Быстрый старт с IDLE Python

Если вы недавно загрузили Python на свой компьютер, то, возможно, заметили новую программу под названием **IDLE**. Вы можете спросить: «Что эта программа делает на моем компьютере? Я её не загружал!». Возможно, вы осознанно не загружали эту программу, но IDLE есть в комплекте дистрибутива Python. IDLE поможет вам начать работу с языком прямо из коробки. Здесь вы узнаете, как работать в IDLE Python, несколько интересных приемов, которые можно использовать в своем путешествии с Python!

- Что такое Python IDLE;
- Как взаимодействовать с Python напрямую, используя IDLE;
- Как редактировать, выполнять и отлаживать файлы Python с помощью IDLE;
- Как настроить IDLE Python на свой вкус.

Содержание

Что есть IDLE Python?

Дистрибутив Python содержит **Integrated Development and Learning Environment – [[Интегрированная среда разработки]] и обучения**, коротко IDLE или даже IDE. Это класс приложений, которые помогают более эффективно писать код. Хотя существует множество IDE, из которых можно что-то выбрать, Python IDLE очень проста, обычно, это идеальный инструмент для начинающего программиста.

Python IDLE входит в дистрибутив для Windows и Mac. Для Linux можно найти и загрузить Python IDLE с помощью вашего менеджера пакетов. После установки можно использовать Python IDLE в качестве интерактивного интерпретатора или редактора файлов.

Интерактивный интерпретатор

Лучшее место для экспериментов с кодом Python – это интерактивный интерпретатор, иначе известный как **shell – оболочка**. Оболочка – это базовая [[REPL]] или Read-Eval-Print Loop. Она **читает** оператор Python, **оценивает** результат этого оператора и затем **выводит результат на экран**. Затем **возвращается** к следующему оператору.

Оболочка Python – отличное место для экспериментов с небольшими фрагментами кода. Доступ к ней можно получить через

терминал или командную строку своего компьютера. Вы можете упростить свой рабочий процесс с Python IDLE, которая сразу же запустит Python при открытии.

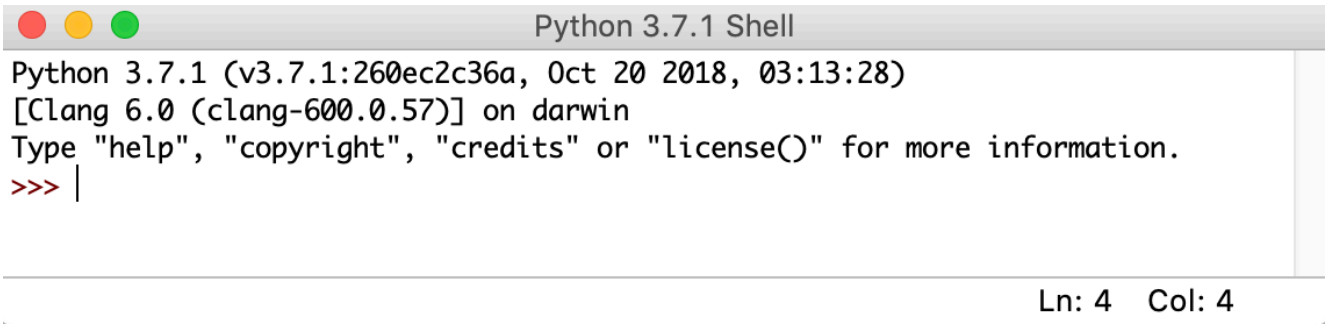
Редактор файлов

Каждый программист должен иметь возможность редактировать и сохранять текстовые файлы. Программы Python – это файлы с расширением .py, которые содержат строчки кода. Python IDLE дает вам возможность легко создавать и редактировать такие файлы.

В Python IDLE есть несколько полезных функций, которые вы увидите в профессиональных IDE, такие как подсветка основного синтаксиса, автозавершение кода и авто-отступ. Профессиональные IDE – более надежное программное обеспечения, для которого крутая кривая обучения. Если вы только начинаете свой путь программирования, то Python IDLE – отличная альтернатива!

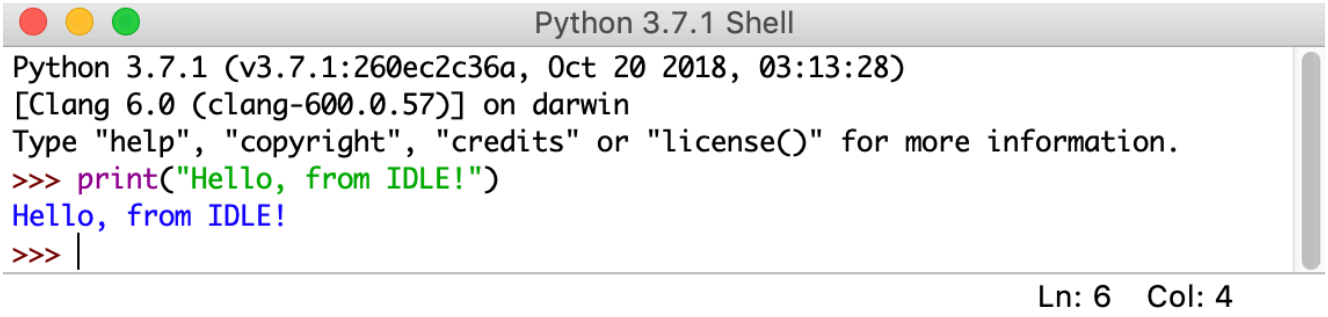
Как использовать оболочку Python IDLE

Оболочка является режимом по умолчанию для Python IDLE. Когда вы нажимаете на значок, чтобы открыть программу, оболочка – это первое, что вы видите:



Интерактивный интерпретатор Python

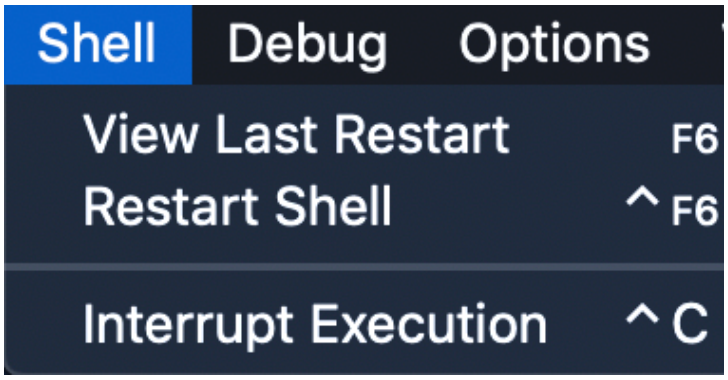
Это пустое окно интерпретатора Python. Вы можете сразу начать «грузить» Python и проверить с помощью короткой строчки кода:



Исполнение кода в интерпретаторе

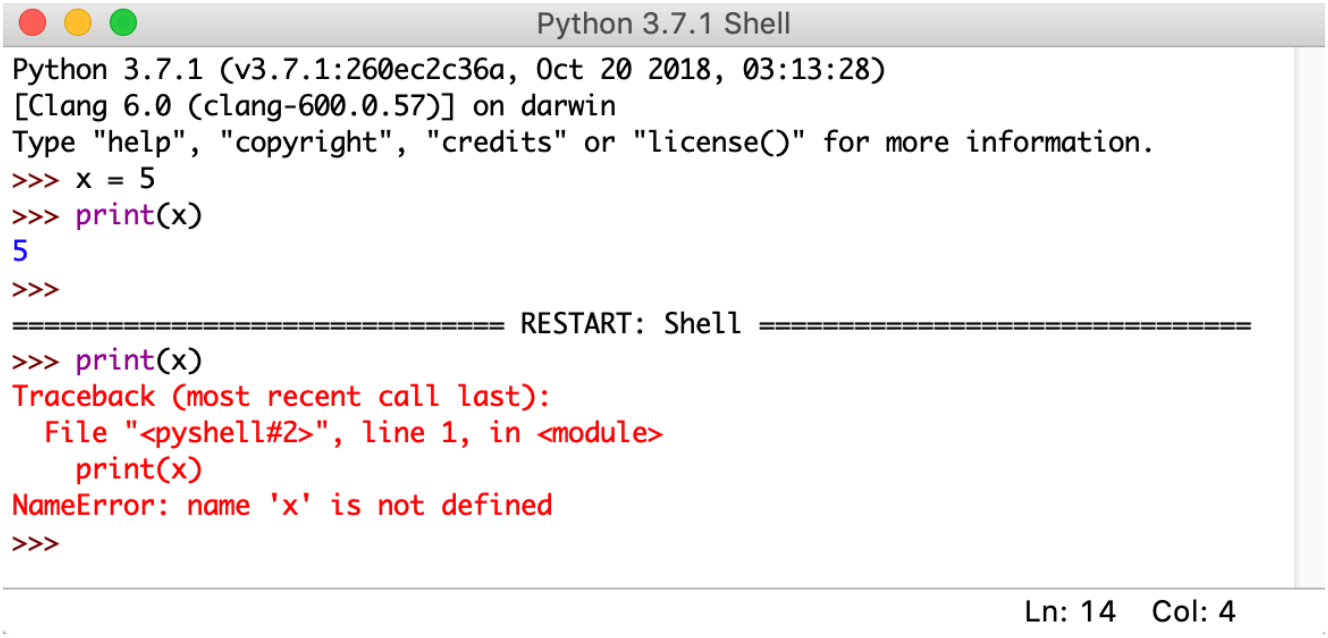
Здесь мы написали вызов функции print() для вывода на экран всего одной строчки "Hello, from IDLE!". Это самый простой способ взаимодействия с Python IDLE. Вы по-одиночке вводите команды, а Python каждый раз отвечает выводом результата на экран.

Далее посмотрите в главное меню и увидите Несколько вариантов использование оболочки:



Несколько вариантов использование оболочки

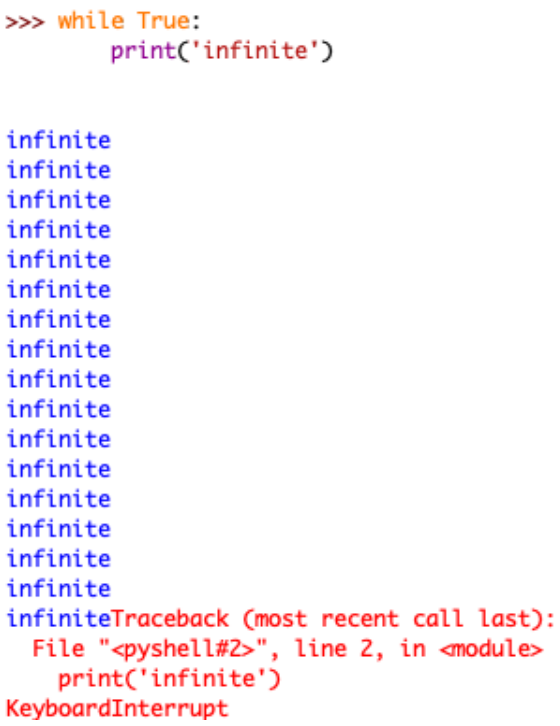
Вы можете **перезапустить оболочку** из этого меню. Если вы выберете этот пункт или одновременно нажмёте клавиши Ctrl + F6, то состояние оболочки очистится и она будет работать так, как будто вы запустили новый экземпляр Python IDLE. Оболочка забудет обо всем, что было раньше:



Перезапустим оболочку

На изображении выше вы сначала объявляете переменную, x = 5. Когда вы вызываете print(x), оболочка показывает правильный результат, число 5. Однако, когда вы перезапускаете оболочку и пытаетесь вызвать print(x), то печатается результат трассировки. Это сообщение об ошибке, в котором говорится, что переменная x не определена. Оболочка забыла обо всем, что было до этого.

Из этого меню можно прервать выполнение любой команды. Это остановит любую программу или оператор, выполняющийся в оболочке на момент прерывания. Посмотрите, что происходит при отправке прерывания с клавиатуры:



Прерывание работы в оболочке

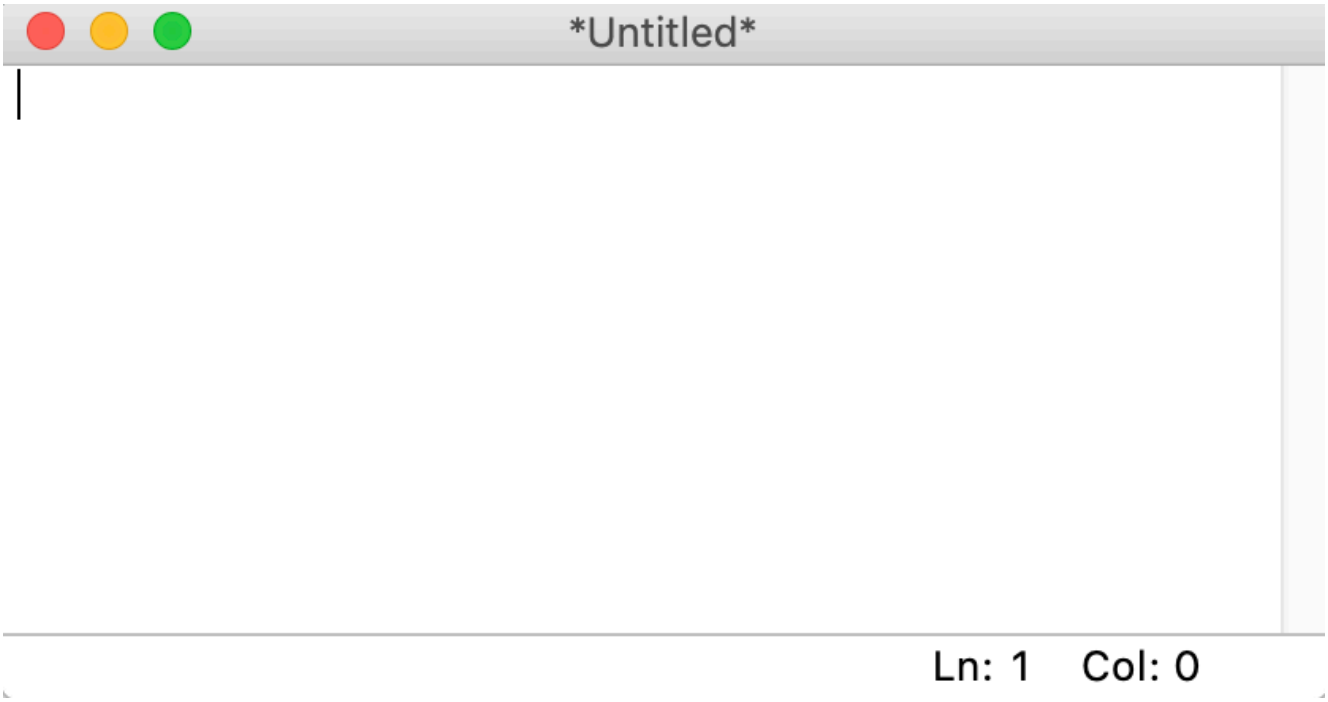
Сообщение об ошибке KeyboardInterrupt отображается красным цветом в нижней части окна. Программа получила прерывание и прекратила выполнение.

Как работать с файлами Python

Python IDLE имеет полноценный текстовый редактор файлов, который дает вам возможность писать и выполнять программы, не выходя из среды. Встроенный редактор имеет несколько встроенных функций, таких как завершение кода и авто-отступы, которые ускоряют процесс кодирования. Для начала, давайте посмотрим, как писать и выполнять программы в Python IDLE.

Открытие файла

Для создания нового файла Python в главном меню выберите *File → New File* или Ctrl + N в Windows. В редакторе будет открыт пустой файл:



Создаём новый файл

Из этого окна вы можете записать новый файл Python. Можно открыть существующий файл, выбрав *File → Open …* в главном меню или Ctrl + O в Windows. Откроется менеджер файлов вашей операционной системы, где можно найти и выбрать нужный для работы файл.

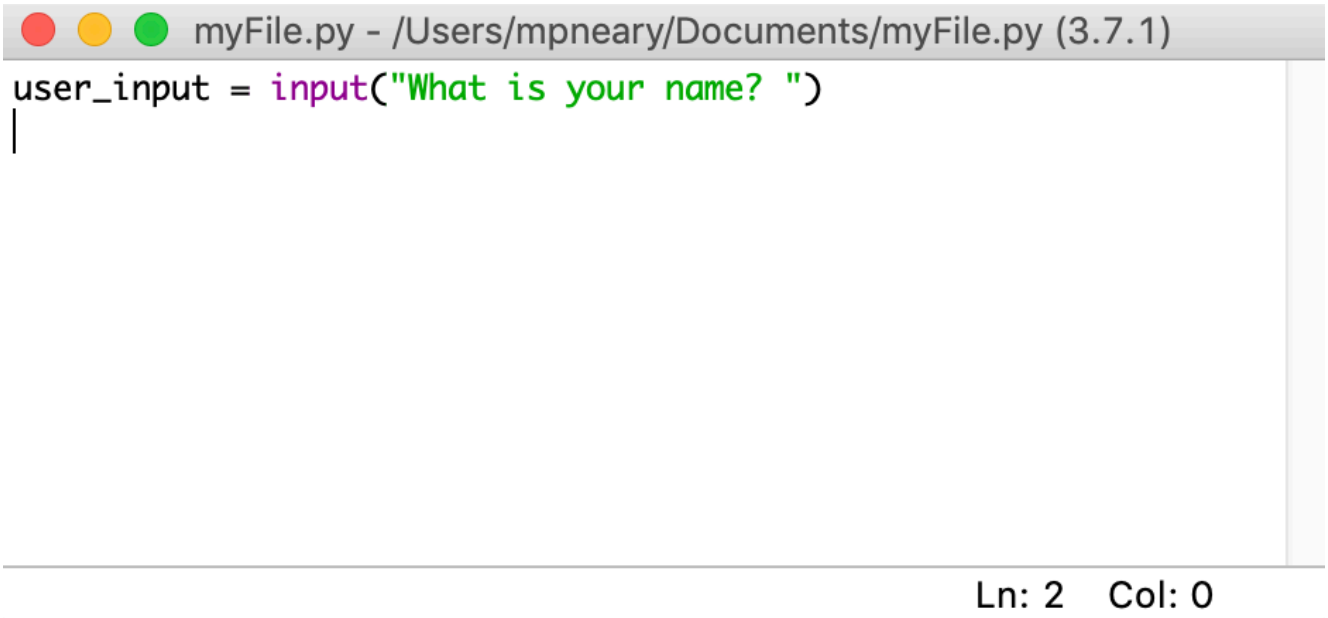
Если вы заинтересованы в чтении исходного кода модуля Python, вы можете выбрать *File → Path Browser*. Это позволит просмотреть модули, которые видет Python IDLE. Если дважды щелкнуть на одном из них, то откроется редактор файлов и вы сможете его прочитать.

Содержимое этого окна будет таким же, как пути, возвращаемые при вызове sys.path. Если вам известно имя определенного модуля, который вы хотите просмотреть, вы можете выбрать *File → Module Browser* или Alt + C в Windows и записать имя модуля в появившемся окне.

Редактирование файла

Открыв файл в Python IDLE, можно внести в него изменения. Когда вы будете готовы отредактировать файл, то увидите что-то

вроде этого:



Редактируем файл в Python IDLE

Содержимое вашего файла отображается в открытом окне. Панель в верхней части окна содержит три важных элемента:

1. **Имя** файла, который вы редактируете
2. **Полный путь** к папке, где вы можете найти этот файл на вашем компьютере
3. **Версия** Python, которую использует IDLE

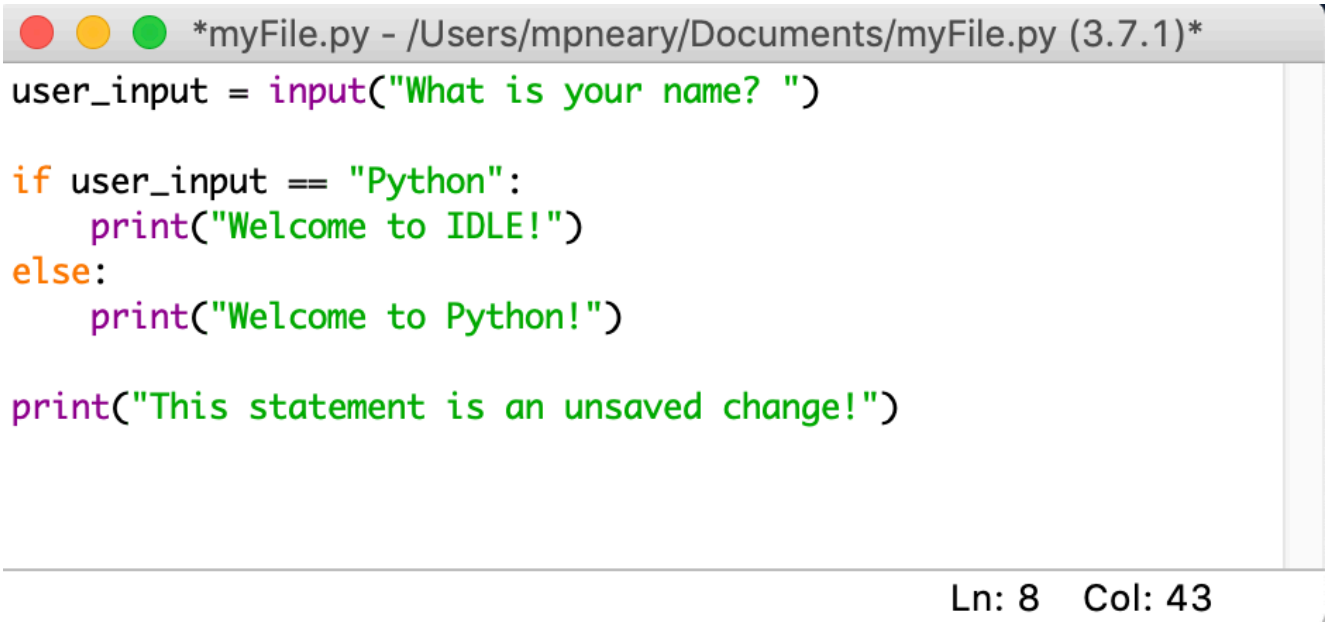
На изображении выше вы редактируете файл myFile.py, который находится в папке Documents. Версия Python 3.7.1, которую вы можете увидеть в скобках.

В правом нижнем углу окна также есть две цифры:

1. **Ln:** показывает номер строки, на которой находится курсор.
2. **Col.** показывает номер столбца, на котором находится курсор.

Полезно видеть эти цифры, чтобы быстрее находить ошибки. Они также помогают убедиться, что вы остаетесь в пределах определенной ширины линии.

В этом окне есть несколько визуальных подсказок, которые помогут вам не забыть сохранить свою работу. Если вы посмотрите внимательно, то увидите, что Python IDLE использует звездочки, чтобы сообщить вам, что в вашем файле есть несохраненные изменения:



Правки файла

Имя файла, отображаемое в верхней части окна IDLE, окружено звездочками. Это означает, что в вашем редакторе есть несохраненные изменения. Вы можете сохранить эти изменения с помощью стандартного сочетания клавиш в вашей системе или выбрать *File → Save* в строке меню. Убедитесь, что вы сохранили свой файл с расширением .py, чтобы подсветка синтаксиса была включена.

Выполнение файла

Если захотите выполнить файл, который создали в IDLE, то сначала надо убедиться, что он сохранен. Помните, что вы можете увидеть, правильно ли сохранен ваш файл, посмотрев звездочки вокруг имени файла в верхней части окна редактора файлов. Не беспокойся! Если забудешь, Python IDLE будет напоминать вам о необходимости сохранения всякий раз, когда вы пытаетесь выполнить несохраненный файл.

Чтобы выполнить файл в режиме IDLE, просто нажмите клавишу F5 на клавиатуре. Вы также можете выбрать *Run → Run Module* в строке меню. Любой из вариантов перезапустит интерпретатор Python, а затем запустит код, который вы написали, с новым интерпретатором. Процесс такой же, как когда вы запускаете python3 -i [имя файла] в своем терминале.

Когда ваш код будет выполнен, интерпретатор будет знать все о вашем коде, включая любые глобальные переменные, функции и классы. Это делает Python IDLE отличным местом для проверки ваших данных, если что-то пойдет не так. Если вам когда-либо понадобится прервать выполнение вашей программы, вы можете нажать Ctrl + C в интерпретаторе, который запускает ваш код.

Как улучшить рабочий процесс

Теперь, когда вы увидели, как писать, редактировать и выполнять файлы в Python IDLE, пришло время ускорить ваш рабочий процесс! Редактор Python IDLE предлагает несколько функций, которые вы увидите в большинстве профессиональных IDE для быстрого кодирования. Эти функции включают в себя авто[5]отступы, завершение кода и помощь в контексте кода.

Авто-отступ

IDLE автоматически сделает отступ в коде, когда должен начаться новый блок. Обычно это происходит после ввода двоеточия (:). Когда вы нажимаете клавишу ввода после двоеточия, курсор автоматически перемещается на определенное количество пробелов и начинает новый блок кода.

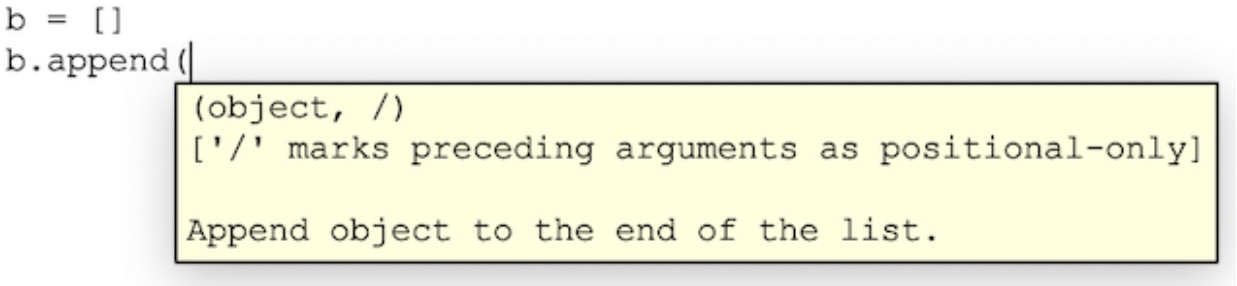
Вы можете настроить, сколько пробелов будет перемещать курсор в настройках, но по умолчанию используются стандартные четыре пробела. Разработчики Python договорились о стандартном стиле для хорошо написанного кода Python, который включает в себя правила для отступов, пробелов и многое другое. Этот стандартный стиль был формализован и теперь известен как **PEP 8**. Чтобы узнать больше об этом, ознакомьтесь со статьёй

Как оформить код.

Дополнение кода и подсказки

Когда вы пишете код для большого проекта или сложной проблемы, вы можете тратить много времени, просто набирая весь необходимый код. **Завершение кода** поможет вам сэкономить время, пытаясь завершить код за вас. Python IDLE имеет базовую функциональность завершения кода такую, как автозаполнение имен функций и классов. Для этого просто нажмите клавишу табуляции в редакторе после записи текста.

В Python IDLE есть подсказки для определенной части кода, которые помогут вспомнить, что нужно этому элементу. После того, как вы введете левую скобку, чтобы начать вызов функции, появится подсказка о вызове, если вы ничего не наберете в течение нескольких секунд. Например, если вы не совсем помните, как добавить к list, после открывающей скобки вы можете сделать паузу, чтобы вызвать подсказку:



Подсказка и дополнение кода

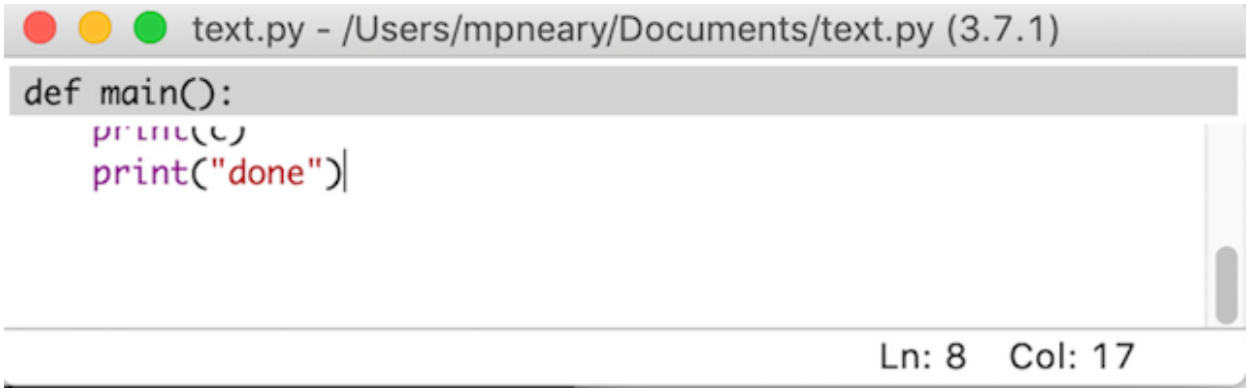
Подсказка о вызове будет отображаться как всплывающая заметка, напоминая вам, как добавить в список. Подобные советы по вызову дают полезную информацию, когда вы пишете код.

Код в контексте

Код в контексте есть отличное свойство редактора файлов Python IDLE. Он покажет вам область действия функции, класса, цикла или иной другой конструкции. Это особенно полезно, когда вы просматриваете длинный файл и нужно отследить, где вы

находитесь, просматривая код в редакторе.

Чтобы включить его, выберите *Options → Code Context* в главном меню. Вы увидите серую полосу в верхней части окна редактора:



Контекст кода включен

Когда вы прокручиваете свой код вниз, **контекст**, содержащий каждую строку кода, будет оставаться внутри этой серой полосы. Это означает, что функции print(), которые вы видите на изображении выше, являются частью основной функции. Когда вы достигнете линии, выходящей за рамки этой функции, полоса исчезнет.

Как отлаживают в IDLE

Ошибка – всегда неожиданная проблема любой программы. Ошибки проявляются по разному, некоторые из них исправить труднее, чем другие. Ошибки достаточно хитры, иногда их нельзя обнаружить, просто прочитав свою код. К счастью, в Python IDLE есть несколько основных инструментов, которые помогут вам при отладке своих программ с легкостью!

Режим отладки интерпретатора

Если вы хотите запустить свой код с помощью встроенного отладчика,то вам нужно будет включить эту функцию. Для этого выберите < em>Debug → Debugger в строке меню Python IDLE. В интерпретаторе вы должны увидеть, что [DEBUG ON] появляется непосредственно перед приглашением (>>>), что означает, что интерпретатор готов и ждет.

Когда вы запустите ваш файл Python, появится окно отладчика:

DEBUG ON

В этом окне вы можете проверить значения ваших локальных и глобальных переменных при выполнении кода. Это дает вам представление о том, как ваши данные обрабатываются во время работы вашего кода.

Вы также можете нажать следующие кнопки для перемещения по коду:

- **Go:** Следующяя [[точка останова]]. Вы узнаете об этом в следующем разделе.
- **Шаг** Нажмите, чтобы выполнить текущую строку и перейти к следующей.
- **Over:** Если текущая строка кода содержит вызов функции, нажмите эту кнопку, чтобы войти в этк функцию. Другими словами, выполните эту функцию и перейдите к следующей строке, но не делайте паузу во время выполнения функции (если не существует точки останова).
- **Out:** Если текущая строка кода находится в функции, нажмите эту кнопку, чтобы выйти их этой функции. Другими словами, продолжайте выполнение этой функции, пока не вернетесь к ней.

Будьте осторожны&nbps;— кнопки реверса нет! Во время отладки

своей программы можно сделать шаг только вперед.

В окне отладки вы увидите четыре флажка:

1. **Globals:** глобальная информация о вашей программе
2. **Locals:** локальная информация о вашей программе во время выполнения
3. **Steck:** функции, которые выполняются во время выполнения
4. **Source:** ваш файл в редакторе IDLE

Выбрав один из них, вы увидите соответствующую информацию в окне отладки.

Breakpoints — точки останова

[[Точки останова]] — это строка, которую вы определили как место, где интерпретатор должен приостановить выполнение кода. Она будет работать только при включенном режиме *DEBUG*, поэтому убедитесь, что вы сделали это в первую очередь.

Чтобы установить точку останова, щелкните правой кнопкой мыши строку кода, которую вы хотите приостановить. Строка кода будет выделена желтым цветом для визуализации. Вы можете установить столько точек останова в своем коде, сколько захотите. Чтобы отменить точку останова, снова щелкните правой кнопкой мыши по той же строке и выберите *Clear Breakpoint*.

После того, как вы установили свои точки останова и включили режим *DEBUG*, вы можете запускать свой код, как обычно. Появится окно отладчика, и вы можете начать пошаговое выполнение кода вручную.

Ошибки и исключения

Когда вы видите сообщение об ошибке в интерпретаторе, Python IDLE позволяет перейти прямо в файле непосредственно к строке, вызвавшей сбой. Все, что вам нужно сделать, это выделить номер строки или имя файла указанным курсором и выбрать *Debug → Go to file/line* в строке меню. Откроет файл на строке с ошибкой.

Эта функция работает независимо от того, включен ли режим *DEBUG*.

Python IDLE также предоставляет инструмент под названием **средство просмотра стека**. Вы можете получить к нему доступ через параметр *Debug* в строке меню. Этот инструмент покажет вам трассировку ошибки, отображаемую в стеке последней ошибки или исключение, с которым столкнулся Python IDLE при запуске вашего кода. Когда происходит неожиданная или интересная ошибка, вам может быть полезно взглянуть на стек. В противном случае эту функцию может быть сложно разобрать, и, вероятно, она вам не пригодится, если вы не пишете очень сложный код.

How to Customize Python IDLE

Есть масса способов придать Python IDLE желаемый внешний вид. По умолчанию он основан на цветах в логотипе Python. Если это вам не нравится, то, почти всегда, ситуацию можно исправить на свой вкус.

Чтобы открыть окно настройки, выберите *Options → Configure IDLE* в строке меню. Чтобы просмотреть результат изменения, которое вы хотите внести, нажмите *Apply*. Когда вы закончите настройку Python IDLE, нажмите *OK* чтобы сохранить все ваши изменения. Если вы не хотите сохранять свои изменения, просто нажмите *Cancel*.

5 пользовательских настроек Python IDLE:

1. Fonts/Tabs – Шрифты/Закладки
2. Highlights – Подсветка кода
3. Keys – Клавиши
4. General – Общие
5. Extensions – Расширения

Let’s take a look at each of them now.

Fonts/Tabs

Первая вкладка позволяет изменять стиль, размер и цвет шрифта. Можно подобрать любой понравившийся вам стиль из вашей операционной системы. Окно настроек шрифта выглядит так:

Настройка Шрифты/Закладки

You can use the scrolling window to select which font you prefer. (I recommend you select a fixed-width font like Courier New.) Pick a font size that’s large enough for you to see well. You can also click the checkbox next to *Bold* to toggle whether or not all text appears in bold.

This window will also let you change how many spaces are used for each indentation level. By default, this will be set to the PEP 8 standard of four spaces. You can change this to make the width of your code more or less spread out to your liking.

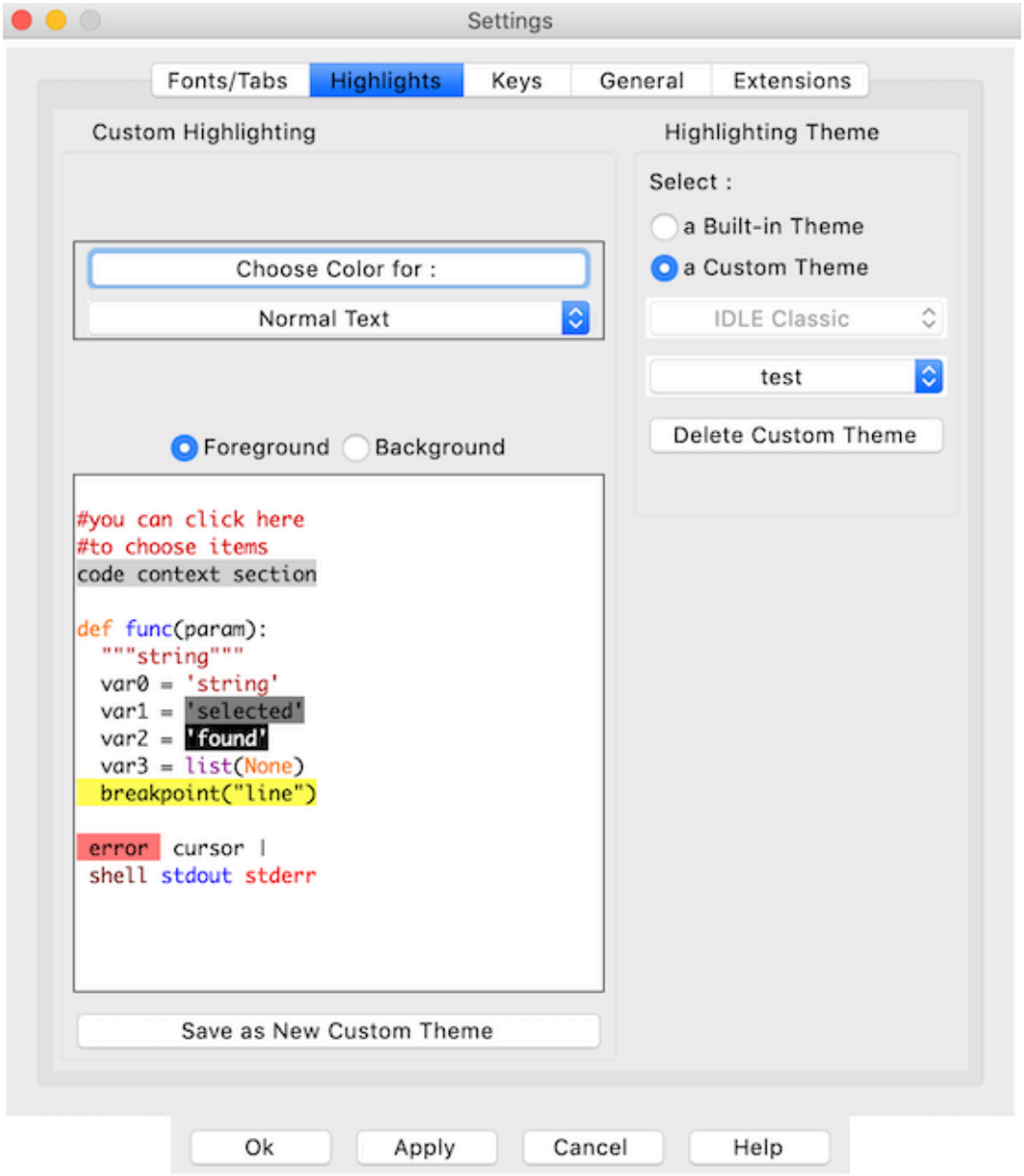
Highlights

Вторая вкладка настройки позволит изменить основные параметры **Подсветки синтаксиса**. Это важная функция любой IDE, которая помогает визуально различать различные конструкции Python и данные, используемые в вашем коде.

Python IDLE позволяет полностью настроить внешний вид вашего кода Python. Он предустановлен с тремя различными темами подсветки:

1. IDLE Day
2. IDLE Night
3. IDLE New

You can select from these pre-installed themes or create your own custom theme right in this window:



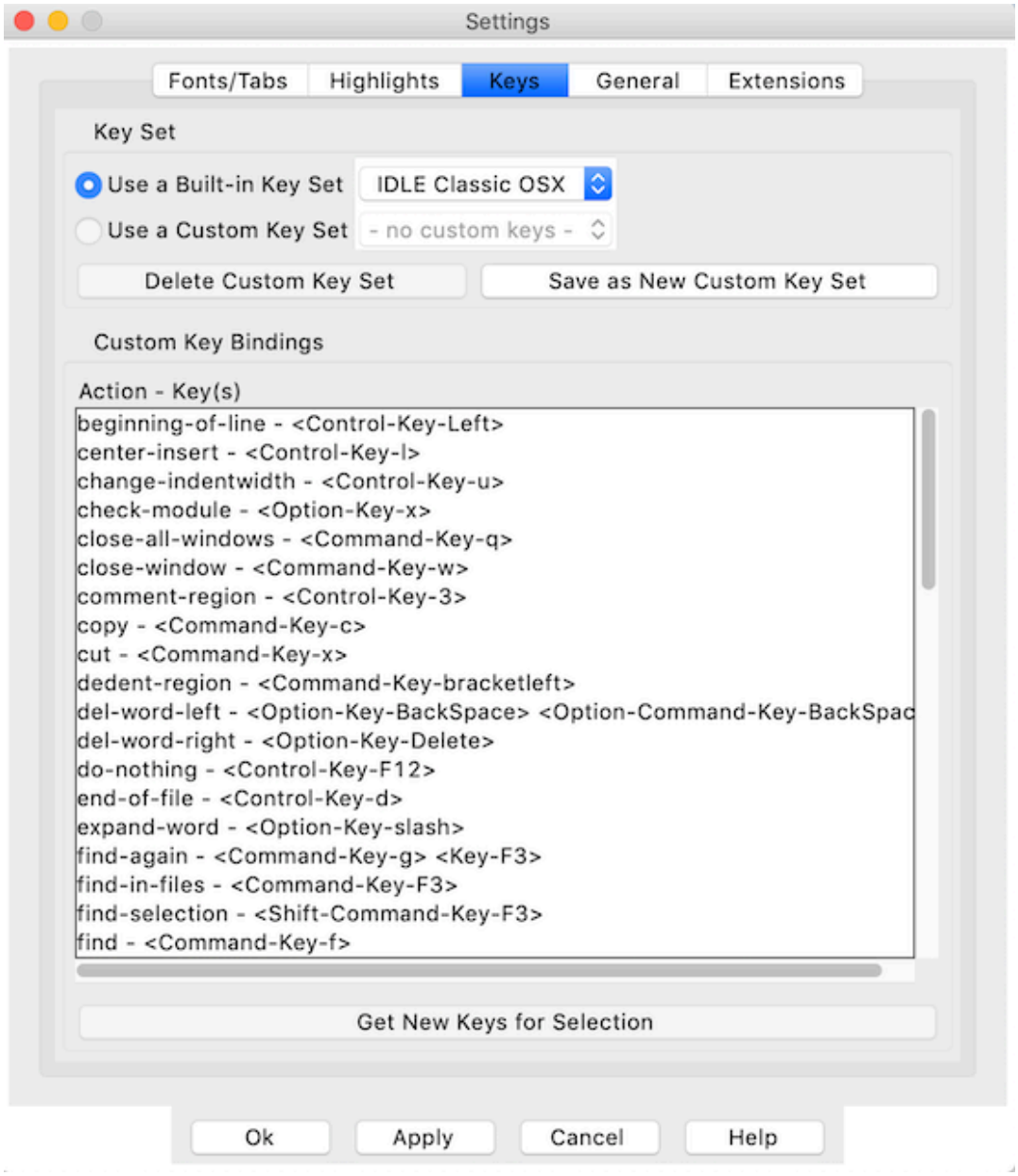
Настройка Подсветки синтаксиса

К сожалению, IDLE не позволяет устанавливать пользовательские темы из файла. Вы должны создать таможенную тему из этого окна. Для этого вы можете просто начать менять цвета для разных предметов. Выберите элемент и нажмите *Choose color for*. Вы попадете в палитру цветов, где сможете выбрать именно тот цвет, который хотите использовать.

Затем вам будет предложено сохранить эту тему как новую пользовательскую тему, и вы сможете ввести имя по вашему выбору. Затем вы можете продолжить изменять цвета разных предметов, если хотите. Не забудьте нажать *Apply*, чтобы увидеть изменения в действии!

Keys

Третья вкладка настройки позволяет сопоставить различные нажатия клавиш с действиями, также известными как **shortkey**. Это жизненно важный компонент вашей производительности, когда вы используете IDE. Вы можете использовать свои собственные сочетания клавиш или те, которые встроены в IDLE. Предварительно установленные shortkey — хорошее место для начала:



Настройка клавиш быстрого доступа

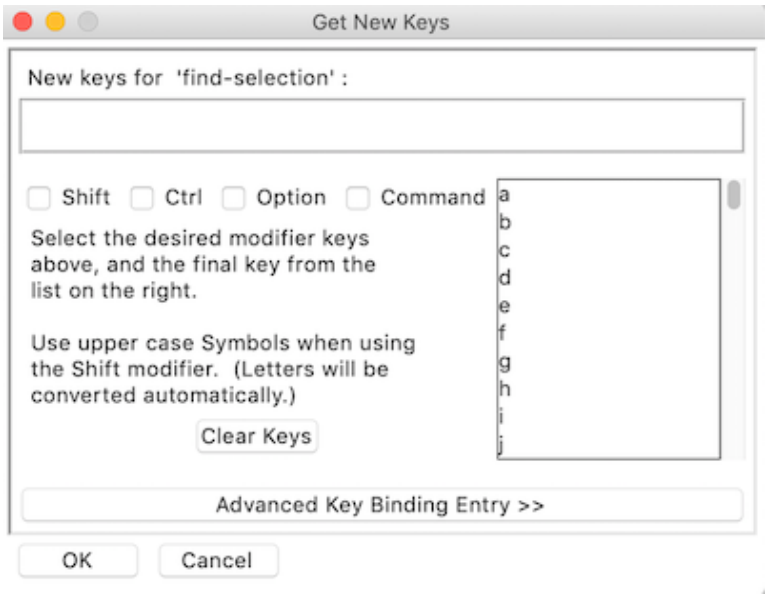
Сочетания клавиш перечислены в алфавитном порядке по действию. Они перечислены в формате *Action — Shortcut*, где *Action* — это

то, что произойдет, когда вы нажмете комбинацию клавиш в *Shortcut*. Если вы хотите использовать встроенный набор ключей, выберите сопоставление, соответствующее вашей операционной системе. Обратите особое внимание на различные клавиши и убедитесь, что они есть на вашей клавиатуре!

Creating Your Own Shortcuts

Настройка сочетаний клавиш очень похожа на настройку цветов подсветки синтаксиса. К сожалению, IDLE не позволяет устанавливать пользовательские сочетания клавиш из файла. Вы должны создать собственный набор ярлыков на вкладке *Keys*.

Выберите одну пару из списка и нажмите *Get New Keys for Selection*. Появится новое окно:

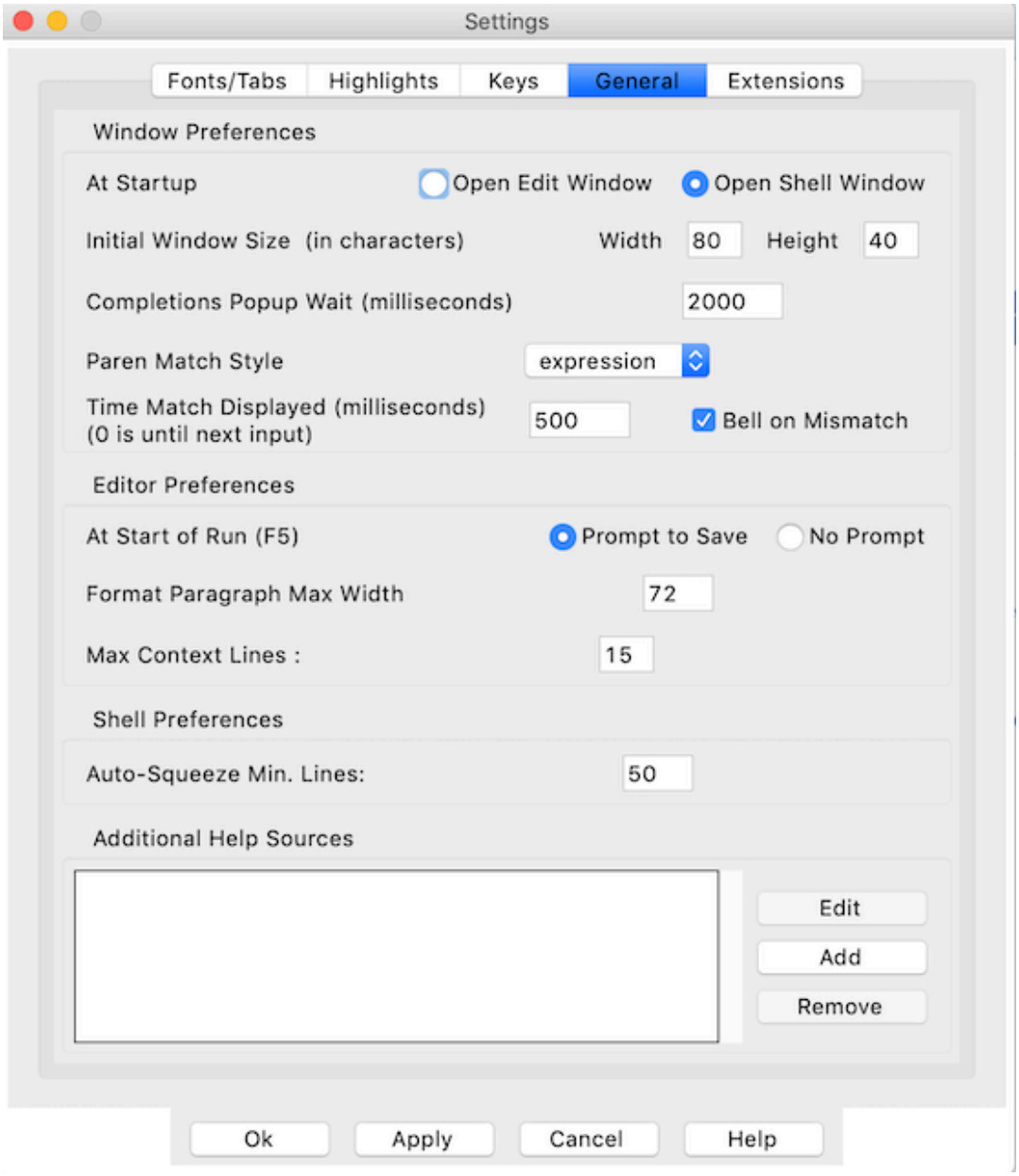


Создание пользовательских клавиш быстрого доступа

Здесь вы можете использовать флажки и меню прокрутки, чтобы выбрать комбинацию клавиш, которые вы хотите использовать. Вы можете выбрать *Advanced Key Binding Entry >>*, чтобы вручную ввести команду. Обратите внимание, что это не может забрать ключи, которые вы нажимаете. Вы должны буквально ввести команду, как вы видите, он отображается для вас в списке.

General

Четвертая вкладка окна настройки – это место для небольших общих изменений. Вкладка общих настроек выглядит следующим образом:



Общие настройки IDLE

Здесь вы можете настроить такие параметры, как размер окна и то, открывается ли оболочка или редактор файлов сначала при запуске Python IDLE. Большинство вещей в этом окне не так интересно менять, поэтому вам, вероятно, не нужно будет много возиться с ними.

Дополнения

Пятая вкладка окна настройки позволяет добавлять расширения в Python IDLE. Расширения позволяют добавлять новые, удивительные функции в Редактор и окно интерпретатора. Вы можете скачать их из интернета и установить их прямо в Python IDLE.

Чтобы посмотреть, какие расширения установлены, выберите *Options → Configure IDLE — & gt; Extensions*. В интернете есть масса расширений, о которых вы можете прочитать здесь, найти, которые вам нравятся и добавьте их в Python IDLE!

Заключение

В этом туториале вы изучили все основы использования **IDLE** для программирования на Python. Теперь вы знаете, что такое Python IDLE и как его можно использовать, как работать с файлами и настраивать Python IDLE по своему вкусу.

Здесь раскрыты следующие темы:

- Работа с оболочкой Python IDLE;
- Использование Python IDLE в качестве редактора файлов;
- Улучшение процесса разработки с помощью функций, которые помогут быстрее создавать код;
- Отладка кода, просмотр ошибок и исключений;
- Настройка Python IDLE по своему вкусу.

Теперь вы вооружены новым инструментом, который позволит вам продуктивно писать код Pythonic и сэкономит вам бесчисленные часы в будущем. Удачного программирования!