

аксиома.ГИС

**Руководство
разработчика**

Этот документ перечисляет средства программирования в среде Аксиомы.ГИС.

В комплект документации Аксиомы.ГИС входят:

- Руководство пользователя
- Установка и активация
- **Руководство разработчика** (настоящий документ)
- Создание собственных проекций
- Редактор стилей линий (настоящий документ)

Последние обновлённые версии Аксиомы.ГИС и документацию можно скачать с сайта:

www.axioma-gis.ru

На этом сайте вы также можете узнать последние новости, получить ответы на свои вопросы.

Адрес технической поддержки:

support@axioma-gis.ru

Пишите нам, мы будем благодарны за предложения по развитию Аксиомы.ГИС.

Желаем вам успехов!

Команда разработчиков ООО «ЭСТИ»

Содержание

О программировании на языке Python	5
Средства Python в Аксиоме.ГИС	6
Окно Python.....	6
Настройки.....	9
Модули	10
Консоль Axioma.Runtime	13
Плагины на языке Python	16
metadata.json.....	16
__init__.py.....	17

О программировании на языке Python

Внимание:

Этот документ предназначен для программистов и опытных пользователей.

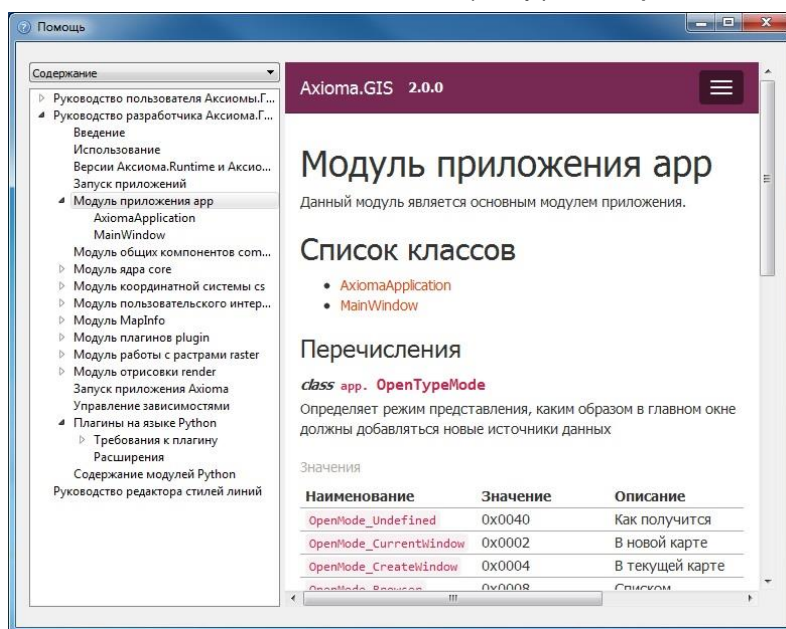
В среде Аксиома.ГИС вы можете запускать собственные процедуры и программы расширяющие ее возможности за пределы стандартной функциональности.

Для этого Аксиома.ГИС предоставляет API на языке Python, с помощью которого можно создавать собственные приложения, обращаться к внутреннему функционалу Аксиомы.ГИС и использовать библиотеку PyQt5, а также любые другие сторонние библиотеки на языке Python.

Приложения для Аксиомы.ГИС одинаково работают на платформах: Windows, Linux, macOS и Android.

Для разработки вы можете пользоваться следующими источниками информации:

- настоящее руководство, описывающее средства программирования на языке Python в среде Аксиома.ГИС
- руководства PyQt5 и Python 3, описывающие синтаксис языка Python
- Справочная система Аксиомы.ГИС (Help), содержащая описание классов API

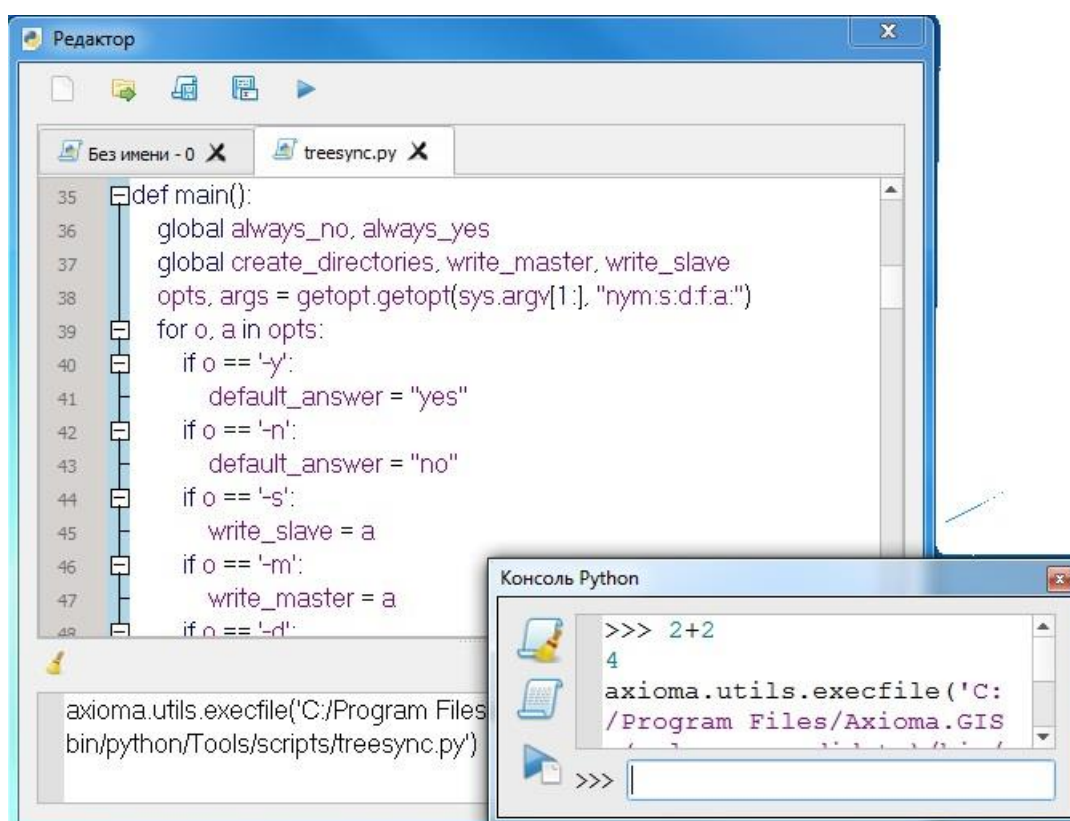


Средства Python в Аксиоме.ГИС

Окно Python

Чтобы открыть окна поддержки языка Python, выполните команду **Панели > Консоль Python**.

Эта команда открывает окно **Консоль Python**.



В левой части консоли расположены три кнопки:



Очистить консоль

Удаляет все содержимое в окне консоли.



Открыть редактор

Открывает окно редактора (см. ниже).



Запустить из файла

Позволяет запускать программные модули из файлов с расширением PY.

В консоли вы можете ввести любую команду на языке Python в окошке >>> и нажать ENTER. В верхней части консоли будет продублирована ваша команда, а также результат ее выполнения интерпретатором Python, либо вы увидите сообщение об ошибке, если ввели некорректную с точки зрения синтаксиса Python команду.

Панель снабжена собственным редактором, в котором вы можете открывать сценарии (программы) Python, редактировать их, сохранять и запускать на исполнение. Для каждого сценария создается собственное окно с вкладкой.

Кнопки в инструментальной панели редактора выполняют следующие действия:

**Новый файл**

Открывает новую вкладку для сценария Python.

**Открыть**

Открывает сценарий Python из файла с расширением PY.

**Сохранить**


Сохраняет сценарий Python в файле с расширением PY.

**Сохранить как**

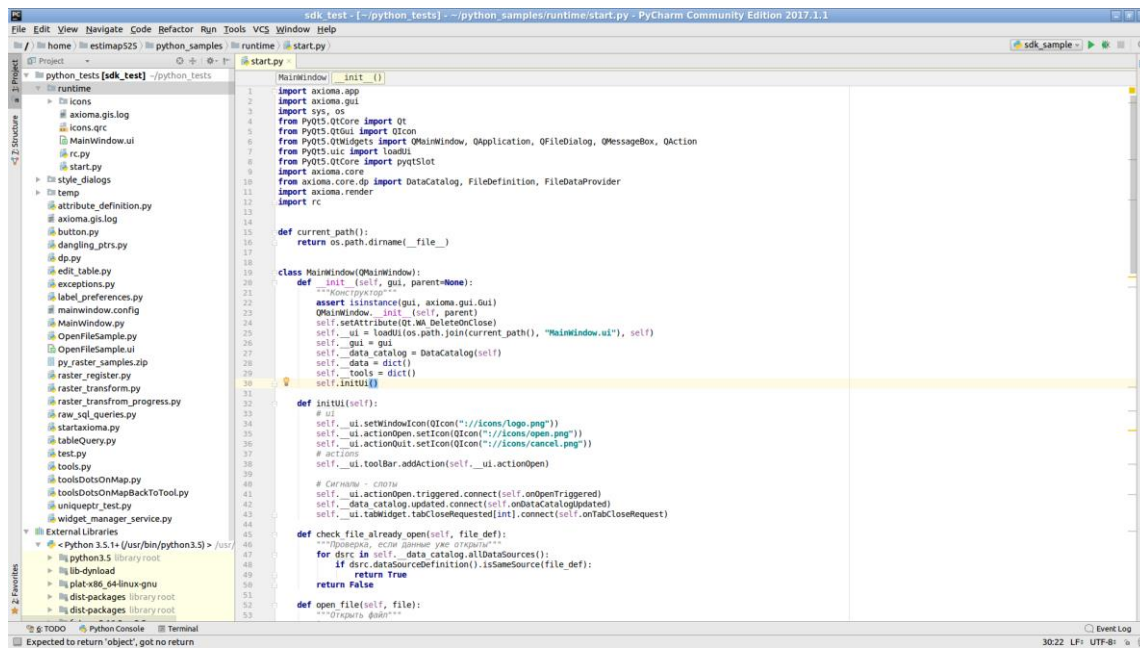
Сохраняет сценарий Python с расширением PY с другим именем.

**Запустить сценарий**

Запускает сценарий из данной вкладки. В нижней части окна редактора показывается результат выполнения сценария и выводятся сообщения об ошибках.

Нажав на кнопку , вы можете очистить окно с результатами.

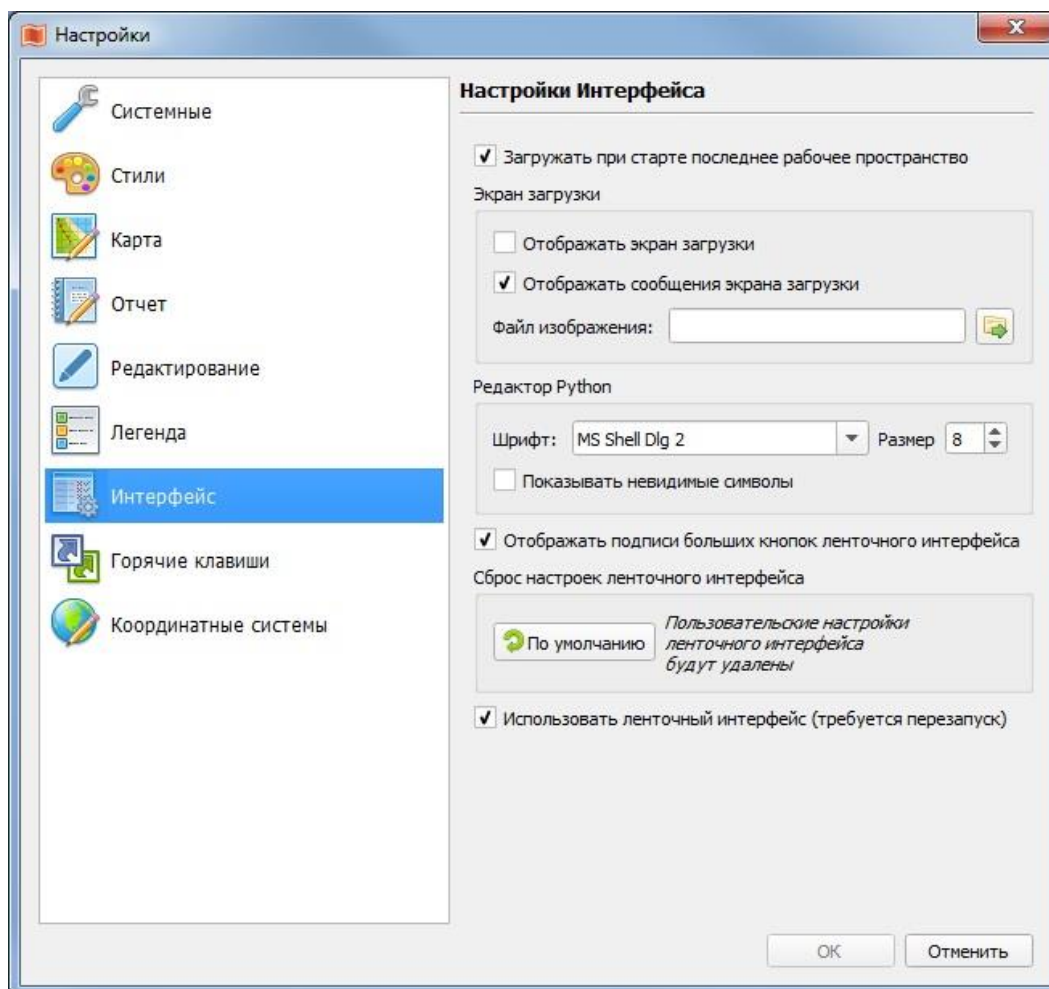
В редакторе вы можете отлаживать сравнительно небольшие тексты. Для более сложных задач, требующих отладки, рекомендуется использовать IDE, например, PyCharm.



```
1  MainWindow.__init__()
2  import axioma.app
3  import axioma.gui
4  import sys, os
5  from PyQt5.QtCore import Qt
6  from PyQt5.QtGui import QIcon
7  from PyQt5.QtWidgets import QMainWindow, QApplication, QDialog, QMessageBox, QAction
8  from PyQt5.QtCore import pyqtSlot
9  import axioma.core
10 from axioma.core.dp import DataCatalog, FileDefinition, FileDataProvider
11 import axioma.render
12 import rc
13
14 def current_path():
15     return os.path.dirname(__file__)
16
17
18 class MainWindow(QMainWindow):
19     """Мок PyQt5"""
20     def __init__(self, gui, parent=None):
21         assert isinstance(gui, axioma.gui.Gui)
22         QMainWindow.__init__(self, parent)
23         self.setAttribute(Qt.WA_DeleteOnClose)
24         self._ui = loadUi(os.path.join(current_path(), "MainWindow.ui"), self)
25         self._gui = gui
26         self._data_catalog = DataCatalog(self)
27         self._data = dict()
28         self._tools = dict()
29         self._initUI()
30
31     def initUI(self):
32         # ui
33         self._ui.setWindowTitle(QIcon(":/icons/logo.png"))
34         self._ui.actionOpen.setIcon(QIcon(":/icons/open.png"))
35         self._ui.actionQuit.setIcon(QIcon(":/icons/cancel.png"))
36         # actions
37         self._ui.toolbar.addAction(self._ui.actionOpen)
38
39         # События - слоты
40         self._ui.actionOpen.triggered.connect(self._onOpenTriggered)
41         self._data_catalog.updated.connect(self._onDataCatalogUpdated)
42         self._ui.tabWidget.tabCloseRequested[int].connect(self._onTabCloseRequest)
43
44     def check_file_already_open(self, file_def):
45         """Проверка, если данные уже открыты"""
46         for dsrc in self._data_catalog.allDataSources():
47             if dsrc.dataSourceDefinition().isSameSource(file_def):
48                 return True
49         return False
50
51     def open_file(self, file):
52         """Открыть файл"""
53
```


Настройки

Команда Аксиомы.ГИС Основные > Параметры > Интерфейс открывает диалог, в котором вы можете настроить некоторые параметры окна **Консоль Python** и редактора.



Раздел Редактор Python

Шрифт

Шрифт для окна редактора Python.

Размер

Размер букв в окне редактора Python.

Показывать невидимые символы

Если флажок установлен, то в окне редактора Python пробелы показываются в виде точек в середине строки.

Модули

В Аксиоме.ГИС вы можете загружать модули (плагины), разработанные вами или сторонними разработчиками на языке Python.

Модули могут создавать свои закладки на ленте и инструментальные кнопки.

Эти программные модули можно:

- загрузить во время работы сеанса Аксиомы.ГИС;
- сделать загружаемыми автоматически при старте Аксиомы.ГИС;

Вы можете отключить автоматическую загрузку модулей, также добавлять папки, в которых хранятся модули, доступные Аксиоме.ГИС.

Модули, доступные Аксиоме.ГИС с момента старта программы, находятся в папке *C:\Program Files\Axioma.GIS\bin\python_plugins*. Вы можете добавлять другие папки. Стандартная папка для хранения скриптов Python задается в диалоге **Основные > Параметры > Системные**.

В настоящее время доступны примеры загружаемых модулей:

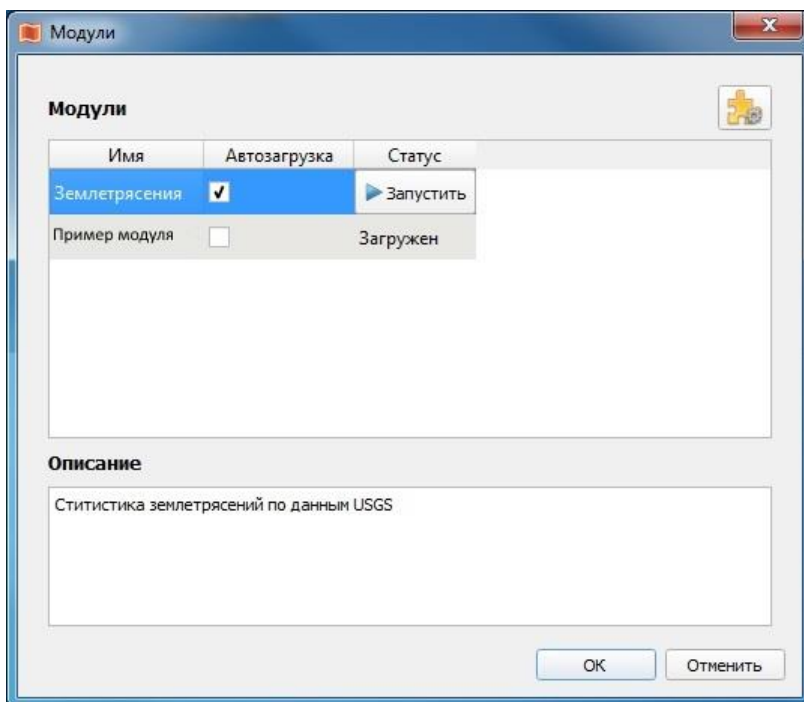
Пример модуля – демонстрирует использование инструментальных кнопок и создание новой вкладки на ленте.

Землетрясения – показывает в окне карты обозначение землетрясений по данным USGS.

Чтобы открыть список доступных модулей:

1. Выберите **Модули** на ленте **Основные**.

Откроется диалог **Модули**.



2. Доступные модули представлены в разделе **Модули**. В разделе Описание приведена информация о выбранном модуле.
3. Выберите мышкой один из модулей.
4. Выберите режимы работы выбранного модуля в полях **Автозагрузка** и **Статус** и нажмите **ОК**.

Нажатие на кнопку **Отменить** закрывает диалог без изменений.

Чтобы загрузить модуль:

1. Выберите **Модули** на ленте **Основные**.

Откроется диалог **Модули**.

2. Выберите один из модулей.
3. Нажмите кнопку **Запустить** в поле **Статус**.

Модуль будет загружен и запущен. В поле **Статус** появится надпись **Загружен**.

Если модуль добавляет кнопки или вносит другие изменения в ленту, то они проявятся немедленно.

4. Закройте диалог нажатием на **ОК** или **Отменить**.

Чтобы сделать модуль автоматически загружаемым:

1. Выберите **Модули** на ленте **Основные**.

Откроется диалог **Модули**.

2. Выберите один из модулей.
3. Установите флажок в поле **Автозагрузка** в записи.
4. Закройте диалог нажатием на **ОК** или **Отменить**.
5. Закройте Аксиому.ГИС и откройте снова.

Модуль будет запущен при старте программы. Если модуль добавляет кнопки или вносит другие изменения в ленту, то они будут показаны на ленте.

Чтобы исключить модуль из автозагрузки:

1. Выберите **Модули** на ленте **Основные**.

Откроется диалог **Модули**.

2. Выберите один модулей.
3. Сбросьте флажок в поле **Автозагрузка** в записи.
4. Закройте диалог нажатием на **ОК** или **Отменить**.
5. Закройте Аксиому.ГИС и откройте снова.

Модуль не будет запускаться при старте программы. Если модуль добавлял кнопки или вносил другие изменения в ленту, то они не будут показаны на ленте.

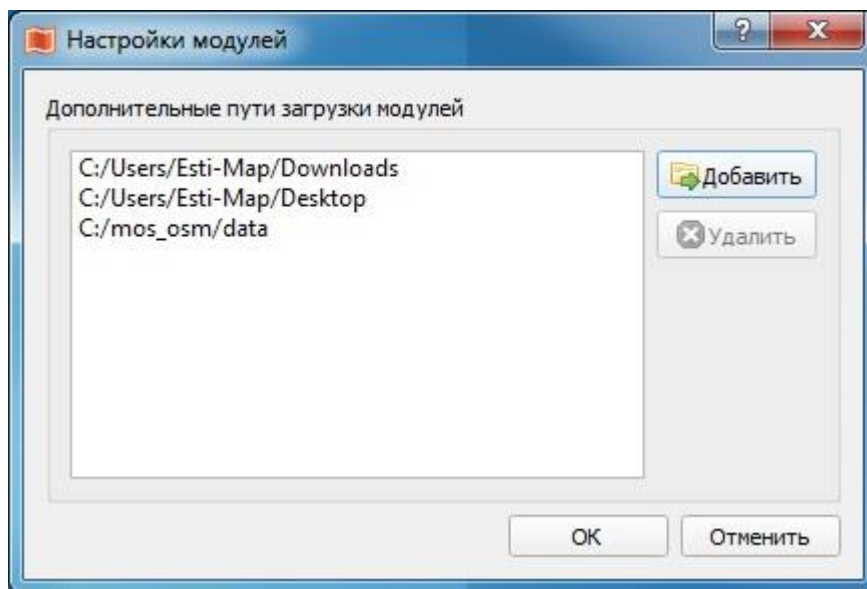
Чтобы добавить модули:

1. Выберите **Модули** на ленте **Основные**.

Откроется диалог **Модули**.

2. Нажмите на кнопку .

Откроется диалог **Настройки модулей**.



3. Чтобы добавить папку, в которой размещаются модули, доступные в Аксиоме.ГИС, нажмите **Добавить** и выберите папку в стандартном диалоге выбора папок.
4. Чтобы удалить папку, в которой размещаются доступные в Аксиоме.ГИС модули, выберите ее в разделе **Дополнительные пути загрузки модулей** и нажмите **Удалить**.
5. Закройте Аксиому.ГИС и откройте снова.

Список папок будет считан заново и Аксиоме.ГИС будут доступны только те модули, которые хранятся в папках из обновленного списка.

Консоль Axioma.Runtime

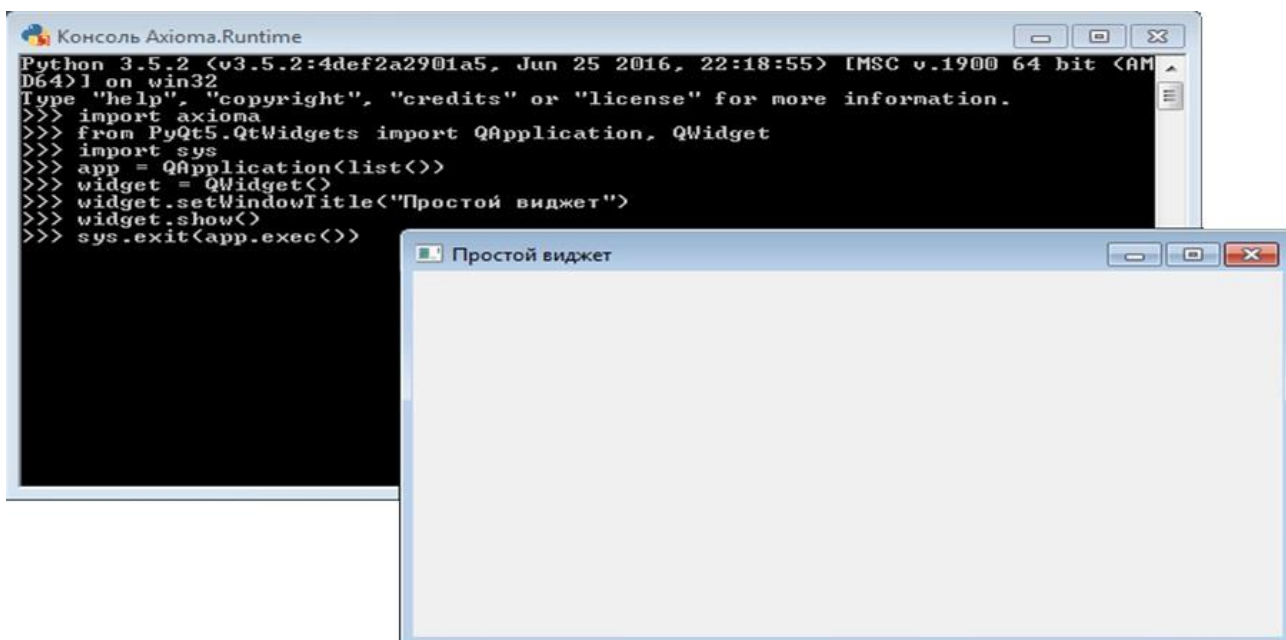
Существует две версии поставки программного продукта Аксиома: Аксиома.ГИС и Аксиома.Runtime. Аксиома.ГИС полностью включает в себя Аксиома.Runtime и дополнительно имеет:

- стандартный графический интерфейс (главное окно);
- Python API стандартного графического интерфейса.

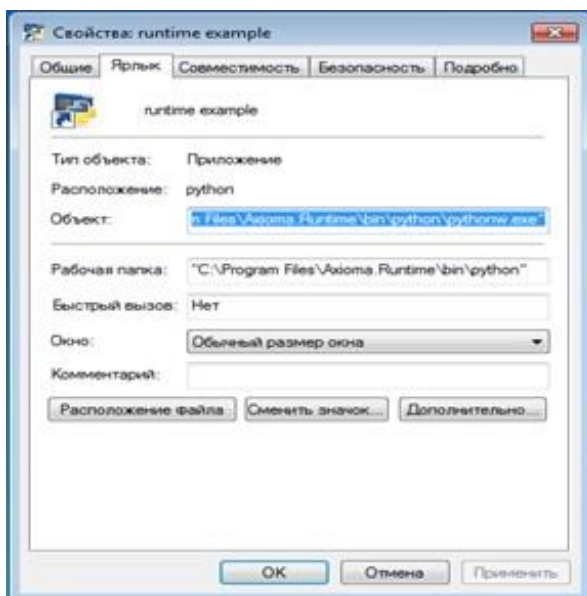
Открыть окно консоли Axioma.Runtime можно, выбрав в стартовой панели Windows **Пуск** иконку консоли:



Пример использования консоли Axioma.Runtime:

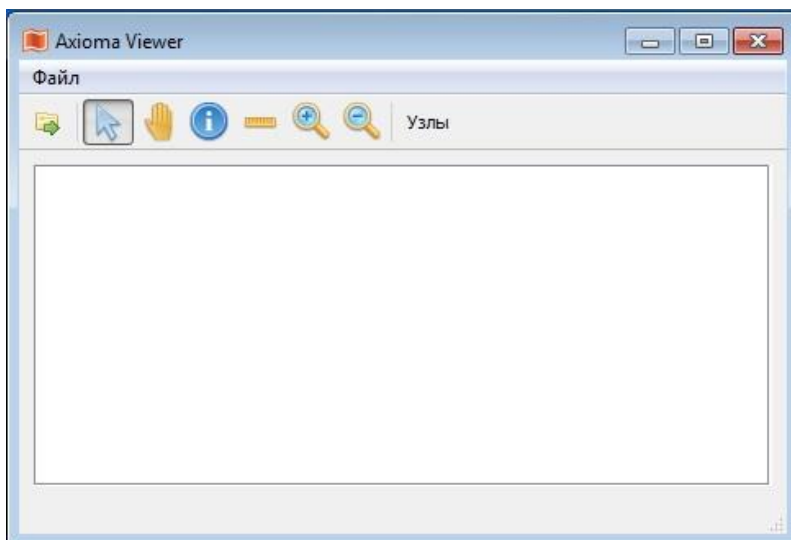


Для запуска программы, хранящейся в отдельном файле, в этой консоли, вы можете создать ярлык на рабочем столе, в свойствах которого указан дисковый маршрут к исполняемому файлу Axioma.Runtime, а также к папке, в которой находится файл на языке Python start.py, например:



В этом диалоге в поле **Объект** указан дисковый маршрут в файлу python.exe и к файлу со сценарием: "C:\Program Files\Axioma.Runtime\bin\python\pythonw.exe" "C:\Program Files\Axioma.Runtime\share\axioma\python_examples\runtime\start.py"

Если нажать **ОК**, то следующее окно:



Плагины на языке Python

В дистрибутив Аксиомы.ГИС входят примеры плагинов, исходный код которых можно посмотреть в папке `bin/python_plugins`

Каждый плагин представлять собой папку, содержащую как минимум два файла: `metadata.json` и `__init__.py`.

metadata.json

Это файл с метаданными в формате JSON.

Корневым элементом этого файла должен быть объект, который может иметь следующие поля:

Название	Тип	Описание
<code>id</code>	Строка	Уникальный идентификатор плагина. Для обеспечения уникальности рекомендуется использовать способ наименования аналогичный пространствам имен <code>java</code> , например, <code>ru.mycompanyname.HelloWorldPlugin</code>
<code>displayName</code>	Строка	Имя плагина, отображаемое пользователю
<code>description</code>	Строка	Описание предназначения плагина
<code>defaultAutoload</code>	Логический	Флаг, определяющий, загружается ли плагин по умолчанию
<code>hidden</code>	Логический	Флаг, определяющий, показывать ли плагин пользователю в настройках

Строковые параметры, предназначенные для отображения в интерфейсе, могут быть локализованы. Для этого нужно использовать JSON-объект, который содержит локализованные строки в полях, соответствующих названию локали.

Например: `{"ru" : "Русский текст", "en" : "English text", "en_GB" : "Fancy English text"}`

Пример файла с метаданными:

```
{
    "id" : "ExamplePlugin",
    "displayName" : {"ru" : "Пример модуля"},
    "description" : {"ru" : "Демонстрация возможностей Python"},
    "defaultAutoload" : false
}
```

__init__.py

С точки зрения Python плагин является пакетом (package), и при его импорте интерпретатор выполняет код из файла `__init__.py`. Кроме того, плагин может содержать специальные методы:

- *init()* - вызывается при загрузке плагина, если инициализировать плагин не удалось, метод должен вернуть False, иначе True.
- *createExtensions()* - этот метод должен вернуть список расширений, которые будут загружены в приложение.

Расширения для метода *createExtensions()*:

- *RibbonExtension* - позволяет создавать вкладки и группы на панели ленточного интерфейса в окне приложения Аксиома.ГИС
- *ActionExtension* - позволяет добавлять действия (*QAction*) в интерфейс
- *ToolControllerExtension* - позволяет добавлять инструменты для работы с окнами
- *BasicToolExtension* - упрощенный вариант *ToolControllerExtension*