サイバー大学IT総合学部
専門応用科目
JavaScriptフレームワークによるWebプログラミング

第5回
Express実習2

小薗井康志

第5回　学習目標

- Expressのルーティングについて理解し説明できる
- Expressで新しいページを作成できる

第5回　授業構成

- 第1章　Expressルーティング概要
- 第2章　Expressルーティング実習

JavaScriptフレームワークによるWebプログラミング
第5回 Express実習2

第1章
Expressルーティング概要

第1章　学習目標

- Expressのルーティングについて理解し説明できる

Express前回のおさらい

- 作業内容
    - Express generatorでアプリを作成
    - ライブラリをインストールしてローカル環境で実行
    - ブラウザで動作確認
    - ソースコードをGitHubにアップロード
    - サーバにログイン
    - サーバにGithubからソースコードをClone
    - サーバでアプリを実行、動作確認

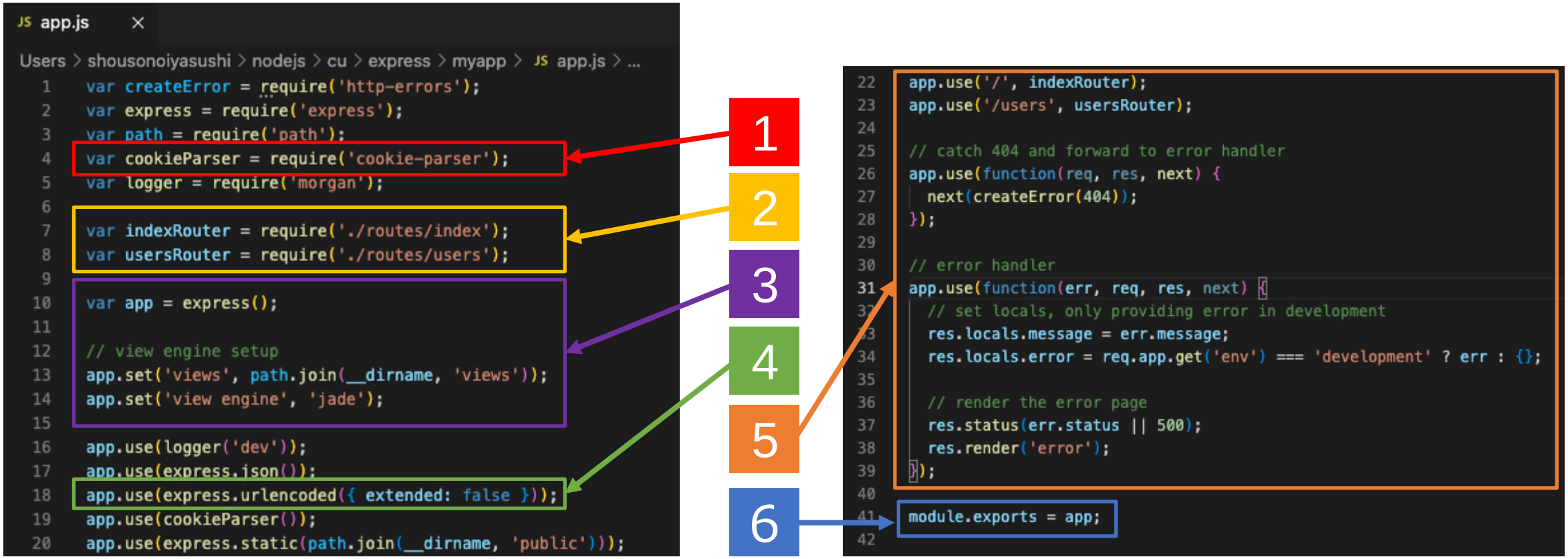
Express Generatorで作成されたファイルを確認（app.js）

・このアプリのメインとなるプログラム

主な流れ

- 1.必要なモジュールをロード
- 2.ルート用モジュールのロード
- 3.Express オブジェクトの作成と基本設定
- 4.関数の組み込み
- 5.ルート用、エラー用のapp.use
- 6.Module.expressの設定

Express Generatorで作成されたファイルを確認（app.js）



Express Generatorで作成されたファイルを確認（app.js）

1.必要なライブラリーをロード

- http-errors: HTTPエラーの対処を行うもの
- express: express本体
- path: ファイルパスを扱う
- cookie-parser: クッキーのパース（値の変換）
- morgan: HTTPリクエストのログ出力に関するもの

Express Generatorで作成されたファイルを確認（app.js）

2.ルート用モジュールのロード

```
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
```

- ルートごとに用意されているプログラムをロードする作業
- routesフォルダにindex.js, users.jsがあり、それをモジュールとしてロードしている。
- index.jsは/indexにアクセスした時の、users.jsは/usersにアクセスした時の処理がまとめられている。

Express Generatorで作成されたファイルを確認（app.js）

3.Express オブジェクトの作成と基本設定

```
var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');
```

- Expressのオブジェクトを作成
  またテンプレートエンジンの設定も行っている
  ここではjadeというview engineを使用

Express Generatorで作成されたファイルを確認（app.js）

4.関数の組み込み

```
app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
```

- 1でロードしたモジュールの機能を呼び出せるようにしたもの
  app.useで関数を設定

```
var cookieParser = require('cookie-parser');
```

```
app.use(cookieParser());
```

Express Generatorで作成されたファイルを確認（app.js）

5.ルート用、エラー用のapp.use

```
app.use('/', indexRouter);
app.use('/users', usersRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});
```

- 上の2つはそれぞれ/(ルート)、/users用

Express Generatorで作成されたファイルを確認（app.js）

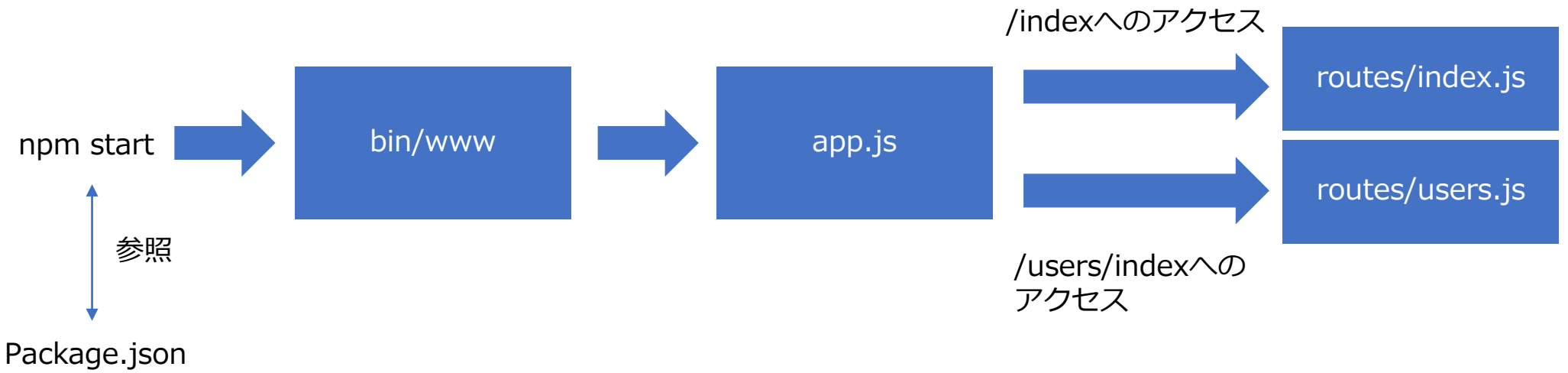
6.Module.expressの設定

```
module.exports = app;
```

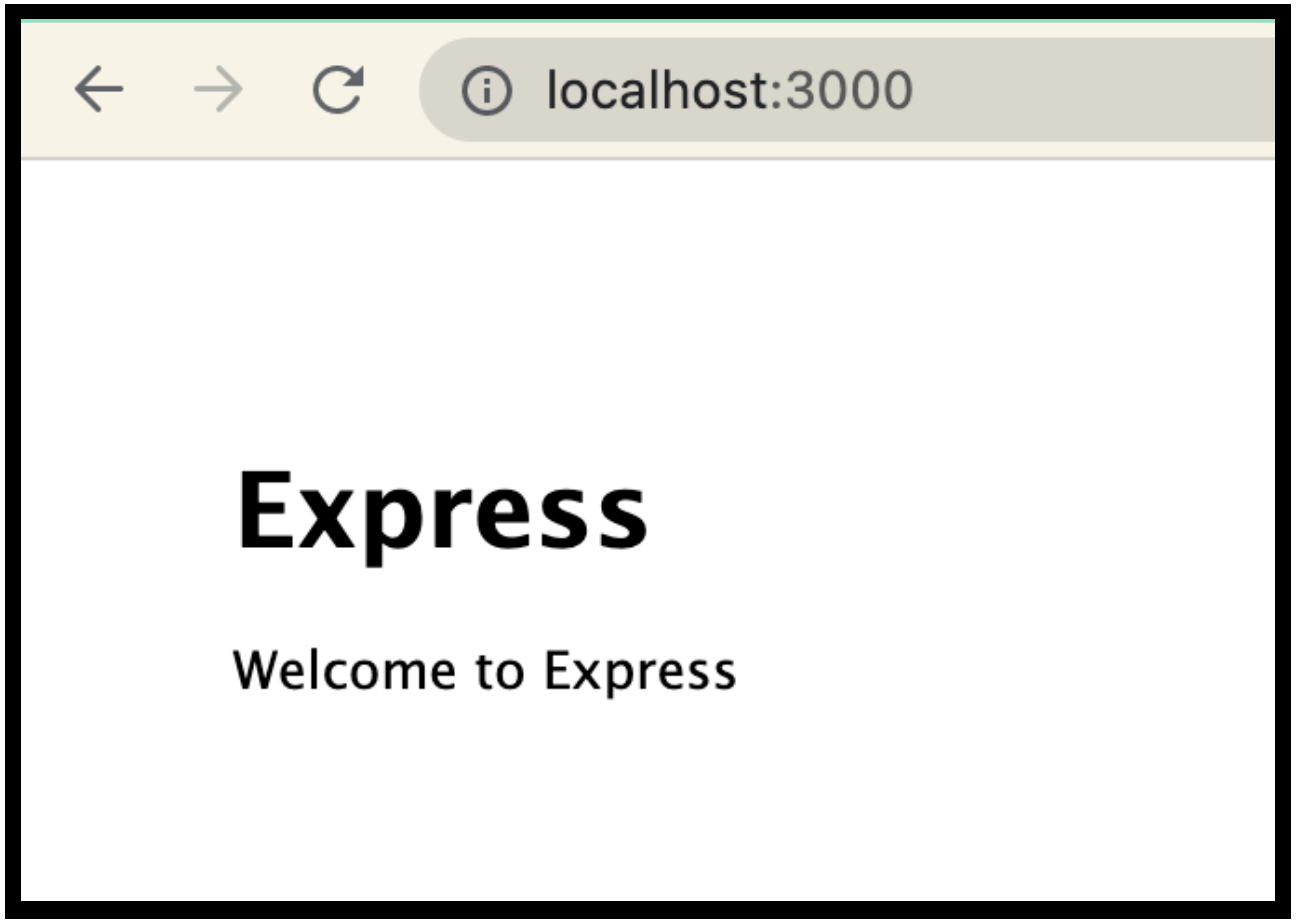
-最後にexpressにappオブジェクトを設定します。

スクリプトの流れ

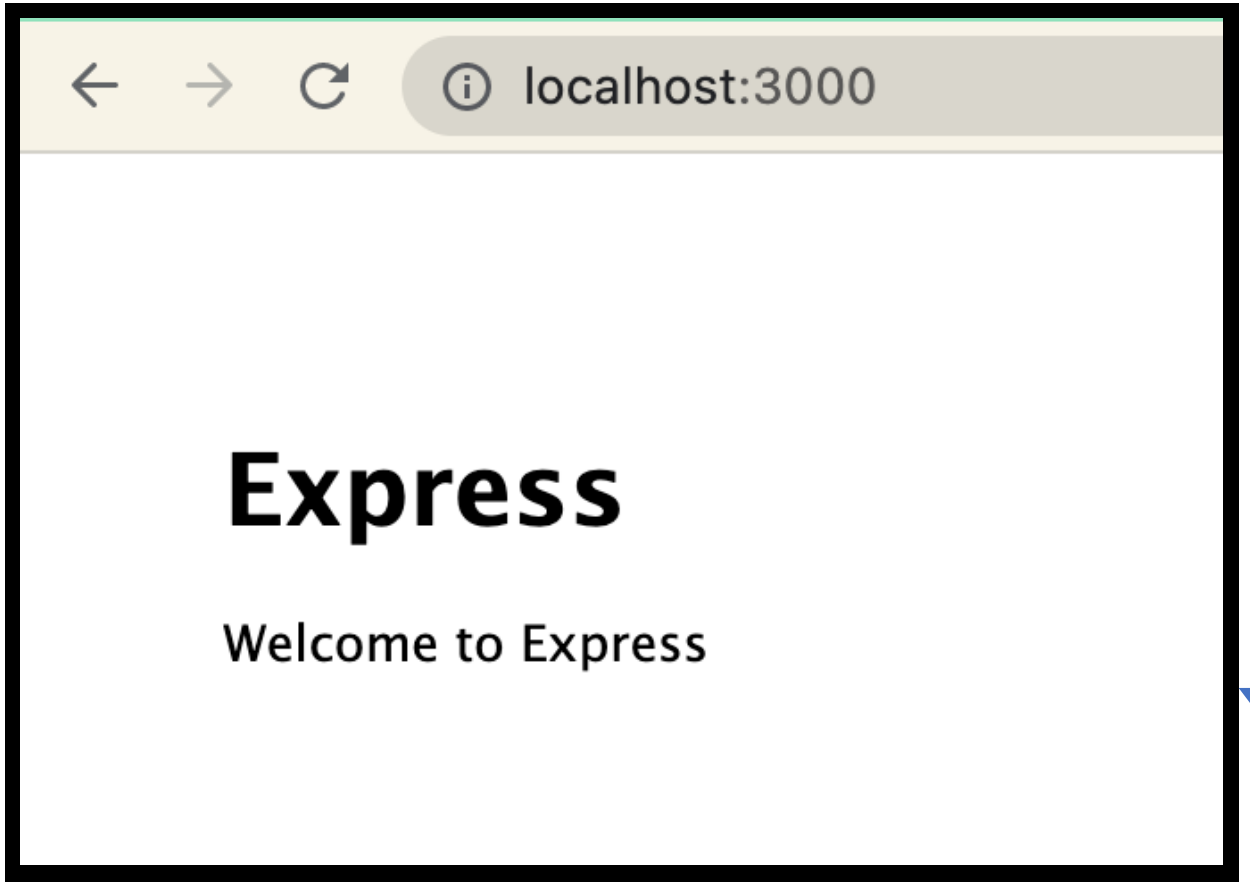
- Express Generatorでのスクリプトの流れ
- wwwでプログラムを起動
- app.jsで設定を読み込み
- Routeフォルダー内のスクリプトで処理



デフォルトで作られるページ



デフォルトで作られるページ



routes/index.js

```
routes > JS index.js > ...
1   var express = require('express');
2   var router = express.Router();
3
4   /* GET home page. */
5   router.get('/', function(req, res, next) {
6     res.render('index', { title: 'Express' });
7   });
8
9   module.exports = router;
10
```

参照

代入

views/index.js

```
views > index.jade
1   extends layout
2
3   block content
4     h1= title
5     p Welcome to #{title}
6
```

実習：index.jsを編集してExpress 2と表示するようにしてみましょう

ページの編集

手順

- index.jsの編集
- ローカル環境での動作確認
- Githubへのプッシュ
- サーバへのログイン
- githubからソースをpull (git pull コマンドを使用)
- サーバで動作確認（npm start)
- Ctrl+cで動作を停止

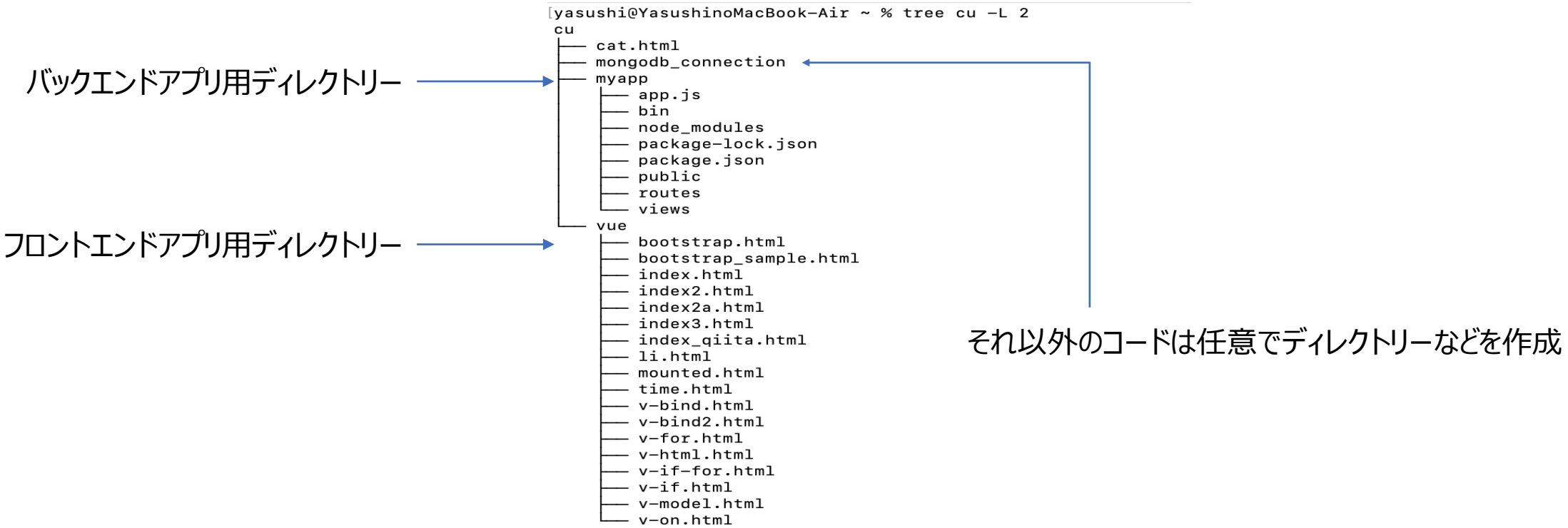
git pullについて

git pullとは、リモートリポジトリから最新の状態を
ローカルリポジトリに反映するコマンドのことです。

実習時間

・10分程度を目安に動画を止めて前ページまでの実習をしてください。
・作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリー構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。



第1章　まとめ

- Expressのルーティングについて学習した。
  - -1.必要なライブラリーをロード
  - -2.ルート用モジュールのロード
  - -3.Express オブジェクトの作成と基本設定
  - -4.関数の組み込み
  - -5.ルート用、エラー用のapp.use
  - -6.Module.expressの設定

- どのような設定、構成でプログラムが連動しているかを学習した。
  - -wwwでプログラムを起動
  - -app.jsで設定を読み込み
  - -Routeフォルダー内のスクリプトで処理

第1章
Expressルーティング概要

終わり

JavaScriptフレームワークによるWebプログラミング
第5回 Express実習2

第2章
Expressルーティング実習

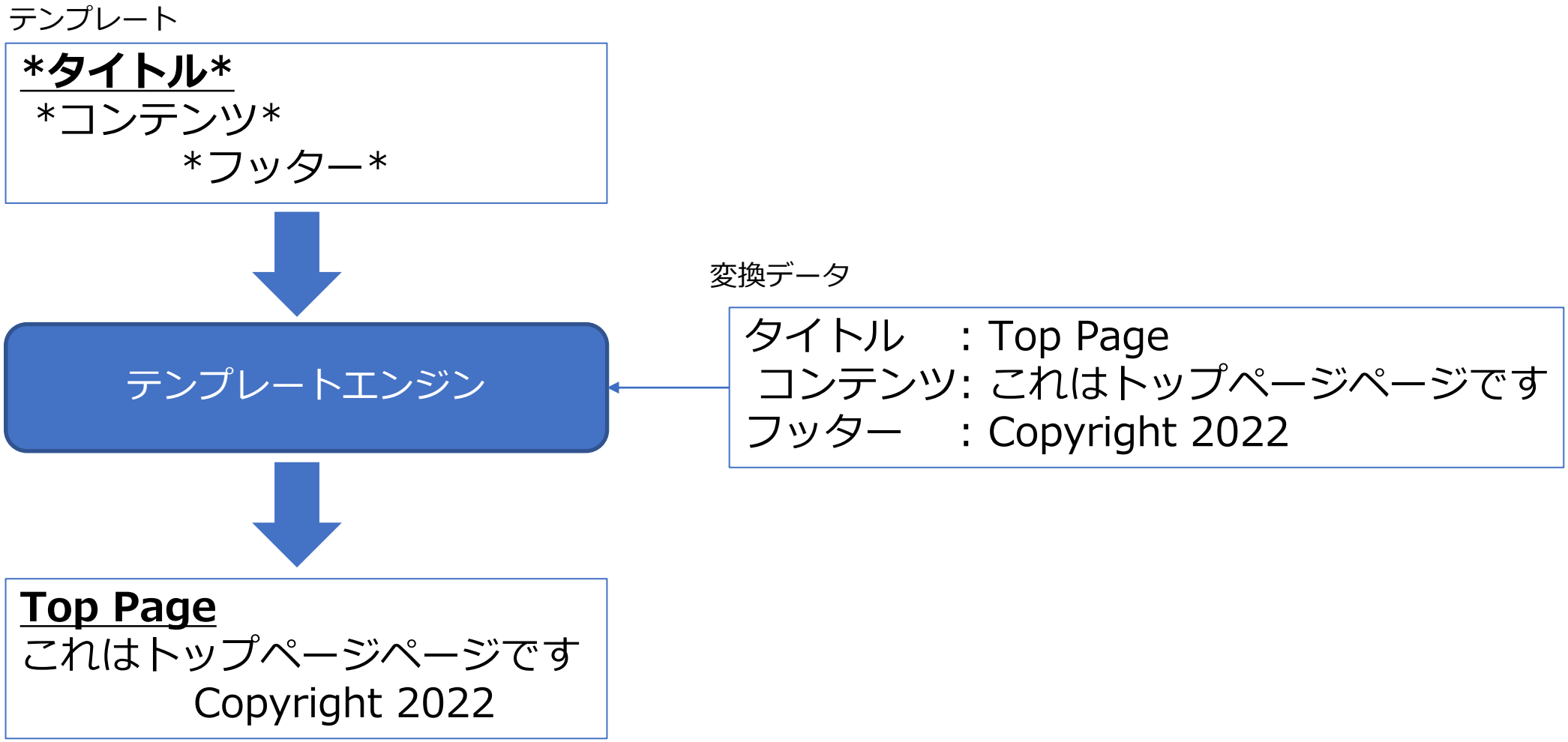
第2章　学習目標

・画面を作成するテンプレートについて理解し、説明できる
・Expressを使用して新しいページを作成できる

テンプレートエンジンとは

- Expressでは、“テンプレートエンジン”を使用してNode.jsのスクリプトからWebページの表示（HTMLの出力）を行っている。
- “テンプレートエンジン”とは“テンプレート”というものを使用して、表示するコンテンツを生成するための仕組み。
- “テンプレート”には変数や値などを記述する仕組みが用意されており、それを利用して仮の値が埋め込まれている。
- Webページを表示する際にはテンプレートをテンプレートエンジンが読み込むことでHTMLに変換する。変換によって必要な値が全て読み込まれた状態のHTMLコードが生成され、生成されたコードが画面に出力される。

テンプレートエンジンとは



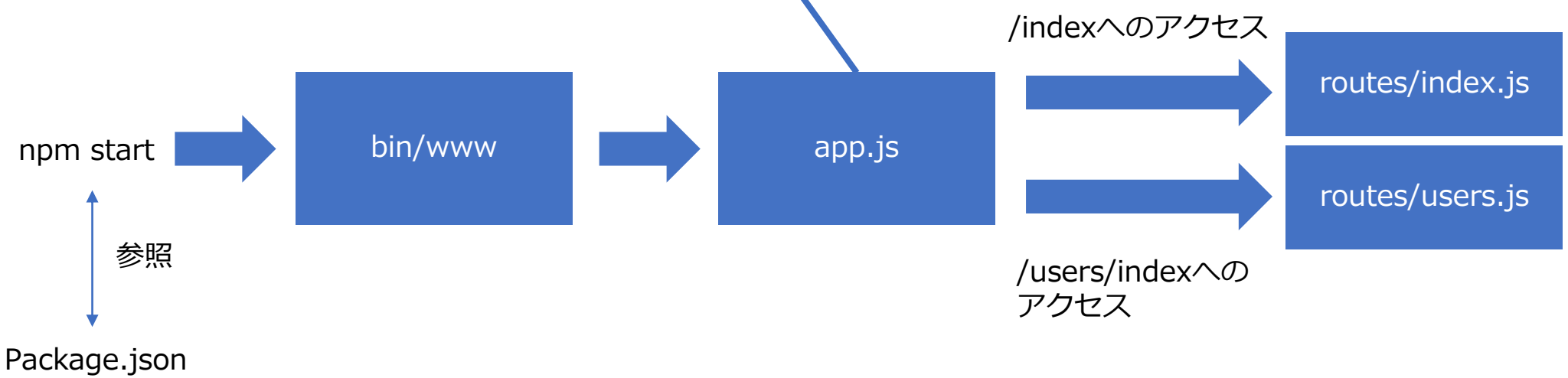
Express Generatorで作成されたスクリプト

index.jade | index.js | app.js

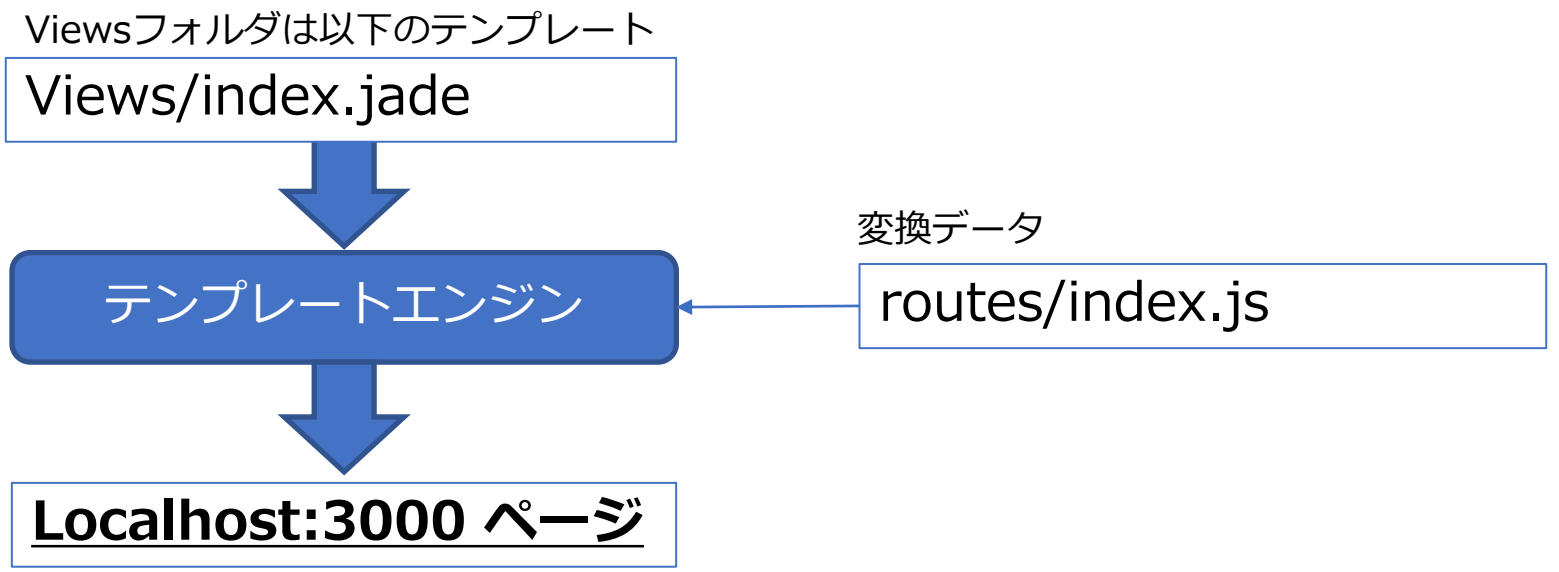
Users > shousonoiyasushi > nodejs > cu > express > myapp > app.js > ...

```
7   var indexRouter = require('./routes/index');
8   var usersRouter = require('./routes/users');
```

```
22  app.use('/', indexRouter);
23  app.use('/users', usersRouter);
```



Express Generatorで作成されたスクリプト



Express Generatorで作成されたスクリプト

Index.jadeでレイアウトと仮の値(title)を指定

Index.jsで値を入れてテンプレートエンジンで表示

index.jade

Users > shousonoiyasushi > nodejs > cu > express > myapp > views > index.jade

```
1  extends layout
2
3  block content
4    h1= title
5    p Welcome to #{title}
6
```

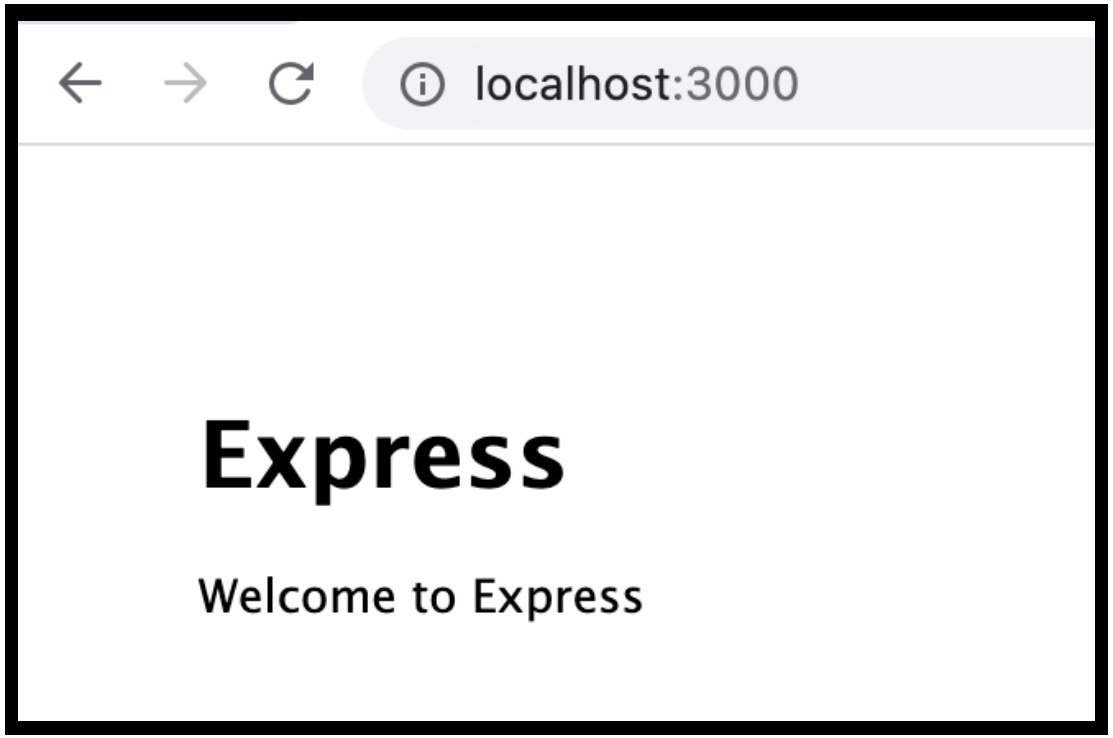
index.jade　index.js

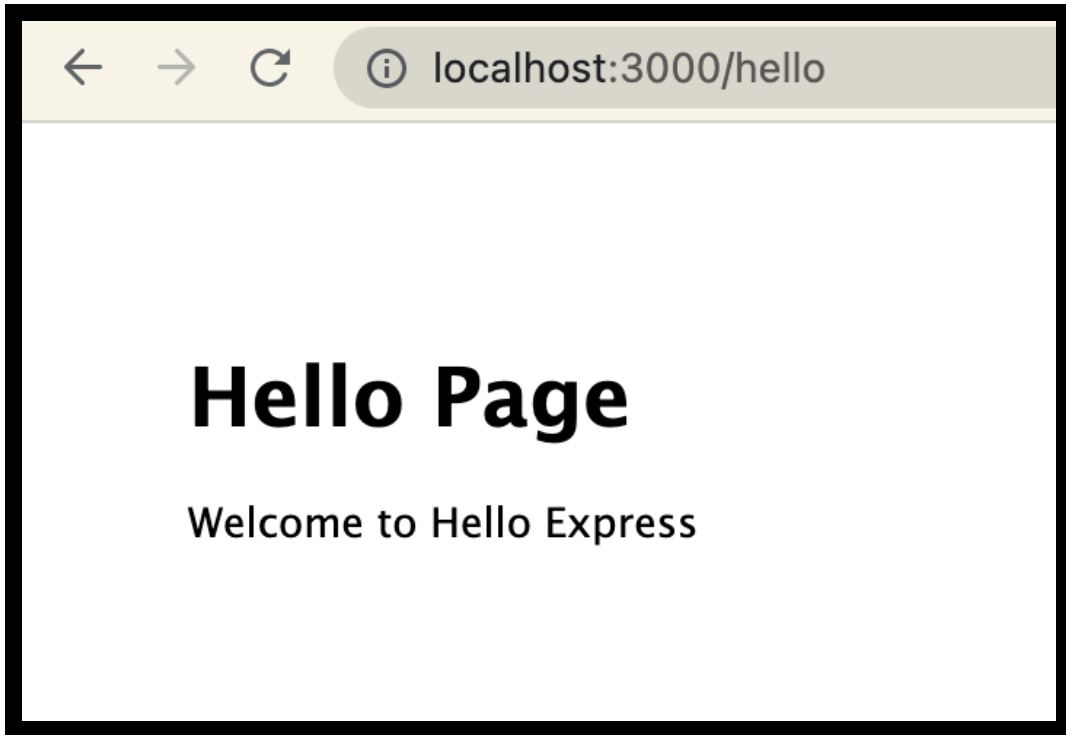
Users > shousonoiyasushi > nodejs > cu > express > myapp > routes > index.js > ...

```
1  var express = require('express');
2  var router = express.Router();
3
4  /* GET home page. */
5  router.get('/', function(req, res, next) {
6    res.render('index', { title: 'Express' });
7  });
8
9  module.exports = router;
10
```

Expressでの仕組みで新しいページを作成する

・前項までの仕組みを理解して新しいページhelloを作成する





Expressでの仕組みで新しいページを作成する

- routesフォルダにhello.jsを作成
    - res.render(‘index……をres.render(‘hello……
- viewsフォルダにhello.jadeを作成
- app.jsに以下の２行を追加
    - var helloRouter = require('./routes/hello');
    - app.use('/hello', helloRouter);

Expressでの仕組みで新しいページを作成する

hello.js

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('hello', { title: 'Hello Express' });
});

module.exports = router;

```

hello.jade

views > hello.jade

```
extends layout

block content
  <h1> Hello Page </h1>
  p Welcome to #{title}

```

app.js

app.js > ...

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var helloRouter = require('./routes/hello');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/hello', helloRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});
```

Githubにpush（アップロード）

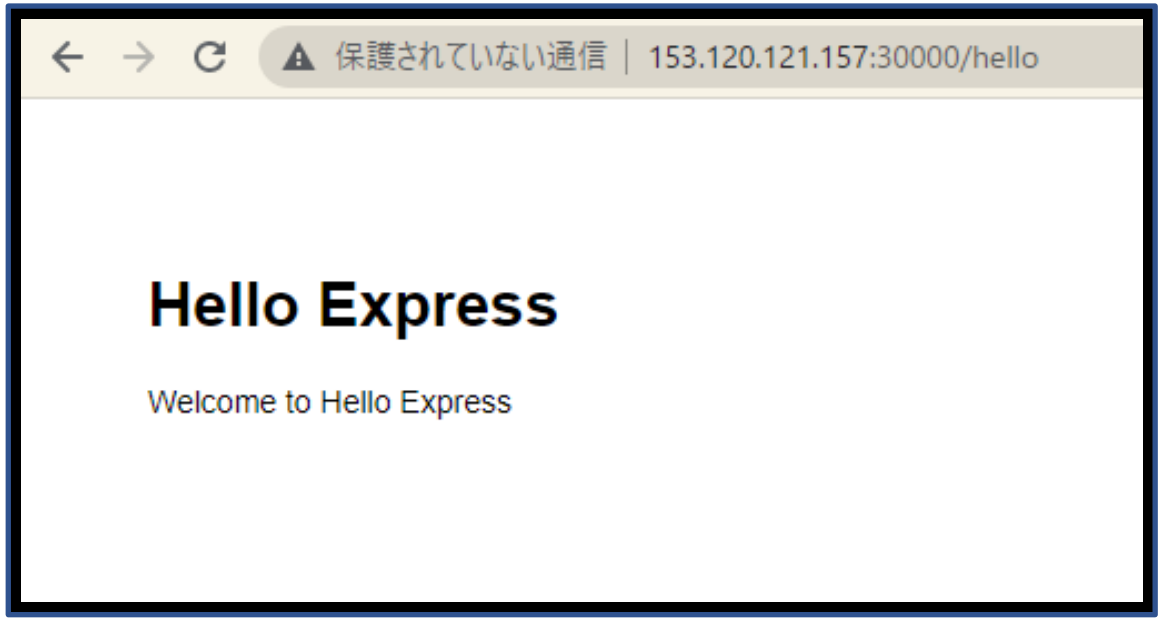
- Githubにpush
  git add .
  (git status　で確認しながらやるといい)
  git commit –m “add hello page”
  git push origin main

サーバで動作させてみる

サーバにログイン後（sshコマンドを使用）
アプリのディレクトリに入る（例　cd myapp）
GitHubのプログラムをアップデート（git pull コマンドを使用）

＊git pull コマンドでうまくいかない場合
　フォルダーごと消去してみる（例　rm -r -f myapp）

サーバで動作させてみる



実習時間

・10分程度を目安に動画を止めて前ページまでの実習をしてください。
・作業が終わったらビデオを再開して学習を進めてください。

※第1回4章で説明した通り、実習のファイル作成時には以下のディレクトリー構造を推奨しています。必要に応じて参照し、演習フォルダを整理してください。

バックエンドアプリ用ディレクトリー

フロントエンドアプリ用ディレクトリー

```
yasushi@YasushinoMacBook-Air ~ % tree cu -L 2
cu
├── cat.html
├── mongodb_connection
├── myapp
│   ├── app.js
│   ├── bin
│   ├── node_modules
│   ├── package-lock.json
│   ├── package.json
│   ├── public
│   ├── routes
│   └── views
└── vue
    ├── bootstrap.html
    ├── bootstrap_sample.html
    ├── index.html
    ├── index2.html
    ├── index2a.html
    ├── index3.html
    ├── index_qiita.html
    ├── li.html
    ├── mounted.html
    ├── time.html
    ├── v-bind.html
    ├── v-bind2.html
    ├── v-for.html
    ├── v-html.html
    ├── v-if-for.html
    ├── v-if.html
    ├── v-model.html
    └── v-on.html
```

それ以外のコードは任意でディレクトリーなどを作成

第2章　まとめ

Expressルーティング実習

- Expressのルーティングを使った新しいページの
  作成時方法を学習した

-“テンプレート”、“テンプレートエンジン”によるWebページ（HTML）の出力
-routesフォルダ（テンプレートエンジン）への変換データ（hello.js）の追加
-viewsフォルダへのテンプレート（hello.jade）ファイルの追加
-app.jsの編集による、新しく作成したページへのルーティング

第5回　まとめ

- Expressのルーティングについて学習した
  - -1.必要なライブラリーをロード
  - -2.ルート用モジュールのロード
  - -3.Express オブジェクトの作成と基本設定
  - -4.関数の組み込み
  - -5.ルート用、エラー用のapp.use
  - -6.Module.expressの設定

- Expressを使用した新しいページの作成方法を学習した。
  - -“テンプレート”、“テンプレートエンジン”によるWebページ（HTML）の出力
  - -上記の仕組みを使用した新ページの作成及びルーティング設定

第2章
Expressルーティング実習

終わり