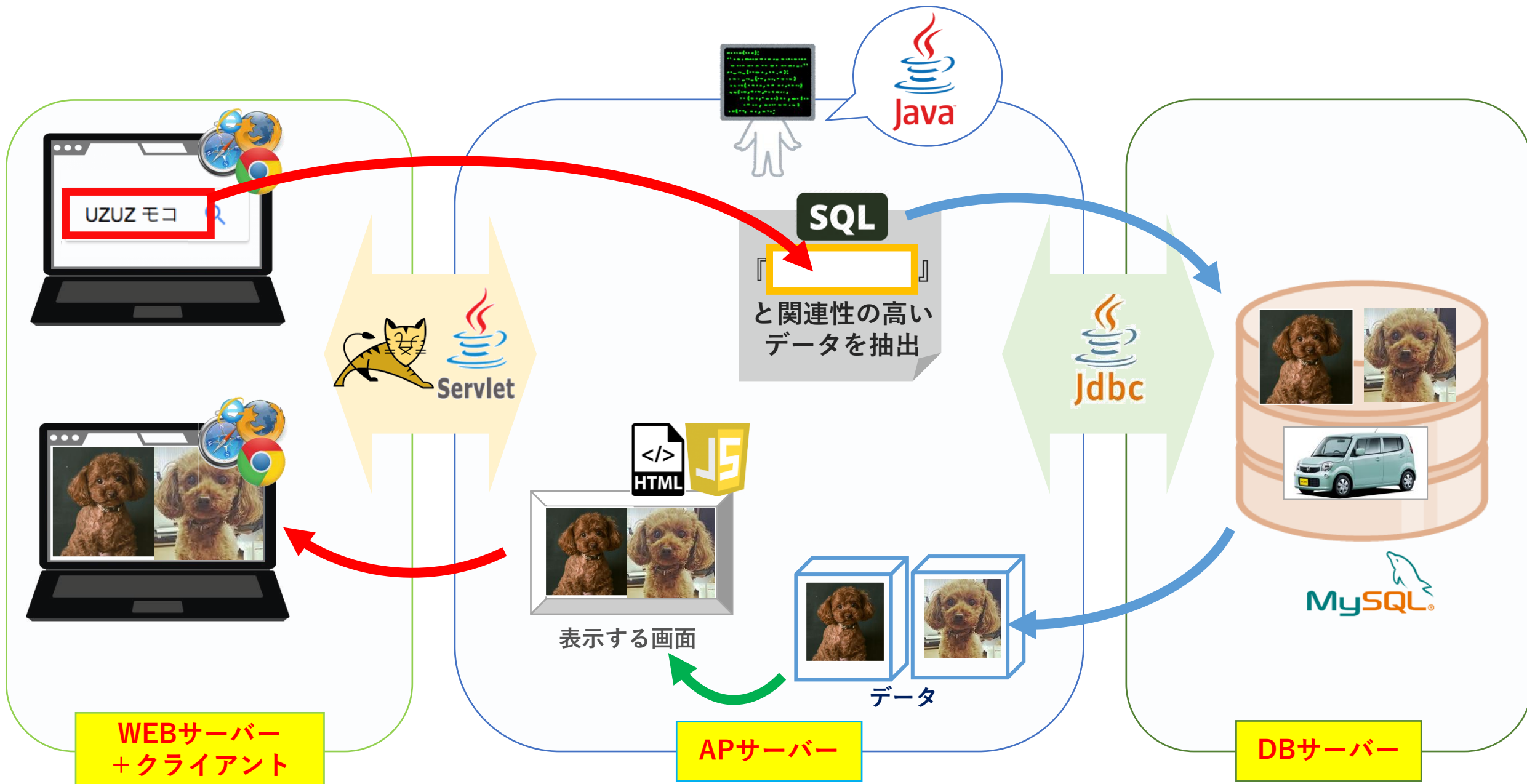
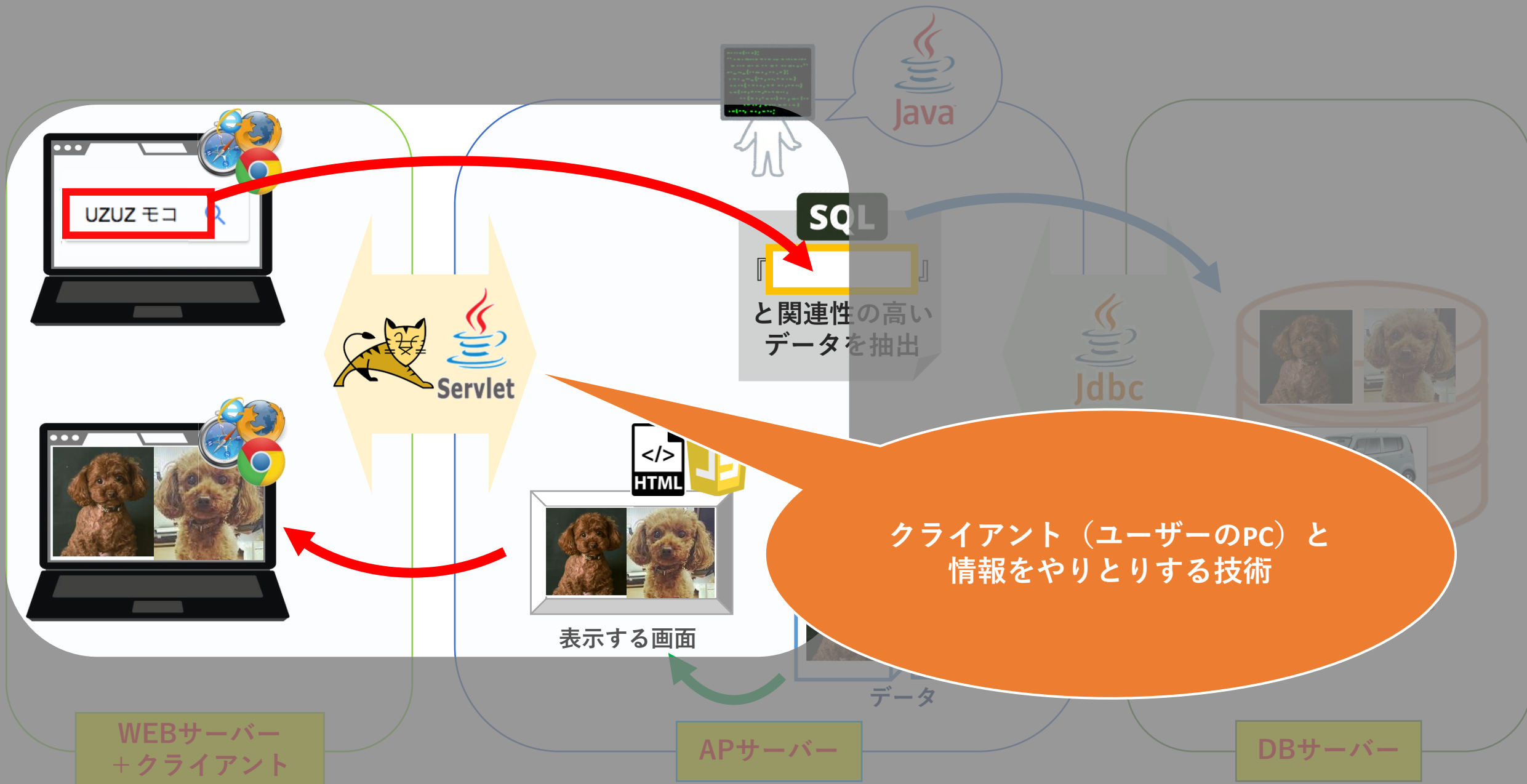


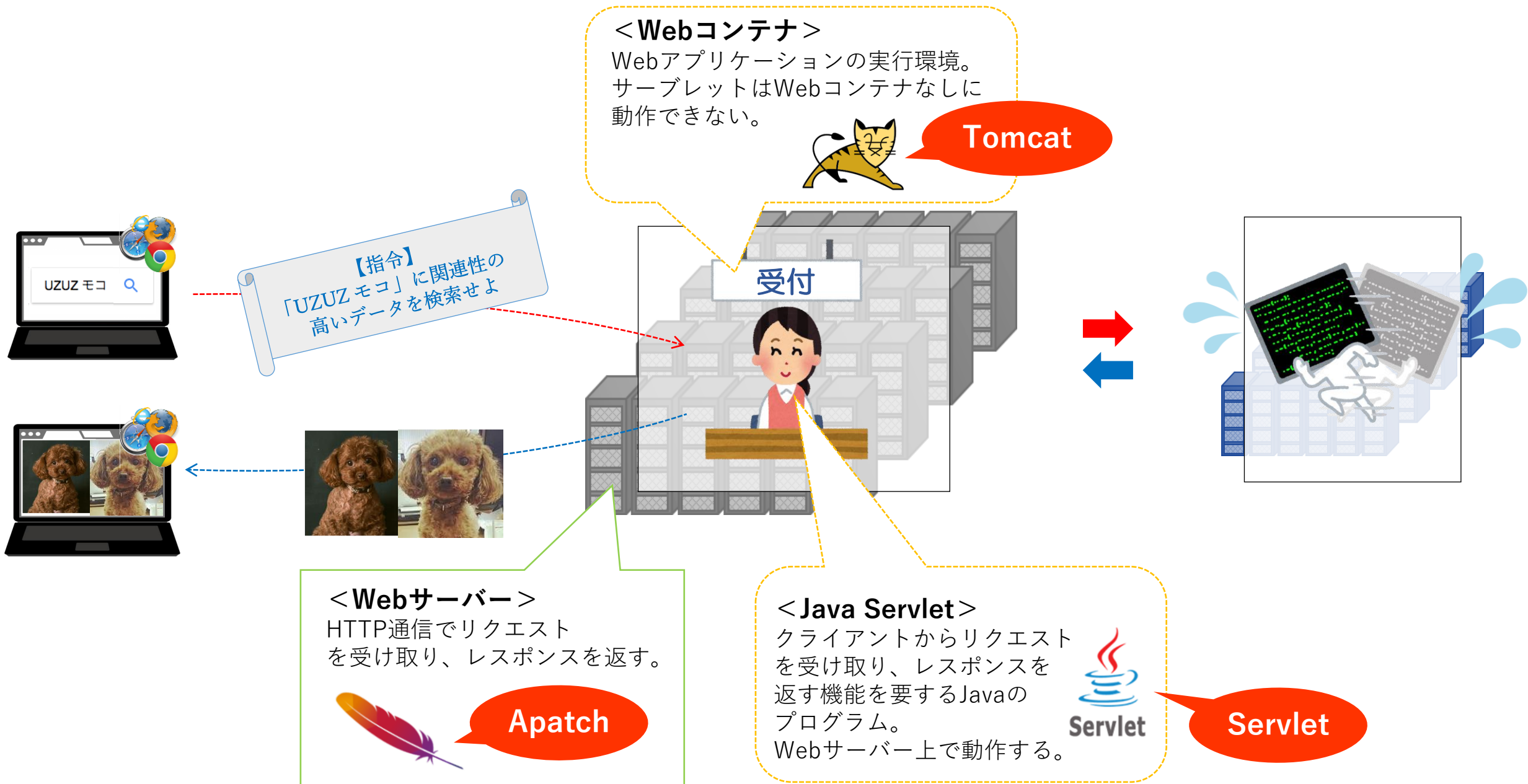
～Webアプリケーション開発講座～
Java Servletの基本とWebページの表示





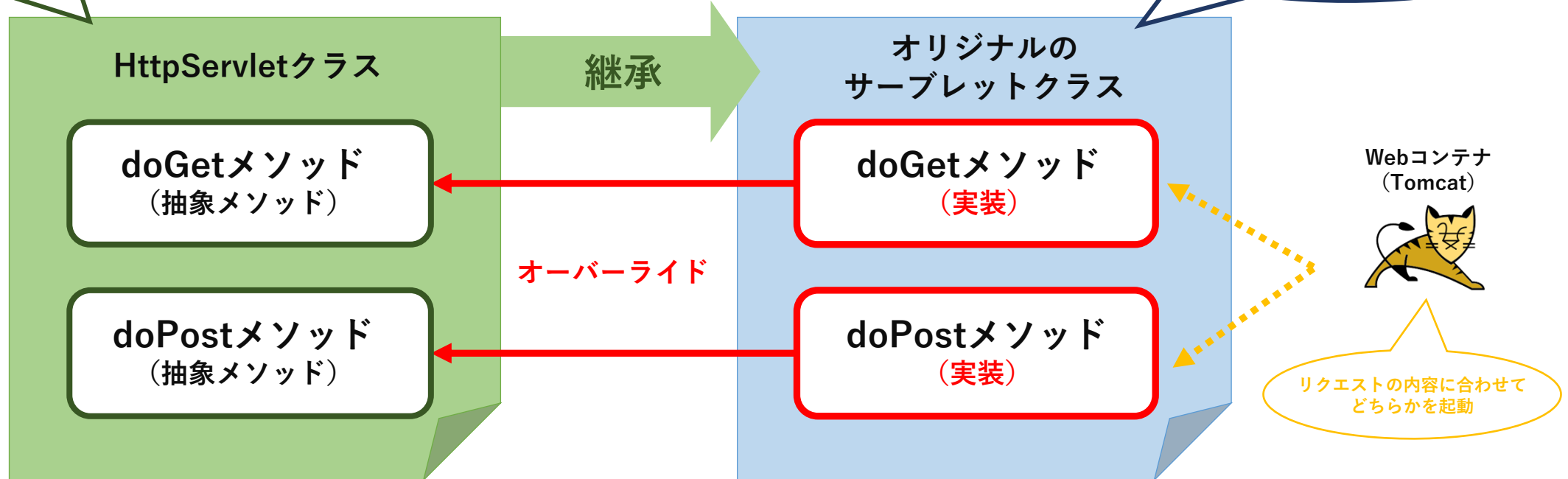


～ Webサーバー周辺の技術 ～



～ サーブレットクラスの構造 ～

Webコンテナ (Tomcat) を介してクライアントとやりとりする複雑な処理は既にこちらでほとんど書いてくれている。



《ポイント》

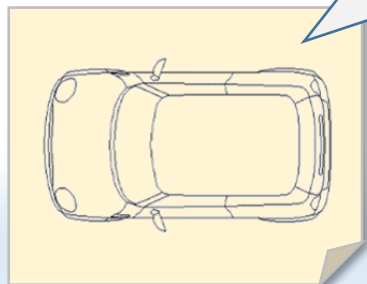
- Webコンテナ (Tomcat) がリクエストの中身を読み取り、doGetメソッド／doPostメソッドのうち適切な方を呼び出す。
- **サーブレットは必ずHttpServletクラスを継承**して作成される。
HttpServletでクライアントとやりとりするための複雑な処理は既に定義しているため、作成したサーブレットでは基本的に**doGetメソッド／doPostメソッドを実装**するだけでよい。
- doGet/doPostメソッドでは基本的に以下の2処理を実装する。
 - リクエストの処理
 - レスポンスの設定

復習

抽象メソッド

車ならこんな機能（メソッド）があるべきだ！
（具体的な処理内容はオーバーライドして定義してね！）

```
//acceleratorメソッド（現在の速度を上げる）←  
abstract void accelerator() ;←  
←  
//brakeメソッド（現在の速度を下げる）←  
abstract void brake() ;←  
←  
//paintメソッド（色を塗る）←  
abstract void paint( String cl ) ;←
```



Sample2_07_1_AbstractCar

抽象クラス

仕様のみを定めた
インスタンス不可で継承前提のクラス

継承



Sample2_07_1_Car
インスタンス

抽象メソッドをオーバーライド
していないとエラー！

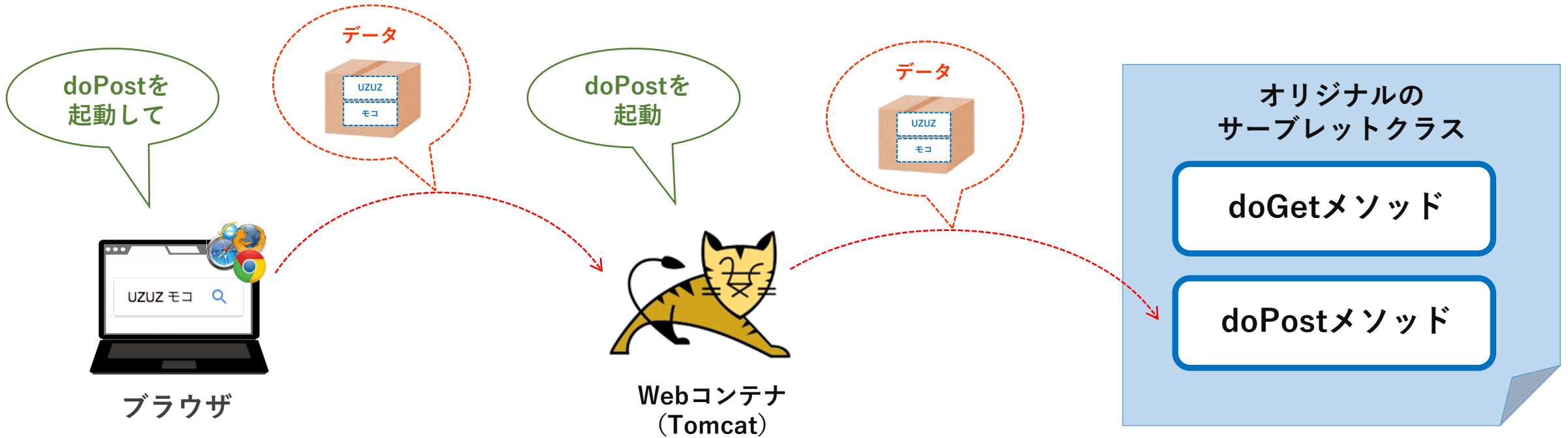
≪ 抽象クラス (abstract) ≫

- 「このクラスを継承したサブクラスには〇〇というメソッドがあるべきだ」という**仕様のみを定義した継承前提のクラスを抽象クラス**と言います。
抽象クラスは**インスタンス化されて使用されることを想定していない**ため、インスタンス化しようとするとうエラーになります。
classの前に**abstract修飾子**を付与することで抽象クラスとして扱うことが可能になります。
- 抽象クラスでは**抽象メソッド**という機能の仕様を定義するためのだけの**具体的な処理内容のないメソッド**を定義することが可能です。
以下のように記述し、大きな特徴として**{}**が存在せず、代わりに「;」が置かれます。

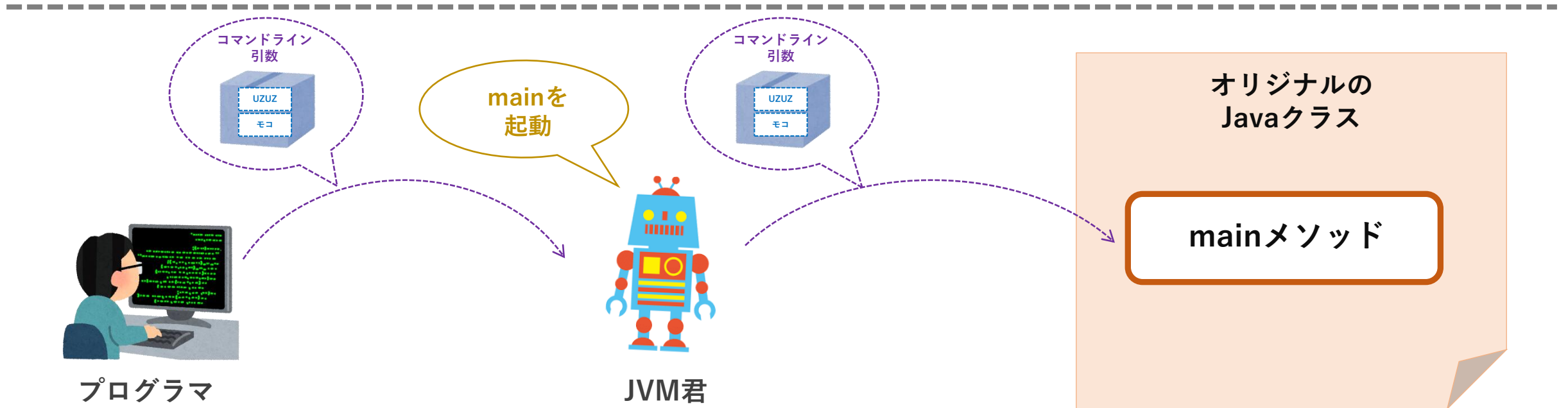
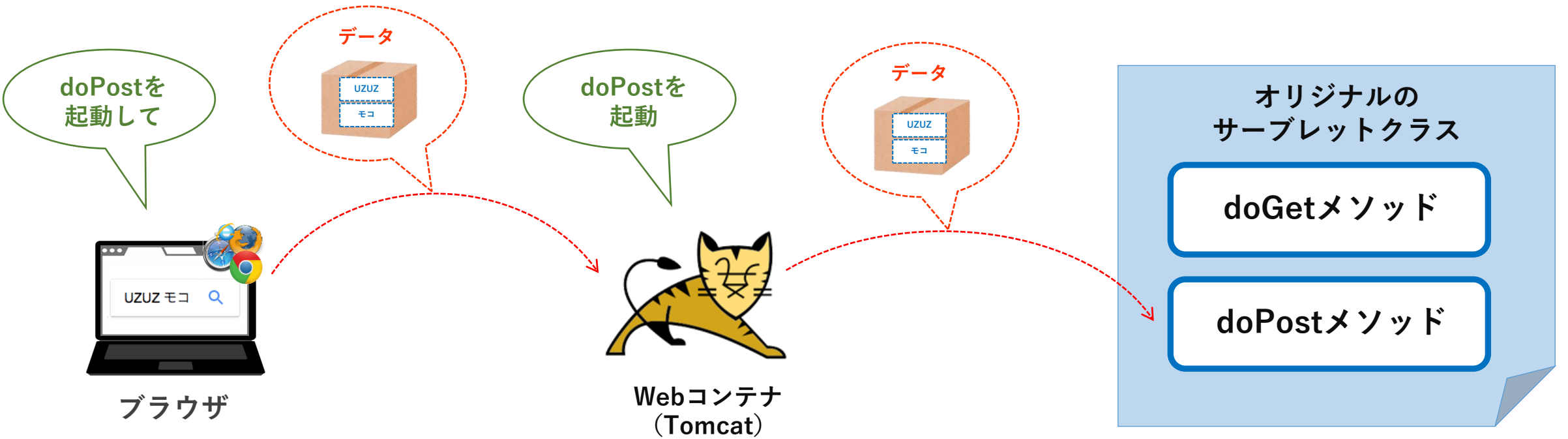
[アクセス修飾子] **abstract** 戻り値の型 **メソッド名**(仮引数の型 仮引数名);

- 抽象メソッドは**オーバーライドして使用されることを前提**とします。
抽象メソッドをオーバーライドして具体的な処理内容を定義することを**実装**と言い、抽象クラスを継承して作成されたインスタンス内に**実装されていない抽象メソッドが1つでもあればインスタンス化の際にエラーとなります**。
抽象クラスを継承した全クラスに抽象メソッドの実装を強制することができるため、そのクラスで定義すべき機能が漏れたり、好き勝手な書き方で定義されないよう抑制できるといったメリットがあります。

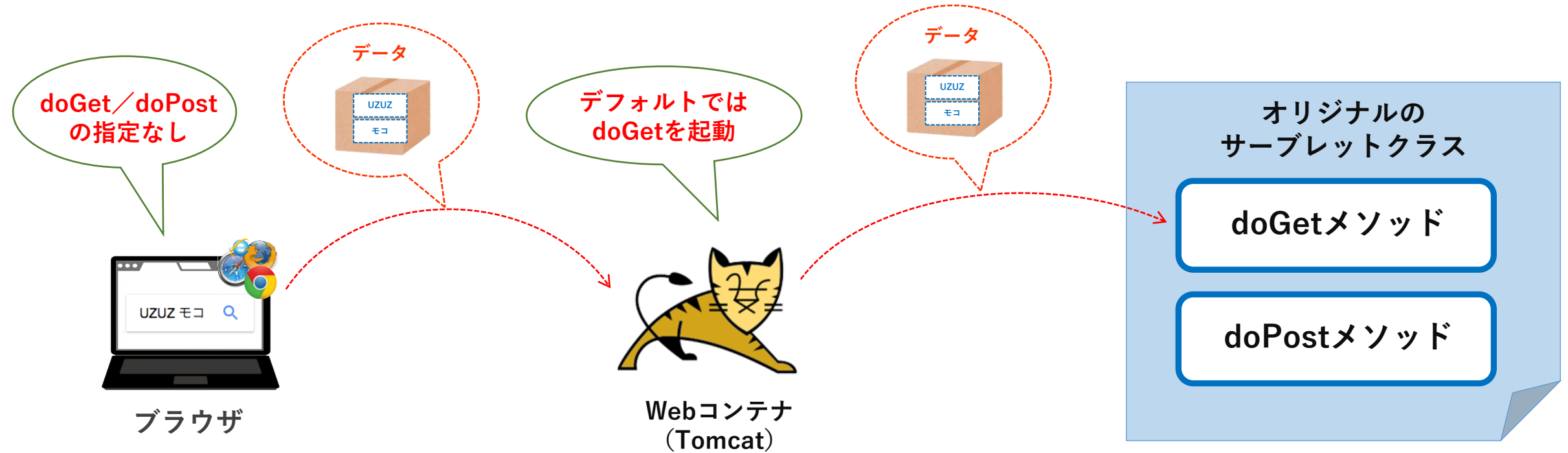
～ doGet/doPostが起動されるまでの流れ ～



～ doGet/doPostが起動されるまでの流れ ～



～ doGet/doPostが起動されるまでの流れ ～



～ サンプルを動かしてみよう ～



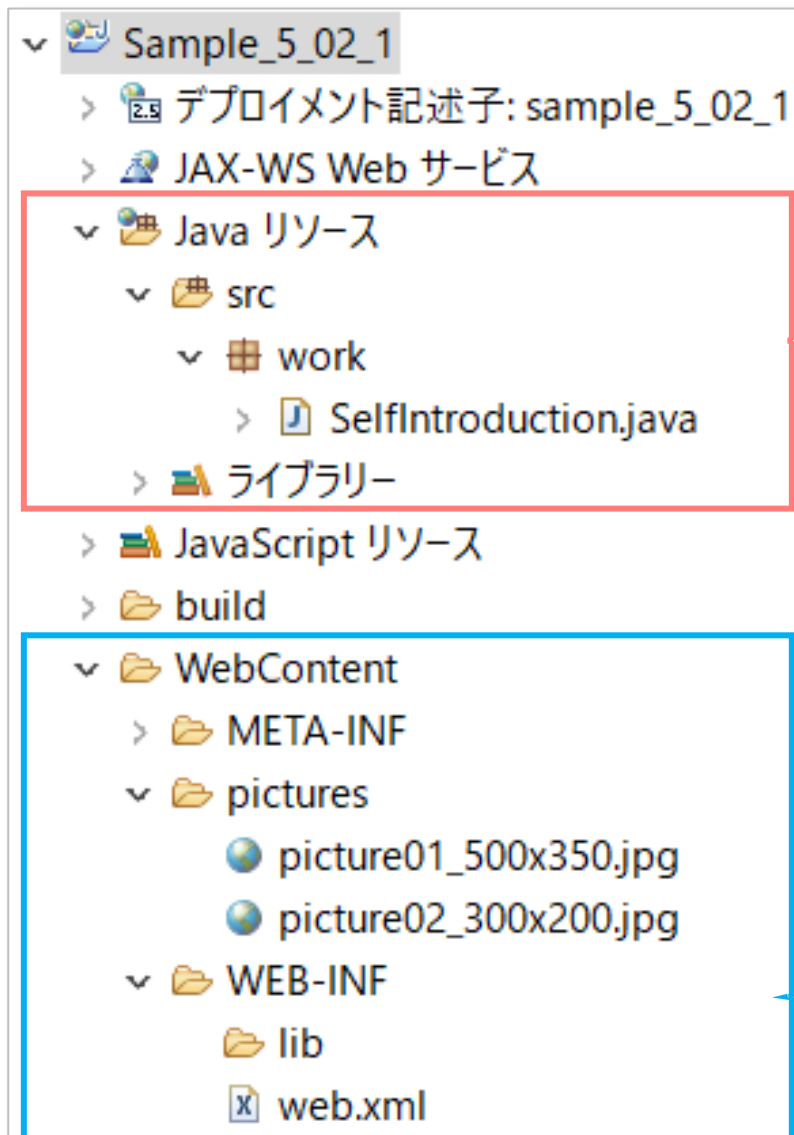
左のような自己紹介ページを
eclipseから動作させてみましょう！

～ サンプルを動かしてみよう ～

手順

- (1) eclipse上に新しい動的Webプロジェクトを作成する。
プロジェクト名：Sample5_02_1
- (2) プロジェクトの srcパッケージ直下に workパッケージを作成する。
- (3) ドライブのServletSample00フォルダからソースコードを取得して
workパッケージ直下にインポートする。
 - SelfIntroduction.java
- (4) 同じく web.xmlをドライブから取得して置き換える。
WebContent > WEB-INF > web.xml
- (5) WebContentフォルダ直下にpicturesフォルダを作成し、以下を置く。
 - picture01_500x350.jpg
 - picture02_300x200.jpg
- (6) Tomcatサーバーを起動し、ブラウザを開いて次のURLを入力する
http://localhost:8080/Sample_5_02_1/SelfIntroduction

～ 「動的Webプロジェクト」 の概要 ～



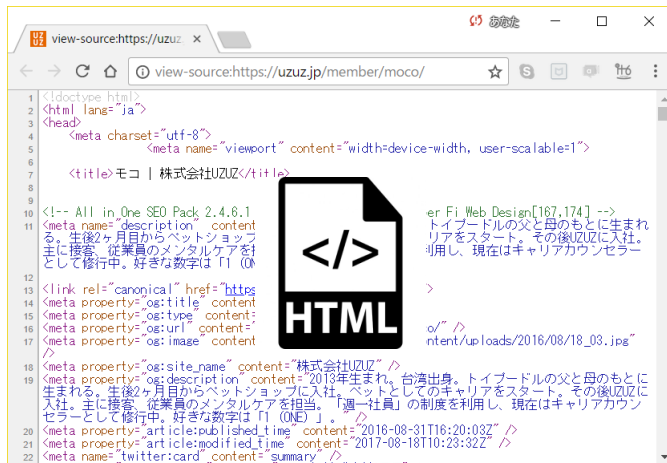
Javaのソースコードファイルを格納するディレクトリ。

Webに公開する情報を格納するディレクトリ。
『WEB-INF』にはweb.xmlをはじめとした公開のために必要な情報が格納されます。

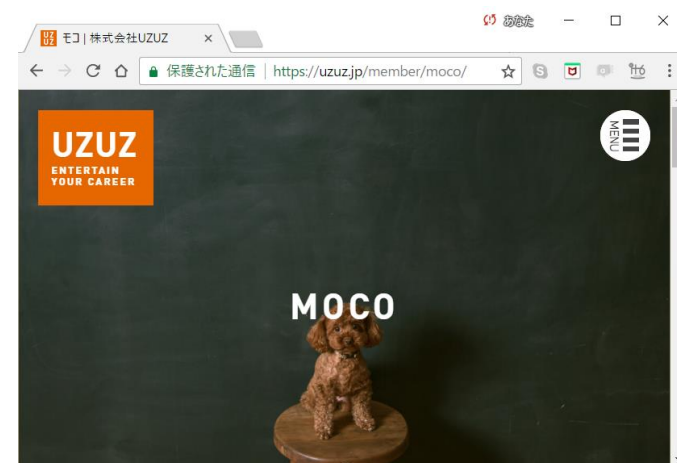
～ 画面が表示される仕組み（HTML） ～

<HTML>

Hyper Text Markup Languageの略。**ブラウザに表示する画面の設計図**を記述するための言語。主にHTMLタグ（</>）を使って記述を行う。ウェブブラウザはHTMLを受け取ると、指定されたとおりの画面を描画し、表示する。この一連の動作を**レンダリング**という。



ウェブ
ブラウザ



解釈 & 表示
(レンダリング)

～ 画面が表示される仕組み（HTML） ～

HTMLの書き方

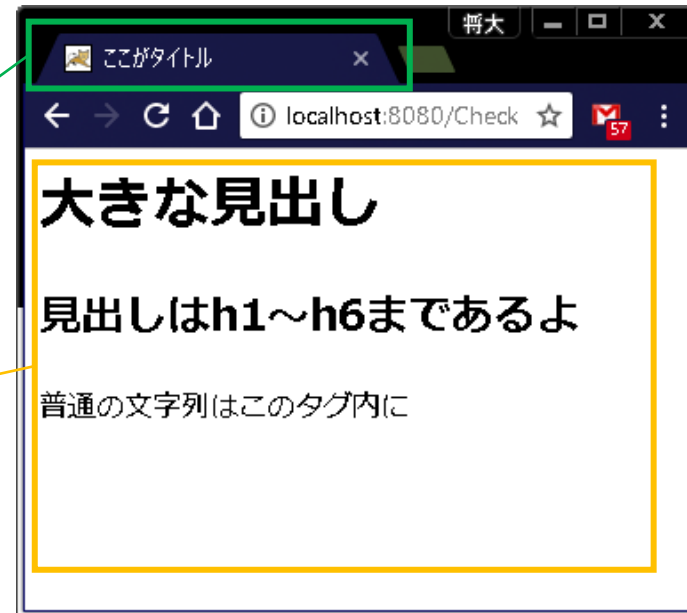
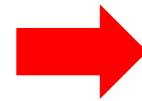
HTMLタグを使って、画面に表示させる部品を一つひとつ記述していく。

< 部品名 > 表示させたい文字列 < /部品名 >

←このひとかたまりを
『要素』と呼びます

サンプルHTMLファイル

```
<html>  
  <head>  
    <title>ここがタイトル</title>  
  </head>  
  <body>  
    <h1>大きな見出し</h1>  
    <h2>見出しはh1～h6までであるよ</h2>  
    <p>普通の文字列はこのタグ内に</p>  
  </body>  
</html>
```



～ 画面が表示される仕組み（HTML） ～

HTMLの書き方

タイトル(titleタグ)
<title>自己紹介</title>

1番大きい見出し(h1タグ)
<h1>自己紹介ページ</h1>

2番目に大きい見出し(h2タグ)
<h2>M O C O</h2>


画像(imgタグ)

段落(pタグ)
<p>・誕生 : 2013年</p>


罫線(hrタグ)
<hr width = ¥"400¥" align = ¥"left¥">

自己紹介ページ

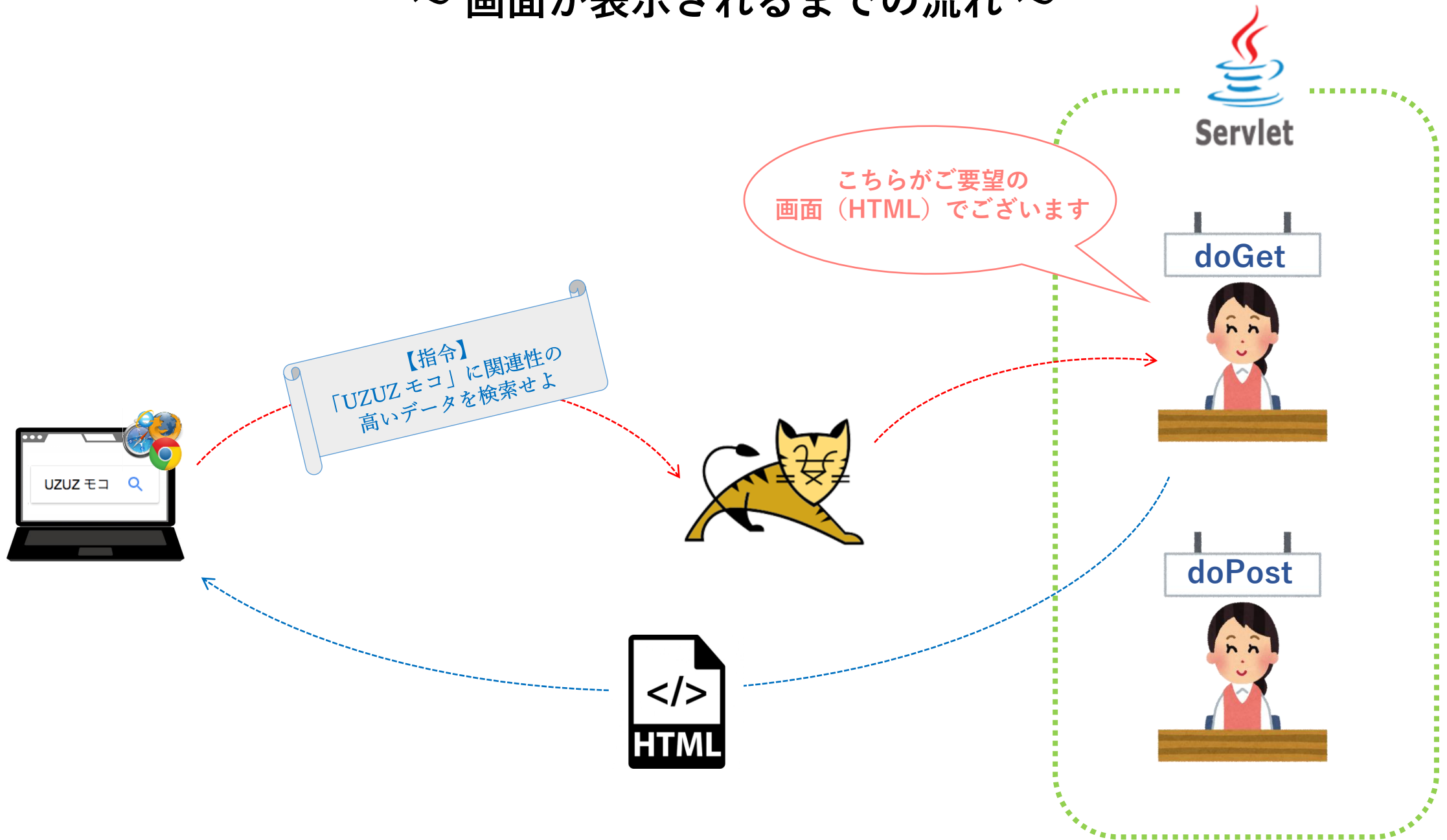
M O C O



・誕生 : 2013年
: 台湾
: 就活犬（接客、メンタルケア）
数字：「1（ONE）」
: 岡本



～ 画面が表示されるまでの流れ ～



～ 画面が表示されるまでの流れ ～



承ります！

doGet

doGetさん
リクエスト来てますよ～

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

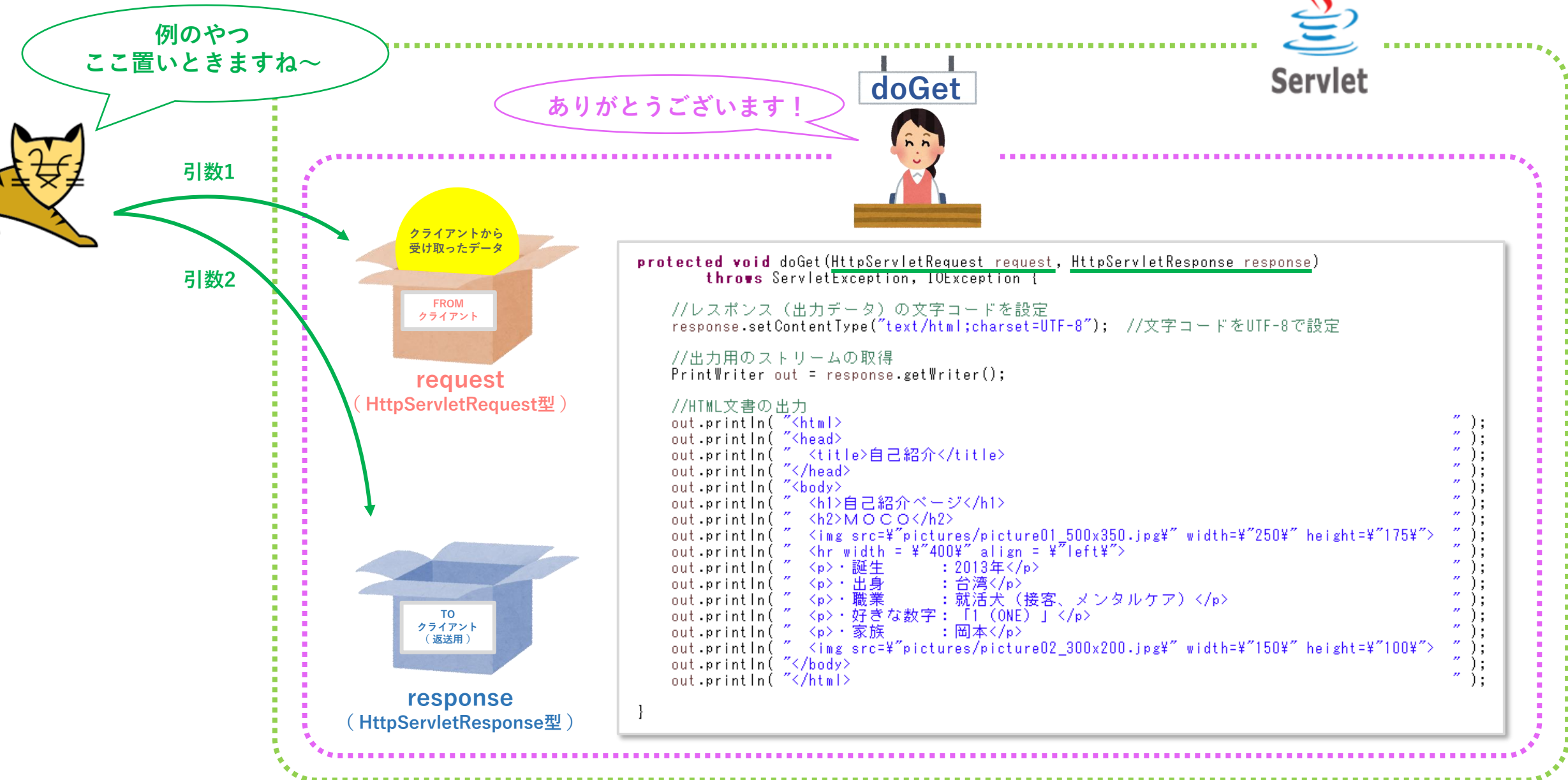
    //レスポンス（出力データ）の文字コードを設定
    response.setContentType("text/html;charset=UTF-8"); //文字コードをUTF-8で設定

    //出力用のストリームの取得
    PrintWriter out = response.getWriter();

    //HTML文書の出力
    out.println("<html>");
    out.println("<head>");
    out.println("    <title>自己紹介</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("    <h1>自己紹介ページ</h1>");
    out.println("    <h2>M O C O</h2>");
    out.println("    <img src=\"%pictures/picture01_500x350.jpg\" width=\"%250\" height=\"%175\">");
    out.println("    <hr width = \"%400\" align = \"%left\">");
    out.println("    <p>・誕生      : 2013年</p>");
    out.println("    <p>・出身      : 台湾</p>");
    out.println("    <p>・職業      : 就活犬（接客、メンタルケア）</p>");
    out.println("    <p>・好きな数字 : 「1（ONE）」</p>");
    out.println("    <p>・家族      : 岡本</p>");
    out.println("    <img src=\"%pictures/picture02_300x200.jpg\" width=\"%150\" height=\"%100\">");
    out.println("</body>");
    out.println("</html>");

}
```

～ 画面が表示されるまでの流れ ～



～ 画面が表示されるまでの流れ ～



doGet



request

クライアントから送信されたデータが詰め込まれたインスタンス (HttpServletRequest型)。



request
(HttpServletRequest型)

response

クライアントに返送するデータを格納するためのインスタンス (HttpServletResponse型)。
doGet/doPostの処理終了後にクライアントへと送信される。



response
(HttpServletResponse型)

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    //レスポンス（出力データ）の文字コードを設定
    response.setContentType("text/html;charset=UTF-8"); //文字コードをUTF-8で設定

    //出力用のストリームの取得
    PrintWriter out = response.getWriter();

    //HTML文書の出力
    out.println("<html>");
    out.println("<head>");
    out.println("    <title>自己紹介</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("    <h1>自己紹介ページ</h1>");
    out.println("    <h2>MO&O&O</h2>");
    out.println("    <img src=\"%pictures/picture01_500x350.jpg\" width=\"%250\" height=\"%175\">");
    out.println("    <hr width = \"%400\" align = \"%left\">");
    out.println("    <p>・誕生      : 2013年</p>");
    out.println("    <p>・出身      : 台湾</p>");
    out.println("    <p>・職業      : 就活犬（接客、メンタルケア）</p>");
    out.println("    <p>・好きな数字 : 「1 (ONE)」</p>");
    out.println("    <p>・家族      : 岡本</p>");
    out.println("    <img src=\"%pictures/picture02_300x200.jpg\" width=\"%150\" height=\"%100\">");
    out.println("</body>");
    out.println("</html>");

}
```

～ 画面が表示されるまでの流れ ～



HTML出力の 3STEP

STEP①

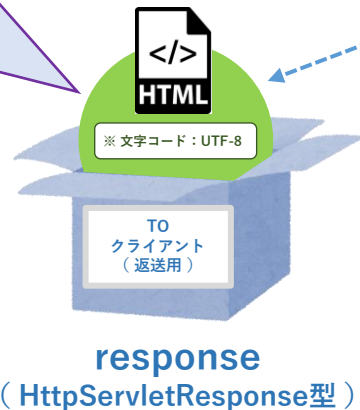
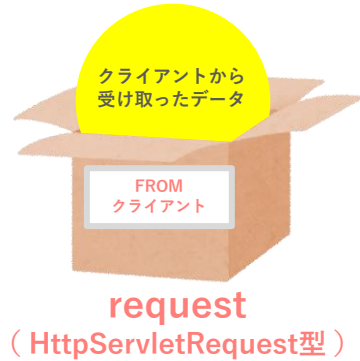
送信するテキストデータ
(HTML) の文字コード
を設定。

STEP②

出力ストリームの取得。

STEP③

出力ストリームを使って
送信するテキストデータ
(HTML) をresponseに
セット。



doGet



```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
```

STEP
①

```
//レスポンス（出力データ）の文字コードを設定
response.setContentType("text/html;charset=UTF-8"); //文字コードをUTF-8で設定
```

STEP
②

```
//出力用のストリームの取得
PrintWriter out = response.getWriter();
```

STEP
③

```
//HTML文書の出力
out.println("<html>");
out.println("<head>");
out.println("  <title>自己紹介</title>");
out.println("</head>");
out.println("<body>");
out.println("  <h1>自己紹介ページ</h1>");
out.println("  <h2>MOCCO</h2>");
out.println("  <img src=\"%pictures/picture01_500x350.jpg\" width=\"%250\" height=\"%175\">");
out.println("  <hr width = \"%400\" align = \"%left\">");
out.println("  <p>・誕生      : 2013年</p>");
out.println("  <p>・出身      : 台湾</p>");
out.println("  <p>・職業      : 就活犬（接客、メンタルケア）</p>");
out.println("  <p>・好きな数字 : 「1 (ONE)」</p>");
out.println("  <p>・家族      : 岡本</p>");
out.println("  <img src=\"%pictures/picture02_300x200.jpg\" width=\"%150\" height=\"%100\">");
out.println("</body>");
out.println("</html>");
```

```
}
```


～ 画面が表示されるまでの流れ ～



HTML出力の 3STEP

STEP①

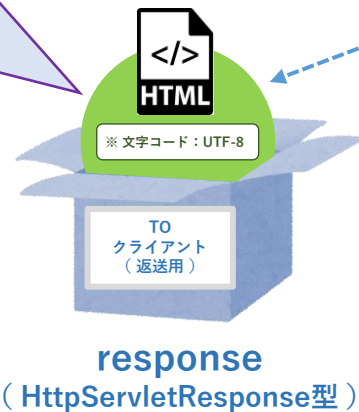
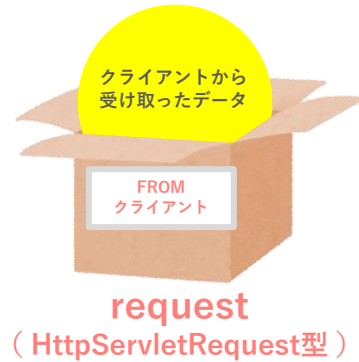
送信するテキストデータ
(HTML) の文字コード
を設定。

STEP②

出力ストリームの取得。

STEP③

出力ストリームを使って
送信するテキストデータ
(HTML) をresponseに
セット。



doGet



```
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {
```

STEP
①

```
//レスポンス（出力データ）の文字コードを設定  
response.setContentType("text/html;charset=UTF-8"); //文字コードを設定
```

STEP
②

```
//出力用のストリームの取得  
PrintWriter out = response.getWriter();
```

STEP
③

```
//HTML文書の出力  
out.println("<html>");  
out.println("<head>");  
out.println("  <title>自己紹介</title>");  
out.println("</head>");  
out.println("<body>");  
out.println("  <h1>自己紹介ページ</h1>");  
out.println("  <h2>MOCCO</h2>");  
out.println("  <img src='\"pictures/picture01_500x350.jpg\"' width='\"250\"\" height='\"175\"\">");  
out.println("  <hr width = \"400\">");  
out.println("  <p>・誕生 : 1985年</p>");  
out.println("  <p>・出身 : 台湾</p>");  
out.println("  <p>・職業 : 就活犬（メンタルケア）</p>");  
out.println("  <p>・好きな数字 : 1 (ONE)</p>");  
out.println("  <p>・家族 : 1人</p>");  
out.println("  <img src='\"pictures/picture02_300x200.jpg\"' width='\"150\"\" height='\"100\"\">");  
out.println("</body>");  
out.println("</html>");
```

出力ストリーム

out
(PrintWriter型)

responseに
テキストデータをセットする
梱包係！

～ 画面が表示されるまでの流れ ～



HTML出力の 3STEP

STEP①

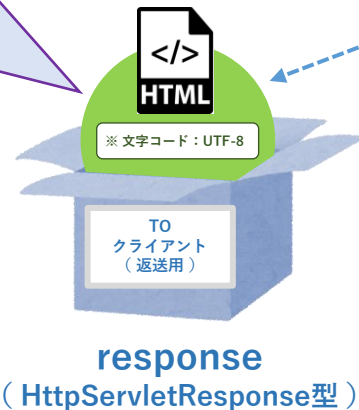
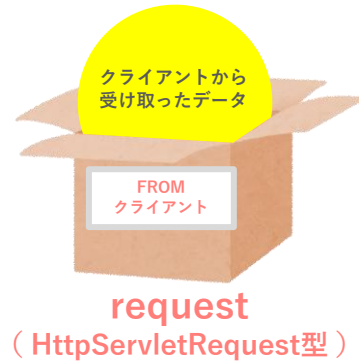
送信するテキストデータ
(HTML) の文字コード
を設定。

STEP②

出力ストリームの取得。

STEP③

出力ストリームを使って
送信するテキストデータ
(HTML) をresponseに
セット。



doGet



```
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {
```

STEP
①

```
//レスポンス（出力データ）の文字コードを設定  
response.setContentType("text/html;charset=UTF-8"); //文字コードをUTF-8で設定
```

STEP
②

```
//出力用のストリームの取得  
PrintWriter out = response.getWriter();
```

STEP
③

```
//HTML文書の出力  
out.println("<html>");  
out.println("<head>");  
out.println("  <title>自己紹介</title>");  
out.println("</head>");  
out.println("<body>");  
out.println("  <h1>自己紹介ページ</h1>");  
out.println("  <h2>MOCCO</h2>");  
out.println("  <img src=\"%pictures/picture01_500.jpg\" width=\"%250\" height=\"%175\">");  
out.println("  <hr width = \"%400\" align = \"%lef");  
out.println("  <p>・誕生      : 2013年</p>");  
out.println("  <p>・出身      : 台湾</p>");  
out.println("  <p>・職業      : 就活犬（接客、メンタルケア）</p>");  
out.println("  <p>・好きな数字 : 「1 (ONE)」</p>");  
out.println("  <p>・家族      : 岡本</p>");  
out.println("  <img src=\"%pictures/picture02_300x200.jpg\" width=\"%150\" height=\"%100\">");  
out.println("</body>");  
out.println("</html>");
```



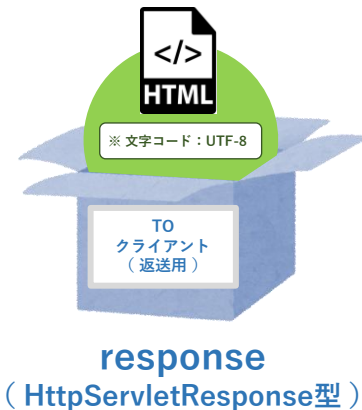
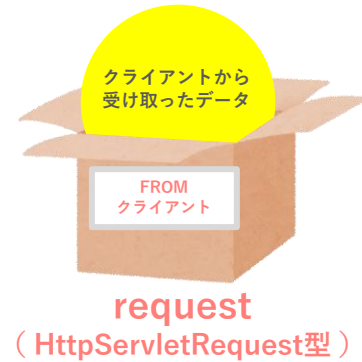
～ 画面が表示されるまでの流れ ～



doGet

クライアントに
返送お願いします！

了解しました！



```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    //レスポンス（出力データ）の文字コードを設定
    response.setContentType("text/html;charset=UTF-8"); //文字コードをUTF-8で設定

    //出力用のストリームの取得
    PrintWriter out = response.getWriter();

    //HTML文書の出力
    out.println("<html>");
    out.println("<head>");
    out.println("    <title>自己紹介</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("    <h1>自己紹介ページ</h1>");
    out.println("    <h2>MOCO</h2>");
    out.println("    <img src=\"%pictures/picture01_500x350.jpg\" width=\"%250\" height=\"%175\">");
    out.println("    <hr width = \"%400\" align = \"%left\">");
    out.println("    <p>・誕生      : 2013年</p>");
    out.println("    <p>・出身      : 台湾</p>");
    out.println("    <p>・職業      : 就活犬（接客、メンタルケア）</p>");
    out.println("    <p>・好きな数字 : 「1 (ONE)」</p>");
    out.println("    <p>・家族      : 岡本</p>");
    out.println("    <img src=\"%pictures/picture02_300x200.jpg\" width=\"%150\" height=\"%100\">");
    out.println("</body>");
    out.println("</html>");

}
```

～ 画面が表示されるまでの流れ ～



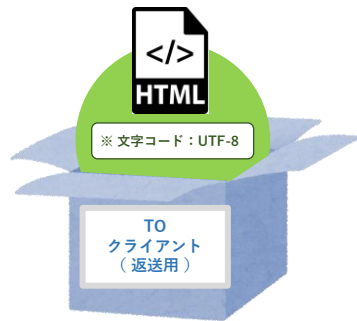
doGet



クライアントに
返送をお願いします！

了解しました！

送信するのは
バイナリデータ
(2進数のデータ)



response
(HttpServletResponse型)

```
doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    // 出力データ) の文字コードを設定
    response.setContentType("text/html; charset=UTF-8"); // 文字コードをUTF-8で設定
    // 出力ストリームの取得
    PrintWriter out = response.getWriter();

    // HTML文書の出力
    out.println("<html>");
    out.println("<head>");
    out.println("    <title>自己紹介</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("    <h1>自己紹介ページ</h1>");
    out.println("    <h2>MOO</h2>");
    out.println("    <img src=\"% pictures/picture01_500x350.jpg\" width=\"%250\" height=\"%175\">");
    out.println("    <hr width = \"%400\" align = \"%left\">");
    out.println("    <p>・誕生      : 2013年</p>");
    out.println("    <p>・出身      : 台湾</p>");
    out.println("    <p>・職業      : 就活犬（接客、メンタルケア）</p>");
    out.println("    <p>・好きな数字 : 「1 (ONE)」</p>");
    out.println("    <p>・家族      : 岡本</p>");
    out.println("    <img src=\"% pictures/picture02_300x200.jpg\" width=\"%150\" height=\"%100\">");
    out.println("</body>");
    out.println("</html>");
}
```

～ 文字コードの設定 ～



再変換
(デコード)



変換
(エンコード)

```
"<html>
"<head>
"  <title>自己紹介</title>
"</head>
"<body>
"  <h1>自己紹介ページ</h1>
"  <h2>MOCO</h2>
"  <img src=%"pictures/picture01_500x350.jp
"  <hr width = %"400%" align = %"left%">
"  <p>・誕生      : 2013年</p>
"  <p>・出身      : 台湾</p>
"  <p>・職業      : 就活犬(接客、メンタル
"  <p>・好きな数字 : 「1 (ONE)」</p>
"  <p>・家族      : 岡本</p>
"  <img src=%"pictures/picture02_300x200.jp
"</body>
"</html>
```

テキストデータ
(人が読める)

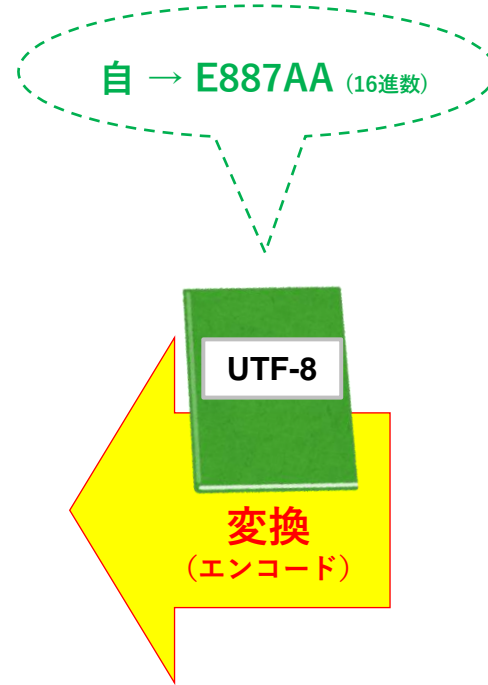
バイナリデータ
(コンピュータが読める)

テキストデータ
(人が読める)

～ 文字コードの設定 ～

00000000011101000
1000011110101010

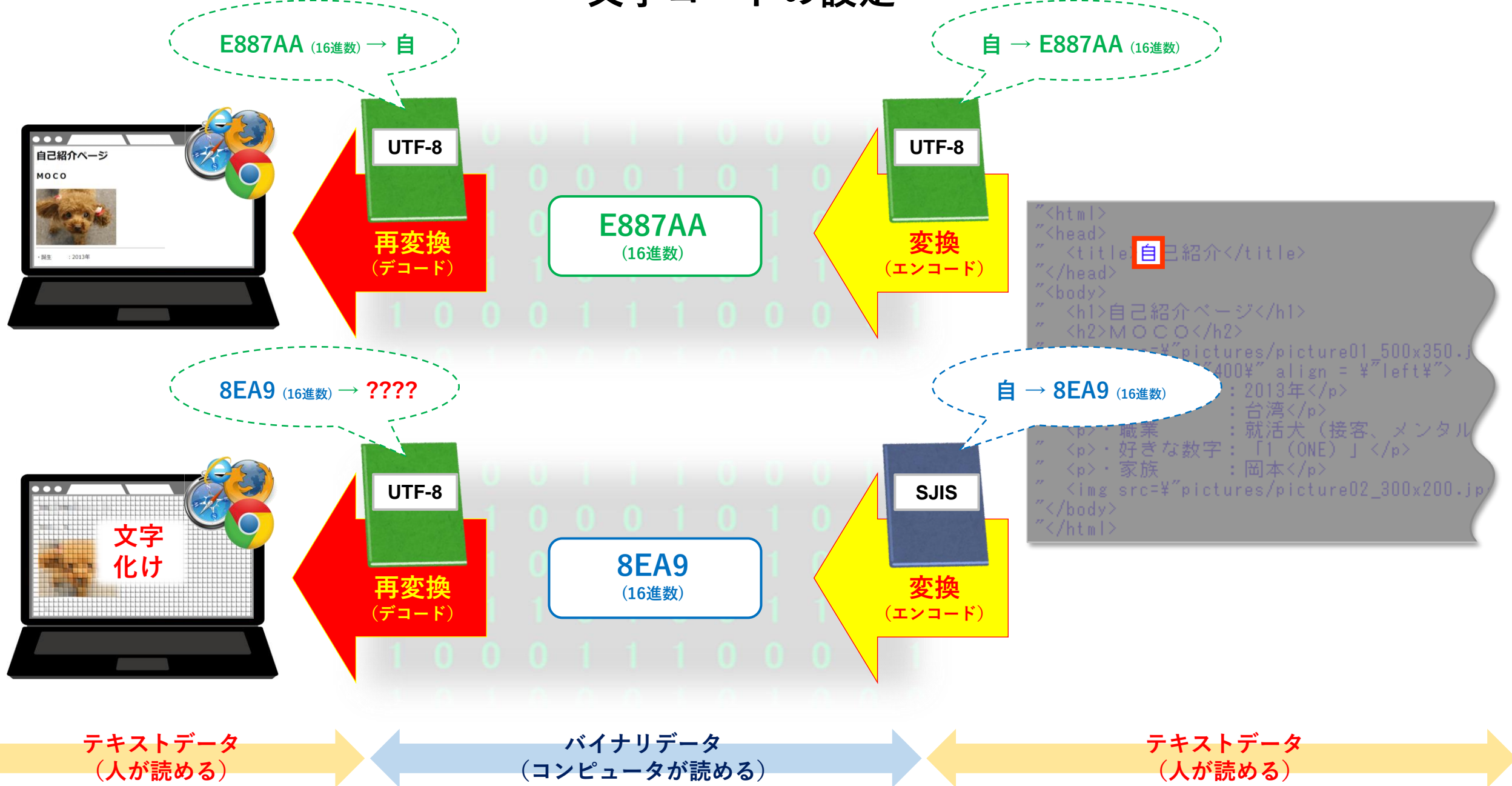
バイナリデータ
(コンピュータが読める)



“ 自 ”

テキストデータ
(人が読める)

～ 文字コードの設定 ～



【演習】

SelfIntroduction.javaを書き換えて
自分の自己紹介ページを作ってみましょう！



 Challenge!! 

《MOCOからの挑戦状》

自己紹介ページに以下の要素も取り入れてみて！
Googleで調べるとすぐに答えに辿り着くよ！

- ・画像を中央寄せで表示
- ・複数行、複数列のテーブル(表)を作成
- ・背景色を好きな色に指定
- ・外部サイトへのリンクをはる
- ・太文字、イタリック文字のどちらかを使う

【演習】

手順

- (1) eclipse上に新しい動的Webプロジェクトを作成する。
プロジェクト名：Ex_5_02
- (2) プロジェクトの srcパッケージ直下に workパッケージを作成する。
- (3) プロジェクト「Sample5_02_1」を参考に演習に取り組む。
 - ・ Javaリソース > src > work 直下にjavaソースコードを配備
 - ・ WebContent > WEB-INF 直下にweb.xmlを配備