

記事

[Mihoko Iijima](#) · 2022年1月6日 4m read

IRIS ターミナルで履歴からのコマンド実行、コマンドのショートカット作成ができるのをご存知ですか？

開発者のみなさん、あけましておめでとうございます 今年もどうぞよろしくお祈りします！

さて、この記事では、IRIS
ターミナルに（こっそり）追加された便利機能をご紹介します！（つい最近知りまして、びっくりしました 😊
）

2023/4/13 追記：Pythonシェルへ切り替えるメソッドのショートカットが追加されていたので返信欄に追記しました。

IRIS ターミナルで以前実行したコマンドを再実行する場合、上矢印キーを連打しながらコマンドを探されていると思うのですが、IRIS 2021.1 から履歴表示と、履歴番号を指定した実行ができるようになっていました！

では早速、履歴（history）の使い方をご紹介します。これがあれば、もう、上矢印キーを連打せずに以前実行したコマンドを再実行できます！！

まずは、履歴を作るため、いくつかコマンドを実行します。

```
?????:_system
?????:***
USER>

USER>write $ZDATE($NOW(),16)
2022?1?6?
USER>

USER>write $ZV
IRIS for Windows (x86-64) 2021.1 (Build 215U) Wed Jun 9 2021 09:39:22 EDT
USER>

USER>write $system.Util.ManagerDirectory()
c:\intersystems\irishealth\mgr\
USER>
```

ここで、`:?` と入力してみます。

（なんと、オプションが表示されます！！）

```
USER>:?
:<number>    Recall command # <number>
:?           Display help
:alias       Create/display aliases
:clear       Clear history buffer
:history     Display command history
:unalias     Remove aliases
```

:history (または **:h**) を実行すると、履歴が表示されます！

```
USER>:history
1: _system
2: write $ZDATE($NOW(),16)
3: write $ZV
4: write $system.Util.ManagerDirectory()
5: :?
6: :history
```

コロン(:)と履歴番号を指定し、以前実行したコマンドを再実行してみます。(:4 と入力しています。)

```
USER>:4
write $system.Util.ManagerDirectory()
c:\intersystems\irishealth\mgr\
USER>
```

上矢印キーを連打せずに、過去のコマンドを実行できました！便利です 😊

続いて、alias オプションを使うとコマンドのショートカットが作成できるようですので、それも試してみます。

以下の実行例では、TESTネームスペースに移動した後、IRISターミナルをSQL実行モードに変更し、SQLを実行しています。

```
USER>set $namespace="test"

TEST>do $system.SQL.Shell()
SQL Command Line Shell
-----

The command prefix is currently set to: <<nothing>>.
Enter <command>, 'q' to quit, '?' for help.
[SQL]TEST>>select * from Training.Person
1.      select * from Training.Person

ID      Email      Name
1       yamada@majorcorp.com      ??????

1 Rows(s) Affected
statement prepare time(s)/globals/cmds/disk: 0.0462s/3307/171368/2ms
```

```
execute time(s)/globals/cmds/disk: 0.0002s/2/675/0ms
      cached query class: %sqlcq.TEST.cls166
```

```
-----
[SQL]TEST>>quit
```

```
TEST>
```

なかなか覚えにくいコマンドに対して、alias を設定しておくとう便利そうですので、ターミナルを SQL の実行環境に変更する `do $system.SQL.Shell()` を設定してみます。

設定方法は以下の通りです。

`:alias` 名称 コマンド

`$system.SQL.Shell()` を `sqlmode` の名称で alias に設定する例は以下の通りです。

```
TEST>:alias sqlmode do $system.SQL.Shell()
```

```
TEST>:alias
      sqlmode          do $system.SQL.Shell()
```

早速、設定した alias を使ってみましょう！

```
TEST>:sqlmode
do $system.SQL.Shell()
SQL Command Line Shell
```

```
-----
The command prefix is currently set to: <<nothing>>.
Enter <command>, 'q' to quit, '?' for help.
[SQL]TEST>>quit
```

```
TEST>
```

無事、実行できました！

(alias 名は省略できるので、上記登録の場合 `:s` でも実行できるようでした。)

メモ：2021.2プレビュー版では、`do $system.SQL.Shell()` が alias 名 `sql` で事前に登録されているため、`:sql` で SQL 実行モードに切り替えできました。

履歴 (history) や alias の例については、ドキュメントにも記載がありますので、併せてご参照ください！

[前のコマンドの繰り返し](#)

[#ターミナル #InterSystems IRIS](#)

ソースURL:

<https://jp.community.intersystems.com/post/iris-%E3%82%BF%E3%83%BC%E3%83%9F%E3%83%8A%E3%83%AB%E3%81%A7%E5%B1%A5%E6%AD%B4%E3%81%8B%E3%82%89%E3%81%AE%E3%82%B3%E3%83%9E%E3%83%B3%E3%83%89%E5%AE%9F%E8%A1%8C%E3%80%81%E3%82%B3%E3%83%9E%E3%83%B3%E3%83%89%E3%81%AE%E3%82%B7%E3%83%A7%E3%83%BC%E3%83%88%E3%82%AB%E3%83%83%E3%83%88%E4%BD%9C%E6%88%90%E3%81%8C%E3%81%A7%E3%81%8D%E3%82%8B%E3%81%AE%E3%82%92%E3%81%94%E5%AD%98%E7%9F%A5%E3%81%A7%E3%81%99%E3%81%8B%EF%BC%9F>