complex Markdown features, notably tables and images.

Using a pre-translated file

If you prefer hand-edited translations over machine translation, provide a pre-translated Markdown file with the **same structure** (same number and order of headings and paragraphs) as the source:

```
bilingual_pdf source_fr.md \  
--translation source_es.md
```

The app warns if the block counts don't match and pads the shorter side with empty cells.

For developers only

How it works

1. **Parse** the input Markdown into structural blocks (headings and paragraphs)
2. **Translate** each block to the target language (automatically via Google Translate, or using a pre-translated file you supply)
3. **Render** a 2-column HTML table where each row pairs a source block with its translated counterpart
4. **Convert** the HTML to an A4 PDF

Build, test and deploy

Use the go build, test and install commands or use the Makefile targets.

```
# Main Title
```

```
## Section
```

A paragraph of text. Multiple lines in the source are joined into a single paragraph.

Another paragraph, separated by a blank line

A web link:
[OpenAI](<https://www.openai.com>)

La aplicación no admite funciones de Markdown más complejas, en particular tablas e imágenes.

Usando un archivo pretraducido

Si prefiere las traducciones editadas a mano en lugar de la traducción automática, proporcione un archivo Markdown pretraducido con la **misma estructura** (mismo número y orden de títulos y párrafos) que la fuente:

```
bilingual_pdf source_fr.md \  
--translation source_es.md
```

La aplicación advierte si el recuento de bloques no coincide y rellena el lado más corto con celdas vacías.

Sólo para desarrolladores

Como funciona

1. **Analizar** el Markdown de entrada en bloques estructurales (títulos y párrafos)
2. **Traduce** cada bloque al idioma de destino (automáticamente a través del Traductor de Google o utilizando un archivo pretraducido que usted proporcione)
3. **Renderizar** una tabla HTML de 2 columnas donde cada fila empareja un bloque fuente con su contraparte traducida.
4. **Convierta** el HTML a un PDF A4

Construya, pruebe e implemente

Utilice los comandos go build, test and install o utilice los objetivos Makefile .

Use the resulting `bilingual_pdf` CLI tool to run the app on the sample Markdown files in `testdata/`. The generated PDFs and intermediate HTML files are saved in the same directory. You can inspect these to understand how the app works and to debug any issues.

Run `scripts/smoketest.sh` to verify that the app runs successfully with valid arguments and that it fails with invalid arguments.

```
# quick tests without network access  
./scripts/smoketest.sh  
  
# include tests using the translation API  
./scripts/smoketest.sh --full  
  
# keep generated files for inspection  
./scripts/smoketest.sh --full --keep  
  
# remove generated files  
./scripts/smoketest.sh --clean
```

Utilice la herramienta CLI `bilingual_pdf` resultante para ejecutar la aplicación en los archivos Markdown de muestra en `testdata/`. Los archivos PDF generados y los archivos HTML intermedios se guardan en el mismo directorio. Puede inspeccionarlos para comprender cómo funciona la aplicación y solucionar cualquier problema.

Ejecute `scripts/smoketest.sh` para verificar que la aplicación se ejecute correctamente con argumentos válidos y que falle con argumentos no válidos.

```
# quick tests without network access  
./scripts/smoketest.sh  
  
# include tests using the translation API  
./scripts/smoketest.sh --full  
  
# keep generated files for inspection  
./scripts/smoketest.sh --full --keep  
  
# remove generated files  
./scripts/smoketest.sh --clean
```

This document was created by [bilingual_pdf](#)