

```
<html>
<head>
<title>パフォーマプロフィール</title>
</head>
<body>
```

```
<fss:errors error="false">
```

（条件）エラーがない場合の処理を開始 …①-A

```
<p>パフォーマプロフィール</p>
```

```
<fss:set var="out" value="<span style='color:red'>ひみつ</span>" />
```

「ひみつ」という文字を「out」とする

「out」は任意の半角英数を指定する

使用モジュール パフォーマモジュール

名前：

```
<fss:out value="${performer.name}" />
```

前のページ（zExplanation3）のリンクから指定されたパフォーマの名前を表示する

```
<br>
```

職業：

```
<fss:set var="jobChk" value="${performer.job!=''}" />
```

「performer.job!='」を「jobChk」とする

「jobChk」は任意の半角英数を指定する

「jobChk」を設定しない場合は「performer.job!='」を直接「fss:if」の条件文に記入する

```
<fss:if test="${jobChk}">
```

（条件）パフォーマの職業が登録されている場合の処理 …②-A

```
<fss:out value="${performer.job}" />
```

パフォーマの職業を表示する

```
</fss:if>
```

②-A の処理終了

```
<fss:if test="${!jobChk}">
```

（条件）②-A の条件の反対の処理（職業が登録されていない場合の処理） …②-B

```
<fss:out value="${out}" />
```

「out」を表示する（始めに「out」には「ひみつ(赤色)」を設定している）

```
</fss:if>
```

②-B の処理終了

誕生日 (年齢) :

```
<fss:set var="ageChk" value="{performer.age!=0}" />
```

「performer.age!=0」を「ageChk」とする

「ageChk」は任意の半角英数を指定する

```
<fss:if test="{ageChk}">
```

(条件) パフォーマの年齢が登録されている場合の処理 …③-A

```
<fss:out value="{performer.birth.formatDate}" />
```

パフォーマの生年月日を「yyyy-mm-dd」で表示する

```
(<fss:out value="{performer.birth.ageString}" />)
```

パフォーマの年齢を「歳」をつけて表示する

星座 :

```
<fss:out value="{performer.birth.constellation}" />
```

パフォーマの星座を表示する

```
</fss:if>
```

③-A の処理終了

```
<fss:if test="{!ageChk}">
```

(条件) ③-A の条件の反対の処理 (年齢が登録されていない場合の処理) …③-B

```
<fss:out value="{out}" />
```

「out」を表示する

```
</fss:if>
```

③-B の処理終了

血液型 :

```
<fss:set var="bloodChk" value="{performer.bloodString!=''}" />
```

「performer.bloodString!='」を「boolChk」とする

「boolChk」は任意の半角英数を指定する

```
<fss:if test="{bloodChk}">
```

(条件) パフォーマの血液型が登録されている場合の処理 …④-A

```
<fss:out value="{performer.bloodString}" />
```

パフォーマの血液型を表示する

```
</fss:if>
```

④-A の処理終了

```
<fss:if test="{!bloodChk}">
```

(条件) ④-A の条件の反対の処理 (血液型が登録されていない場合の処理) …④-B

```
<fss:out value="{out}" />
```

「out」を表示する

```
</fss:if>
```

④-B の処理終了

```
<br>
```

```
<fss:set var="imgChk" value="{performer.thumbnailImageFile3.exists}" />
```

「performer.thumbnailImageFile3.exists」を「imgChk」とする

「imgChk」は任意の半角英数を指定する

```
<fss:if test="{imgChk}">
```

(条件) パフォーマの画像が登録されている場合の処理 …⑤-A

```
">
```

パフォーマの画像を表示する

```
</fss:if>
```

⑤-A の処理終了

```
<fss:if test="{!imgChk}">
```

(条件) ⑤-A の条件の反対の処理 (画像が登録されていない場合の処理) …⑤-B

```
<font color="red">
```

がぞうなし

```
</font>
```

```
</fss:if>
```

⑤-B の処理終了

```
<br>
```

ひとこと :

```
<fss:set var="msgChk" value="{performer.message!=''}" />
```

「performer.message!='」を「msgChk」とする

「msgChk」は任意の半角英数を指定する

```
<fss:if test="{msgChk}">
```

(条件) パフォーマのひとことが登録されている場合の処理 …⑥-A

```
<fss:out value="{performer.message}" />
```

パフォーマのひとことを表示する

```
</fss:if>
```

⑥-A の処理終了

```
<fss:if test="{!msgChk}">
```

(条件) ⑥-A の条件の反対の処理 (ひとことが登録されていない場合の処理) …⑥-B

```
<fss:out value="{out}" />
```

```
「out」を表示する
</fss:if>
    ⑥-B の処理終了
<br>
</fss:errors>
    ①-A の処理終了

<fss:errors error="true" var="errors">
    (条件) ①-A の反対の処理 (エラーがある場合の処理) ...①-B
<span style="color:red">
<fss:forEach items="{errors}" var="error">
    (ループ) 順番にエラーの処理を開始 ...⑦
    「error」はこれから処理をするエラーを1つずつ参照する
    「error」は任意の半角英数を指定する
<fss:out value="{error}"/><br>
    エラーの内容を表示する
</fss:forEach>
    ⑦のループ処理終了
</span>
</fss:errors>
    ①-B の処理終了

</body>
</html>
```