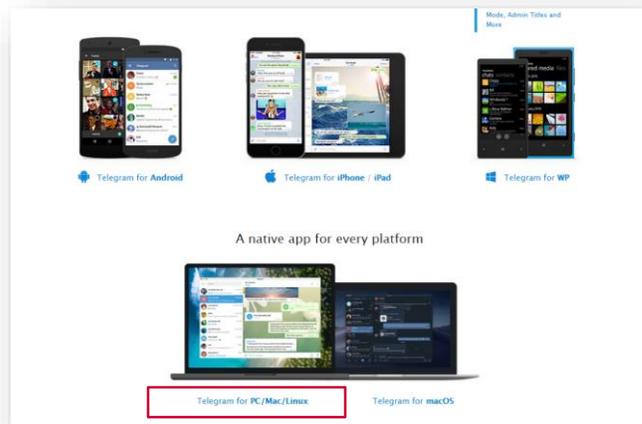


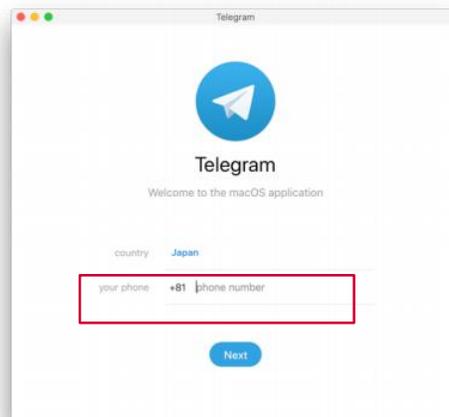
# Telegramのインストール方法

1. <https://telegram.org> にアクセスし, アプリをダウンロード (Mac/Windows/Linux いずれの OS でも動作可)

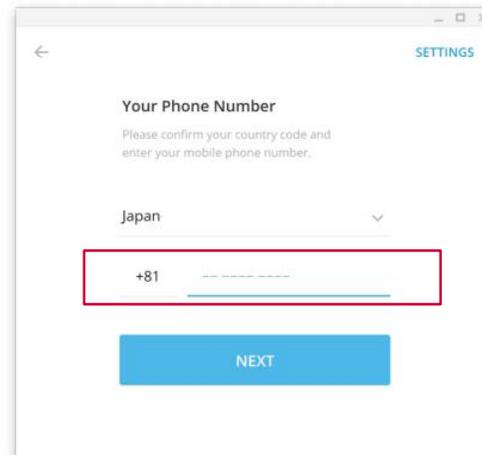


2. Telegramを立ち上げる
3. 認証コードを受け取るための電話番号を入力する

**Mac**



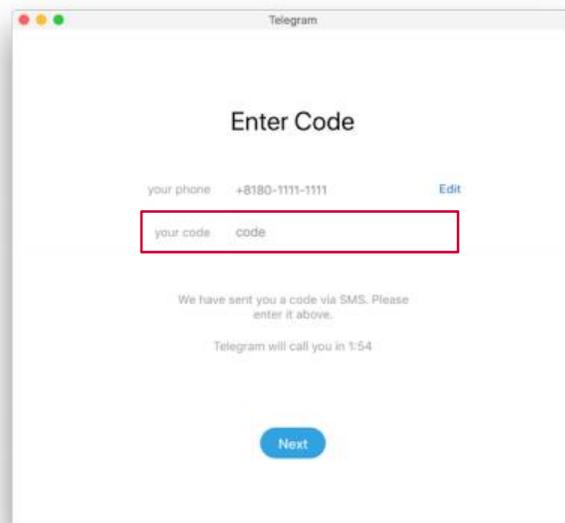
**Windows**



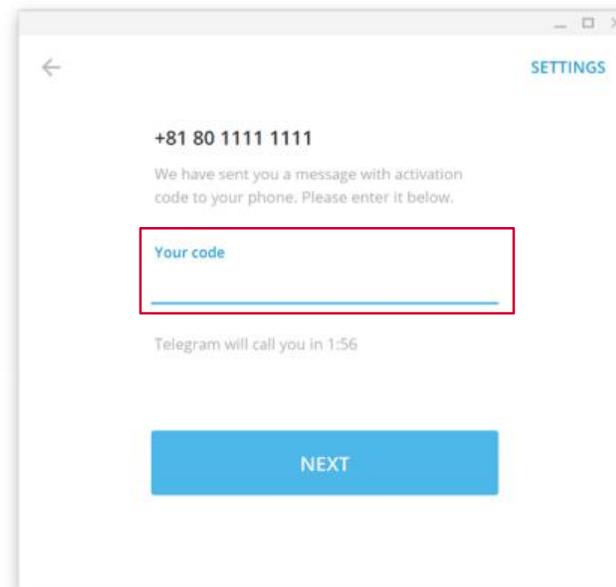
# Telegramのインストール方法

4. 携帯電話に送られてきた数字 5 桁のコードを入力し, Next ボタンをクリック

Mac



Windows

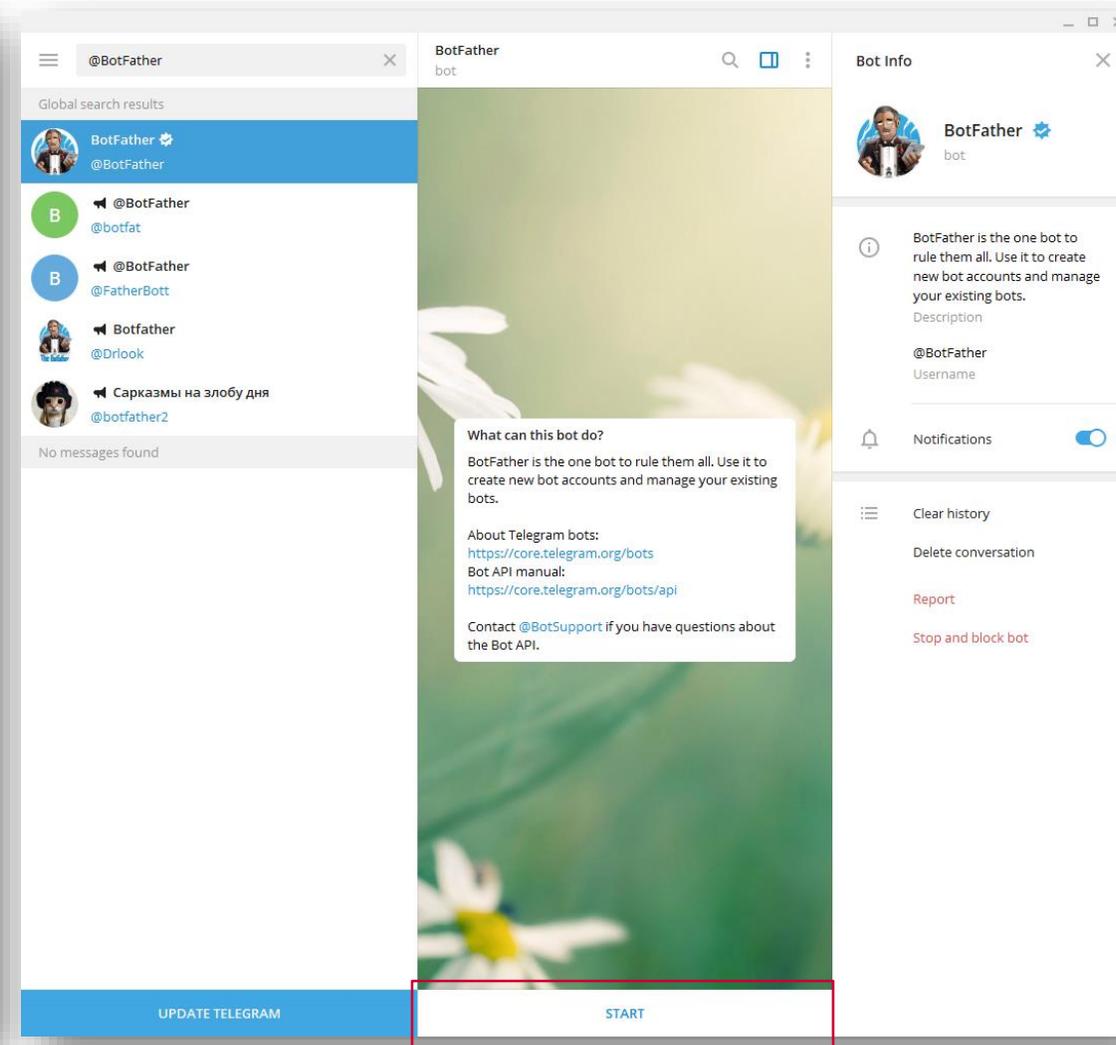
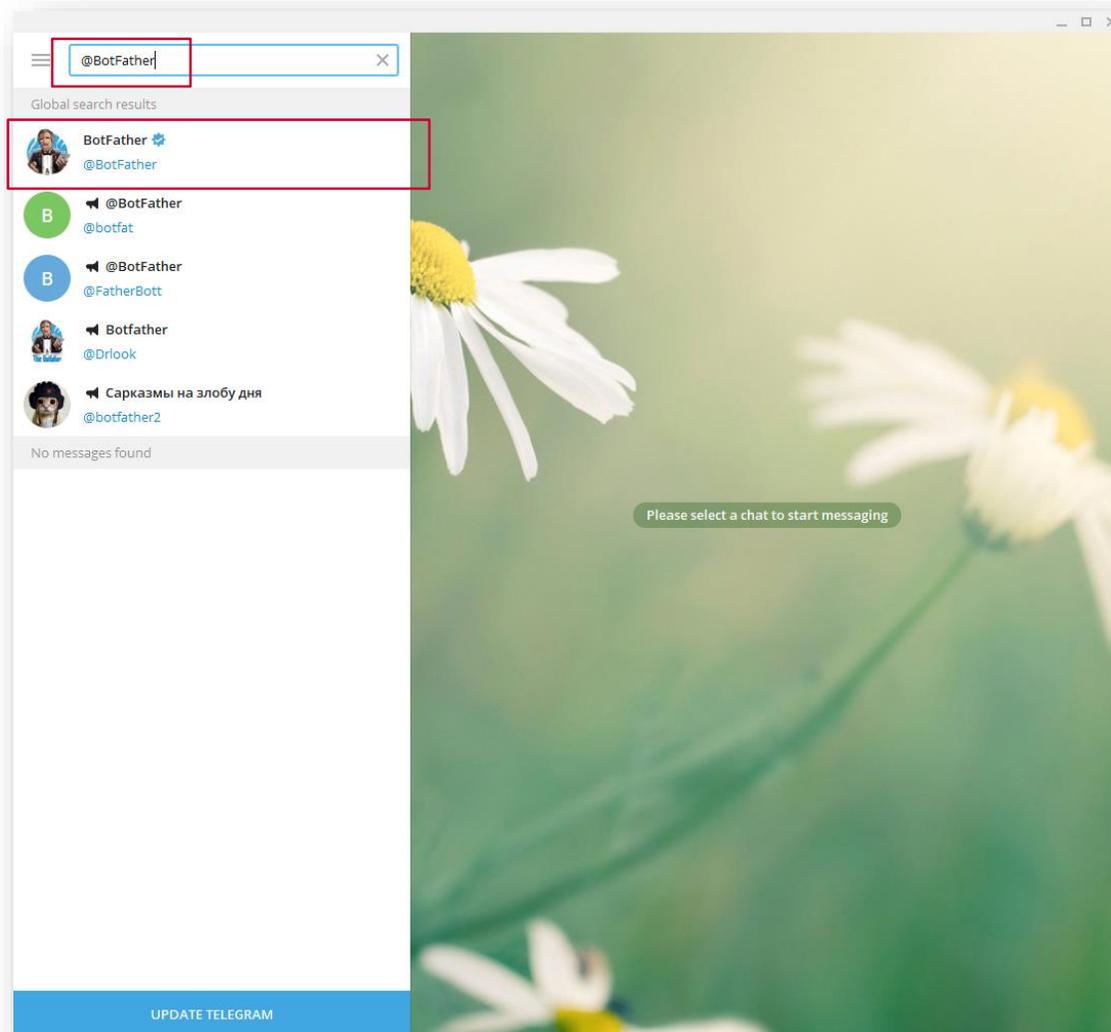


5. コードで認証されればアプリ画面に移る

※ ここで “Phone number is not registered. If you don't have a Telegram account yet, please sign up with your mobile device.” というメッセージがでて先に進めない場合は, 先にスマートフォン版で, 同じ電話番号を使って, Telegram アカウントをセットアップする. その後, パソコン版アプリで作業を再開する.

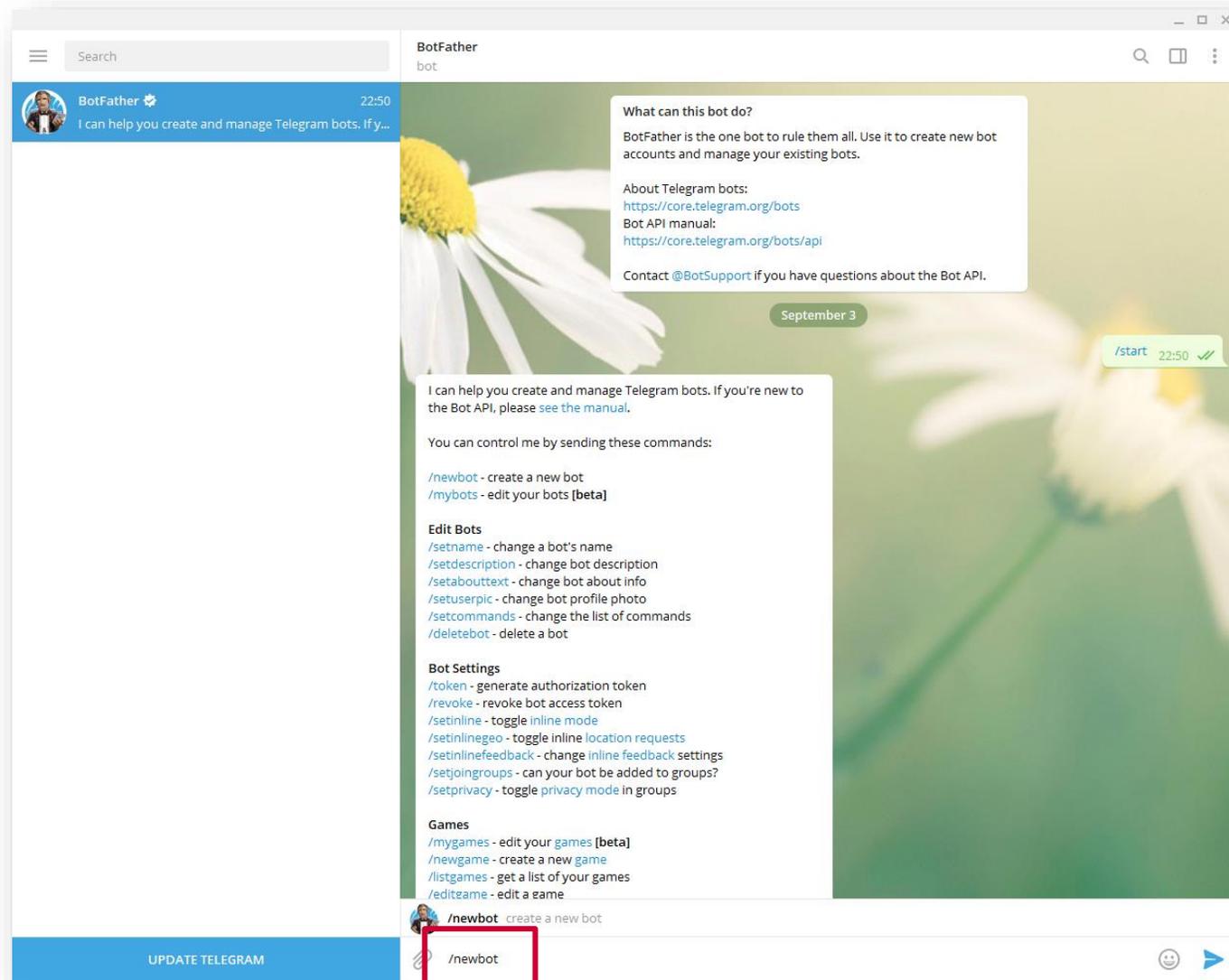
# Telegramボットの登録方法

1. Telegramでユーザ名 “@BotFather” を検索し, クリック.
2. STARTをクリックしてフレンドに追加



# Telegramボットの登録方法

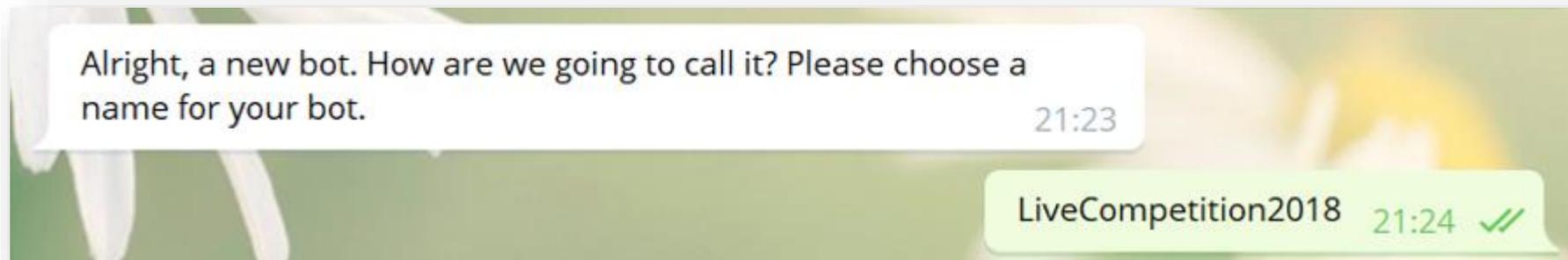
## 3. @BotFatherに “/newbot” と送信



# Telegramボットの登録方法

4. BotFatherの「How are we going to call it? Please choose a name for your bot.」に対して、ボット名を入力

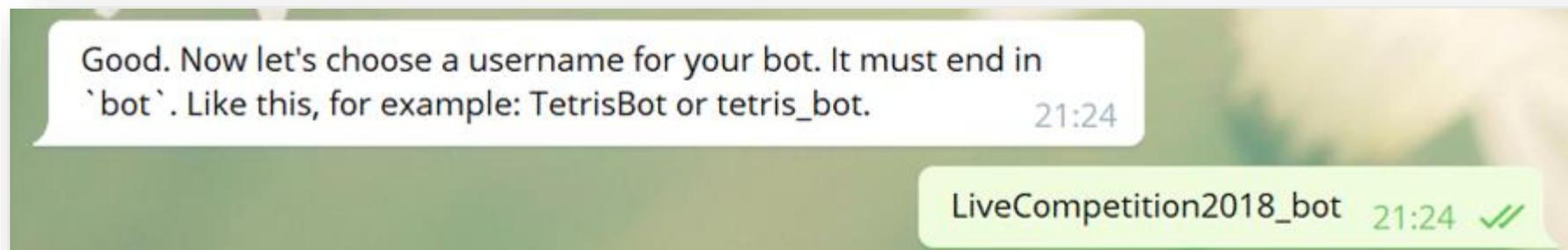
※任意の文字列が使用できるが、本コンペで使用するボットは、半角英数字のみ使用。



5. BotFatherの「Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris\_bot.」に対して、ボットのusernameを入力

※半角英数字のみ使用可で、最後は'bot'で終わる名前にする。（例：TetrisBot や tetris\_bot）

※他に同じusernameのボットがある場合は登録できない。

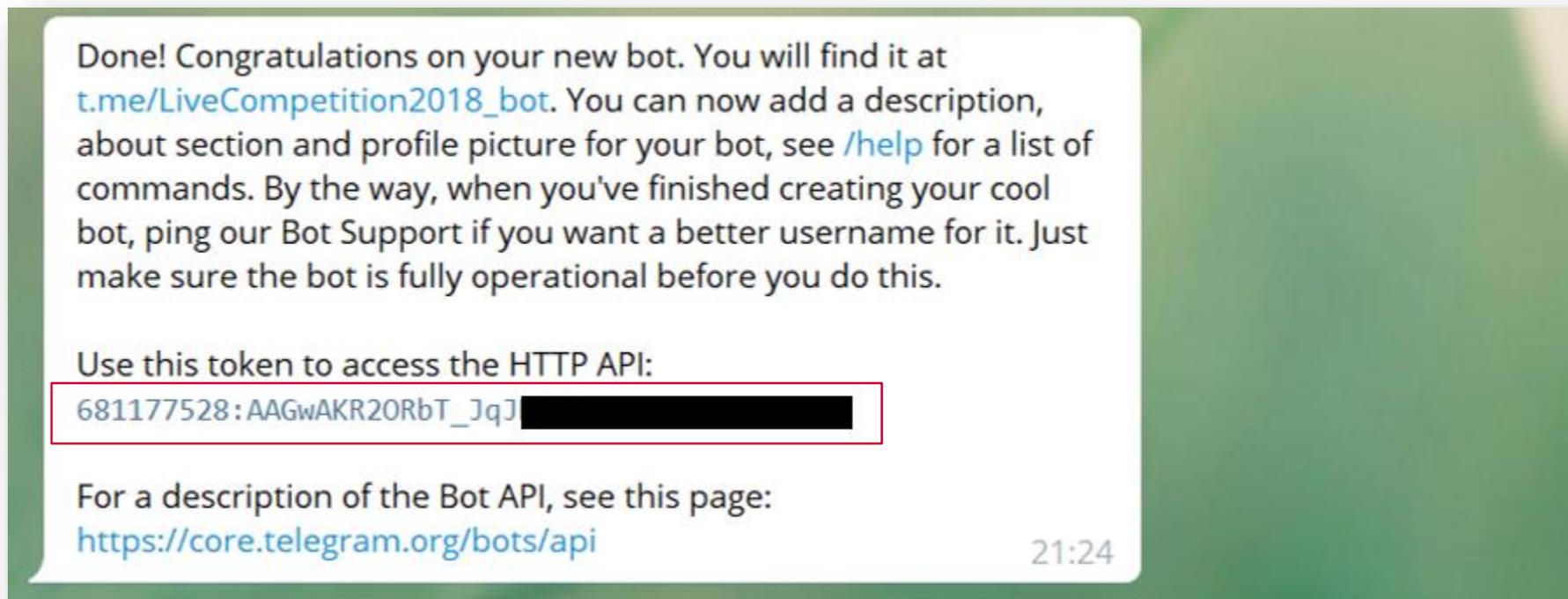


# Telegramボットの登録方法

6. アクセストークンが発行される.

※プログラム上でこのアクセストークンを指定する必要がある.

※同一の token を用いて複数の対話サーバを起動した場合, 正しく動かなくなるので注意



ライブコンペのHP (<https://dialog-system-live-competition.github.io/dslc2/gettingstart.html>) で配布しているサンプルスクリプト (samplebot9.py) をベースに, Repl-AI連携用に変更したスクリプト (**samplebot9\_4repl.py**) を使用してTelegramのボットを実装します.

# 事前準備 (Windows10向け)

- python3系のインストールパッケージをダウンロード (<https://www.python.org/downloads/>)

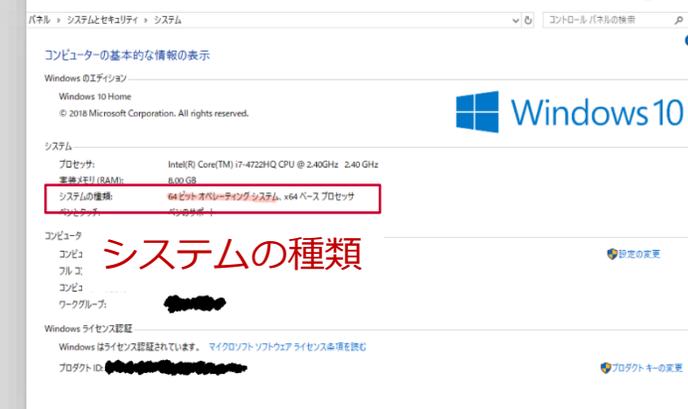
Version	Operating System
<a href="#">Gzipped source tarball</a>	Source release
<a href="#">XZ compressed source tarball</a>	Source release
<a href="#">Mac OS X 64-bit/32-bit installer</a>	Mac OS X
<a href="#">Windows x86-64 executable installer</a>	Windows
<a href="#">Windows x86-64 web-based installer</a>	Windows
<a href="#">Windows x86 embeddable zip file</a>	Windows
<a href="#">Windows x86 executable installer</a>	Windows
<a href="#">Windows x86 web-based installer</a>	Windows

64bit→ Windows x86-64 executable installer

32bit→ Windows x86 executable installer

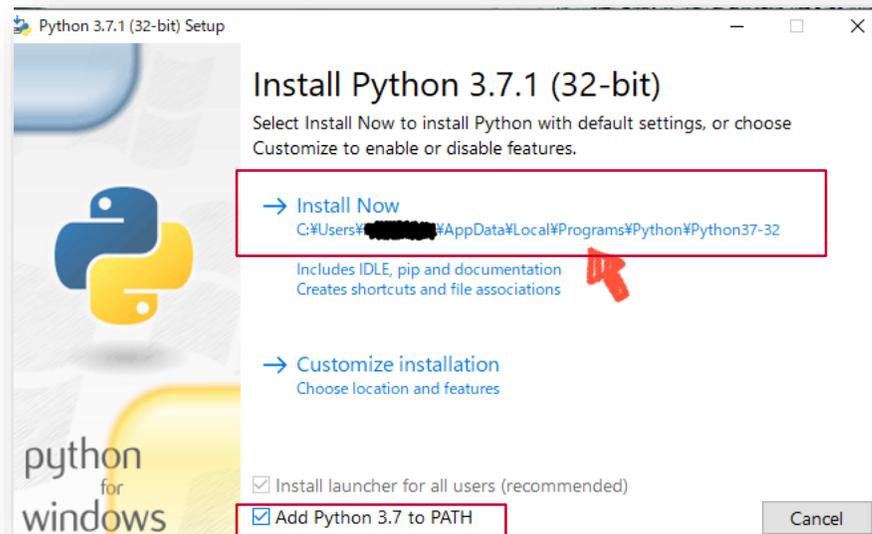
## PCスペックの確認方法

コントロールパネルを開く



# 事前準備 (Windows10向け)

2. インストーラーを起動して、『**Add Python 3.7 to Path**』にチェックを入れてから『Install now』をクリック



3. インストールが完了したら、インストーラーを閉じる
4. コマンドプロンプトを起動 (windows+r で検索窓を起動し, "cmd"と打ち込む)
5. コマンドプロンプトで「python -V」と入力して実行. 『Python 3.7.2』と表示されればインストール完了



# 事前準備 (Windows10向け)

6. Telegramと接続するためのpythonライブラリをインストール
7. コマンドプロンプトで下記を実行
  1. `pip install python-telegram-bot`
  2. `pip install requests`

参考 :

[https://qiita.com/New\\_enpitsu\\_15/items/ee95bde0858e9f77acf0](https://qiita.com/New_enpitsu_15/items/ee95bde0858e9f77acf0)

<https://qiita.com/kituneazami/items/4d2db7726bdc0eb37f2b>

# Telegramボットの実装方法

## 1. Repl-AIのボットID・シナリオID・ボットIDを確認する

The screenshot shows the Repl-AI dashboard. On the left is a sidebar with navigation options: プロジェクト一覧 (highlighted), マイページ, 使い方, APIリファレンス, and 新着情報. The main content area is titled 'ボット一覧' (Bot List) and includes a 'サンプル' (Sample) section with a '+ ボットを追加する' button. Below this is a table of scenarios with columns for 'シナリオ名(シナリオID)', 'シナリオタイプ', and '公開ステータス (2017/11/28)'. The scenarios listed are: docomoAPI連携 (docomoapi), LINE連携 (line), シンプル (simple), テスト1 (s4uidhnszq9e0an), and 挨拶 (aisatsu). The 'aisatsu' scenario is highlighted with a red box, and a red text label '(追加したシナリオの) シナリオID' points to it. Below the table is a '+ シナリオを追加する' button. At the bottom, the '稼働状況' (Operational Status) section indicates the project is active, with a '一時停止する' button. The '開発者向け' (Developer) section shows an 'APIキー' (API Key) as 'kufuPsKILF...' with a red box around it and a red text label 'APIキー'.

シナリオ名(シナリオID)	シナリオタイプ	公開ステータス (2017/11/28)
docomoAPI連携 (docomoapi)	自由文脈	公開
LINE連携 (line)	自由文脈	公開
シンプル (simple)	自由文脈	公開
テスト1 (s4uidhnszq9e0an)	自由文脈	非公開
挨拶 (aisatsu)		

ユーザの初期発話が入るシナリオID (トピック) を指定する

(追加したシナリオの) シナリオID

APIキー

# Telegramボットの実装方法

## 2. スクリプト (samplebot9\_4repl.py) のパラメータを変更する

```
10  ### telegram~
11  # アクセストークン~
12  TOKEN = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
13  ト
14  ### Repl-AI~
15  # APIキー~
16  APIKEY = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
17  # BOTID~
18  BOTID = "sample"
19  # 最初のTopicId~
20  TopicId = "aisatsu"
```

Telegramで払いだされたアクセストークン

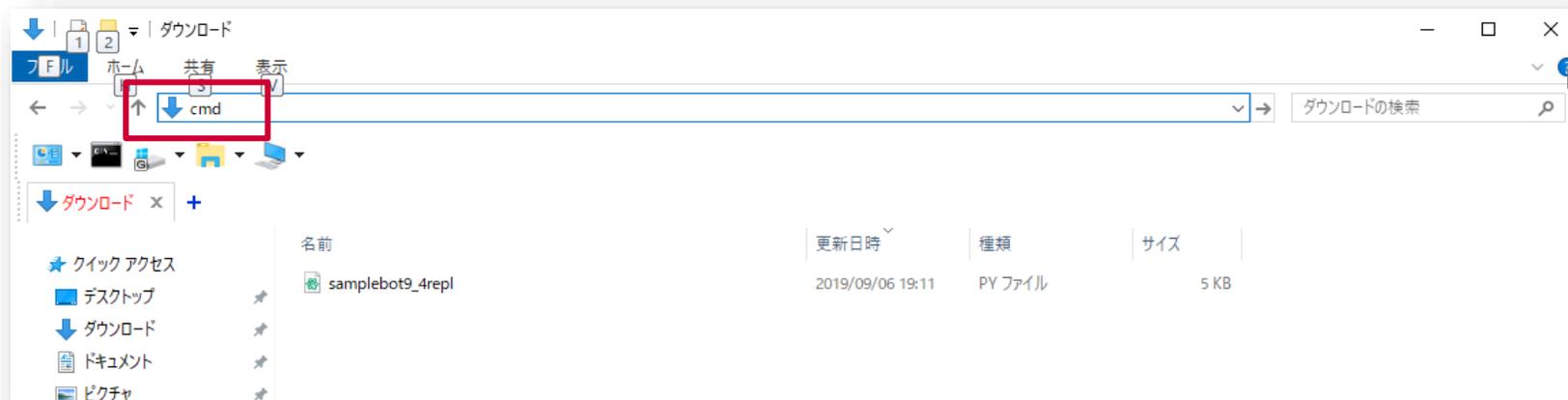
Repl-AIのAPIキー

Repl-AIのBOTID

ユーザの初期発話が入るRepl-AIのシナリオID

# Telegramボットを起動する

1. samplebot9\_4repl.py を置いたディレクトリでコマンドプロンプトを起動  
例) ダウンロードに置いた場合  
ダウンロードのディレクトリをエクスプローラーで開き,  
アドレスバーに「cmd」と打ち込んでエンターを押し、コマンドプロンプトを起動

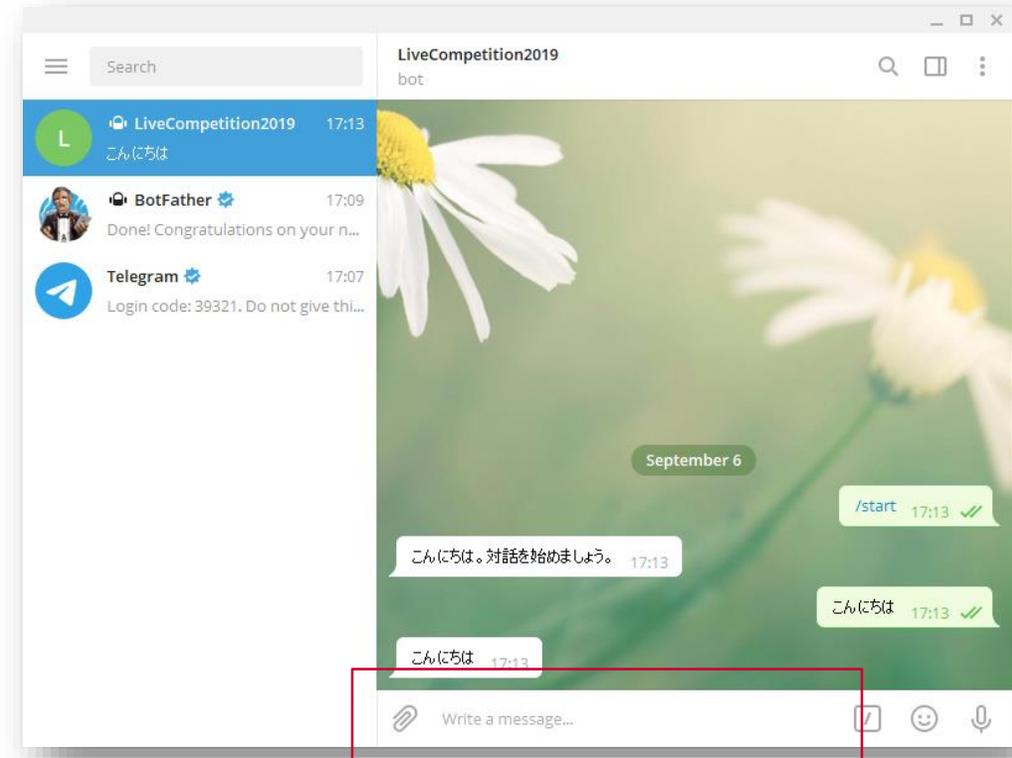
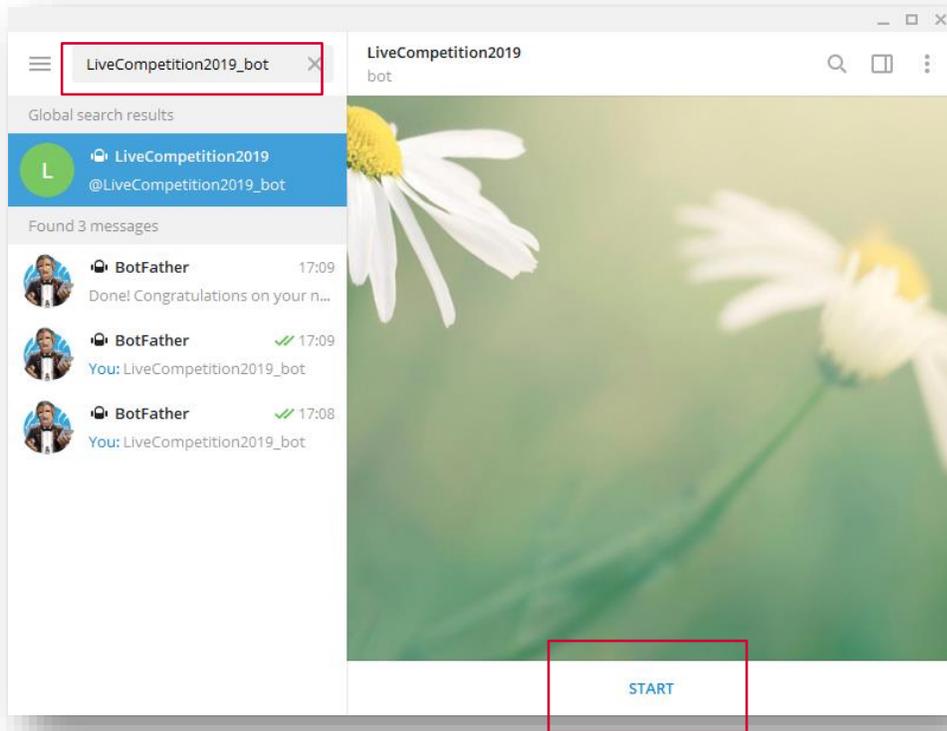


2. コマンドプロンプトで「python samplebot9\_4repl.py」を実行

※同一の トークンを用いて複数の対話サーバを起動した場合, 正しく動かないので注意

# Telegram上で対話を行う

1. 作成したBOT名（TetrisBot や tetris\_bot）を検索して，選択する
2. 「START」を押して対話を開始する



自由に発話を入力（エンターキーで送信）

※新規に対話を行いたい場合は，「/start」を入力する

- Repl-AIでは表現できないような、さらに複雑なシナリオを書きたい方は、SUNABA (<https://docs.xaiml.docomo-dialog.com/>) がおすすめです (無料)
  - xaimlという言語を使って、対話シナリオを記載します
  - チュートリアルが充実しているので、初心者から始められます



- 雑談シナリオを書くのが大変 (返せないユーザ入力が多すぎる…) と思った方は、かたらい (<https://www.katar.ai/>) がおすすめです
  - シナリオ作成不要ですぐ雑談機能を付与できるAPIです
  - 約4,000万相当のシナリオを持っているので、あらゆる話題に応答できます

