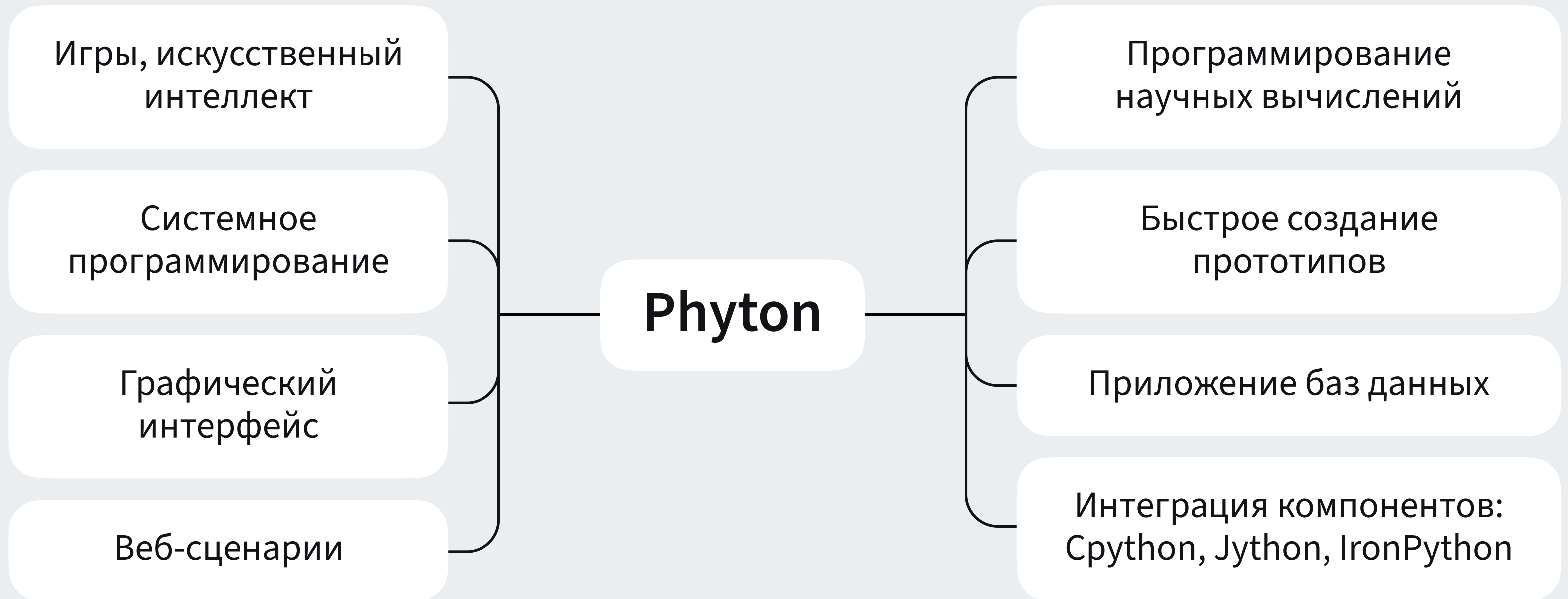


Занятие 1

# Введение в программу. Настройка IDE



**o Phython**



**Python** — это высокоуровневый язык программирования общего назначения. Его можно использовать в разных сферах: для разработки веб и мобильных приложений, обработки данных, написания системных скриптов и т.д.

Python был создан в конце 80-х голландским программистом **Гвидо ван Россумом**.

Python отличается чистым синтаксисом и хорошо структурированным кодом, который легко читается.



Согласно рейтингам  
**RedMonks** за июнь 2022  
года, Python является  
первым по популярности  
языком  
программирования  
в мире.



Python легко изучить  
и с ним легко  
работать.



Сотни библиотек  
и фреймворков.



Python гибок  
и универсален.

**о нашем курсе**

# обозначения в курсе



важная информация, запомни!



вопросы по изученному материалу



время практики



повторим пройденный материал



домашнее задание



время подумать и высказать мысли



# правила поведения на занятии



не флудить в чате



смело просить помощи, если что-то не понятно



относиться к остальным участникам с уважением



не отвлекаться на посторонние вещи

# Какие модули изучим?

Программи-  
рование  
на Python

ООП,  
алгоритмы  
и структуры  
данных

Проекты WEB  
и API

Модуль Flask.  
Проект

# Модуль 1: Программирование на Python



IDLE



настройка среды IDLE



назначение модуля PyQt



основные функции модуля PyQt



существующий набор виджетов  
графического интерфейса



стили виджетов



доступ к базам данных с помощью SQL  
(ODBC, MySQL, PostgreSQL, Oracle)



QScintilla, основанный на Scintilla виджет  
текстового редактора



поддержка интернационализации (i18n)



парсер XML



поддержка SVG



интеграция с WebKit, движком  
рендеринга HTML



поддержка воспроизведения  
видео и аудио

# Модуль 2: ООП, алгоритмы и структуры данных



класс и экземпляр класса



поиск информации по заданному критерию



создание производного класса,  
применение экземпляров базового и производного класса



основные алгоритмы сортировки и поиска



функция isinstance

# Модуль 3: проекты WEB и API



форматы файлов (json, xml)



понятие исключения, обработка исключений



собственные исключения



работа с командной строкой (скрипты, аргументы)



периодические задачи (модуль shedule)



библиотека argparse



работа с протоколом HTTP



функционал Discord



функционал VK API

# Модуль 4: Модуль Flask. Проект



Фреймворк Flask



Обработка html-форм



Шаблоны. flask-wtf



Flask-sqlalchemy



REST-API

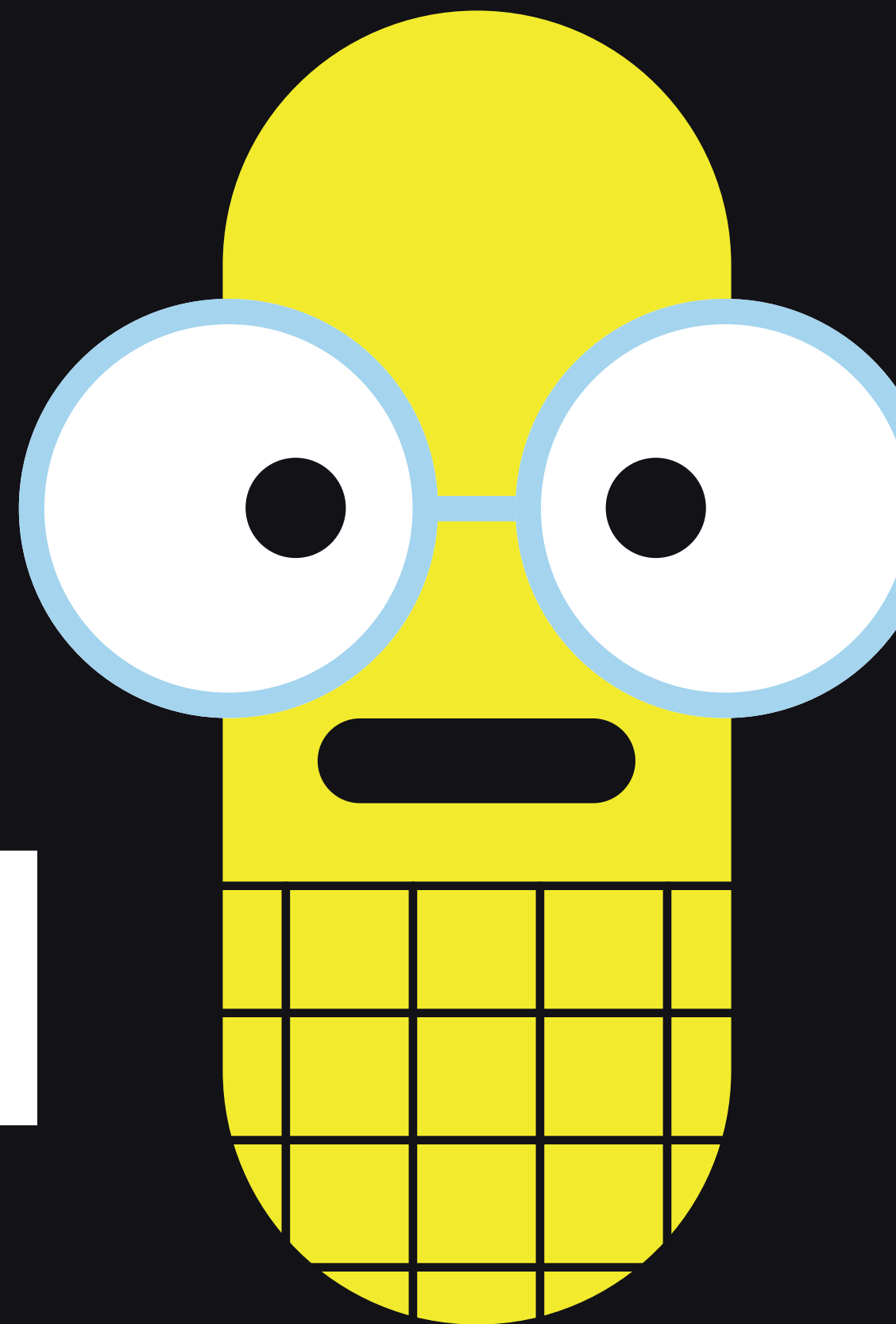


Flask-restful



Итоговый проект

теория



# Что такое Python IDLE



**Integrated Development and Learning Environment** — Интегрированная среда разработки и обучения, коротко IDLE или даже IDE.

Phyton Shell



```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>> x = 5
```

```
>>> print(x)
```

```
5
```

```
>>>
```



# Что такое Python IDLE

Лучшее место для экспериментов с кодом Python — это интерактивный интерпретатор, иначе известный как shell-оболочка.

**Оболочка Python** — отличное место для экспериментов с небольшими фрагментами кода.

# Что такое Python IDLE

Доступ к ней можно получить через терминал или командную строку своего компьютера.

Командная строка



Microsoft Windows [Version 10.0.18362.535]

(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C: \Users\webgy>

# Что такое Python IDLE

Программы Python — это файлы с расширением `.py`, которые содержат строки 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.

# Как взаимодействовать с Python напрямую, используя IDLE

Когда вы нажимаете на значок, чтобы открыть программу, оболочка — это первое, что вы видите:

File Edit Shell Debug Options Window Help

Python 3.7.0 (v3.7.0:lb9cc5093, Jun 27 2018, 04:59:58)  
[MSC v.1914 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

# Как взаимодействовать с Python напрямую, используя IDLE

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

File Edit Shell Debug Options Window Help

Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)

[Clang 6.0 (clang-600.0.57)] on darwin

Type "help", "copyright", "credits" or "license()" for more information.

```
>>> print("Hello, from IDLE!")
```

```
Hello, from IDLE!
```

```
>>>
```

# Открыть файл



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

File Edit Shell Debug Options Window Help



```
Python 3.7.0 (v3.7.0:lb9cc5093, Jun 27 2018, 04:59:58)
[MSC v.1914 64 bit (AMD64)] on win32
```

```
Type "copyright", "credits" or "license()" for more information.
```

```
>>>
```

# Редактирование файла

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



Имя файла, который вы редактируете



Полный путь к папке, где вы можете найти этот файл  
на вашем компьютере



Версия Python, которую использует IDLE

# Редактирование файла

В правом нижнем углу окна также есть две цифры:

**Ln:**

показывает номер строки,  
на которой находится  
курсор.

**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.



# Редактирование файла

Имя файла, отображаемое в верхней части окна IDLE, окружено звездочками. Это означает, что в вашем редакторе есть несохраненные изменения.

Убедитесь, что вы сохранили свой файл с расширением **.py**, чтобы подсветка синтаксиса была включена.

# Редактирование файла

File Edit Shell Debug Options Window Help



```
user_input = input("What is your name? ")

if user_input == "Python":
    print("Welcome to IDLE!")
else:
    print("Welcome to Python!")

print("This statement is an unsaved change!")
```

# Выполнение файла

Чтобы выполнить файл в режиме IDLE, просто нажмите клавишу **F5** на клавиатуре. Вы также можете выбрать **Run → Run Module** в строке меню.

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.

# Как улучшить рабочий процесс

**Авто отступ**

**Дополнение кода  
и подсказки**

**Код в контексте**

# Режим отладки интерпретатора

Если вы хотите запустить свой код с помощью встроенного отладчика, то вам нужно будет включить эту функцию. Для этого выберите **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.

# Режим отладки интерпретатора

Будьте осторожны — кнопки реверса нет! Во время отладки своей программы можно сделать шаг только вперед.

В окне отладки вы увидите четыре флажка:

## Globals

глобальная  
информация о вашей  
программе

## Locals

локальная информация  
о вашей программе во  
время выполнения

## Steck

функции, которые  
выполняются во время  
выполнения

## 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.

degub\_example.py – C:/Users/davea/Deskt...

```
for i in range(1, 4):
```

```
    j = i * 2
```

```
    print(f"i is {i} and j is {j}")
```

# 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 в строке меню:

**Debug → Go to file/line**



# Настройка Python IDLE

5 пользовательских настроек Python IDLE:

**Fonts / Tabs**

Шрифты /  
Закладки

**Highlights**

Подсветка кода

**Keys**

Клавиши

**General**

Общие

**Extensions**

Расширения

# 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:



# 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:



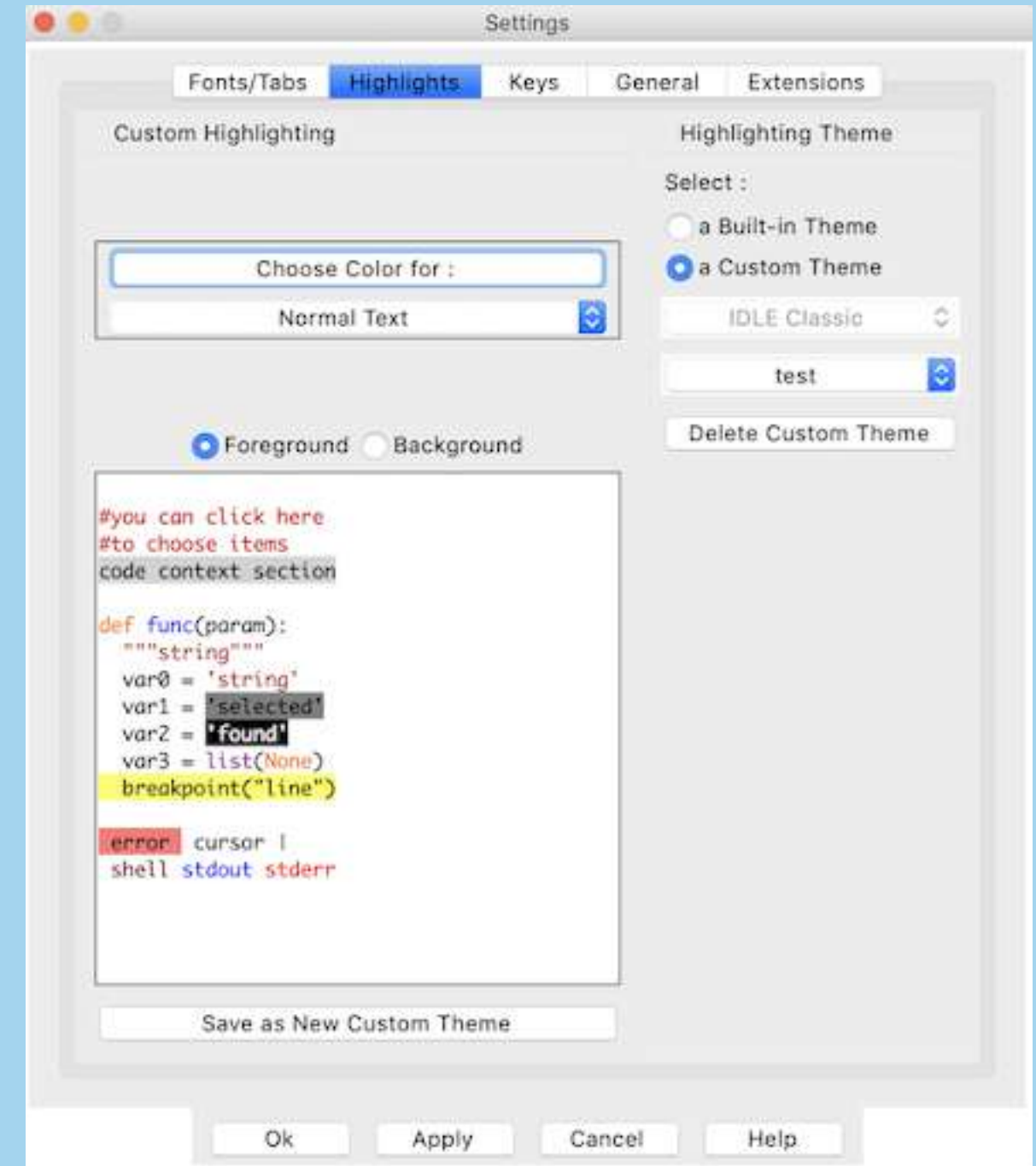
IDLE Day



IDLE Night



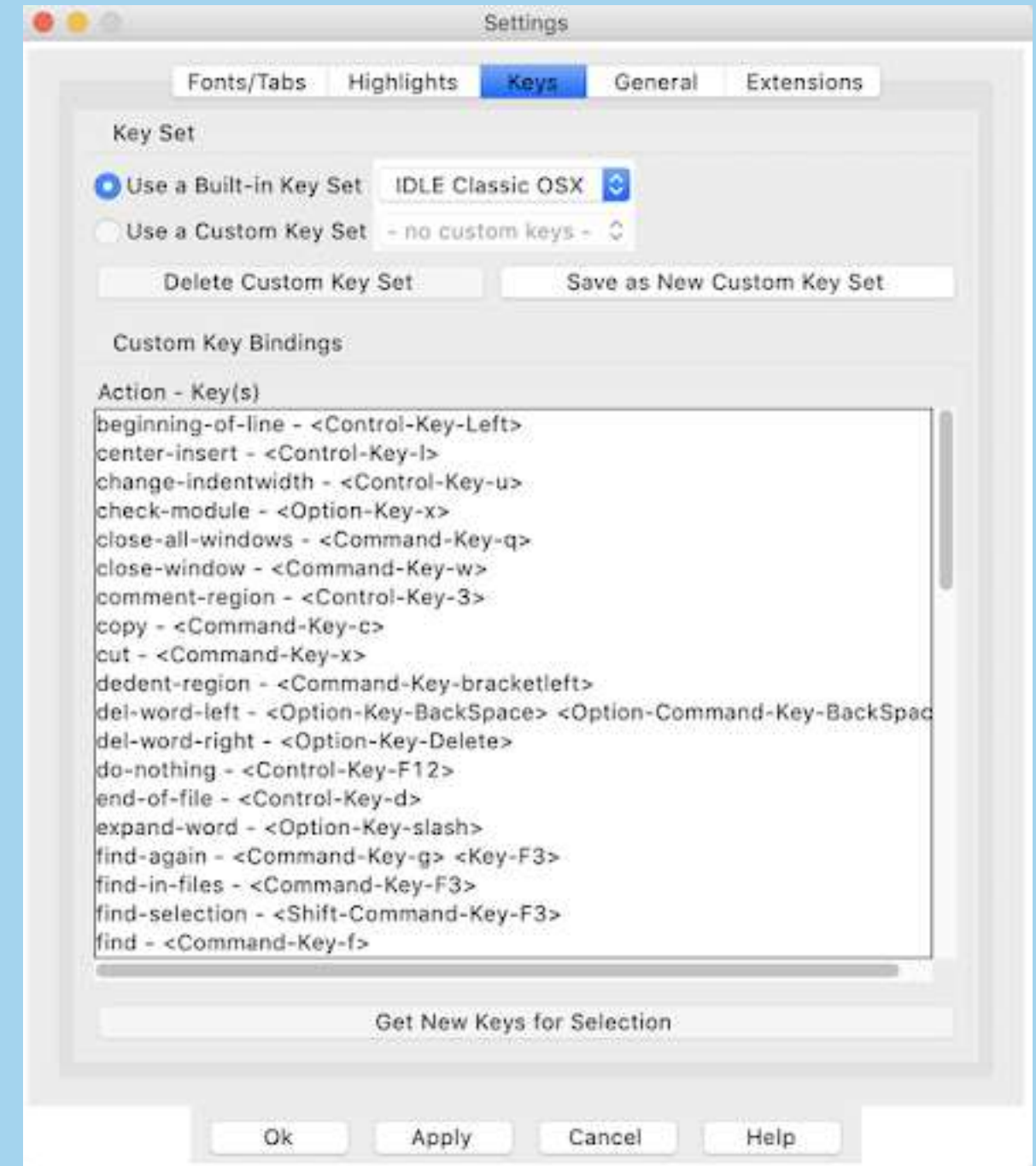
IDLE New



# Keys

Третья вкладка настройки позволяет сопоставить **различные нажатия клавиш с действиями**, также известными как `shortcut`.

Это жизненно важный компонент вашей производительности, когда вы используете IDE. Вы можете использовать свои собственные сочетания клавиш или те, которые встроены в IDLE.



# General

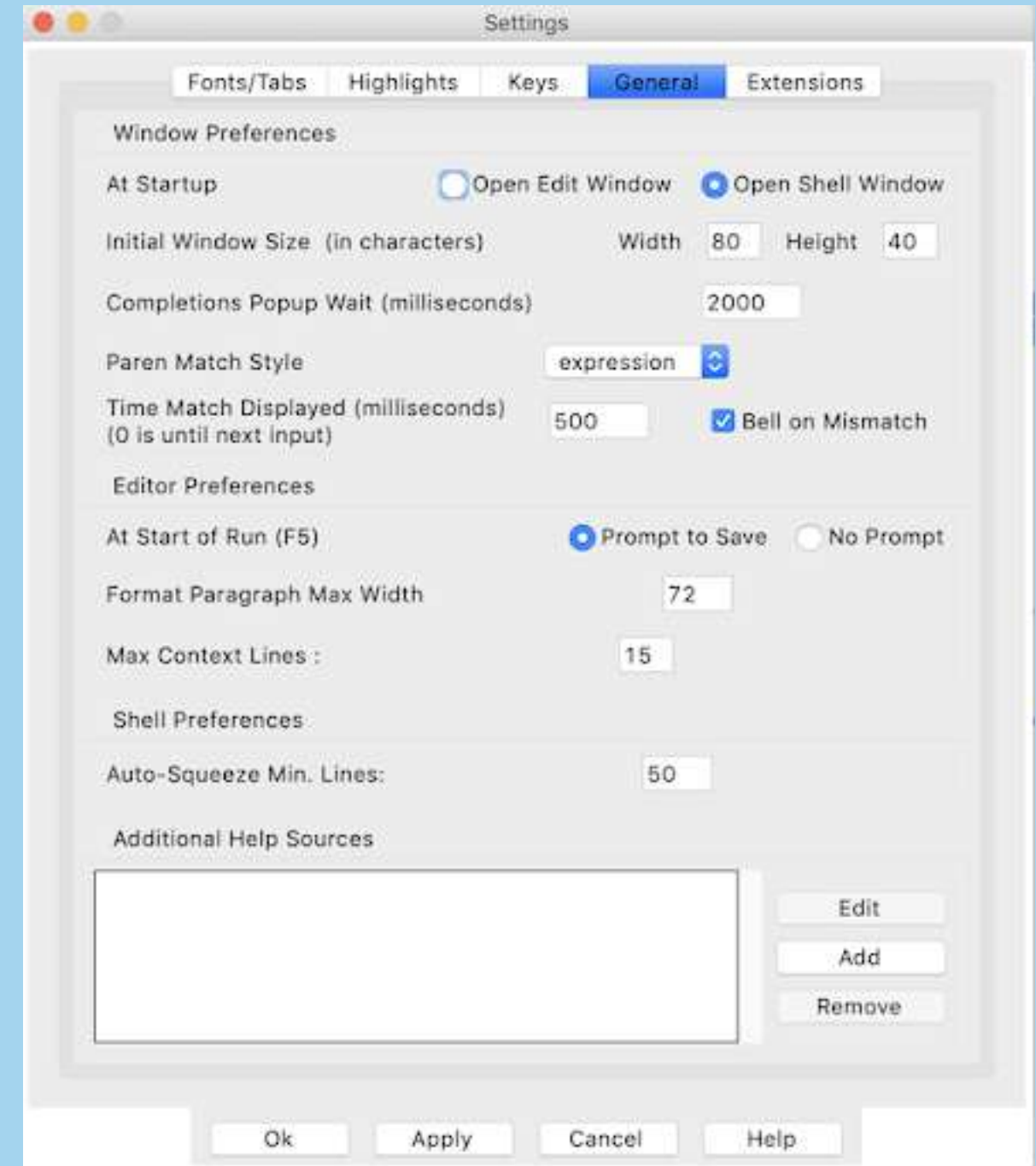
Четвертая вкладка окна настройки — это место для **небольших общих изменений**. Здесь вы можете настроить такие параметры, как:



размер окна,



открывается ли оболочка или редактор файлов сначала при запуске Python IDLE.



# 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.