**Урок 6**

**Создание клавиатуры ReplyKeyboardMarkup**

На предыдущем уроке мы создали бота, который отправлял пользователю предсказание, составленное случайным образом из фраз-шаблонов.

Сегодня мы поработаем над меню: добавим для пользователей клавиатуру с быстрыми ответами. Но сначала разберёмся с видами возможных клавиатур.

**Виды клавиатур**

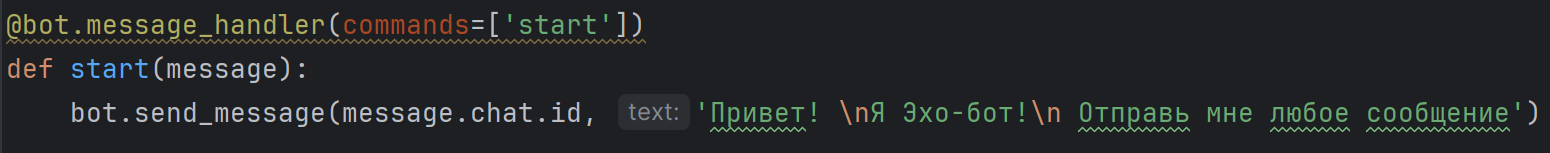
Модуль **types** позволяет создать на Python клавиатуры двух видов, отличающиеся друг от друга расположением кнопок:

* **Reply-кнопки** для шаблонных ответов, которые закрепляются вместо основной клавиатуры на экране. Часто используются в чат-ботах как меню. Создаются с помощью метода ReplyKeyboardMarkup.
* **Inline-кнопки**, связанные с сообщениями в чате. При этом пользователь видит и основную клавиатуру. Создаются с помощью метода InlineKeyboardMarkup.

Мы познакомимся с reply-кнопками на примере программы «Эхо-бота», которую мы делали на втором уроке.

**Reply-кнопки**

Ранее при получении команды **/start** наш бот представлялся как «Эхо-бот».

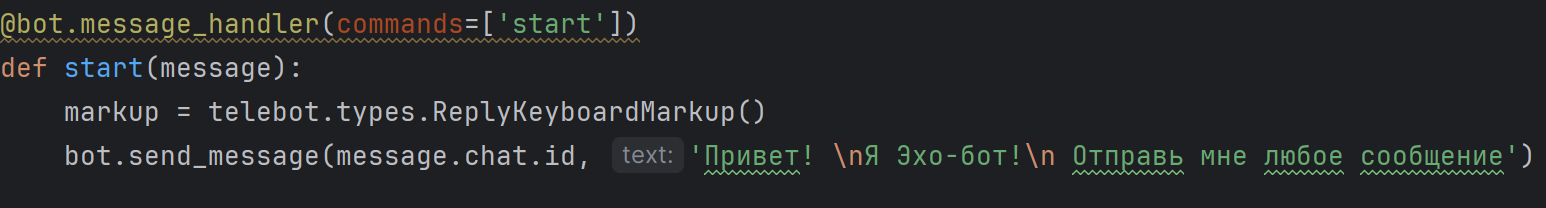


Давайте же теперь будем при получении команды **/start** будем отправлять нашему пользователю тот же самый текст, но только уже с кнопками.

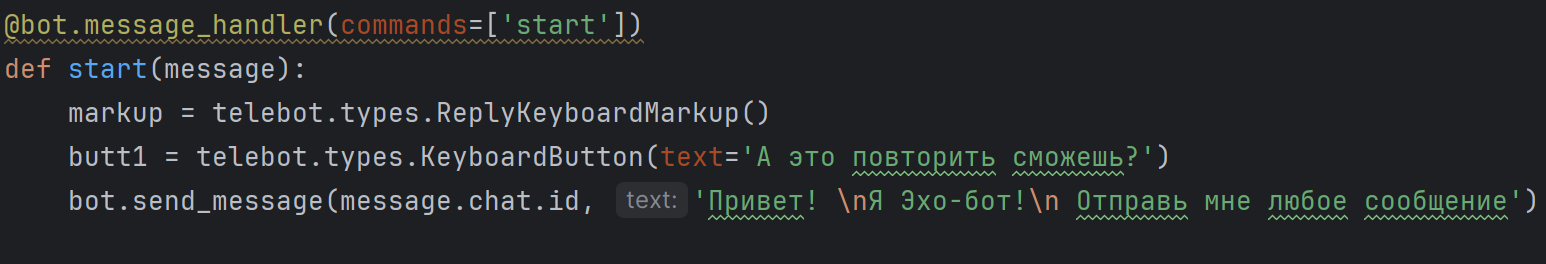
Для начала создадим некий объект, который назовем, например, **markup**, далее нам надо через библиотеку **telebot** обратиться к такому значению как **types**.

В модуле **types** есть классы **ReplyKeybordRemove** и **ReplyKeyboardMarkup** позволяют создавать и удалять клавиатуру, а класс **KeyboardButton** используется для добавления кнопок.

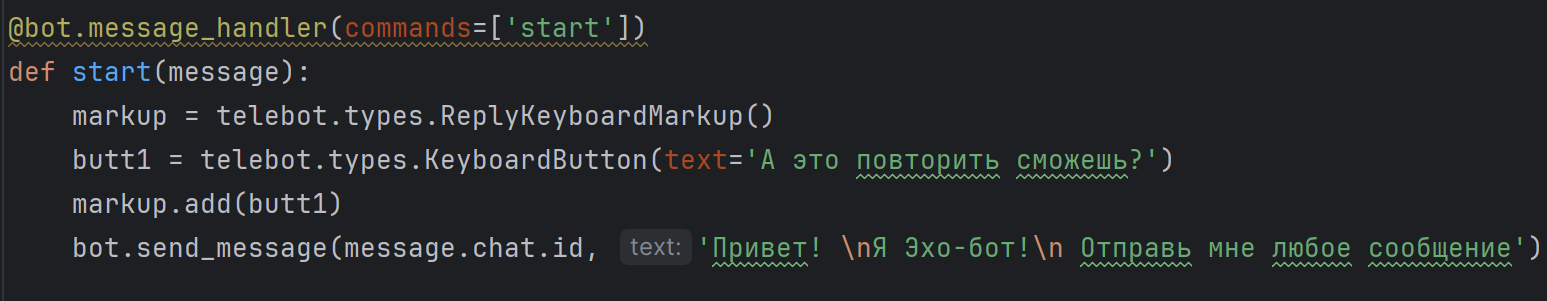
Мы будем создавать с вами встроенные кнопки, поэтому нас интересует класс, создающий клавиатуру — **ReplyKeyboardMarkup.**



Чтобы создать некую новую кнопку, мы опять обращаемся к **types**, далее говорим какую кнопку мы будем создавать — **KeyboardButton**. В качестве параметров мы должны указать текст для кнопки, например, у меня это будет «А это повторить сможешь?»

****

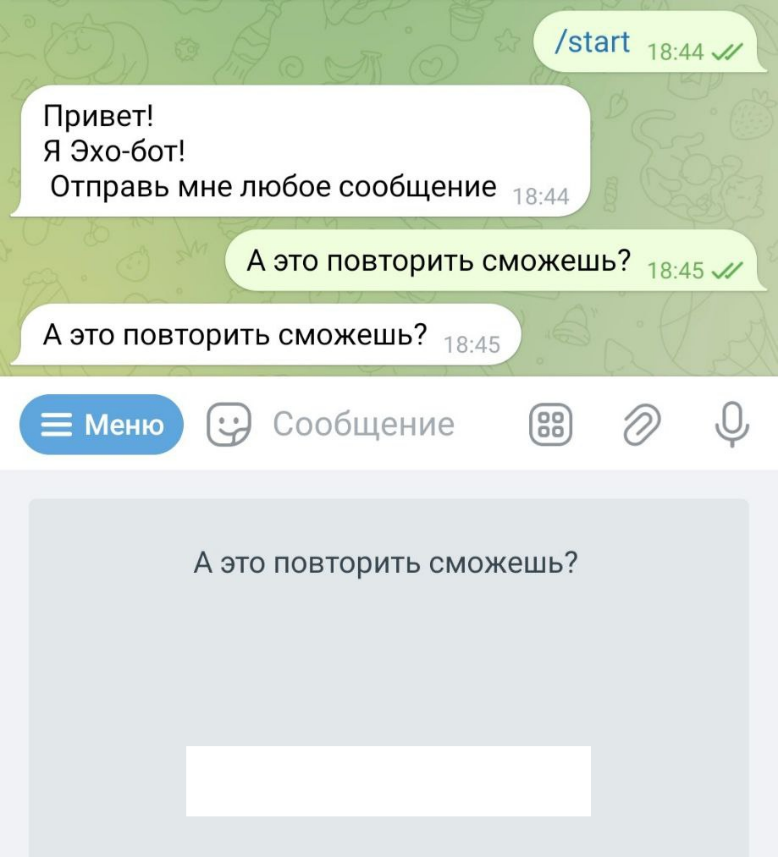
Мы создали кнопку, но не добавили её в клавиатуру. Чтобы это сделать мы обращаемся к нашему **markup**, к методу **add** и указываем кнопку.



Теперь давайте попробуем вывести нашу кнопку с сообщением. Для этого в методе **send\_message** мы передаем еще один параметр, который называется **reply\_markup** и в качестве значения передаем наш объект **markup**.

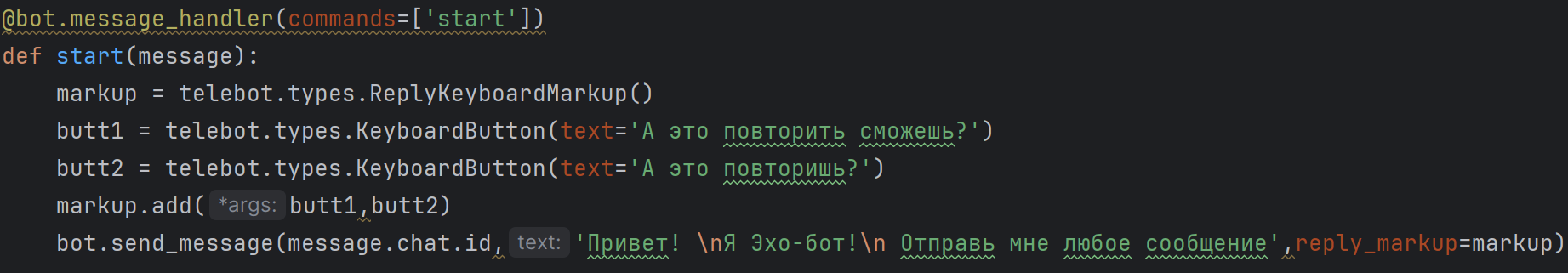


Теперь при запуске бота, когда мы введем команду **/start**, наш бот выдаст нам текст, а также нашу кнопку. Если нажать на неё, текст кнопки отправится в чат, а Telegram-бот пришлёт в ответ эту же фразу:

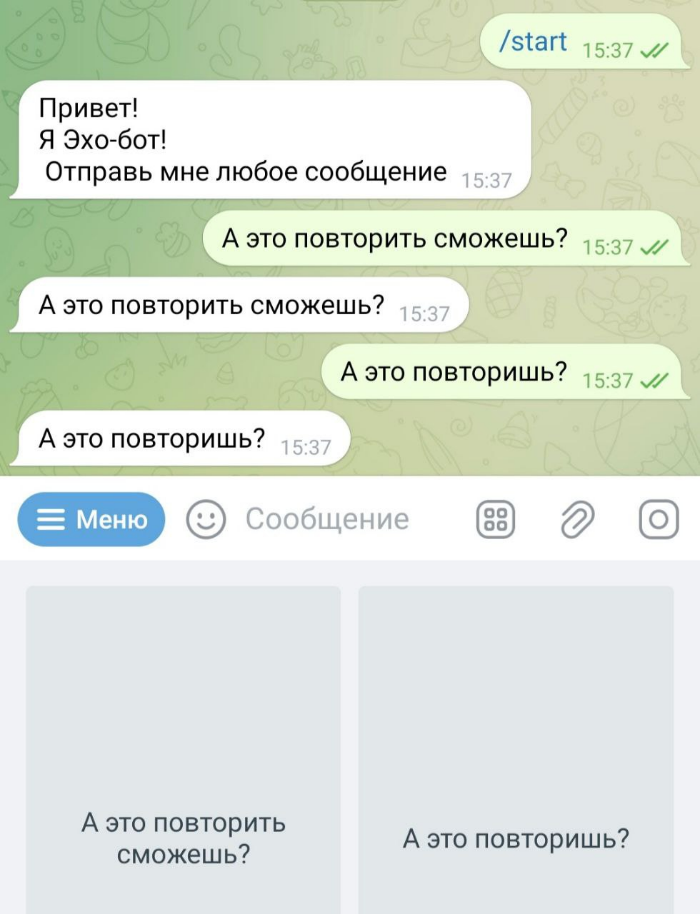


**Самостоятельная работа**

Добавьте вторую кнопку. Для этого необходимо создать вторую переменную, обратиться к классу KeyboardButton**.** Далее к методу **add** добавляем эту кнопку.



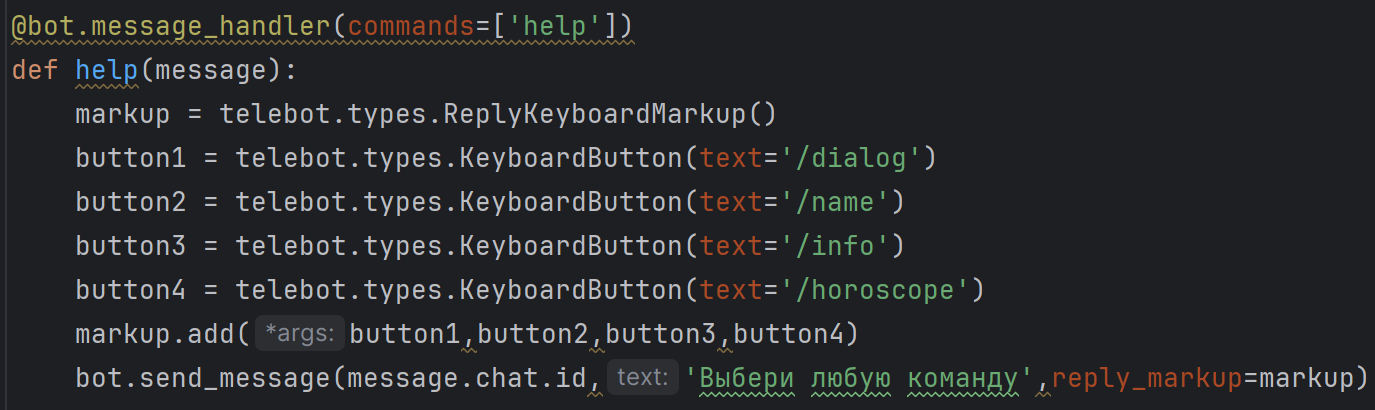
Вот что у нас получилось:

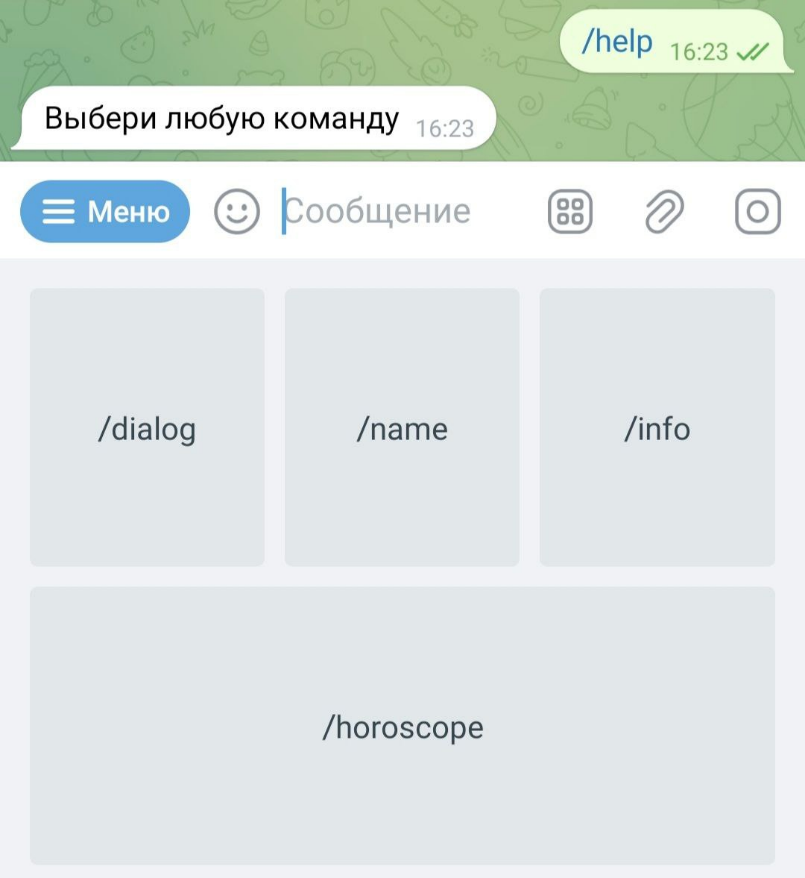


Наш бот научился делать разные вещи (теперь он не только Эхо-бот), поэтому пользователь может легко запутаться в том, какие команды он может использовать. Мы добавим кнопки-подсказки с названиями команд, которые будут появляться после отправления команды **/help.**

**Самостоятельная работа**

Запрограммируйте бота так, чтобы при команде **/help** появлялись кнопки с названиями всех команд, которые мы добавляли для нашего бота ранее.





**Рефлексия**

* Сегодня мы создали меню для бота
* Разобрались с видами возможных клавиатур
* Добавили для пользователей клавиатуру с быстрыми ответами
* Познакомились с библиотекой types, классами ReplyKeyboardMarkup и KeyboardButton

***Тайминг.***

|  |  |  |
| --- | --- | --- |
|  | Раздел урока | Время с начала урока, мин. |
|  | Повторение материала предыдущего урока. Постановка цели урока | 0 |
|  | Виды клавиатур | 10 |
|  | Reply-кнопки | 20 |
|  | Самостоятельная работа | 40 |
|  | Бескомпьютерная деятельность | 55 |
|  | Самостоятельная работа | 60 |
|  | Рефлексия | 85 |