

動画に関しては[こちら](#)からご覧ください。

<https://youtu.be/bdl1ubs-TLU>

※ご利用のブラウザによって、外部サイトへのリンクが機能しない場合がございます。
その際はお手数ですがファイルをダウンロードしご利用下さい。

The image shows a YouTube video player interface. At the top, there is a channel banner for 'OSS協議会 | OSS勉強会 - Python - (2022年2月...)'. Below the banner, the video title 'IBM i で Python やってみた. 3' is displayed in large black text. To the right of the title, there is a play button icon and a small box containing 'Ver. 3.0' and '2022-02-25'. Below the title, there is a graphic showing the IBM i logo (a blue circle with an 'i') plus the Python logo (two interlocking snakes, one blue and one yellow) followed by the word 'python' in lowercase. This is followed by an equals sign and a hand-drawn lightbulb icon with a hand holding a pen, symbolizing an idea or solution. In the bottom right corner, there is a logo for 'T&Trust' with the tagline 'Tenderness and Trust' and the text 'ティアンドトラスト株式会社 北原 啓夫' below it.

IBM i で Python やってみた. 3



+  python™



ティアンドトラスト株式会社 北原 征夫

アジェンダ

- python 概説
- Web アプリケーション
- おまけ

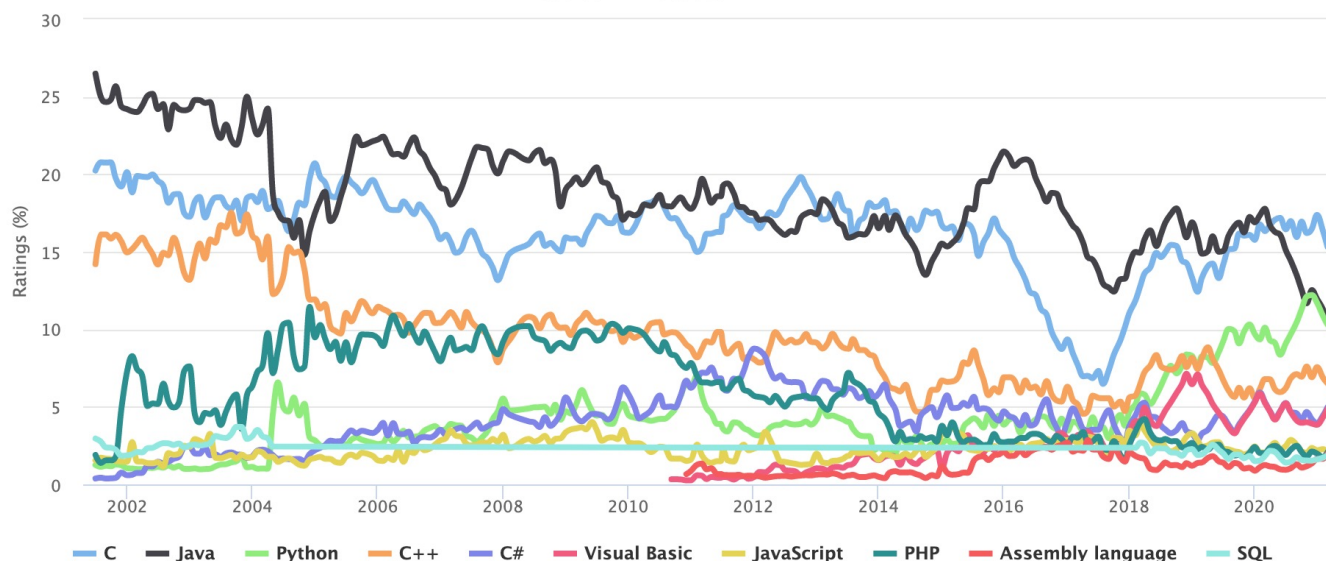
Python 概説

Python 概説

■ 2021年3月 人気度：TIOBE

TIOBE Programming Community Index

Source: www.tiobe.com



Mar 2021	Mar 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	15.33%	-1.00%
2	1	▼	Java	10.45%	-7.33%
3	3		Python	10.31%	+0.20%
4	4		C++	6.52%	-0.27%

AI (機械学習, 統計, 分析)

での利用が起因?

- 第三次 AIブーム (2006年?~)
- 非プログラマーによる利用



- ・ 習得の容易性
- ・ 豊富なライブラリー
- ・ 活用範囲の広さ

2018年ごろから急激な増加
→ Java に変わる言語として注目か

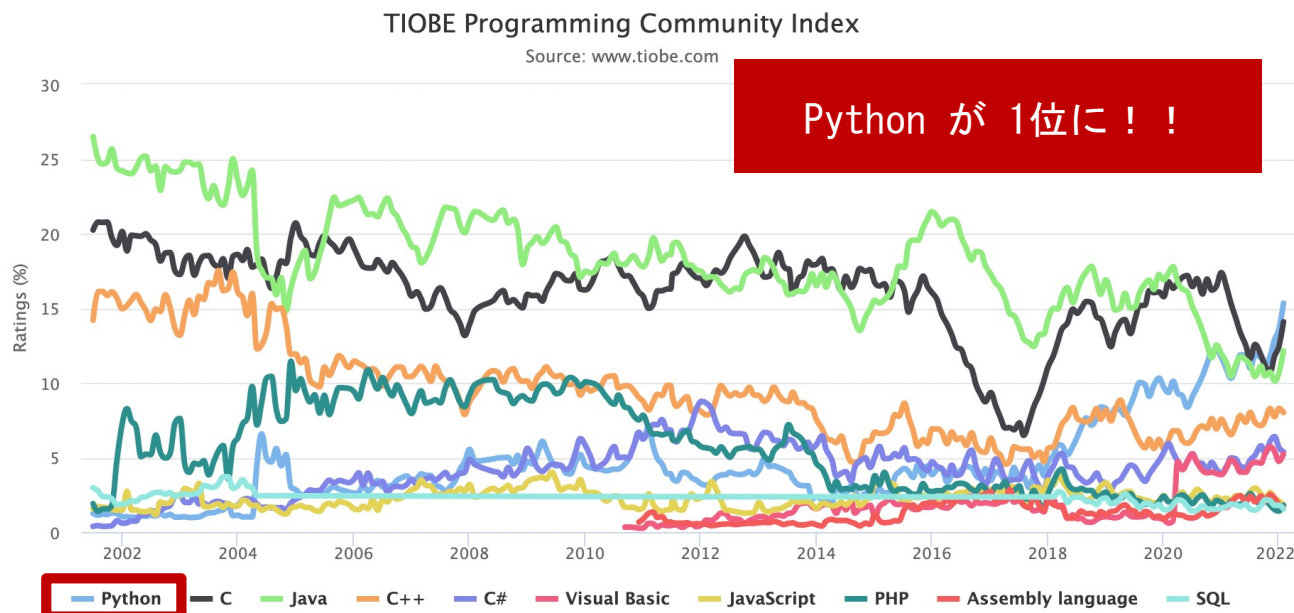
※1) TIOBE のソース
条件を満たした 25の検索エンジンのヒット数
(詳細は以下のリンクを参照)

<https://www.tiobe.com/tiobe-index/>

<https://www.tiobe.com/tiobe-index/programming-languages-definition/>

Python 概説

■ 2022年2月 人気度：TIOBE



Feb 2022	Feb 2021	Change	Programming Language	Ratings	Change
1	3	▲	Python	15.33%	+4.47%
2	1	▼	C	14.08%	-2.26%
3	2	▼	Java	12.13%	+0.84%
4	4		C++	8.01%	+1.13%

AI (機械学習, 統計, 分析)

での利用が起因?

- 第三次 AIブーム (2006年?~)
- 非プログラマーによる利用



- ・ 習得の容易性
- ・ 豊富なライブラリー
- ・ 活用範囲の広さ

2018年ごろから急激な増加

→ Java に変わる言語として注目か

※1) TIOBE のソース
条件を満たした 25の検索エンジンのヒット数
(詳細は以下のリンクを参照)

<https://www.tiobe.com/tiobe-index/>

<https://www.tiobe.com/tiobe-index/programming-languages-definition/>



Web アプリケーション

Python で Web アプリケーションを実現するには

■ 簡単な方法 : CGI

cgi_db.py

```
1 #!/QOpenSys/pkgs/bin/python
2 import sys
3 import io
4 import ibm_db_dbi as db
5 sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
6
7 print("Content-type:text/html; charset=utf-8;\n")
8 conn = db.connect("DATABASE=*LOCAL", "KIT", "KITA")
9 cursor = conn.cursor()
10 cursor.execute("SELECT * FROM KIT.TOKMSP")
11 ret = ""
12 for r in cursor.fetchall():
13     print(r[0] + ' ' + r[1] + ' ' + r[2] + '<br>')
14 conn.close()
15
```

ファイルの配置

```
/home/KIT
|-- cgi-bin
|   |-- cgi_db.py
```

注意点

- ・cgi-bin ディレクトリにプログラムは配置する
- ・コードの先頭に以下を記載する
#!/QOpenSys/pkgs/bin/python

Webサーバーの実行

```
[bash-5.1$ cd /home/KIT
[bash-5.1$ python -m http.server 8085 --cgi
Serving HTTP on 0.0.0.0 port 8085 (http://0.0.0.0:8
```

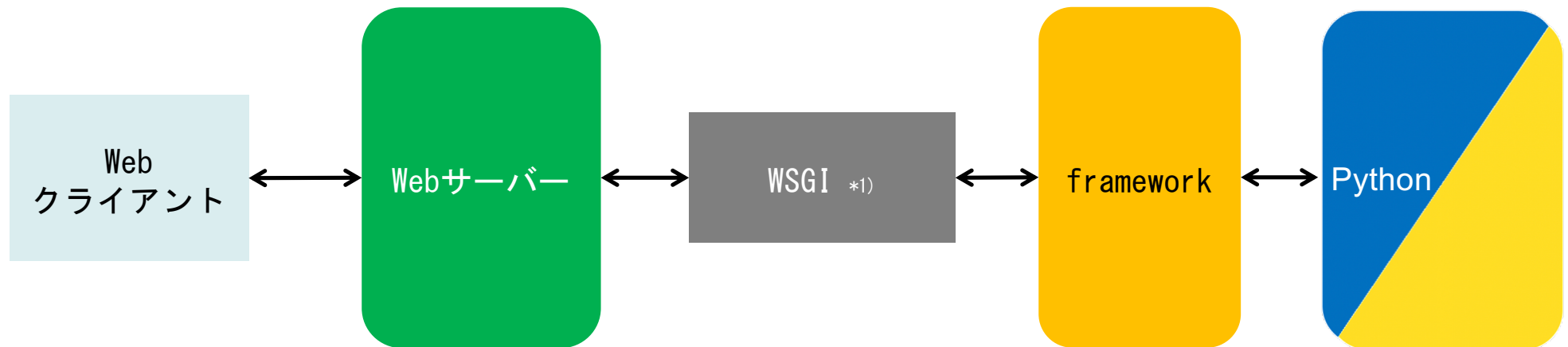
← → ↻ ▲ 保護されていない通信 | 172.23.0.236:8085/cgi-bin/cgi_db.py

```
01010 アイリコフ 日本語です。
01020 アイコギョウ 阿井工業
01040 アイリコフヤ 阿井旅行社
01070 アイカカメラ 相川カメラ
01080 アイカコウカクK.K 相川広告K. K
01150 アイカウユK.K 相川運輸K. K
01230 アイカコウカフヤ 相川広告社
01240 アイカホウカクK.K 相川包装K. K
01250 アイデアサキョウK.K アイデア産業K. K
01280 アイデアエッセイK.K アイデア協会K. K
```

簡単にWeb化できる

Python で Web アプリケーションを実現するには

■ 実際の Python Web アプリケーション構成

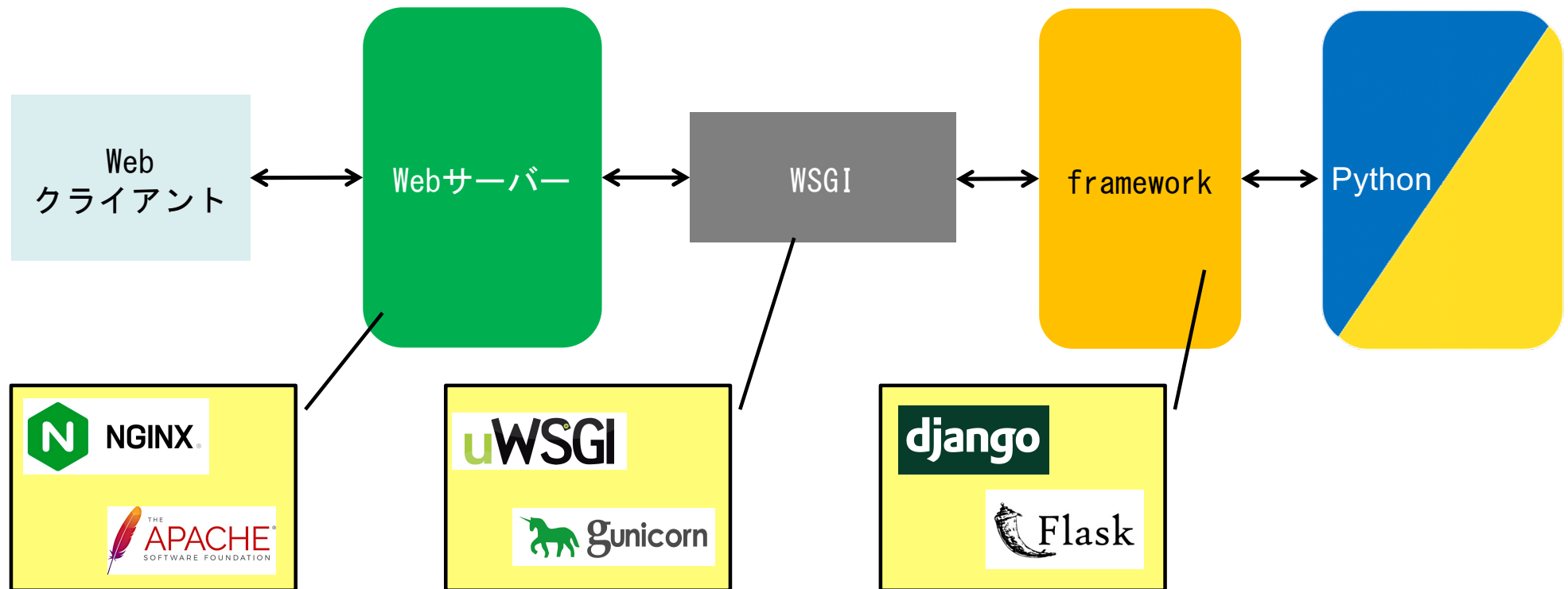


*1) Web Server Gateway Interface

WebサーバーとWebアプリケーションを接続するための標準化されたインターフェイス

Python で Web アプリケーションを実現するには

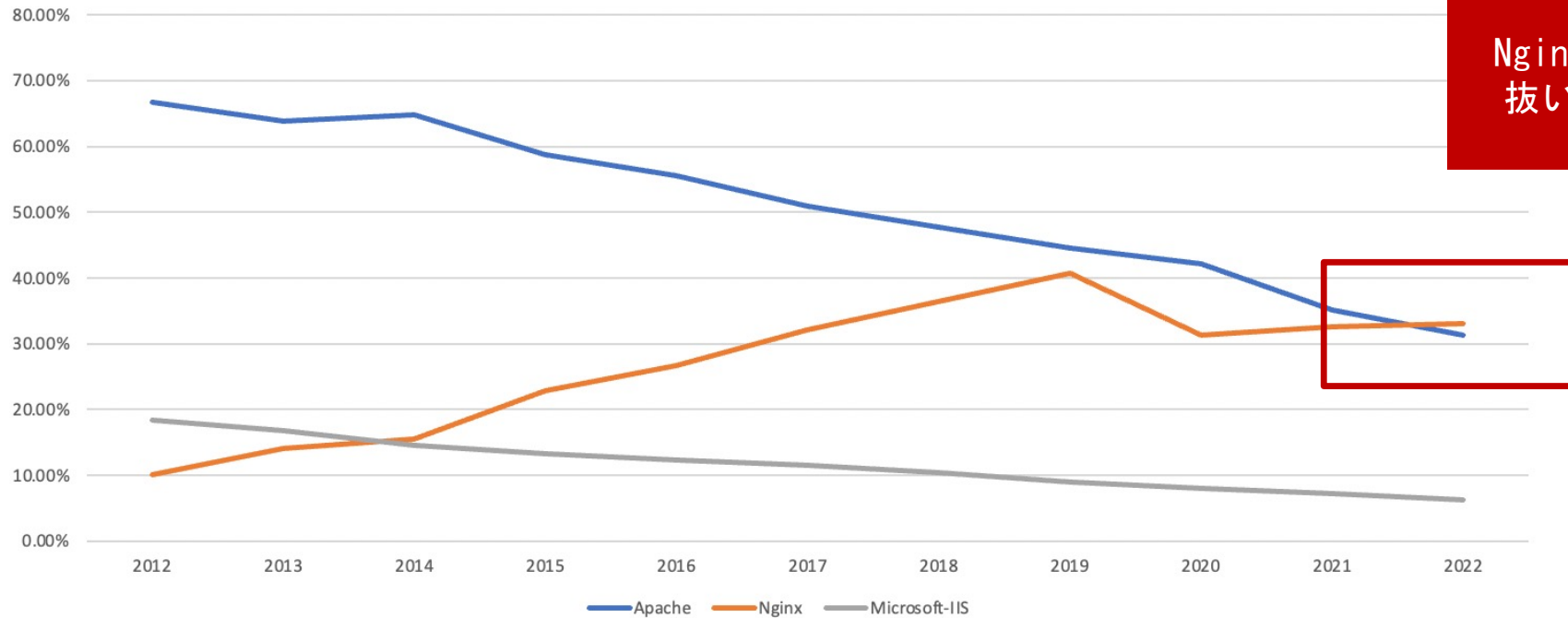
■ Python Web アプリケーション構成



参考：Webサーバーのシェア

■ 世界の Webサーバー 市場シェア 2012年～2022年

Historical yearly trends in the usage statistics of web servers



Nginx が Apache を抜いて首位に！！

参考：W3Techs web technology surveys

- https://w3techs.com/technologies/history_overview/web_server/ms/y

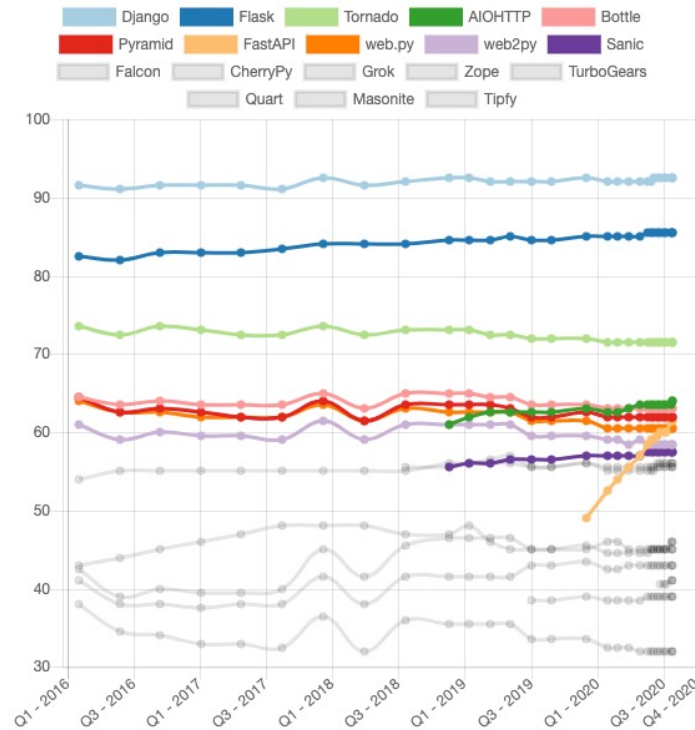
※注 グラフは情報の一部を抜粋しグラフ化したものです。

参考 : Python フレームワークのシェア

■世界の Python Framework 市場シェア 2016年～2022年

Python

Framework	Score
Django	92
Flask	85
Tornado	71
AIOHTTP	64
Bottle	63
Pyramid	62
FastAPI	61
web.py	60
web2py	58
Sanic	57
Falcon	56
CherryPy	55
Grok	46
Zope	45
TurboGears	43



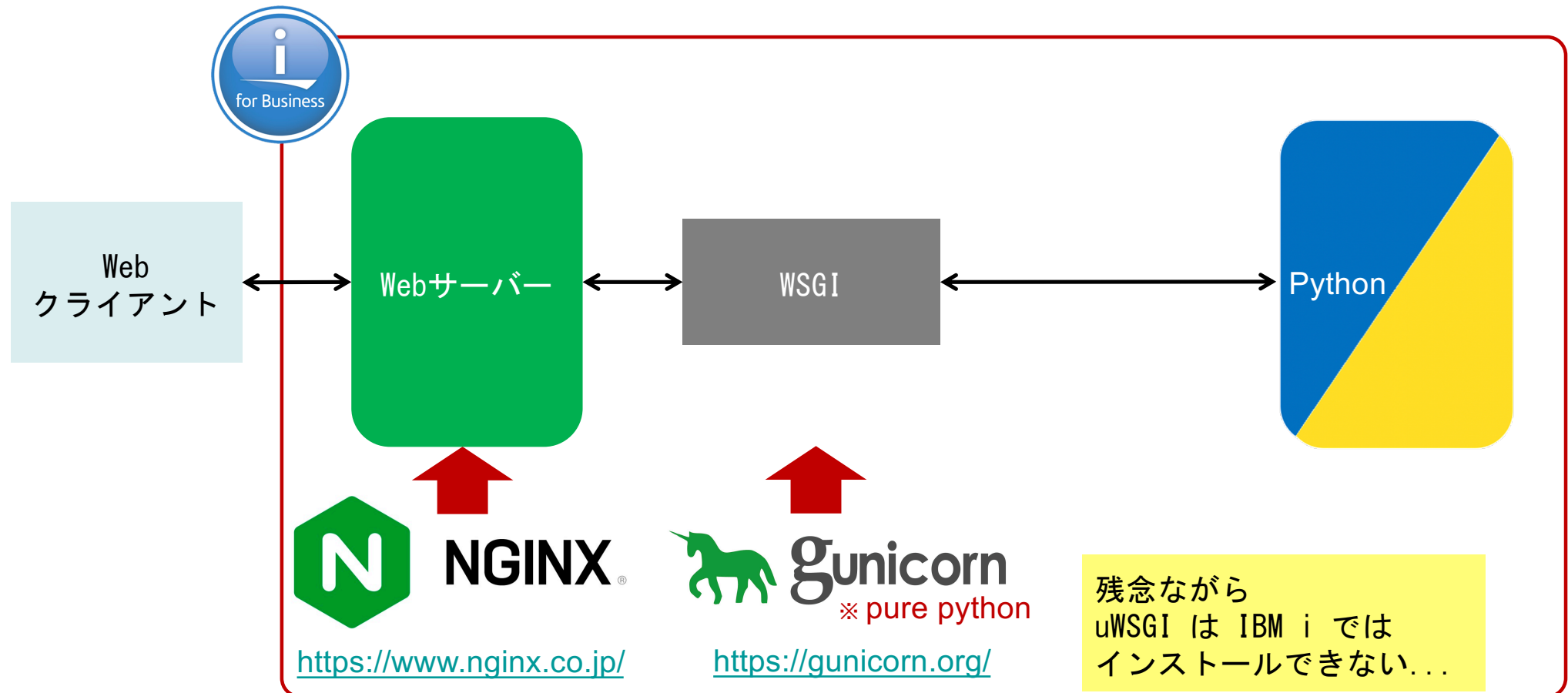
2022年現在

Framework	Score
Django	93
Flask	86
Tornado	71
FastAPI	66
AIOHTTP	64
Bottle	63
Pyramid	62
web.py	60
Sanic	58
web2py	58
CherryPy	55
Falcon	55
Grok	46
Zope	45
TurboGears	43
Quart	42
Masonite	39
Tipfy	32

参考: HotFrameworks - <https://hotframeworks.com/languages/python/>

Python で Web アプリケーションを実現するには

■ 今回試した構成1



Python で Web アプリケーションを実現するには

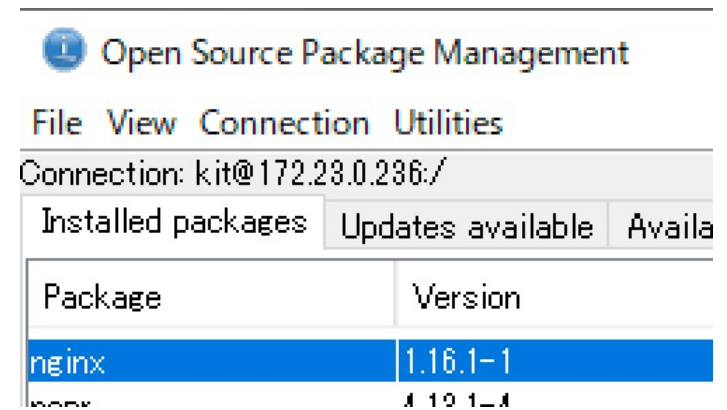
■ Nginx インストール

- ACS の場合
 - ACS で nginx を選択し Install
- yum コマンドの場合
 - yum install nginx

■ Gunicorn インストール

- pip コマンド *2)
 - pip install gunicorn

```
bash-4.4$ pip install gunicorn
Collecting gunicorn
  Downloading https://files.pythonhosted.org/packages/69/ca/926f7cd3a2014b16870086b2d0fdc84a9e49473c68a8dff8b57f7c156f43/gunicorn-20.0.4-py2.py3-none-any.whl (77kB)
    100% |#####| 81kB 1.2MB/s
Requirement already satisfied: setuptools>=3.0 in /QOpenSys/pkgs/lib/python3.8/site-packages (from gunicorn)
Installing collected packages: gunicorn
Successfully installed gunicorn-20.0.4
```



Open Source Package Management

File View Connection Utilities

Connection: kit@172.23.0.236:/

Package	Version
nginx	1.16.1-1
ssm	1.10.1-1

*2) Pip Install Package : Python のパッケージ管理システム

Python で Web アプリケーションを実現するには

■ 設定

■ nginx.conf

```
worker_processes 1;

events {
    worker_connections 512;
}

http {
    server {
        listen 9123;
        server_name INFRA-PRACTICE-NGINX;
        charset UTF-8;

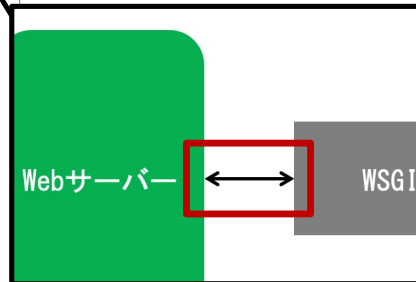
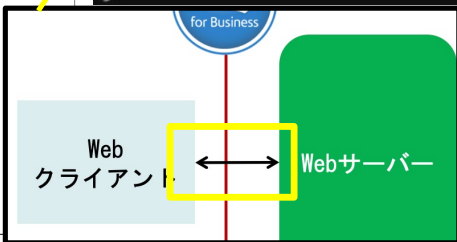
        proxy_set_header    Host    $host;

        location / {
            proxy_pass http://127.0.0.1:9876;
        }
    }
}
```

■ gsettings.py

```
import os

bind = '127.0.0.1:' + str(os.getenv('PORT', 9876))
proc_name = 'Infrastructure-Practice'
workers = 1
```



各ファイルの配置

```
/home/KIT/gunicorn
|-- server
|   |-- gsettings.py
|   |-- nginx.conf
```

Python で Web アプリケーションを実現するには

gni_db.py

```
1 import ibm_db_dbi as db
2
3 def application(env, start_response):
4     start_response('200 OK', [('Content-Type', 'text/html')])
5     conn = db.connect("DATABASE=*LOCAL","KIT","KITA")
6     cursor = conn.cursor()
7     cursor.execute("SELECT * FROM QEOL.TOKMSP")
8     ret = ""
9     for r in cursor.fetchall():
10         ret += r[0] + ' ' + r[1] + ' ' + r[2] + '</br>'
11     return [bytes(ret,'utf-8')]
12
```

各ファイルの配置

```
/home/KIT/gunicorn
|-- gni_db.py
|-- server
|   |-- gsettings.py
|   |-- nginx.conf
```


Python で Web アプリケーションを実現するには

■ Nginx の起動

```
$ nginx -c /home/KIT/gunicorn/server/nginx.conf
```

■ Gunicorn (wsgi) の起動

```
$ cd /home/KIT/gunicorn/  
$ gunicorn gni_db:application -c server/gsettings.py
```

各ファイルの配置

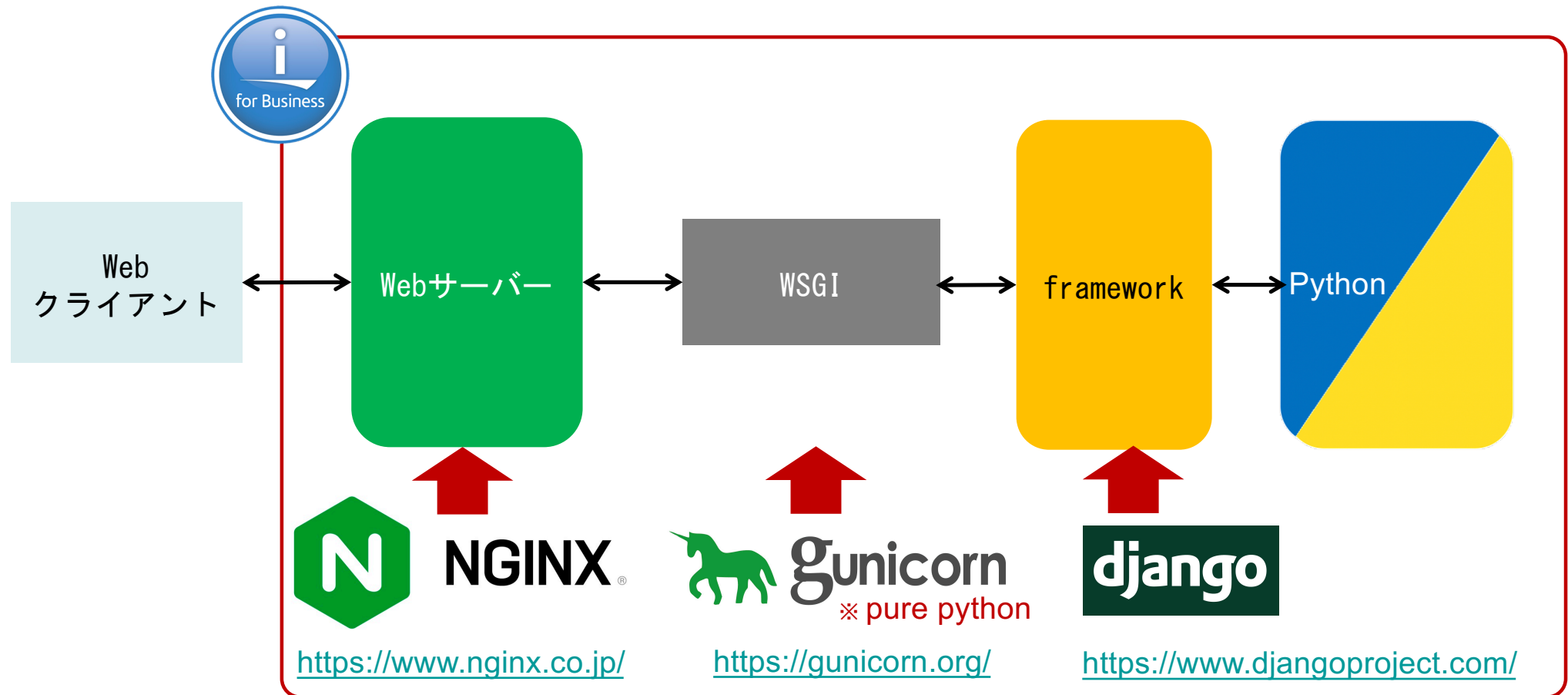
```
/home/KIT/gunicorn  
|-- gni_db.py  
|-- server  
|   |-- gsettings.py  
|   |-- nginx.conf
```

← → ↻ ▲ 保護されていない通信 | 172.23.0.236:9123

01010 アイリオン 阿井旅館
01020 アイコギョ 阿井工業
01030 アイカコギョ 相川工業
01040 アイリョウヤ 阿井旅行社
01050 アイショクダウK.K 阿井食品K. K
01060 アイジドウヤ 阿井自動車
01070 アイカカメラ 相川カメラ
01080 アイカコウカK.K 相川広告K. K
01090 アイカデンキK.K 相川電機K. K
01100 アイカガキヤ 相川製菓

Python で Web アプリケーションを実現するには

■ 今回試した構成2



Python で Web アプリケーションを実現するには

■ Django インストール

- pip コマンド

- pip install django

```
bash-5.1$ pip install Django
Collecting Django
  Downloading Django-3.2.12-py3-none-any.whl (7.9 MB)
    |#####| 7.9 MB 5.9 MB/s
Requirement already satisfied: pytz in /QOpenSys/pkgs/lib/python3.6/site-packages (from Django) (2018.5)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
    |#####| 42 kB 252 kB/s
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Requirement already satisfied: typing-extensions in /QOpenSys/pkgs/lib/python3.6/site-packages (from asgiref<4,>=3.3.2->Django) (3.7.4.3)
Installing collected packages: sqlparse, asgiref, Django
Successfully installed Django-3.2.12 asgiref-3.4.1 sqlparse-0.4.2
WARNING: Running pip as root will break packages and permissions. You should install packages reliably by using venv: https://pip.pypa.io/warnings/venv
```

Python で Web アプリケーションを実現するには

■ Django プロジェクト作成

- `django-admin.py startproject <プロジェクト名>`

```
$ django-admin.py startproject django01
```

- 作成されるディレクトリ構造

```
/home/KIT/django
|-- django01
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
`-- manage.py
```

※一部抜粋

Python で Web アプリケーションを実現するには

■ Django アプリケーション作成

- 作成したプロジェクトディレクトリで以下を実行
- `python manage.py startapp <アプリケーション名>`

```
$ cd django01/  
$ python manage.py startapp app_django01
```

- 作成されるディレクトリ構造

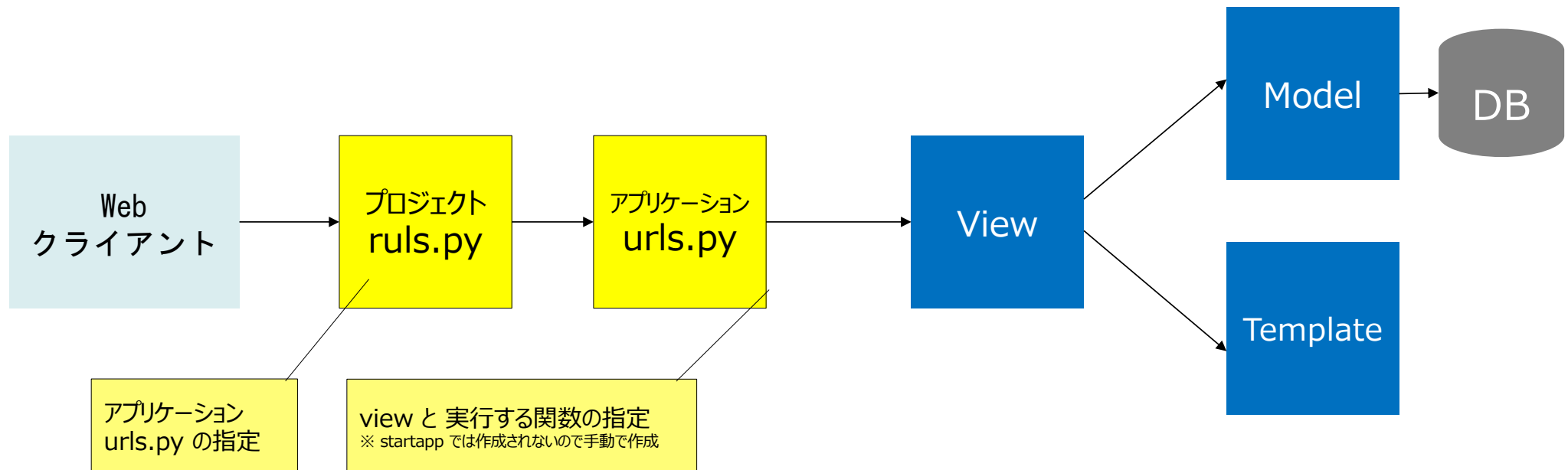
```
/home/KIT/django  
|-- app_django01  
|   |-- models.py  
|   `-- views.py
```

※一部抜粋

Python で Web アプリケーションを実現するには

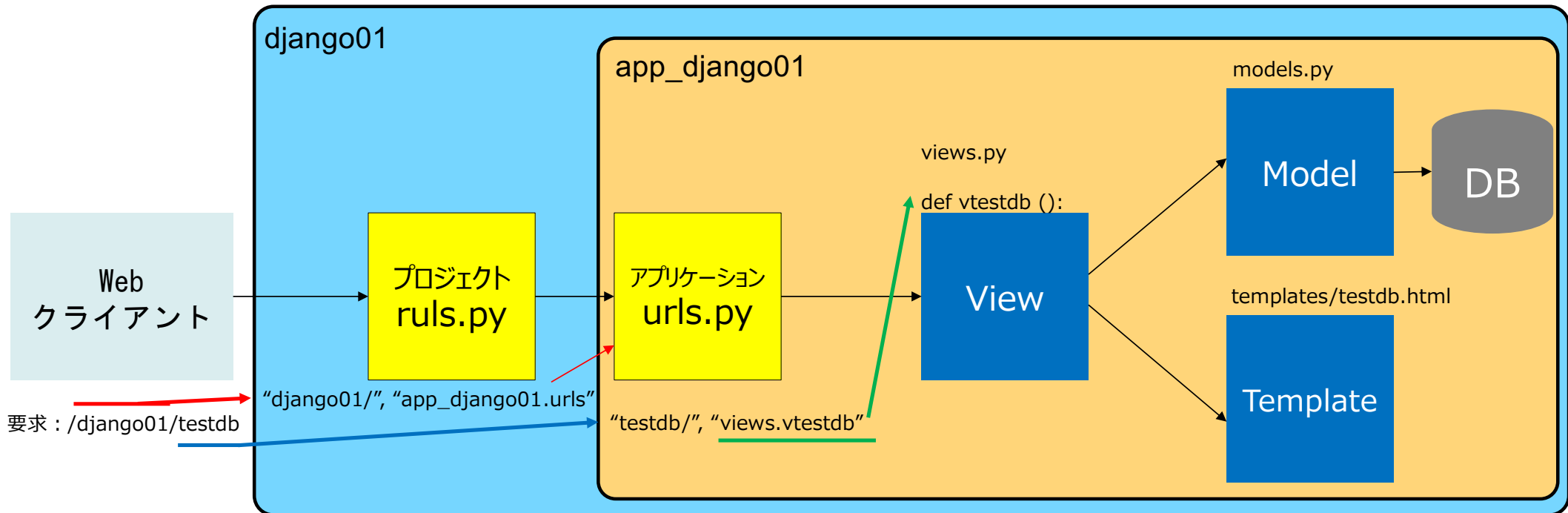
■ Django の要点

- ルーティングの設定：クライアントからの呼び出し方を設定
- MVT (Model-View-Template)：プログラム構造の考え方
 - MVC (Mode-View-Controller) と対比すると... M=M, V=T, C=V



Python で Web アプリケーションを実現するには : ルーティング例

- 例 : プロジェクト=django01, アプリケーション=app_django01

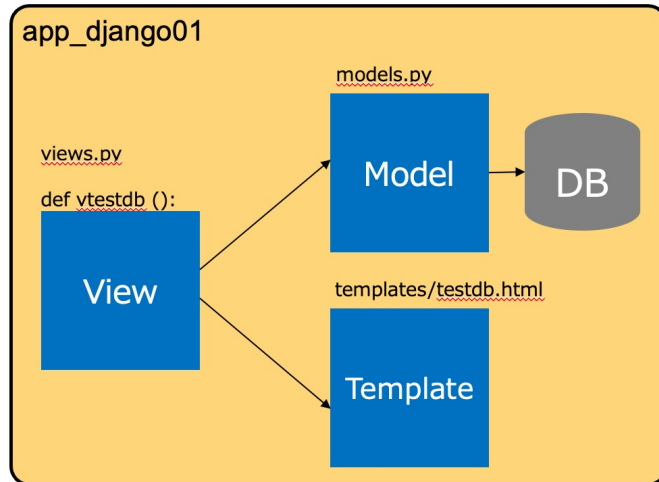


ルーティングの流れ

1. 要求を受ける
2. プロジェクトの url.py が参照 (django01/urls.py)される
3. "django01/" でアプリケーション ruls.py の参照ファイル (app_django01/urls.py)が決まる
4. アプリケーションの urls.py さ参照され "testdb/" で実行する view ファイルと関数 (views.py の vtestdb()関数)が決まる

Python で Web アプリケーションを実現するには : プログラム例

- 例 : プロジェクト=django01, アプリケーション=app_django01



各ファイルの配置

```
|-- app_django01
|   |-- models.py
|   |-- templates
|   |   |-- testdb.html
|   |-- urls.py
|   |-- views.py
```

※一部抜粋

※ templates ディレクトリは手動で作成

views.py

```
1 from django.shortcuts import render
2 from app_django01.models import Tokmsp
3
4 def vtestdb(request):
5     tokmsp = Tokmsp()
6     rec = tokmsp.getTokmsp()
7     return render(request, 'testdb.html', {"recs":rec})
```

models.py

```
1 from django.db import models
2 import ibm_db_dbi as db
3
4 class Tokmsp:
5     def getTokmsp(self):
6         conn = db.connect("DATABASE=*LOCAL", "KIT", "KITA")
7         cursor = conn.cursor()
8         cursor.execute("SELECT * FROM KIT.TOKMSP")
9         ret = cursor.fetchall()
10        return ret
```

templates/testdb.html

```
1 {% for rec in recs %}
2     {{ rec.0 }} {{ rec.1 }} {{ rec.2 }} </br>
3 {% endfor %}
```


Python で Web アプリケーションを実現するには : 全体像

各ファイルの配置

```
|-- django
|   |-- django01
|   |   |-- app_django01
|   |   |   |-- models.py
|   |   |   |-- templates
|   |   |   |-- testdb.html
|   |   |   |-- urls.py
|   |   |   |-- views.py
|   |   |-- django01
|   |   |   |-- settings.py
|   |   |   |-- urls.py
|   |   |   |-- wsgi.py
|   |   |-- manage.py
|   |-- server
|   |   |-- gsettings.py
|   |   |-- nginx.conf
```

※一部抜粋

settings.py ※修正点のみ抜粋

```
28 ALLOWED_HOSTS = ["*"]
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'app_django01',
41 ]
```

※ wsgi.py は変更しないが、利用する

gsettings.py ※修正点のみ抜粋

```
1 import os
2
3 bind = '127.0.0.1:' + str(os.getenv('PORT', 9876))
4 proc_name = 'Infrastructure-Practice'
5 workers = 1
```

nginx.conf

```
1 worker_processes 1;
2 error_log /home/KIT/django/server/error.log warn;
3 pid /home/KIT/django/server/nginx.pid;
4
5 events {
6     worker_connections 512;
7 }
8
9 http {
10     server {
11         listen 9123;
12         server_name INFRA-PRACTICE-NGINX;
13         charset UTF-8;
14
15         proxy_set_header host $host;
16
17         location / {
18             proxy_pass http://127.0.0.1:9876;
19         }
20     }
21 }
```

Python で Web アプリケーションを実現するには

■ Nginx の起動

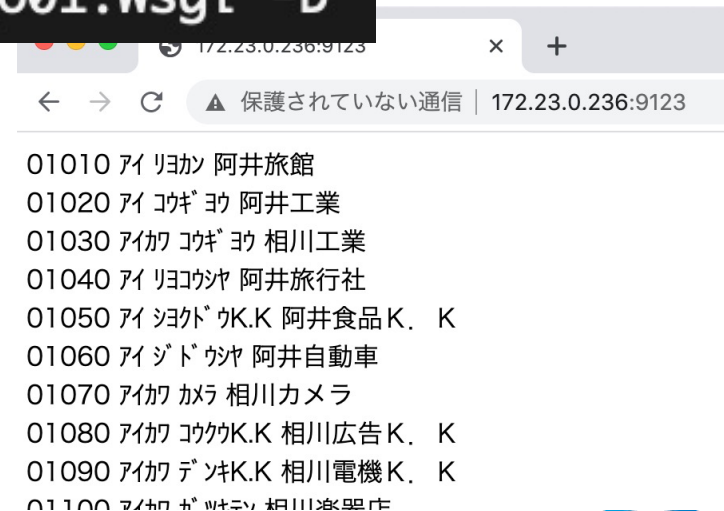
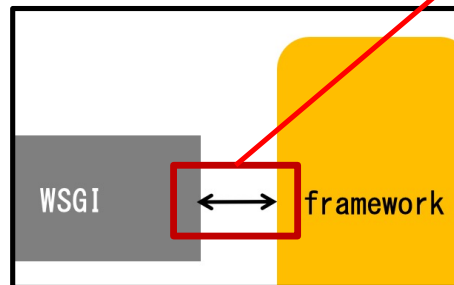
```
$ nginx -c /home/KIT/django/server/nginx.conf
```

■ Gunicorn (wsgi) の起動

```
$ cd /home/KIT/django  
$ gunicorn -c ../server/gsettings.py django01.wsgi -D
```

```
-- urls.py  
-- wsgi.py  
-- manage.py
```

※ wsgi.py は変更しないが、利用する





おまけ

IBM i の オープンソース環境をブラウザ(Chrome)で利用

参考 : Chrome secure shell

参考 : Chrome Secure Shell

■ Chrome Secure Shell の利用

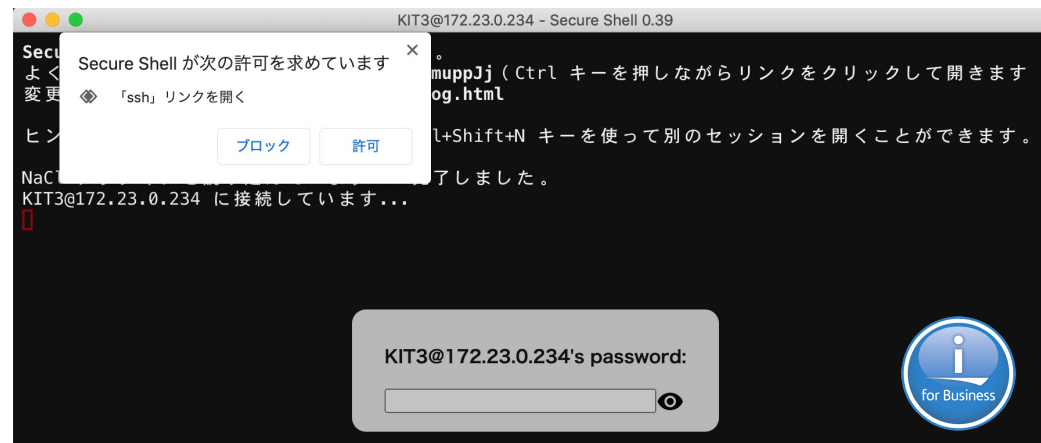
1. IBM i にユーザーのホームディレクトリを作成（作成済みの場合は不要、`MKDIR DIR('/home/<ユーザー>')`）
2. IBM i で SSHD を開始（開始済みの場合は不要）
3. Chrome ブラウザで「chrome ウェブストア」を表示
 - <https://chrome.google.com/webstore>
4. 「ストアを検索」に “secure shell app” と入力し Enter
5. 一覧から “Secure Shell” (以下のアイコン)をクリック
6. 「Chrome に追加」をクリック
7. 確認画面で「拡張機能を追加」をクリック
8. Chrome の右上にある「拡張機能アイコン」をクリック
9. 表示された「接続ダイアログ」をクリック
10. ユーザー、IBM I の IPアドレス、ポート(22)を入力し [ENTER]接続
11. 接続後、パスワードを入力（初回のみ接続確認があるので yes で回答）
12. 環境の作成（初回のみ）

参考 : Chrome Secure Shell

[補足] 10. ユーザー、IBM I の IPアドレス、ポート(22)を入力し [ENTER]接続



The screenshot shows the Chrome Secure Shell configuration interface. It includes a text input field for the host address containing 'KIT3@172.23.0.234'. Below this are fields for the username 'KIT3', the IP address '172.23.0.234', and the port '22'. There is a section for 'SSH 中継サーバーのオプション' with a dropdown menu for 'ID' set to '[default]' and an 'インポート...' button. Below that are fields for 'SSH 引数' (set to 'その他のコマンドライン引数') and '現在のプロファイル' (set to 'デフォルト'). At the bottom, there are buttons for '[DEL] 削除', 'オプション', 'フィードバックを送信', 'SFTP', and '[ENTER] 接続'.



参考 : Chrome Secure Shell

[補足] 12. 環境の作成(初回のみ)

事前に以下を対応しておくとう使いやすいです。

※ Secure shell にログイン後以下を実行

```
$ pwd
```

```
/home/KIT3
```

作成したホームディレクトリになっている事を確認

```
$ echo bash > .profile
```

bash シェルを利用するための設定 (.profile)

```
$ echo alias python=python3 > .bashrc
```

以下、python を利用するための設定 (.bashrc)

```
$ echo alias pip=pip3 >> .bashrc
```

```
$ echo PATH=/QOpenSys/pkg/bin:/QOpenSys/pkg/sbin:$PATH >> .bashrc
```

実行後、以下の様に cat コマンドで作成したファイル (.profile, .bashrc) を確認

```
$ cat .profile
```

```
bash
```

```
$ cat .bashrc
```

```
alias python=python3
```

```
alias pip=pip3
```

```
PATH=/QOpenSys/pkg/bin:/QOpenSys/pkg/sbin:/QOpenSys/usr/bin:/usr/ccs/bin:/QOpenSys/usr/bin/X11:
```

```
/usr/sbin:./usr/bin
```

exit コマンドで一度終了し、再度接続する事で、作成した環境設定が反映されます。

環境設定の適用後は終了する際、exit を2回実行します(bashの終了とシステムからのログアウト)



IBM i のオープンソース環境を無料で利用できる Microsoft Visual Studio Code で利用

参考 : VSCode

参考 : VSCode (Microsoft Visual Studio Code)

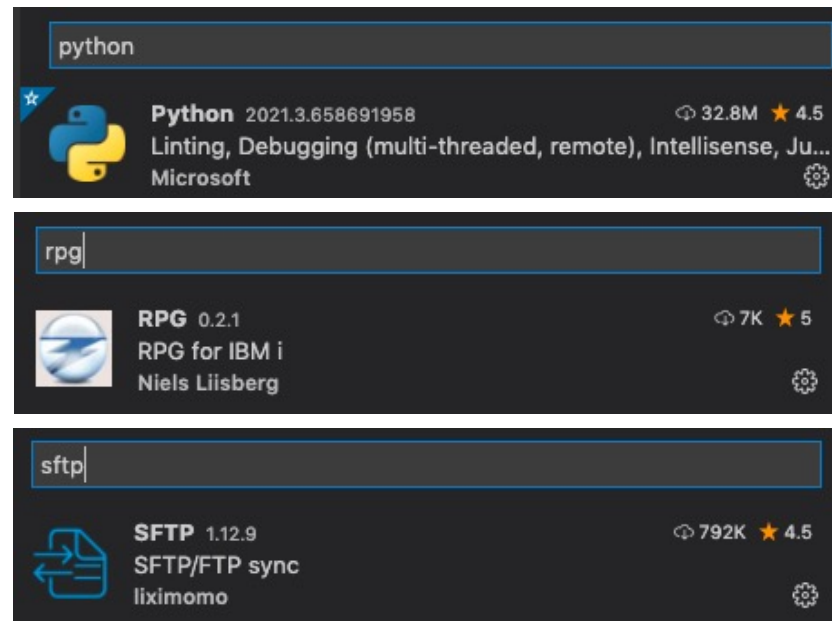
■ VSCode の利用

1. VSCode のインストール (無償)

- ダウンロード : <https://code.visualstudio.com/>

2. 拡張機能を導入

- python
- rpgle
- sftp



参考 : VSCode (Microsoft Visual Studio Code)

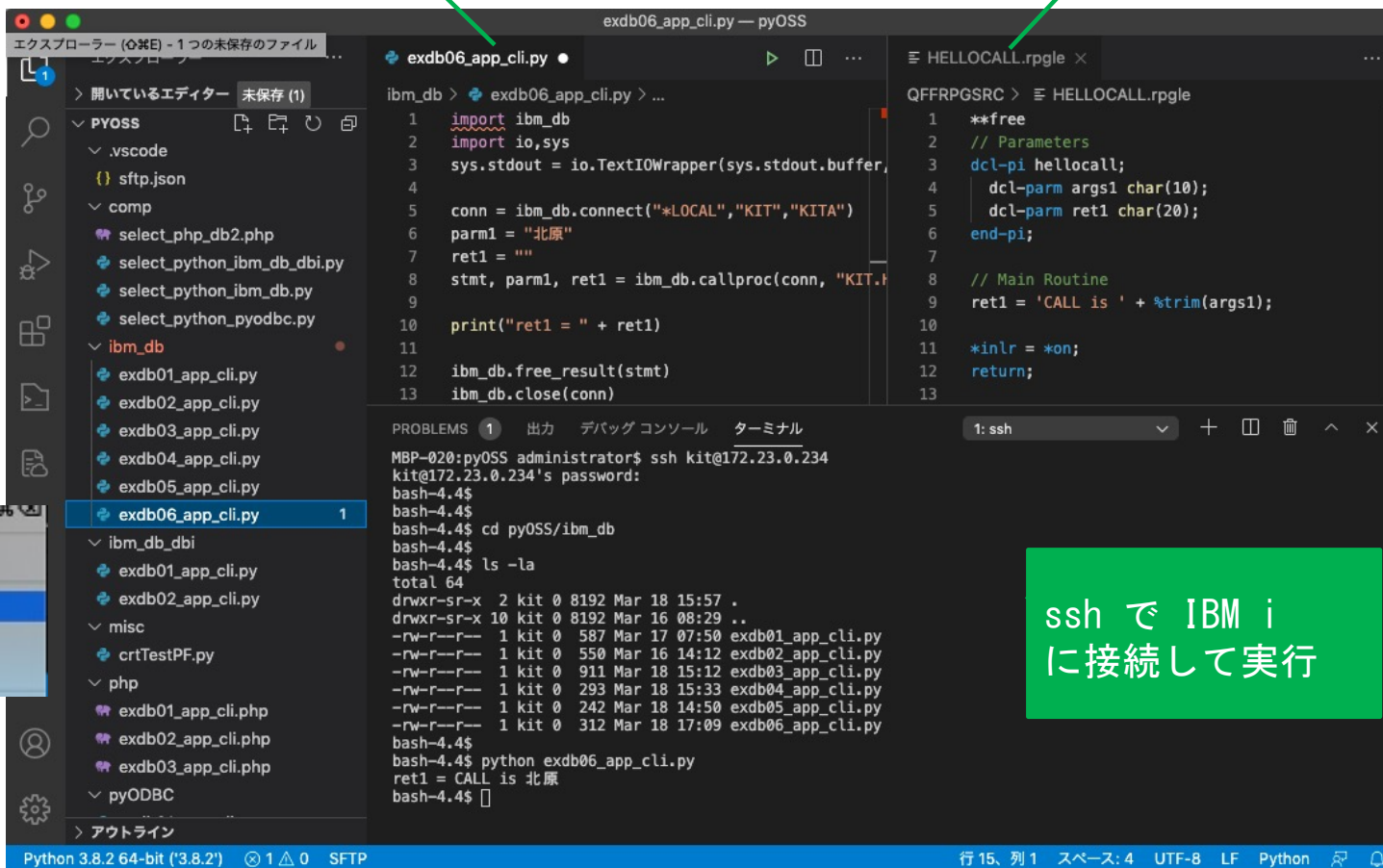
■ 利用のイメージ

python のコーディング

free form RPG のコーディング

右クリックで
IFS に配置
(sftp)

Create Maven Project
Sync Local -> Remote
Sync Remote -> Local
Sync Both Directions



ssh で IBM i
に接続して実行

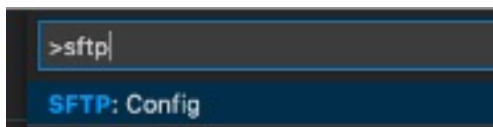
参考 : VSCode (Microsoft Visual Studio Code)

■ sftp の設定例

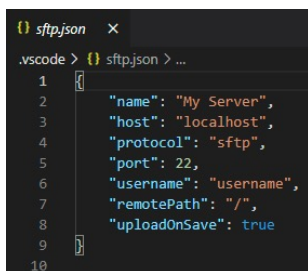
1. 管理→コマンドパレット



2. 画面上部に表示された入力域に “sftp” と入力し、sftp:config をクリック



3. 設定画面が表示される



参考 : VSCode (Microsoft Visual Studio Code)

■ sftp の設定例

```
1 {
2   "name": "OSS Python Subcommittee",
3   "host": "172.23.0.234",
4   "protocol": "sftp",
5   "port": 22,
6   "username": "KIT",
7   "password": "<password>",
8   "remotePath": "/home/KIT/pyOSS/",
9   "ignore": [
10    ".vscode",
11    ".git",
12    ".DS_Store",
13    ".svn"
14  ],
15   "syncOption": {
16     "delete": false,
17     "skipCreate": false,
18     "ignoreExisting": false,
19     "update": true
20   },
21   "uploadOnSave": false,
22   "watcher": {
23     "files": "/Users/kitahara/Documents/pyOSS/*",
24     "autoUpload": false,
25     "autoDelete": false
26   }
27 }
```

IBM i の情報

ローカルPCの情報