

idle の使い方

Ver 1.0

Seiichi Nukayama

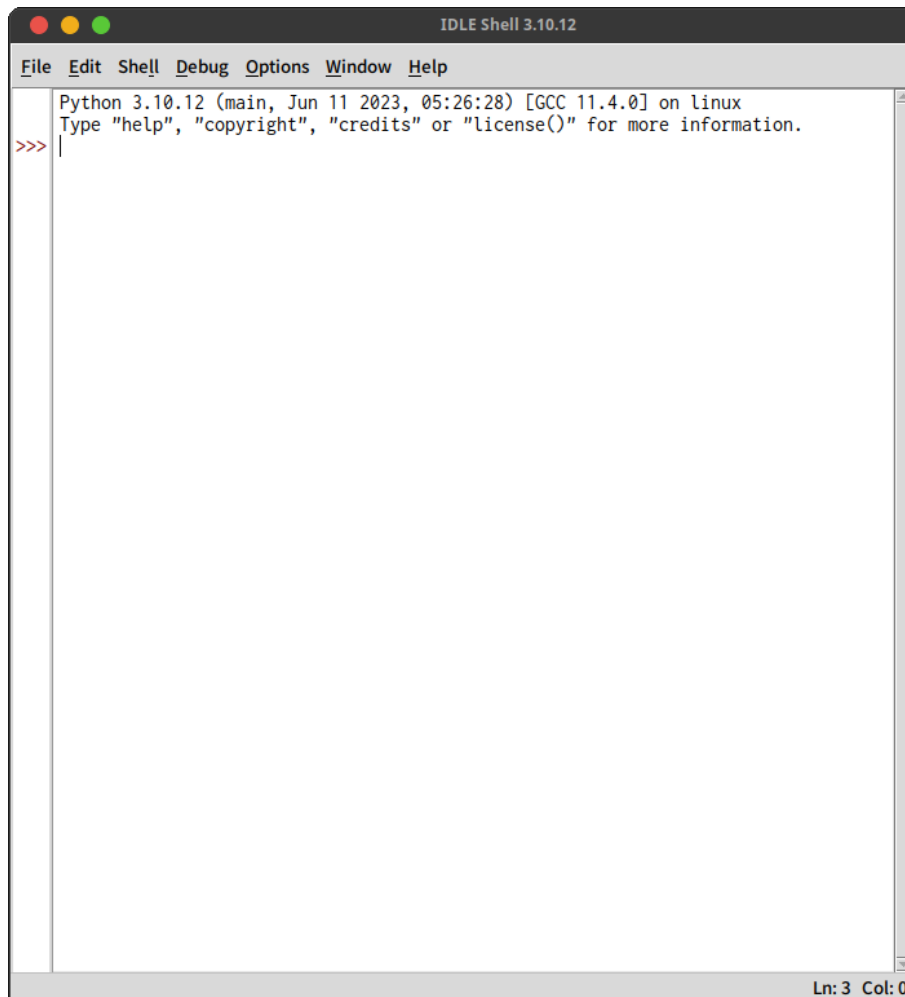
2023 年 8 月 15 日

目次

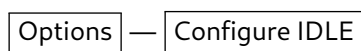
1	IDLE Shell	1
1.1	起動直後の画面	1
1.2	設定	1
2	プログラムの入力	8
2.1	IDLE Shell で実行	8
2.2	ファイルの作成	9
3	デバッガの使い方	12
3.1	サンプルコードの入力	12
3.2	実行	12
3.3	コードの中の変数の値を調べる	13
3.4	デバッガで 1 行ずつ実行する	13
3.5	ブレークポイントで止める	16

1 IDLE Shell

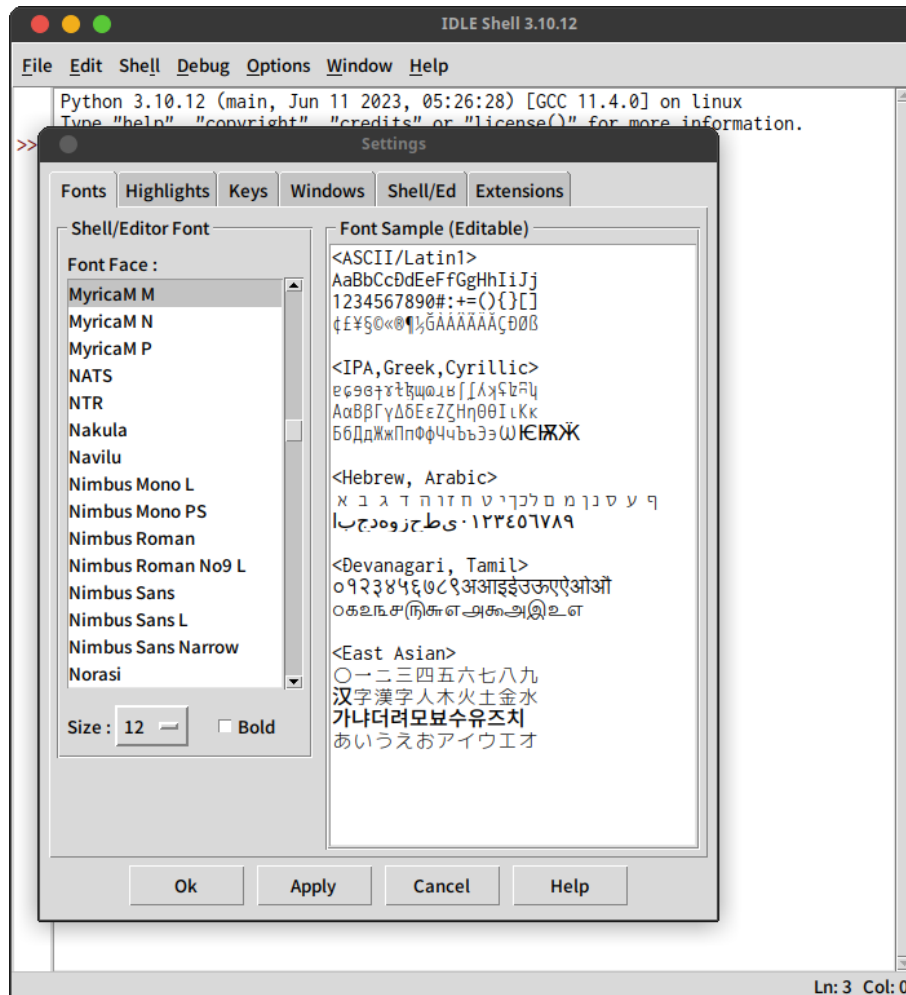
1.1 起動直後の画面



1.2 設定

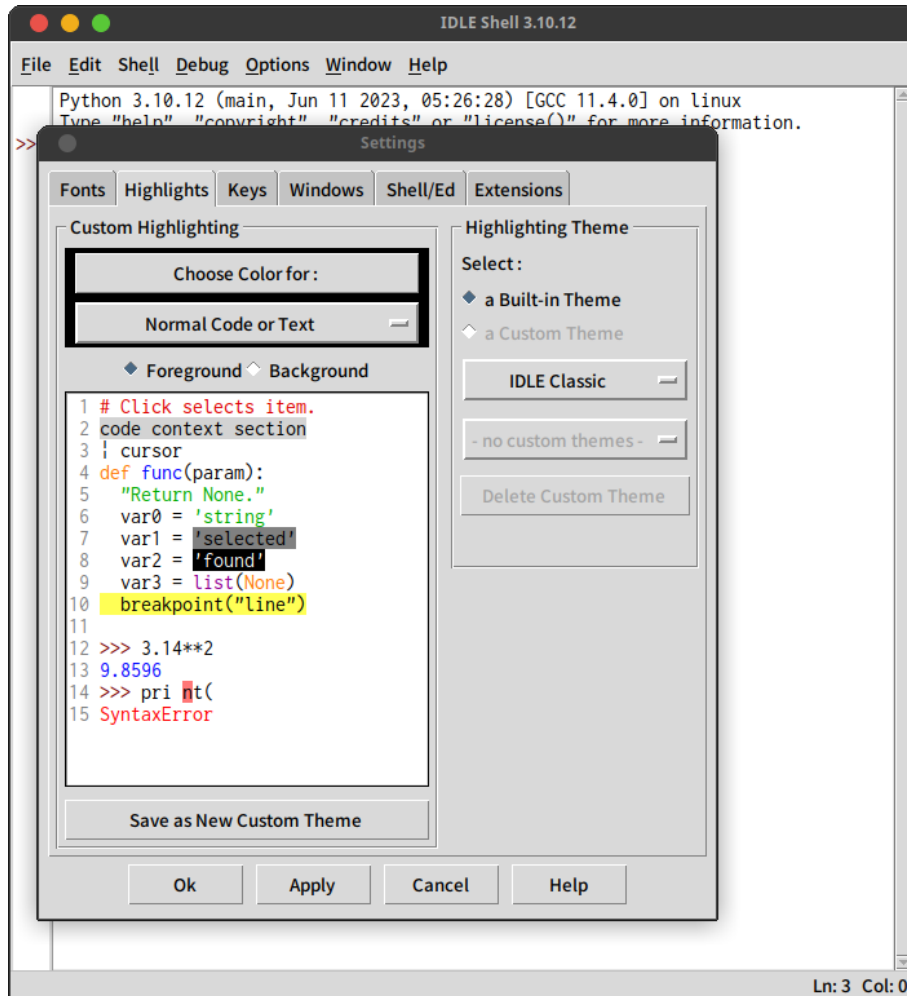


1.2.1 Fonts



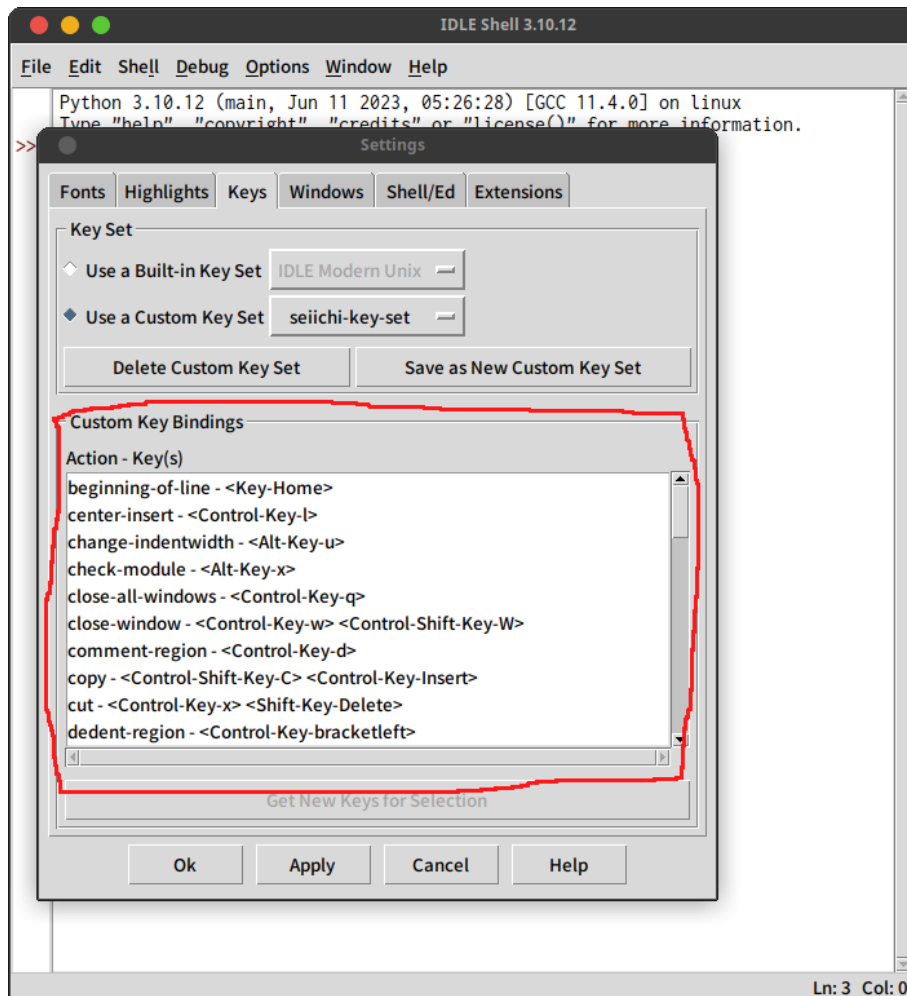
ここでは“MyricaM M” — Size 12 を指定している。

1.2.2 Highlights



デフォルトのまま。

1.2.3 keys

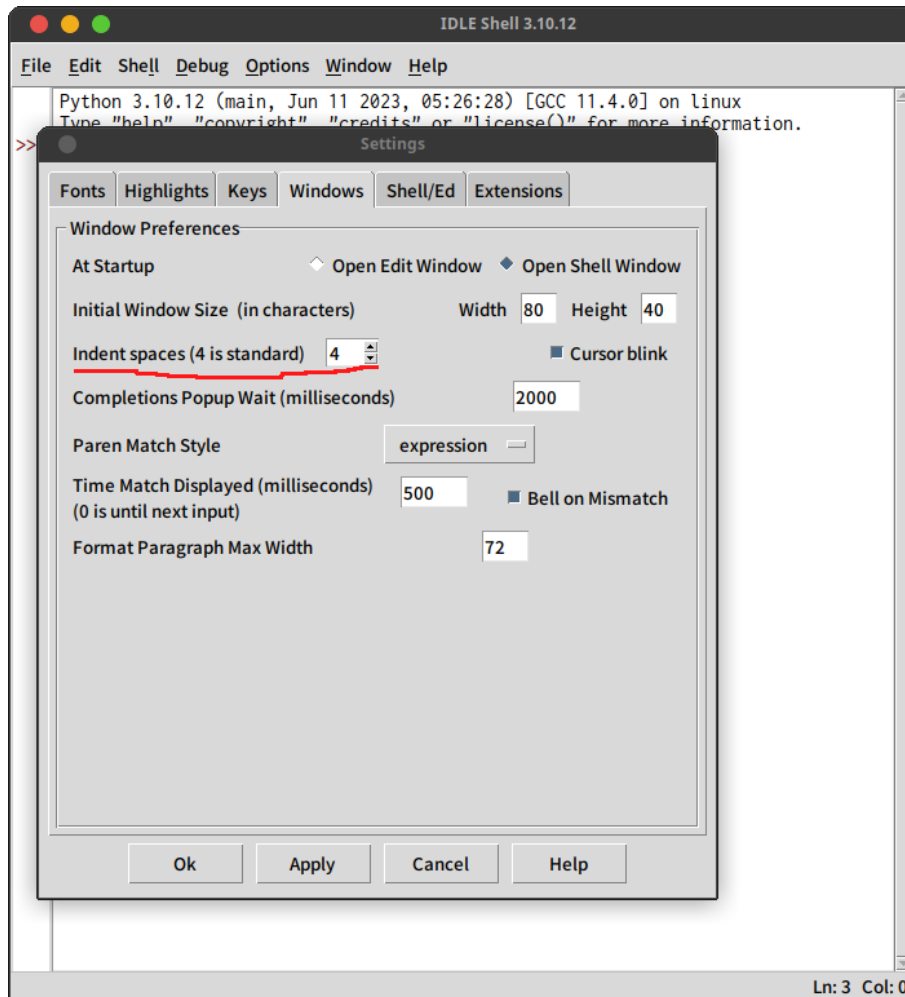


デフォルトのまま。

Custom Key Bindings の Action - Key(s) にさまざまなショートカットが登録されている。

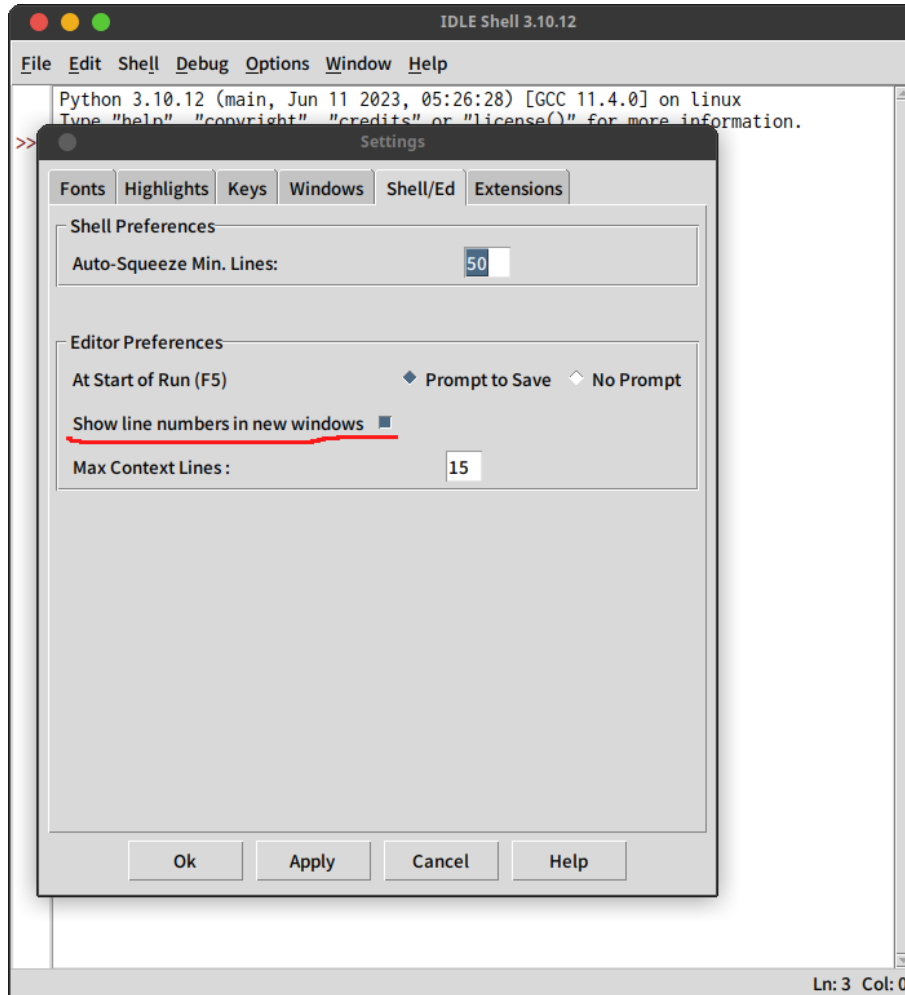
comment-region	コメント (指定範囲)	Ctrl-d
goto-line	行ジャンプ	Ctrl-g
history-next	ヒストリ次	Alt-n
history-previous	ヒストリ前	Alt-p
indnet-region	インデント増 (指定範囲)	Ctrl-]
python-context-help	オンラインヘルプ	Shift-F1
uncomment-region	コメント取消 (指定範囲)	Ctrl-Shift-D
undo	取消	Ctrl-z

1.2.4 Windows



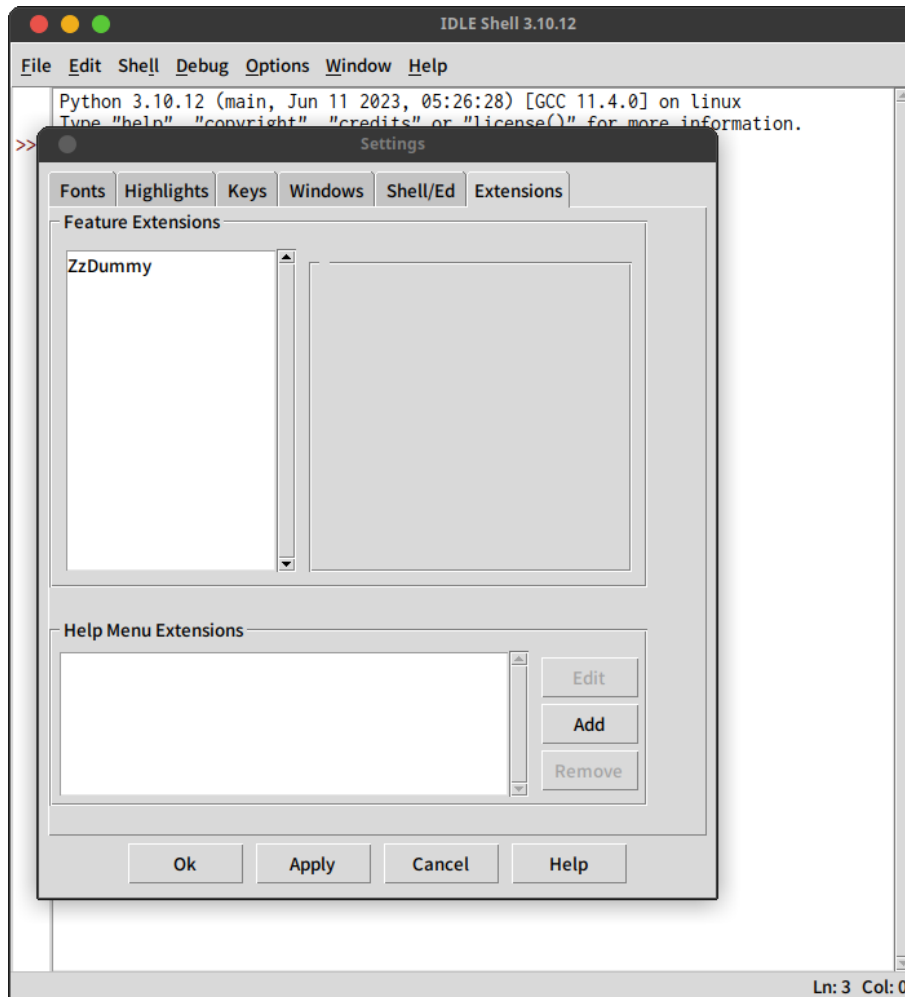
Indent spaces (4 is standard) は、“4” が Python ではデフォルトである。

1.2.5 Shell/Ed



Show line numbers in new windows にチェックを入れておく。そうすることで、行番号が表示される。

1.2.6 Extensions



ここでは特にすることはない。

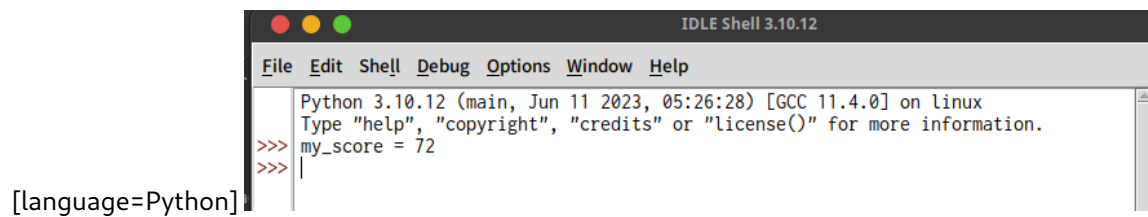
2 プログラムの入力

2.1 IDLE Shell で実行

IDLE Shell を起動する。

以下のように 入力する。

```
1 my_score = 72
```



続けて、以下のように入力する。

```
1 if my_score >= 80:
2     print("Grade: A")
3 elif myscore >= 70:
4     print("Grade: B")
5 else:
6     print("Grade: C")
```

最後に [Enter] キーを押下すると、以下のようなエラーメッセージが出力される。

Traceback (most recent call last):

File "/usr/lib/python3.10/idlelib/run.py", line 578, in runcode

exec(code, self.locals)

File "<pyshell#1>", line 3, in <module>

NameError: name 'myscore' is not defined. Did you mean: 'my_score'?

最後の行でアドバイスしてくれている。Did you mean: 'my_score'?

Alt - **p** で履歴を呼び出せる。

カーソルキーで "myscore" のところに移動し、"my_score" に修正する。

その後、カーソルキーを末尾に移動し、[Enter] キーを 2 度押下する。

コードが実行されて、"Grade: B" と表示される。

```
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> my_score = 72
>>> if my_score >= 80:
...     print("Grade: A")
... elif myscore >= 70:
...     print("Grade: B")
... else:
...     print("Grade: C")
...
Traceback (most recent call last):
  File "/usr/lib/python3.10/idlelib/run.py", line 578, in runcode
    exec(code, self.locals)
  File "<pyshell#1>", line 3, in <module>
NameError: name 'myscore' is not defined. Did you mean: 'my_score'?
>>> if my_score >= 80:
...     print("Grade: A")
... elif my_score >= 70:
...     print("Grade: B")
... else:
...     print("Grade: C")
...
Grade: B
>>>
```

2.2 ファイルの作成

以下のフォルダで作業をしているとする。

`C:\Users\user\Desktop\python-work` ← デスクトップに "python-work" というフォルダを作成したのである。

以下のファイルを作成する。

`test01.py`

このファイルを "IDLE" で開く。(IDLE を起動して、そのウィンドウの中に test01.py をドラッグするというやり方もある)

その中に以下のコードを書く。

test01.py

```
1 name = input('名前を入力してください\n')
2 print(name + 'さん、こんにちは。')
```

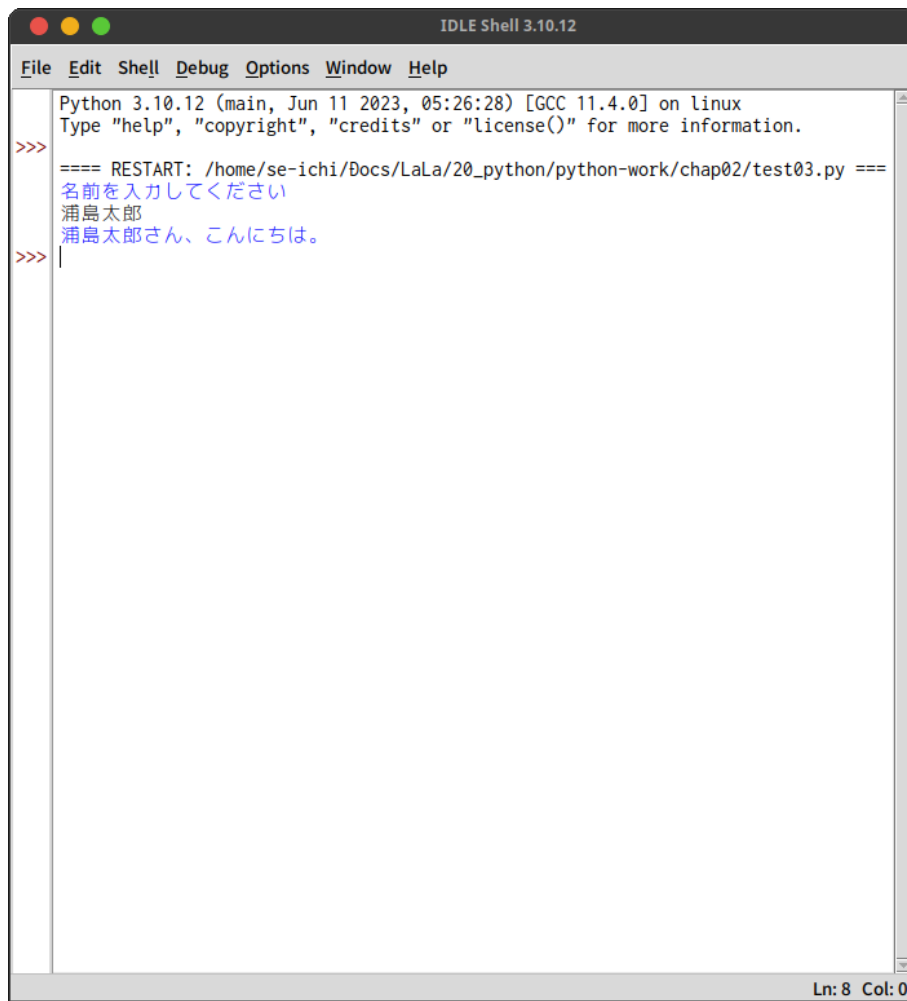
上書き保存する。(Ctrl-z)

`Run` — `Run Module` を選択する。

自動的に“IDLE Shell”が起動して、“名前を入力してください”と表示される。



ここでは、“浦島太郎”と入力した。



The image shows a screenshot of an IDLE Shell 3.10.12 window. The title bar at the top reads "IDLE Shell 3.10.12". Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main text area contains the following text:

```
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /home/se-ichi/Docs/LaLa/20_python/python-work/chap02/test03.py ====
名前を入力してください
浦島太郎
浦島太郎さん、こんにちは。
>>> |
```

At the bottom right of the window, the status bar displays "Ln: 8 Col: 0".

3 デバッガの使い方

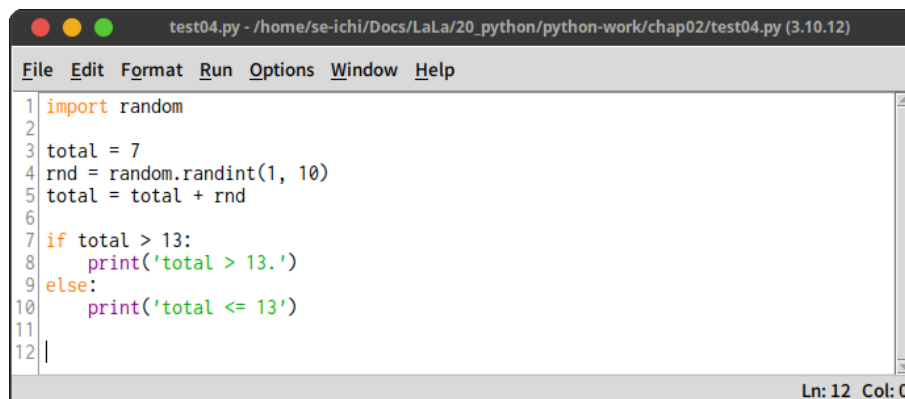
3.1 サンプルコードの入力

以下のフォルダで作業していることとする。

C:\Users\user\Desktop\python-work ← (デスクトップに python-work というフォルダを作成)
このフォルダに "test02.py" というファイルを作成し、以下の内容とする。

test02.py

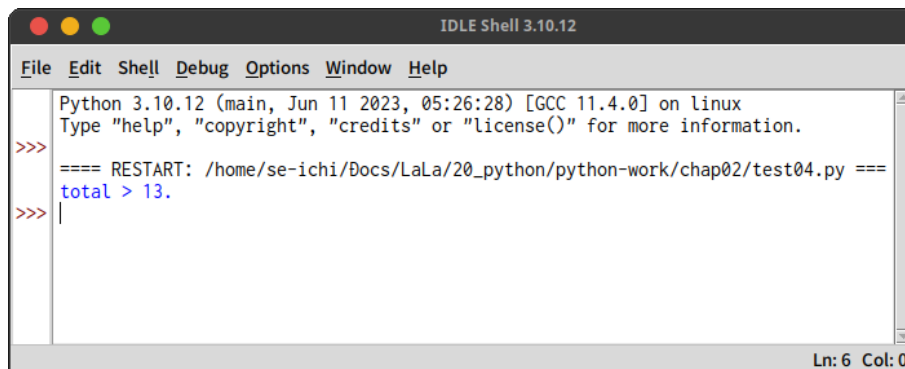
```
1 import random
2
3 total = 7
4 rnd = random.randint(1, 10)
5 total = total + rnd
6
7 if total > 13:
8     print('total > 13.')
9 else:
10    print('total <= 13')
```



3.2 実行

Run — **Run Module** で実行する。

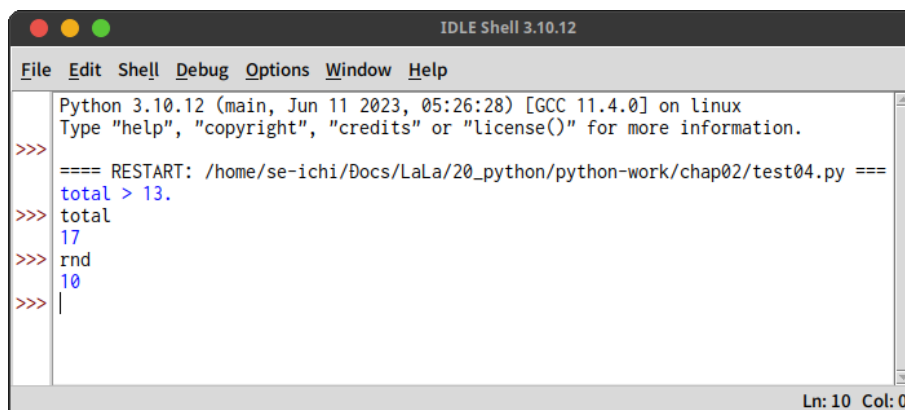
IDLE-Shell が起動して、プログラムコードが実行される。



```
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /home/se-ichi/Docs/LaLa/20_python/python-work/chap02/test04.py ====
>>> total > 13.
>>> |
```

3.3 コードの中の変数の値を調べる

プログラム実行中に変数がどのような値を持つのかを調べるには、デバッガを使う。
しかし、IDLE の場合、IDLE-Shell でも調べることができる。
その変数名を Shell の中で入力するだけでよい。



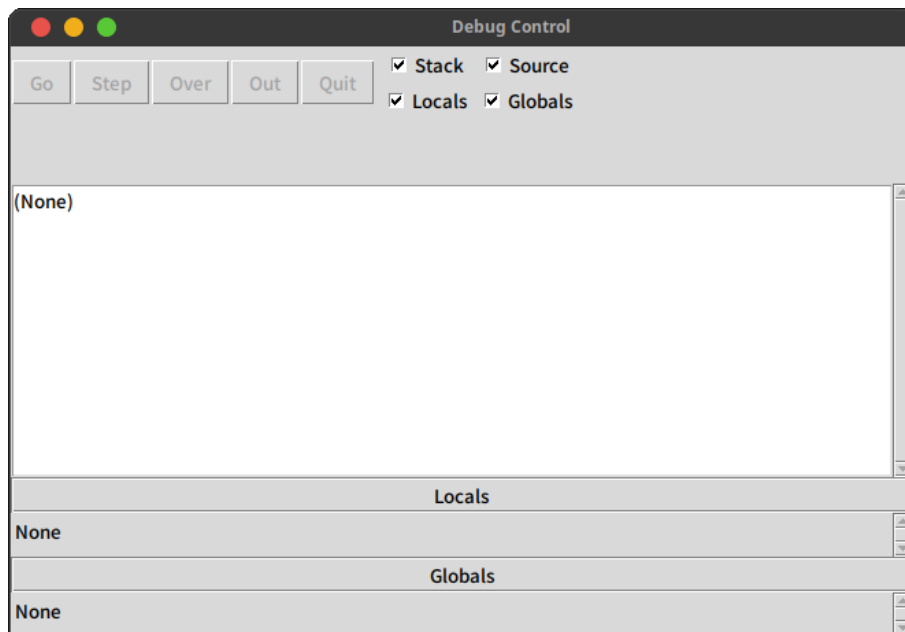
```
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /home/se-ichi/Docs/LaLa/20_python/python-work/chap02/test04.py ====
>>> total > 13.
>>> total
17
>>> rnd
10
>>> |
```

3.4 デバッガで 1 行ずつ実行する

コードの中の変数がどういう値を持っているのか、デバッガを使って 1 行ずつ実行してみる。

IDLE-Shell の `Debug` — `Debugger` を選択する。

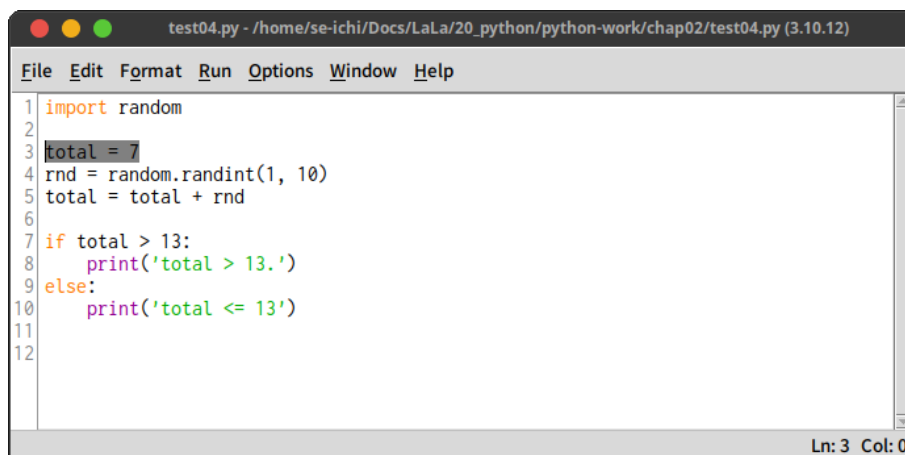
以下のようなウィンドウ (Debug Control) が開くので、`Stack`、`Locals`、`Source`、`Globals` にチェックを入れる。



プログラムコードが書かれているウィンドウの **Run** — **Run Module** を選択する。

すると、コードの 1 行目が灰色に反転している。(現在実行中の行)

Debug Control ウィンドウの **Over** をクリックする。コードの灰色が 1 つ進んで、total = 7 が現在実行中であることがわかる。



さらに **Over** をクリックする。現在行の反転が 1 つ進む。

```
1 import random
2
3 total = 7
4 rnd = random.randint(1, 10)
5 total = total + rnd
6
7 if total > 13:
8     print('total > 13.')
9 else:
10    print('total <= 13')
11
12
```

Ln: 3 Col: 0

Debug Control ウィンドウが変化して、Globals のところに total 7 が現れる。

Debug Control

Go Step Over Out Quit ☒ Stack ☒ Source ☒ Locals ☒ Globals

test04.py:4: <module>()

'<__main__>'.run(), line 598: exec(cmd, globals, locals)
> '<__main__>'.<module>(), line 4: rnd = random.randint(1, 10)

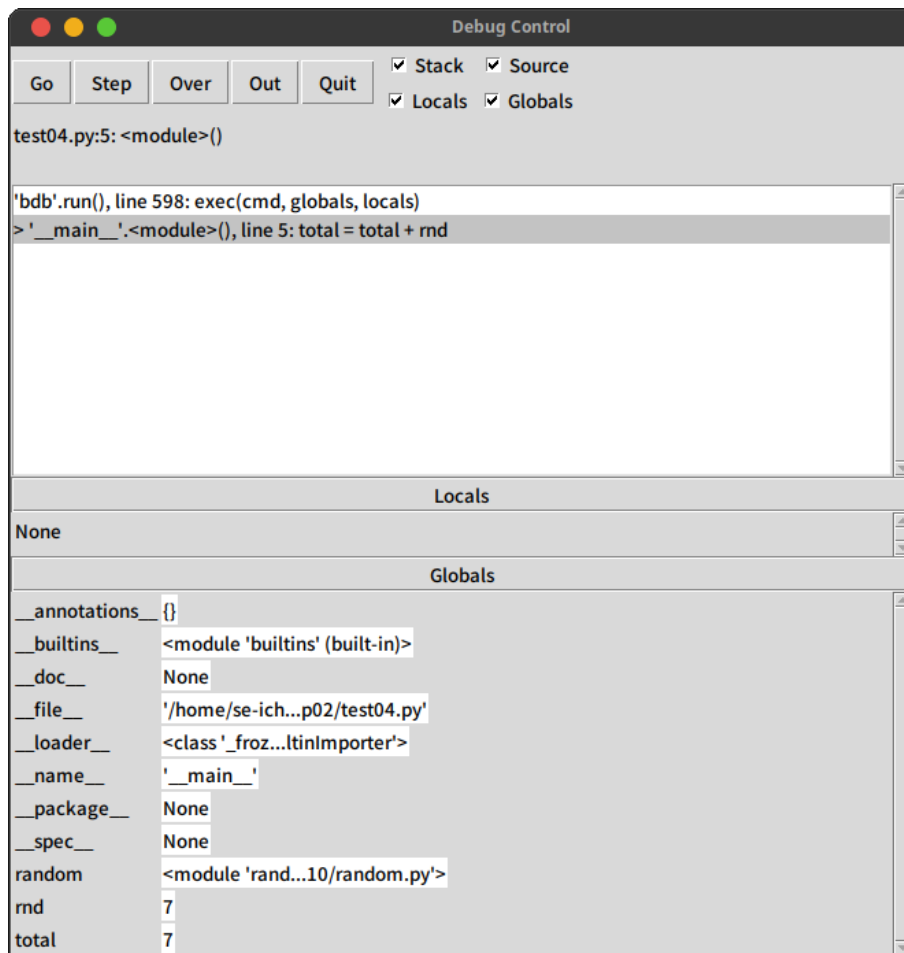
Locals

None

Globals

__annotations__	{}
__builtins__	<module 'builtins' (built-in)>
__doc__	None
__file__	'/home/se-ich...p02/test04.py'
__loader__	<class '_froz...ltinImporter'>
__name__	'__main__'
__package__	None
__spec__	None
random	<module 'rand...10/random.py'>
total	7

さらに Over をクリックすると、rnd にどんな数が代入されたのかがわかる。



Quit をクリックしてデバッガから抜ける。しかし、デバッグモードは ON のままである。
 デバッグモードを OFF にするには、IDLE-Shell の **Debug** の **Debugger** を再度クリックすればよい。
 あるいは、**Debut Control** ウィンドウを閉じることで OFF にできる。

3.5 ブレークポイントで止める

プログラムコード中の任意の行で実行を止めることもできる。

たとえば、`rnd = random.randint(1, 10)` の行で止めたい場合は、その行を右クリックして、**Set Breakpoint** とする。すると、その行が黄色く反転する。



```
test04.py - /home/se-ichi/Docs/LaLa/20_python/python-work/chap02/test04.py (3.10.12)
File Edit Format Run Options Window Help
1 import random
2
3 total = 7
4 rnd = random.randint(1, 10)
5 total = total + rnd
6
7 if total > 13:
8     print('total > 13.')
9 else:
10    print('total <= 13')
11
12
Ln: 4 Col: 10
```

IDLE-Shell で **Debug** — **Debugger** をクリックして、**Debug Control** を再度表示させる。

エディタ で **Run** — **Run Module** とする。

プログラムの実行が始まる。

Debug Control で **Go** ボタンをクリックすると、ブレークポイント行まで進む。

Over ボタンをクリックすることで、その行から 1 行ずつ実行させることができる。