**DF Config**

**Languages:**

Русский (Original)

English (SOON – Google Translator)

RU

DF Config – Простая библиотека для создания простых конфигураций.

Чтобы создать конфигурацию, создайте файл с расширением *.dfc*

*Пример : file.dfc*

Для редактирования этого файла используйте любой текстовый редактор/редактор кода.

Перейдем к созданию переменной. Все переменные должны иметь названия без пробелов. Данное правило необязательно, но желательно его соблюдать для улучшения читаемости конфигурации.

Данные типа String указываются в двойных кавычках

*Пример : "Текст"*

Данные типа Integer(int), Float(float), Boolean(boolean) указываются без кавычек. В предпоследнем неполная часть отделяется точкой

*Пример : 2023*

*Пример : 3.14*

Теперь запишем саму переменную. В файле конфигурации переменная пишется в формате \_название\_

*Пример : \_variable\_*

В самом же коде переменная указывается без "\_". К коду мы перейдем позже.

Чтобы указать, чему равна переменная, необходимо сделать следующую запись:

*Переменная = значение*

Обратите внимание, что знак „=“ ОБЯЗАТЕЛЬНО отделяется пробелами с обеих сторон, а после значения не должно быть ничего. Только при соблюдении этих правил вы сможете получить корректные значения при помощи кода.

*Пример:*

*\_text\_ = "Hello World!"*

*\_a\_ = 2023*

*\_pi\_ = 3.14*

Все нужные данные записали. Теперь перейдём к коду. Сначала библиотеку необходимо импортировать. Для этого перенесите папку DFConfig(в которой находится этот файл) в папку с вашей программой. Затем добавьте в код следующую строку:

*from DFConfig import dfc*

Теперь вы можете использовать функции библиотеки DFConfig в вашем коде.

Начнём с получения(чтения) данных. Для этого нужно использовать функции типа get(получать). Функции get в DFConfig называются по принципу

*getПолное-название-типа-данных()*

В качестве аргументов нужно указать : файл, название переменной

Для простоты приведу пример. Допустим, у нас есть конфиг с переменной “text”, которая содержит «Hello World!»

*\_text\_ = “Hello world!”*

Тогда, пишем в нашем скрипте(обратите внимание, что основной скрипт находится на один уровень выше файла библиотеки) данную строку

*dfc.getString(“file.dfc”, “text”)*

Поместив данную функцию, например, в print(), получим следующий результат

*Hello world!*

Вот, как называются некоторые другие функции для получения данных:

Целое число — getInteger()

Дробное число — getFloat()

True/False – getBoolean()

Функция возвращает тип данных, соответствующий указанному в названии функции.

Теперь приступим к изменению данных. Принцип, по которому названы функции схожи с «get», но вместо него мы ставим «set». Аргументы так же немного изменились. Теперь вам нужно добавить туда новое значение переменной. Давайте изменим текст ранее созданной переменной. Пусть новым значением будет «Hi!». Для этого нужно написать:

dfc.setString(“file.dfc”, “text”, “Hi!”)

Готово. Изменения сохраняются автоматически после ввода команды