

CAPTER 6

ダイナミックなページ生成

本章ではダイナミックなページ生成について説明します。ダイナミックにページを生成することは、ウェブアプリケーションサーバの基本的な機能です(図 6.1)。

❗ 本章を読むには Java の知識が必要です。付録 A を参照してください。

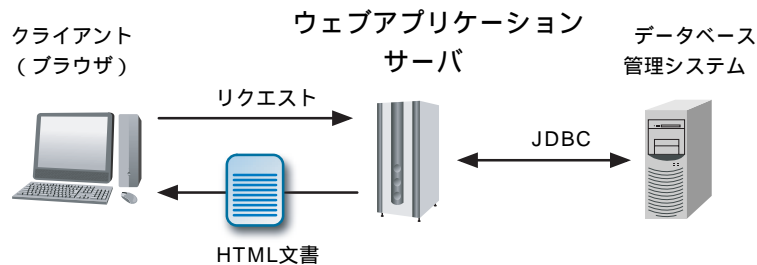


図 6.1 本章で学ぶこと：ダイナミックなページ生成方法

ダイナミックなページ生成とは、具体的にはどういうことを意味するのでしょうか。ダイナミックなページ生成の反対が静的なページ生成です。これは、4.4 節 (p.33) で見たように、あらかじめ作成しておいたページを提示することを言います。それに対して、要求があったその時にページを生成するのがダイナミックなページ生成です。

もう少し詳しく説明しましょう。3.2 節 (p.17) で書いた HelloWorld.html は次のようなものでした (これは妥当な HTML 文書ではありませんが、ここでは気にしません)。

```
<html>
<body>
Hello World!
</body>
</html>
```

ダイナミックにこのページを生成するというのは、次のようなプログラムを実行するということです (これは説明のための擬似コードだと思ってください)。

```
printf "<html>\n"
printf "<body>\n"
printf "Hello World!\n"
printf "</body>\n"
printf "</html>\n"
```

しかしこれでは手間が増えただけのように見えます。もう少し利用価値のわかる例を考えましょう。たとえば、「1 から 100 までの数値を箇条書きにする」という場合はどうで

しょうか。次のようなコードで実現できますが（これも説明のための擬似コードだと思ってください）、li 要素を 100 個書くのに比べればはるかに簡単です。

```
printf "<ul>";
i=1;
while [ $i -le 100 ]
do
  printf "<li>%d</li>" $i;
  i=$((i+1));
done
printf "</ul>";
```

このままでは、ダイナミックに生成する必要のない要素もプログラムで生成しなければなりません。たとえば、4.2.1 項 (p. 23) で紹介した各種宣言を、わざわざ次のように書かなければならないのでしょうか¹⁾。

```
printf '<?xml version="1.0" encoding="UTF-8" ?>';
printf '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN";';
printf ' "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">';
printf '<html xmlns="http://www.w3.org/1999/xhtml">';
...
(面倒)
```

実行前から決まっている部分はそのまま書いて、ダイナミックに生成したいところだけ、プログラムで生成するようにしたいものです。それを可能にするのが、本章で紹介する **JSP** (JavaServer Pages) です。JSP のファイルは、一見ふつうのウェブページですが、ところどころ Java のコード（あるいは特殊なタグ）が埋め込まれます。

6.1 JSP: HTML と Java の融合

6.1.1 JSP 版 Hello World

JSP 版 HelloWorld は次のようになります。

```
<html>
<body>
<p><% out.println("Hello World!"); %></p>
</body>
</html>
```

もちろん、「<p>Hello World!</p>」でいいのですが、それでは単なる HTML 文書になってしまうため、あえて Java のプログラムを書いています。このように、<%と%>の間に Java のプログラムを書いたものを **スクリプトレット** と呼びます²⁾。

- 1) このコードは **シェルスクリプト** なので実行できます。二重引用符と一重引用符で意味が変わるのはよくあることです。
- 2) スクリプトレットには、本文中で用いた「<% Java のコード %>」という書き方 (**JSP シンタックス**) と「<jsp:scriptlet> Java のコード </jsp:scriptlet>」という書き方 (**XML シンタックス**) があります。