



13. EJS のデータ受け渡し

EJS の基本構文

テンプレートエンジンでは、テンプレートファイルにデータを受け渡して、処理することができます。EJS の場合は ASPタグの構文で記述します。

JavaScript の処理

`<% %>` の中に JavaScript の処理を記述できます。

```
<% 処理 %>
```

変数の表示

`<%= %>` で変数を表示します。

```
<%= 変数 %>
```

HTML タグの表示 `<%- %>`

`<%- 変数 %>` で変数を HTML出力します。

```
<%- 変数 %>
```

if() 文

if() 文は、構文の最初と終わりを `<% %>` で囲みます。この間にHTMLを記述します。

```
<% if (条件) { %>

<% } else if (条件) { %>

<% } %>
```

forEach() 文

forEach 文も同様に <% %> で囲みます。

```
<% 配列.forEach((変数, インデックス) => { %>

<% }) %>
```

for(in) 文

for 文も同様に <% %> で囲みます。

```
<% for (インデックス in 配列) { %>

<% } %>
```

while() 文

while 文も同様に <% %> で囲みます。

```
<% while (条件) { %>

<% } %>
```

データの受け渡し

render() メソッドの第2引数にデータを受け渡します。

```
res.render(ファイルパス, データ)
```

データの表示

EJS ファイルで変数を表示するには `<%= %>` タグを利用します。その他の EJS のタグは「EJSの構文」の章で説明します。

```
<%= 変数 %>
```

EJS にデータ受け渡し

トップページのルーティングで、`data.title` を設定して `index.ejs` に受け渡します。

```
router.get('/', (req, res) => {
  let data = {}
  data.title = 'トップページ';
  res.render('index.ejs', data);
})
```

EJS でデータ表示

`index.ejs` の `h1` タグで `title` 変数を表示します。

```
<h1 class="h1"><%= title %></h1>
```

サーバを再起動して、タイトルが表示できるか確認してみましょう。

← → ↺ 🏠 ⓘ localhost:3000

[プロフィール 商品一覧](#)

トップページ

プロフィールページ

EJS にデータ受け渡し

`/profile` ルーティングに `user` データを用意します。 `hobby` は配列で設定しました。 `user` や `title` をオブジェクトにして、`profile/index.ejs` に受け渡します。

```
router.get("/profile", (req, res) => {
  let user = {
    id: 1,
    name: '横浜 太郎',
    birthplace: '横浜',
    hobby: ['旅行', 'グルメ', 'スポーツ'],
  }
  let data = {};
  data.title = 'プロフィール';
  data.user = user;

  res.render('profile/index.ejs', data);
})
```

EJS でデータ表示

`profile/index.ejs` ファイルに変数を表示します。 `hobby` は配列なので `forEach` で繰り返し表示します。

商品一覧

EJS にデータ受け渡し

`/item` ルーティングで商品データを取得し、 `item/index.ejs` に受け渡します。

```
router.get('/item', (req, res) => {
  let data = { items: item.values };
  data.title = '商品一覧';
  res.render('item/index.ejs', data);
})
```

EJS でデータ表示

`item/index.ejs` で商品一覧を表示します。

```
...
<div class="container">
  <h3 class="h3"><%= title %></h3>
  <dl>
    <% items.forEach ((item) => { %>
      <dd><a href="/item/<%= item.id %>"><%= item.name %></a></dd>
```

```
      <dd><%= item.price %>円</dd>
    <% }) %>
  </dl>
</div>
...
```

商品詳細

EJS にデータ受け渡し

`/item/:id` ルーティングで商品データを取得し、`item/show.ejs` に受け渡します。

```
...
router.get('/item/:id', (req, res) => {
  const id = req.params.id;

  let data = {};
  data.item = item.find(id);
  data.title = '商品情報';
  res.render('item/show.ejs', data);
})
...
```

EJS でデータ表示

`item/show.ejs` で商品情報を表示します。

```
...
<h3 class="h3">商品情報</h3>
<% if (item) { %>
  <dl>
    <dt>商品名</dt>
    <dd><%= item.name %></dd>
    <dt>価格</dt>
    <dd><%= item.price %>円</dd>
  </dl>
<% } else { %>
  <p>商品がみつかりませんでした</p>
<% } %>
...
```

ソース

routes.js

```
const express = require('express');
const router = express.Router();
const item = require('./models/item');

router.get('/', (req, res) => {
  let data = {}
  data.title = 'トップページ';
  res.render('index.ejs', data);
})

router.get("/profile", (req, res) => {
  let user = {
    id: 1,
    name: '横浜 太郎',
    birthplace: '横浜',
    hobby: ['旅行', 'グルメ', 'スポーツ'],
  }
  let data = {};
  data.title = 'プロフィール';
  data.user = user;

  res.render('profile/index.ejs', data);
})

router.post('/auth', (req, res) => {
  const login_name = req.body.login_name;
  const password = req.body.password;

  let message = 'ログインできませんでした';
  if (login_name == process.env.LOGIN_NAME && password == process.env.PASSWORD) {
    message = 'ログインしました';
  }
  res.send(message);
})

router.get('/item', (req, res) => {
  let data = { items: item.values };
  data.title = '商品一覧';
  res.render('item/index.ejs', data);
})

router.get('/item/:id', (req, res) => {
  const id = req.params.id;

  let data = {};
  data.item = item.find(id);
  data.title = '商品情報';
  data.message = ''

  if (!data.item) data.message = '商品が見つかりませんでした';
});
```

```
res.render('item/show.ejs', data);  
})  
  
module.exports = router
```

当サイトの教材をはじめとするコンテンツ（テキスト、画像等）の無断転載・無断使用を固く禁じます。これらのコンテンツについて権利者の許可なく複製、転用等する事は法律で禁止されています。尚、当ウェブサイトの内容をWeb、雑誌、書籍等へ転載、掲載する場合は「ロジコヤ」までご連絡ください。