Парсинг: модуль argparse | Я.Шпора

Модуль `argparse` парсит команды в терминале — ищет нужные аргументы, даже если они введены в произвольном порядке или часть из них пропущена.

Для работы с командной строкой через модуль `argparse` нужно создать экземпляр класса `ArgumentParser` и выполнить парсинг строчки в терминале при помощи метода `parse_args()`:

```
import argparse

# Инициализация парсера аргументов с описанием.
parser = argparse.ArgumentParser(description='Вежливый скри
# Извлечение аргументов командной строки в переменную args.
args = parser.parse_args()
```

Чтобы просмотреть аргументы, с которыми может работать скрипт, используется команда `-h` или `--help`. Пример использования:

```
$ python arg_parsing.py -h
```

Аргументы добавляются при помощи метода `add_argument()`. Они могут быть именованными и позиционными. Программа определяет тип аргумента по тому, как он написан.

По договорённости позиционные аргументы прописываются без префикса `-`, например, вот так: `add_argument('name')`. Они обязательны для указания.

Аргументы с префиксом `-` — именованные и по умолчанию необязательные.

К аргументам можно добавить описание на русском языке с помощью параметра `help`, которое будет выводиться при запросе справки через команду `-h` или `--help`.

Пример:

```
import argparse
```

```
parser = argparse.ArgumentParser(description='Вежливый скри
parser.add_argument('name', help='Имя')
parser.add_argument('-s', '--surname', help='Фамилия')
args = parser.parse_args()
```

Аргументы с заданным списком значений

Можно задать допустимые значения для аргумента — через список, кортеж или результаты вызова функции. Для этого у метода `add_arguments()` есть параметр `choices`.

Пример:

```
import argparse

parser = argparse.ArgumentParser(description='Вежливый скри
parser.add_argument(
    '-c',
    '--city',
    help='Город',
    choices=['Chekhov', 'Dublin', 'Minsk', 'Simbirsk'],
)
```

Аргументы со значением True или False

Аргумент может вести себя как булева переменная, у которой лишь два значения — `True` или `False`.

Для реализации такого поведения в методе `add_arguments()` используется параметр `action` со значением `store_true`, когда при наличии аргумента нужно передать значение `True`, или `store_false`, когда нужно передать значение `False`. Чтобы передать такой аргумент, нужно ввести его длинную или короткую форму.

Пример:

```
import argparse
# Константа для работы с аргументом.
MURZIK = '=^..^=_____/'

parser = argparse.ArgumentParser(description='Вежливый скри
```

```
parser.add_argument(
    '-m',
    '--murzik',
    action='store_true',
    help=f'Отправить кота Мурзика {MURZIK}'
)
```