

Thonny Python - Déboguer son code

Le débogage de code est une compétence essentielle pour tout développeur informatique. Qu'il s'agisse de résoudre des erreurs de syntaxe simples ou de traquer des bogues complexes, savoir comment déboguer efficacement son code peut faire toute la différence entre un projet réussi et des heures de frustration. Dans cet article, nous allons explorer les concepts de base du débogage de code, ainsi que comment utiliser les outils de débogage disponibles dans l'environnement de développement intégré (IDE) Thonny.

4TTR

4GMS

3TTR



Aventurier

Comprendre le débogage de code

Le débogage de code consiste à **identifier**, **localiser** et **résoudre les erreurs** (ou "bugs") présentes dans un programme informatique. Ces erreurs peuvent prendre différentes formes, telles que des erreurs de syntaxe, des erreurs logiques ou des comportements inattendus du programme. Le processus de débogage implique généralement l'utilisation d'outils spécifiques, tels que des débogueurs, des impressions de messages de débogage ou des tests unitaires, pour isoler et corriger ces erreurs.

Les étapes du débogage de code

- 1. Reproduire le problème:** La première étape pour déboguer un code est de reproduire le problème. Cela implique d'**identifier les conditions ou les actions** qui provoquent l'erreur ou le comportement inattendu dans le programme.
- 2. Isoler la source du problème:** Une fois que le problème a été reproduit, l'étape suivante consiste à **isoler la source du problème** en identifiant les lignes de code spécifiques ou les parties du programme qui contribuent à l'erreur.
- 3. Utiliser des outils de débogage:** Les outils de débogage, tels que les débogueurs intégrés aux IDE, sont **extrêmement utiles** pour examiner l'exécution du programme, surveiller les valeurs des variables et détecter les erreurs à l'exécution.
- 4. Apporter des modifications et tester:** Une fois que la source du problème a été identifiée, le développeur peut apporter des modifications au code pour **corriger l'erreur**. Il est important de tester ces modifications pour s'assurer qu'elles résolvent effectivement le problème sans introduire de nouveaux bugs.

Déboguer son code dans Thonny

Thonny est un IDE convivial et puissant pour Python, qui offre plusieurs outils de débogage pour aider les développeurs à identifier et à résoudre les erreurs dans leur code.

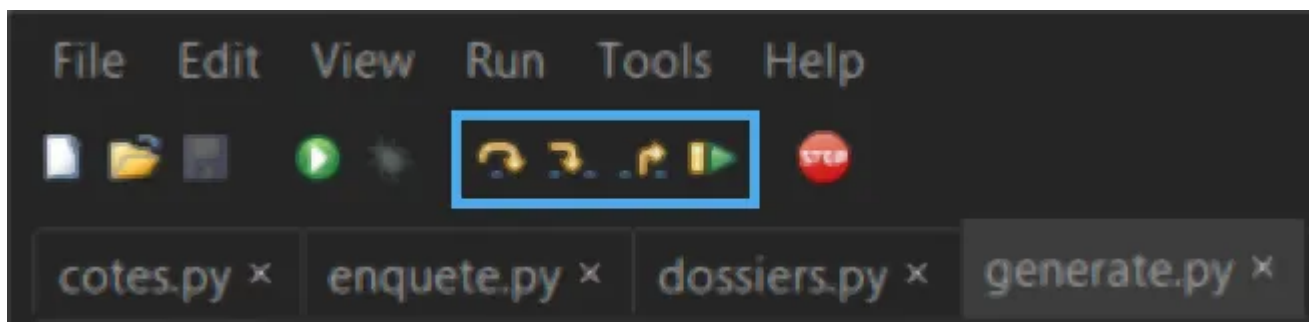
Thonny comprend un débogueur intégré qui permet aux développeurs d'exécuter **leur code pas à pas**, autrement dit **ligne par ligne**, d'**observer les valeurs des variables** à chaque étape, et de **mettre en pause l'exécution du programme** pour inspecter l'état du code.

Pour utiliser le débogueur dans Thonny, il suffit d'exécuter votre programme en **mode "debug"** grâce au bouton de la barre d'outils et/ou de placer des **points d'arrêt** à des endroits stratégiques dans le code, puis de lancer le programme en mode débogage.

Les différentes possibilités d'exécution pas à pas dans Thonny

L'exécution pas à pas, également connue sous le nom de "step-by-step execution" en anglais, est une fonctionnalité essentielle des débogueurs intégrés dans les environnements de développement intégrés (IDE) comme Thonny. Cette fonctionnalité permet aux développeurs **d'exécuter leur code ligne par ligne**, en observant et en analysant l'état du programme à chaque étape.

Ces fonctionnalités sont accessibles via la barre d'outils (et les raccourcis clavier correspondants):



Mode "Step Over"

Thonny offre la possibilité d'exécuter le code pas à pas en mode **"Step Over"** (étape suivante). Lorsque les développeurs utilisent cette option, **le débogueur exécute la ligne de code actuelle, puis s'arrête à la ligne suivante, sans entrer dans les appels de fonction ou de méthode**. Cela permet aux développeurs de suivre l'exécution de leur code de manière linéaire, en évitant les détails internes des fonctions ou des méthodes appelées.

Mode "Step Into"

Thonny propose également l'option d'exécuter le code pas à pas en mode **"Step Into"** (entrer dans). Avec cette option, le débogueur **entre dans chaque appel de fonction ou de méthode rencontré** pendant l'exécution du code, permettant ainsi aux développeurs d'inspecter le comportement interne de ces fonctions ou méthodes. Cela peut être utile pour comprendre le flux d'exécution du programme et identifier les erreurs potentielles à l'intérieur des fonctions appelées.

Mode "Step Out"

Enfin, Thonny offre la possibilité d'exécuter le code pas à pas en mode "**Step Out**" (sortir). Lorsque les développeurs utilisent cette option, le débogueur **exécute le reste du code de la fonction actuelle jusqu'à ce qu'il atteigne la ligne de retour ou la fin de la fonction**, puis s'arrête à la ligne suivante dans la fonction appelante. Cela permet aux développeurs de passer rapidement à la prochaine étape du processus de débogage, en évitant de parcourir chaque ligne de code à l'intérieur de la fonction actuelle.

En utilisant ces différentes options d'exécution pas à pas dans Thonny, les développeurs peuvent analyser et déboguer leur code de manière efficace et précise, ce qui contribue à la création de logiciels de haute qualité et fiables.

Mode "Resume"

Si vous avez terminé de déboguer votre code, vous pouvez demander à Thonny de continuer à exécuter (resume) votre programme jusqu'à la fin, sans s'arrêter.

Les points d'arrêts (ou *breakpoints*)

Les points d'arrêt, également appelés **breakpoints** en anglais, sont des outils essentiels dans le processus de débogage de code. Dans Thonny, les développeurs peuvent placer des points d'arrêt à des endroits spécifiques de leur code où ils souhaitent mettre en pause l'exécution du programme afin d'inspecter l'état des variables et des données.

Pour ce faire, il suffit de cliquer sur le numéro de la ligne de code correspondante. Un marqueur apparaît dans la marge:

Une fois que le programme atteint un point d'arrêt, **l'exécution est suspendue automatiquement** et le développeur peut examiner le contexte de l'exécution, observer les valeurs des variables et évaluer l'état du code.

Les points d'arrêt permettent aux développeurs de suivre pas à pas l'exécution de leur programme, ce qui est particulièrement utile pour identifier les erreurs et les comportements inattendus. En plaçant stratégiquement des points d'arrêt dans leur code, les développeurs peuvent **cibler spécifiquement les parties du programme qu'ils souhaitent examiner** de plus près, ce qui accélère le processus de débogage et améliore l'efficacité globale du développement logiciel.

Affichage des messages de débogage

En plus du débogueur intégré, Thonny permet également aux développeurs d'insérer des messages de débogage dans leur code à l'aide de la fonction `print()`. Ces messages peuvent être utilisés pour afficher des informations sur l'exécution du programme à des points spécifiques, ce qui peut aider à identifier les erreurs ou les comportements inattendus.

Conclusion

Le débogage de code est une compétence essentielle pour tout développeur informatique. En comprenant les concepts de base du débogage et en utilisant les outils appropriés, les développeurs peuvent identifier et résoudre efficacement les erreurs dans leur code, ce qui contribue à la création de logiciels de haute qualité et fiables. Avec son débogueur intégré et ses autres outils de débogage, Thonny est un excellent choix pour les développeurs Python qui cherchent à améliorer leur efficacité et leur précision lors du débogage de leur code.