

Notebookの基礎

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', an 'アップグレード' (Upgrade) button, a notification bell, and a user profile 'Seminar Kenichi Inoue's A...'. Below that, the breadcrumb 'My projects / First Project / First Notebook' is visible. The main toolbar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A blue callout bubble labeled 'Runで実行' (Execute with Run) points to the 'Run' button in the toolbar. The code editor shows a cell with the code 'import this' highlighted in a red box. A red callout bubble labeled '「セル」にPythonコードを入力' (Enter Python code in the cell) points to the code input area. The output area below the code is enclosed in a green box and labeled '実行結果' (Execution Result). It displays the text of 'The Zen of Python' by Tim Peters. At the bottom, there's an empty code input field 'In []:' and a chat icon.

IBM Watson Studio

アップグレード

Seminar Kenichi Inoue's A... SK

My projects / First Project / First Notebook

File Edit View Insert Cell Kernel Help

Run

Runで実行

「セル」にPythonコードを入力

```
In [1]: import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

実行結果

In []: