

# Guide d'Utilisation de l'Outil

## `generate_images.py`

Ce guide vous offre une synthèse complète de l'utilisation de l'outil `generate_images.py`, en détaillant ses fonctionnalités, les arguments disponibles, et des exemples pour vous aider à générer les images dont vous avez besoin.

---

## Description Générale

Le script `generate_images.py` est un outil Python qui génère des images PNG contenant des combinaisons personnalisées de lettres et de nombres. Chaque image affiche une combinaison spécifique, comme "A1", "B2", "-C-3", etc. Le texte est centré dans l'image, avec une police fixe (**Cantarell Bold**) et un contour dont l'épaisseur peut être ajustée.

En plus de générer un fichier JSON (`output.json`) contenant les données des images encodées en base64, le script offre désormais la possibilité de créer un dossier contenant les images PNG individuelles.

---

## Prérequis

- **Python 3.x** doit être installé sur votre système.

Le module **Pillow** (PIL) est requis. Vous pouvez l'installer en exécutant :

```
pip install Pillow
```

- - La police **Cantarell Bold** doit être installée sur votre système. Si elle n'est pas déjà installée, vous pouvez la télécharger et l'installer depuis [Google Fonts](#).
- 

## Utilisation de l'Outil

Le script s'utilise en ligne de commande et accepte plusieurs arguments permettant de personnaliser les images générées.

## Syntaxe Générale

python generate\_images.py [options]

## Arguments Disponibles

### 1. `--start_letter`

- **Description** : Spécifie la lettre de début de la plage de lettres pour les combinaisons.
- **Valeur par défaut** : 'A'
- **Type** : Chaîne de caractères (`str`)
- **Notes** :
  - Peut être une lettre négative en commençant par - (par exemple, -B).
  - Accepte des lettres au-delà de "Z" (par exemple, AA, AB).

### 2. `--end_letter`

- **Description** : Spécifie la lettre de fin de la plage de lettres.
- **Valeur par défaut** : 'Z'
- **Type** : Chaîne de caractères (`str`)
- **Notes** :
  - Peut être une lettre négative.
  - Doit être cohérente avec `--start_letter`.

### 3. `--start_number`

- **Description** : Spécifie le nombre de début de la plage de nombres.
- **Valeur par défaut** : 1
- **Type** : Entier (`int`)
- **Notes** :
  - Peut être un nombre négatif.
  - Le nombre zéro (0) est exclu.

### 4. `--end_number`

- **Description** : Spécifie le nombre de fin de la plage de nombres.
- **Valeur par défaut** : 26
- **Type** : Entier (`int`)
- **Notes** :
  - Peut être un nombre négatif.
  - Doit être cohérente avec `--start_number`.

### 5. `--contour_thickness`

- **Description** : Définit l'épaisseur du contour autour du texte (en pixels).

- **Valeur par défaut :** 1
- **Type :** Entier (`int`)
- **Exemples :**
  - `--contour_thickness=2` pour un contour plus épais.

## 6. `--export_png`

- **Description :** Lorsque cette option est spécifiée, le script génère un dossier contenant les images PNG individuelles.
- **Type :** Option booléenne (flag)
- **Exemple :**
  - `--export_png` pour activer la génération des images PNG.

## Remarques Importantes

- **Lettres et Nombres Négatifs :** Vous pouvez utiliser des lettres et des nombres négatifs en ajoutant un signe `-` devant (par exemple, `-A`, `-10`).
- **Lettres au-delà de "Z" :** Le script gère les lettres au-delà de "Z" en continuant la séquence avec "AA", "AB", "AC", etc.
- **Exclusion du Zéro :** Le nombre zéro (`0`) n'est pas utilisé dans les combinaisons.

**Utilisation du Signe Égal = :** Lors de la spécification d'arguments avec des valeurs négatives, il est impératif d'utiliser le signe égal `=` pour éviter les erreurs de parsing. Par exemple :

```
--start_letter=-B
```

- 
- **Police Fixe :** La police utilisée est **Cantarell Bold** et ne peut pas être modifiée via les arguments.

## Exemples d'Utilisation

### 1. Utilisation Par Défaut

Génère des images pour toutes les combinaisons de lettres de "A" à "Z" et de nombres de 1 à 26 :

```
python generate_images.py
```

## 2. Spécifier une Plage de Lettres Étendue

Génère des images avec des lettres allant de "A" à "AD" :

```
python generate_images.py --end_letter=AD
```

## 3. Inclure des Nombres au-delà de 26

Génère des images avec des nombres allant de 1 à 30 :

```
python generate_images.py --end_number=30
```

## 4. Spécifier des Lettres Négatives

Génère des images avec des lettres de "-Y" à "-O" :

```
python generate_images.py --start_letter=-Y --end_letter=-O
```

## 5. Combinaison de Lettres et Nombres Négatifs

Génère des images avec des lettres de "-B" à "AD" et des nombres de -5 à 30 :

```
python generate_images.py --start_letter=-B --end_letter=AD --start_number=-5  
--end_number=30
```

## 6. Spécifier l'Épaisseur du Contour

Génère des images avec un contour de 2 pixels d'épaisseur :

```
python generate_images.py --contour_thickness=2
```

## 7. Générer les Images PNG Individuelles

Génère les images et crée un dossier `images` contenant les fichiers PNG :

```
python generate_images.py --export_png
```

## 8. Exemple Complet

Génère des images avec :

- Lettres de "-B" à "AD"
- Nombres de -5 à 30
- Épaisseur du contour de 2 pixels
- Génération des images PNG individuelles

```
python generate_images.py --start_letter=-B --end_letter=AD --start_number=-5  
--end_number=30 --contour_thickness=2 --export_png
```

---

## Notes Supplémentaires

- **Emplacement du Fichier `output.json`** : Le fichier est enregistré dans le même répertoire que le script.
  - **Dossier des Images PNG** : Les images PNG sont enregistrées dans un dossier nommé `images`, créé dans le même répertoire que le script. Vous pouvez modifier le nom du dossier en ajustant le script si nécessaire.
  - **Noms des Fichiers PNG** : Les images sont nommées selon la combinaison qu'elles représentent (par exemple, `A1.png`, `-B-5.png`).
  - **Gestion des Erreurs** : Si une erreur survient (lettres invalides, police manquante), le script affiche un message et s'arrête.
  - **Attention aux Grandes Plages** : Spécifier de larges plages de lettres ou de nombres peut entraîner la génération d'un grand nombre d'images, consommant du temps et de l'espace disque.
- 

## Résumé des Arguments

Argument	Type	Par Défaut	Description
<code>--start_letter</code>	<code>str</code>	<code>'A'</code>	Lettre de début de la plage (peut être négative, ex: <code>-B</code> )
<code>--end_letter</code>	<code>str</code>	<code>'Z'</code>	Lettre de fin de la plage (peut être négative, ex: <code>-Y</code> )
<code>--start_number</code>	<code>int</code>	<code>1</code>	Nombre de début de la plage (peut être négatif, ex: <code>-5</code> )
<code>--end_number</code>	<code>int</code>	<code>26</code>	Nombre de fin de la plage (peut être négatif, ex: <code>-1</code> )

<code>--contour_thic</code>	<code>int</code>	<code>1</code>	Épaisseur du contour autour du texte (en pixels)
<code>--export_png</code>	Flag	Non spécifié	Génère un dossier avec les images PNG individuelles

---

## Conseils pour l'Utilisation

- **Vérifiez la Police** : Assurez-vous que **Cantarell Bold** est installée pour éviter des erreurs.

**Utilisez le Signe Égal pour les Valeurs Négatives** : Pour les arguments avec des valeurs négatives, utilisez `=` :

`--start_letter=-B`

- - **Surveillance des Ressources** : Si vous générez un grand nombre d'images, assurez-vous d'avoir suffisamment d'espace disque et de mémoire.
  - **Personnalisation Avancée** : Si vous avez besoin de fonctionnalités supplémentaires, vous pouvez modifier le script en fonction de vos besoins, en respectant les bonnes pratiques de programmation.
- 

## Assistance et Support

Si vous rencontrez des problèmes ou avez des questions, n'hésitez pas à demander romain à la CADO ([cado@gendarmerie.interieur.gouv.fr](mailto:cado@gendarmerie.interieur.gouv.fr))

---

Merci d'utiliser `generate_images.py` !