

Améliorez Vos Affichages avec le module Rich

Le module **Rich** est un outil puissant pour ceux qui souhaitent améliorer l'affichage de leurs programmes Python dans la console. Ce module vous permet de rendre vos sorties plus lisibles et esthétiques en ajoutant des couleurs, des tableaux, des barres de progression, des arbres de dossiers, et bien plus encore. Dans cet article, nous allons explorer quelques-unes des fonctionnalités de base que vous pouvez utiliser avec la fonction `print()`.

 Exploration

Installation du Module Rich

Avant de commencer, vous devez installer le module Rich.

Dans Thonny

Allez dans le menu Outils/Gérer les paquets, cherchez "rich" et installer le paquet.

L'affichage des couleurs n'est pas pris en charge par la console de Thonny. Il faudra exécuter vos programmes dans une console système (`CTRL-T`).

Dans Python (global):

Vous pouvez l'installer facilement via `pip` :

```
1 | pip install rich
```

Une fois installé, vous pouvez commencer à utiliser les fonctionnalités de Rich dans vos scripts Python.

Dans PyCharm (5e et 6e TTR)

Le module est facilement installable depuis le package manager.

Utilisation de Rich avec `print()`

L'un des principaux avantages de Rich est sa compatibilité avec la fonction `print()`. Contrairement à la fonction `print()` standard de Python, Rich permet de formater facilement le texte, de le colorier et de l'organiser en tableaux.

Exemple de Base : Texte Coloré

Pour commencer, voyons comment afficher du texte en couleur. Avec Rich, c'est aussi simple que d'utiliser la fonction `print()` :

```
1 | from rich import print
2 |
3 | print("[bold red]Ceci est un texte en rouge gras[/bold red]")
```

Dans cet exemple, le texte "Ceci est un texte en rouge gras" sera affiché en rouge et en gras. Le module Rich utilise des balises similaires à celles de HTML pour définir le formatage du texte.

Codes couleurs

Voici une liste des principales couleurs disponibles avec le module Rich, que vous pouvez utiliser pour styliser le texte dans vos sorties console :

Couleurs de Base

- **red** : Rouge
- **green** : Vert
- **blue** : Bleu
- **yellow** : Jaune
- **magenta** : Magenta (Rose violacé)
- **cyan** : Cyan (Bleu clair)
- **white** : Blanc
- **black** : Noir

Couleurs Claires

- **bright_red** : Rouge clair
- **bright_green** : Vert clair
- **bright_blue** : Bleu clair
- **bright_yellow** : Jaune clair
- **bright_magenta** : Magenta clair
- **bright_cyan** : Cyan clair
- **bright_white** : Blanc très clair
- **grey** : Gris

Nuances de Gris

- **grey0** : Noir pur
- **grey1** à **grey23** : Différentes nuances de gris allant de plus foncé à plus clair
- **grey100** : Blanc pur

Autres Couleurs Prédéfinies

- **orange1** : Orange vif
- **purple** : Violet
- **brown** : Marron
- **pink** : Rose
- **turquoise** : Turquoise

Ajouter des Styles au Texte

Rich vous permet également d'ajouter plusieurs styles à votre texte. Vous pouvez combiner des couleurs, du gras, de l'italique, du souligné, etc.

```
1 | print("[italic blue]Texte en bleu et italique[/italic blue]")
2 | print("[underline green]Texte en vert souligné[/underline green]")
3 | print("[bold italic yellow]Texte en jaune gras et italique[/bold italic yellow]")
```

Ces options sont idéales pour mettre en évidence des parties spécifiques de vos sorties.

Affichage de Tableaux

L'une des fonctionnalités les plus intéressantes de Rich est la possibilité d'afficher des tableaux. Ceci est particulièrement utile pour afficher des données structurées.

Voici un exemple:

```
1 | from rich.table import Table
2 |
3 | table = Table(title="Résultats des Étudiants")
4 |
5 | table.add_column("Nom", justify="right", style="cyan", no_wrap=True)
6 | table.add_column("Age", justify="right", style="magenta")
7 | table.add_column("Moyenne", justify="right", style="green")
8 |
9 | table.add_row("Alice", "21", "14.5")
10 | table.add_row("Bob", "22", "16.0")
11 | table.add_row("Charlie", "20", "13.8")
12 |
13 | print(table)
```

Ce code créera un tableau stylisé avec des colonnes "Nom", "Age", et "Moyenne", chacune ayant sa propre couleur. Le tableau sera automatiquement ajusté pour être bien lisible dans la console.

Les styles conditionnels

Il est facile de personnaliser vos couleurs et styles en fonction des valeurs que Afficher.

Il suffit de créer une variable `style` et de l'utiliser dans une f-string:

```
1  from rich.table import Table
2  from rich import print
3
4  # Création d'un tableau avec du style conditionnel
5  table = Table(title="Scores avec Mise en Évidence")
6
7  table.add_column("Nom", style="cyan")
8  table.add_column("Score", style="magenta")
9
10 # Ajouter des lignes avec mise en évidence conditionnelle
11 scores = [("Alice", 85), ("Bob", 92), ("Charlie", 58), ("Dave", 88)]
12
13 for name, score in scores:
14     if score >= 90:
15         style = "bold green"
16     elif score < 60:
17         style = "bold red"
18     else:
19         style = "white"
20
21     table.add_row(name, f"[{style}]{score}[/{style}]")
22
23 print(table)
24
```

Créer des encadrés

Il est tout à fait possible d'encadrer du texte avec le module **Rich**. Pour cela, vous pouvez utiliser la classe `Panel` qui permet d'afficher du texte ou même d'autres objets Rich (comme des tableaux, des arbres, etc.) à l'intérieur d'un encadré, avec des bordures personnalisables.

Encadrement d'un texte:

Voici un exemple simple où l'on encadre du texte avec un `Panel` :

```
1 from rich import print
2 from rich.panel import Panel
3
4 # Création d'un texte encadré
5 encadre = Panel("Ceci est un texte encadré")
6
7 # Affichage du texte encadré
8 print(encadre)
```

Changer la couleur de la bordure

Vous pouvez changer la couleur de la bordure grâce au paramètre `boder_style` :

```
1 from rich import print
2 from rich.panel import Panel
3
4 # Création d'un texte encadré avec un titre
5 encadre = Panel("Ceci est un texte encadré", border_style="green")
6
7 # Affichage du texte encadré
8 print(encadre)
```

Ajouter un titre au panel

Voici un exemple où l'on ajoute un titre au texte avec un `Panel` :

```
1 from rich import print
2 from rich.panel import Panel
3
4 # Création d'un texte encadré avec un titre
5 encadre = Panel("Ceci est un texte encadré", title="Encadré", border_style="green")
```

```

6
7 # Affichage du texte encadré
8 print(encadre)

```

Personnalisation du Panel

Le `Panel` offre plusieurs options pour personnaliser l'encadrement :

- `title` : Permet d'ajouter un titre au-dessus du panel.
- `border_style` : Définit la couleur des bordures.
- `padding` : Définit l'espacement intérieur entre le texte et les bordures du panel.
- `expand` : Si `True`, le panel occupera toute la largeur disponible de la console.
- `subtitle` : Ajoute un sous-titre en bas du panel.

Exemple avec expand

```

1 from rich import print
2 from rich.panel import Panel
3
4 # Création d'un texte encadré
5 encadre = Panel(
6     "Voici un texte important avec un encadrement stylisé.",
7     title="Encadré Principal",
8     border_style="bold red",
9     expand=False
10 )
11
12 # Affichage du texte encadré
13 print(encadre)

```

Ces exemples montrent comment vous pouvez utiliser `Panel` pour encadrer du texte ou d'autres éléments, en ajoutant ainsi une touche esthétique et organisée à vos sorties dans la console. Cette fonctionnalité est particulièrement utile pour mettre en avant des informations importantes ou pour structurer des interfaces en ligne de commande plus complexes.

Utiliser des Barres de Progression

Rich facilite également l'affichage de barres de progression, ce qui est très utile pour indiquer l'avancement de certaines tâches.

```

1 from rich.progress import track
2 import time

```

```
3 |  
4 | for step in track(range(10), description="Traitement..."):  
5 |     time.sleep(0.1)
```

Ici, la fonction `track()` de Rich enveloppe une boucle pour afficher une barre de progression pendant que la boucle s'exécute. Cela permet de donner un retour visuel immédiat sur la progression d'une tâche.

Conclusion

Le module Rich est un outil formidable pour tous ceux qui cherchent à améliorer l'affichage de leurs programmes Python dans la console. En utilisant Rich avec la fonction `print()`, vous pouvez facilement ajouter des couleurs, des styles, des tableaux, des barres de progression, et bien plus encore à vos sorties. Ces fonctionnalités permettent de rendre vos programmes non seulement plus esthétiques mais aussi plus lisibles et compréhensibles.

En explorant plus en profondeur les possibilités offertes par Rich, vous pouvez améliorer significativement l'expérience utilisateur de vos scripts Python. Alors, n'hésitez pas à l'essayer dans vos prochains projets !

Ressources

Pour aller plus loin, voici quelques ressources utiles pour apprendre à utiliser le module Rich :

- [Documentation officielle de Rich](#)
- [Référence rapide des balises de style](#)
- [Exemples de code avec Rich sur GitHub](#)

Essayez ces fonctionnalités dans vos projets et vous verrez combien elles peuvent transformer l'affichage de vos scripts Python!

Exercices

Fiche personnage D&D

Vous allez recréer une version simplifiée et colorée de votre fiche personnage D&D à l'aide de Rich:

Votre personnage:

- Nom: Prof
- Race: Humain
- Classe: Guerrier
- Points d'XP: 450
- Niveau: 2
- Objet Magique: Excalibur

Caractéristiques:

- Force: 10
- Dextérité: 18
- Constitution: 17
- BONUS: +3
- Intelligence: 6
- Sagesse: 19
- Charisme: 13
- Points de vie: 13

1. Encadrez les titres de sections
2. Le nom de la caractéristique doit être affiché en blanc.
3. La valeur doit être affichée en jaune

Etape 2:

1. Utilisez des couleurs différentes pour chaque race
2. Affichez les valeurs des caractéristiques <10 en rouge, les autres en vert