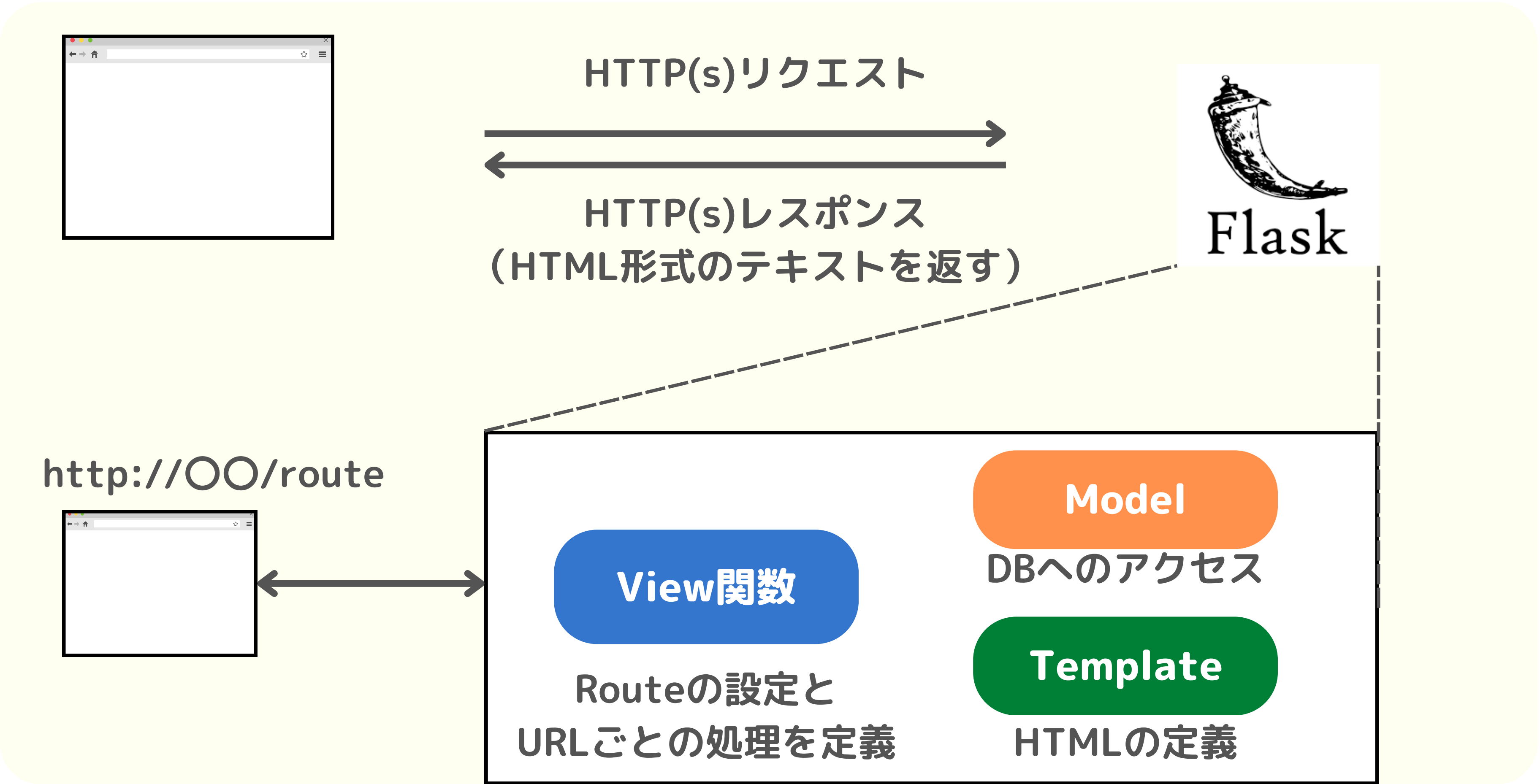
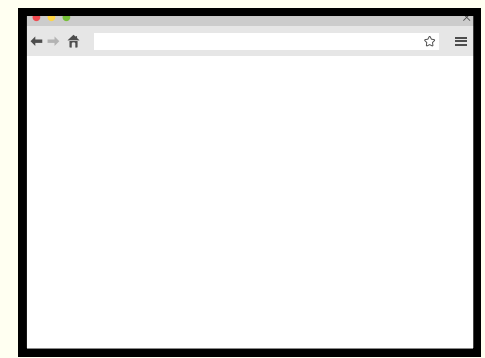


# Flaskの基本的な仕組み



# Flaskの基本的な仕組み



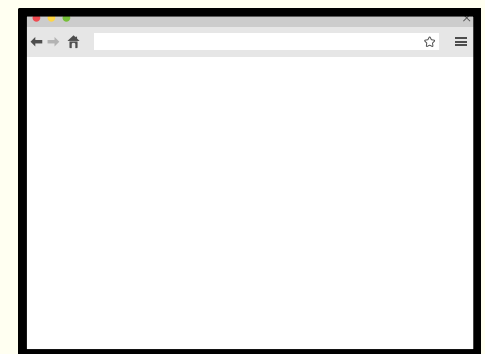
http://〇〇〇〇.〇〇〇〇/home



Flask

View関数  
(home)

View関数  
(login)



http://〇〇〇〇.〇〇〇〇/login



Flask

View関数  
(home)

View関数  
(login)

# Routeの設定

Flaskのルートの設定は以下のようにする

```
from flask import Flask
```

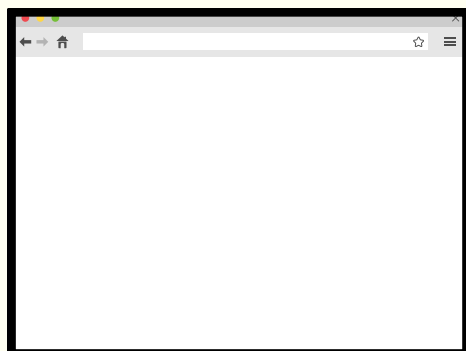
```
# アプリケーション立ち上げのインスタンス(__name__は、テンプレート読み込みの時などに内部で使用)  
app = Flask(__name__)
```

**@app.route():** デコレータ関数(後続の関数に処理を追加)

```
@app.route('/hello')
```

```
def info():  
    return "<h1>Hello world</h1>"
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```



http://〇〇/  
http://〇〇/hello



Flask

**info**関数を実行

# 変数を用いる

動的なページを作成(URLに応じてページが変わる)

```
@app.route('/page/<variable>'):  
def function(variable):  
    return variable
```

http://〇〇〇.〇〇〇/page/abc  
と  
http://〇〇〇.〇〇〇/page/def  
で異なるページを表示

```
@app.route('/post/<int:post_id>')  
def show_post(post_id):  
    return 'Post %d' % post_id
```

# <int:変数名>とすると、int型以外は  
ページに遷移されない

string	デフォルト、スラッシュなし文字列
int	正の整数
float	正の浮動小数点数
uuid	UUID文字列 オブジェクトを一意に識別するための識別子 フォーマット: xxxxxxxx-xxxx-xxxx-xxxx- xxxxxxxxxxxxxx パスワードの再設定のURLに用いることもある
path	スラッシュあり文字列

# Flaskの起動

# app.runを書いているFlaskファイルを実行

```
python XX.py
```

# 環境変数(FLASK\_APPにXX.pyを設定)して、flask run を実行

windowsの場合、set FLASK\_APP=XX.py

mac, Linuxの場合、export FLASK\_APP=XX.py

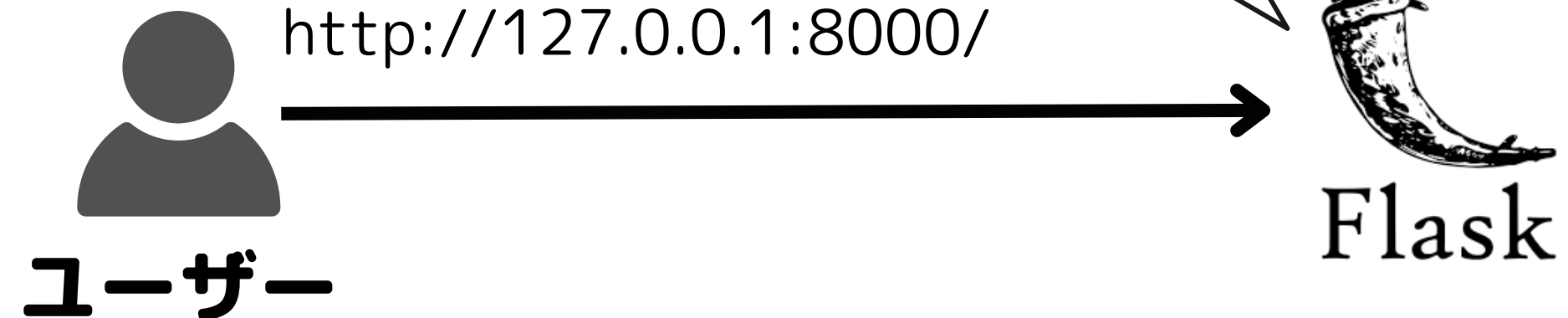
# 待ち受けるホスト名、ポート番号の変更(デフォルトは5000), 0.0.0.0はそのサーバーの全てのネットワークインターフェースで待ち受けることを意味する

```
app.run(host='0.0.0.0', port=80)
```

```
(export FLASK_RUN_HOST=0.0.0.0  
export FLASK_RUN_PORT=5000でも可)
```

# デバッグモード

```
app.run(debug=True)
```



# 演習

以下のようなサイトを作成してください

/hello: でアクセスすると<h1>こんにちは</h1>と表示される

/hello/文字列1/文字列2: でアクセスすると<h1>こんにちは文字列1さん文字列2さん</h1>と表示される

/add/num1/num2: でアクセスすると $\text{num1} + \text{num2}$ の値が表示される(ただし、num1とnum2は整数固定)

/div/num1/num2: でアクセスすると $\text{num1} / \text{num2}$ を小数点以下切り下げた値が表示される(ただし、num1とnum2は浮動小数点数)