

# Support des langues : polices et synthèse vocale

Mon blog prend désormais en charge neuf langues : le japonais (ja), l'espagnol (es), l'hindi (hi), le chinois (zh), l'anglais (en), le français (fr), l'allemand (de), l'arabe (ar), et le chinois traditionnel (hant). Vous pouvez visiter le site à l'adresse <https://lzwjava.github.io>.

Lorsqu'on traite plusieurs langues dans un environnement informatique, plusieurs aspects doivent être pris en compte.

## Gestion des polices

Différentes langues nécessitent des polices spécifiques pour un rendu correct, en particulier lors de la génération de PDFs avec LaTeX. Le code Python suivant montre comment sélectionner les polices appropriées en fonction du système d'exploitation et de la langue :

```
if platform.system() == "Darwin":
    if lang == "hi":
        CJK_FONT = "Kohinoor Devanagari"
    elif lang == "ar":
        CJK_FONT = "Geeza Pro"
    elif lang in ["en", "fr", "de", "es"]:
        CJK_FONT = "Helvetica"
    elif lang == "zh":
        CJK_FONT = "PingFang SC"
    elif lang == "hant":
        CJK_FONT = "PingFang TC"
    elif lang == "ja":
        CJK_FONT = "Hiragino Sans"
    else:
        CJK_FONT = "Arial Unicode MS"
else:
    if lang == "hi":
        CJK_FONT = "Noto Sans Devanagari"
    elif lang == "ar":
        CJK_FONT = "Noto Naskh Arabic"
    elif lang in ["en", "fr", "de", "es"]:
        CJK_FONT = "DejaVu Sans"
    elif lang == "zh":
        CJK_FONT = "Noto Sans CJK SC"
    elif lang == "hant":
        CJK_FONT = "Noto Sans CJK TC"
```

```

elif lang == "ja":
    CJK_FONT = "Noto Sans CJK JP"
else:
    CJK_FONT = "Noto Sans"
command = [
    'pandoc',
    input_markdown_path,
    '-o', output_pdf_path,
    '-f', 'markdown',
    '--pdf-engine', 'xelatex',
    '-V', f'romanfont={CJK_FONT}',
    '-V', f'mainfont={CJK_FONT}',
    '-V', f'CJKmainfont={CJK_FONT}',
    '-V', f'CJKsansfont={CJK_FONT}',
    '-V', f'CJKmonofont={CJK_FONT}',
    '-V', f'geometry:{GEOMETRY}',
    '-V', 'classoption=16pt',
    '-V', 'CJKoptions=Scale=1.1',
    '-V', 'linestretch=1.5'
]

```

Il est important de noter que cette solution n'est pas parfaite. Par exemple, le texte en hindi dans les commentaires des blocs de code pourrait ne pas s'afficher correctement.

## Synthèse vocale

J'utilise Google Text-to-Speech pour générer des versions audio de mes articles de blog. L'extrait de code suivant montre comment je sélectionne le code de langue approprié pour le moteur de synthèse vocale :

```

if filename.endswith('-zh.md'):
    language_code = "cmn-CN"
    voice_language_code = "cmn-CN"
else:
    language_code = "en-US"
    voice_language_code = "en-US"

text_to_speech(
    text=article_text,
    output_filename=output_filename,
    task=task,

```

```
language_code=language_code,  
dry_run=dry_run,  
progress=progress  
)
```

Actuellement, l'audio est généré pour le contenu en chinois et en anglais. Pour étendre le support à d'autres langues, les codes de langue correspondants doivent être configurés.