

第2講 Webアプリケーションの概要、HTMLとフォーム

- Webアプリケーションとは
- PHPで開発するために必要な技術
- Webアプリケーションの開発事例
- HTMLとフォーム
- (HTTP)

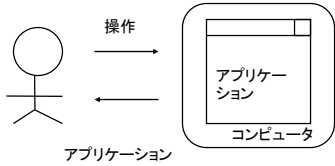
1

Webアプリケーションとは

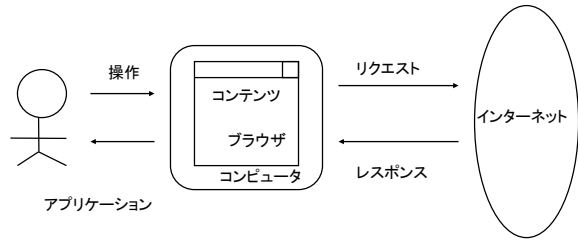
- Webアプリケーション
 - ブラウザでWebページ(ホームページ)を閲覧するのと同じ感覚で操作するアプリケーション
- スタンドアローン型アプリケーション
 - 単一マシン上でネットワークを介さずに起動するアプリケーション

2

通常のアプリケーションvs Webアプリケーション



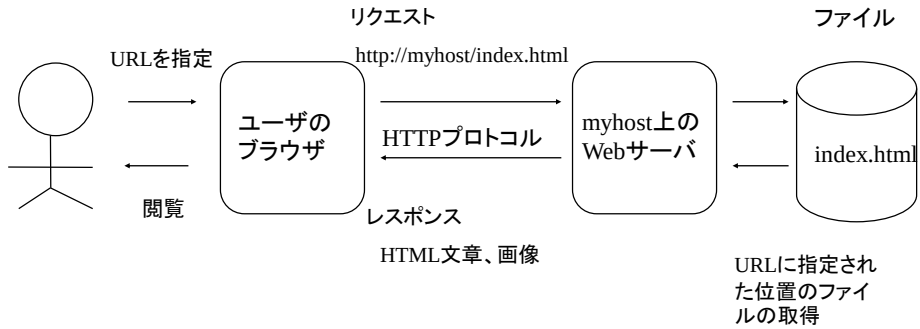
通常のアプリケーション



Webアプリケーション

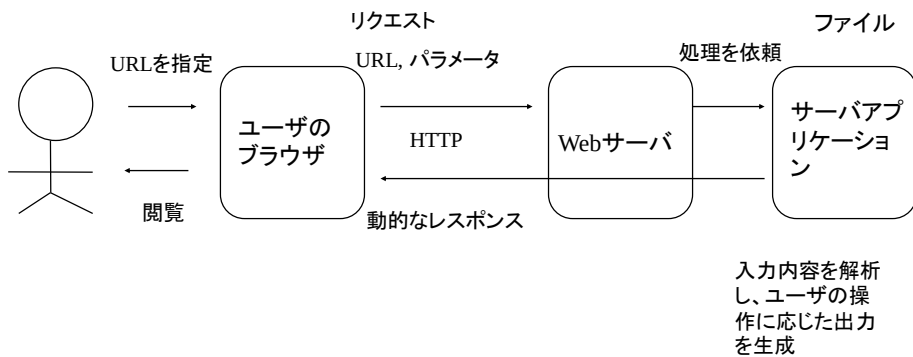
3

ブラウザとWebサーバ間の通信



4

Webアプリケーションの仕組み

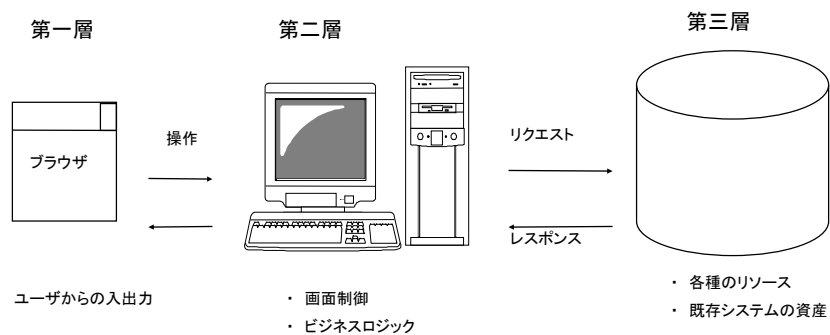


5

Webアプリケーションの階層

Webアプリケーション

- 「N階層モデル」で構成
- 通常は3階層



Webアプリケーションにおける3階層モデル

6

Webアプリケーションの特徴

- ユーザインターフェイスにブラウザを利用
- 入出力画面は基本的にHTMLで記述
- クライアント側(ブラウザ)とサーバ側(Webサーバ+アプリケーションサーバ)に分かれている
- クライアントとサーバ間の通信にHTTPを使用
- サーバ側がさらに、2つの層に分けることができ、クライアント側と合わせ、一般に3階層モデルで構成

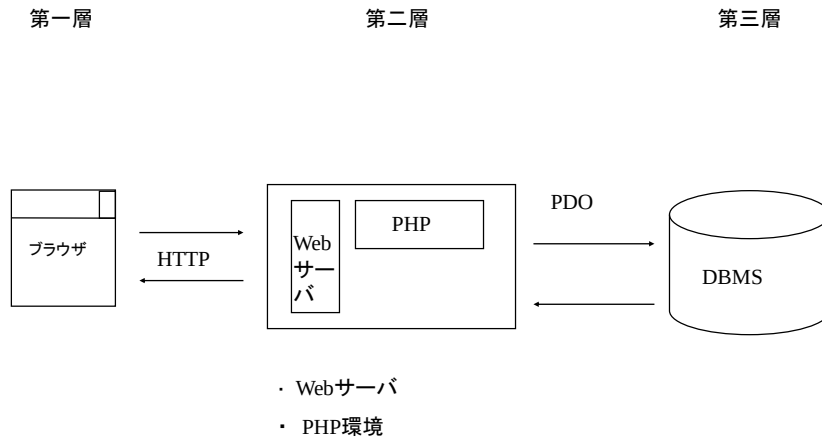
7

PHPで開発するために必要な技術

- (1) Webアプリケーション(ソフトウェア)
- (2) クライアントサーバシステム、Webサーバ
- (3) Webアプリケーションで必要となる基本知識
- (4) サーバサイドスクリプトPHPプログラミングの基本概念
- (5) HTMLとフォームタグ
- (6) CGIとの比較
- (7) データベース、MySQLの基本知識
- (8) PHPプログラミングとデータベースとの接続
- (9) PHPによるWebシステム開発

8

PHPによるWebアプリケーションの構成図



9

Webアプリケーションの開発事例

- ・ イン트라ネット系
 - ・ 既存の社内システムをWebアプリケーション化したもの
 - ・ 販売管理
 - ・ 人事、給与管理
 - ・ 受発注管理
 - ・ 帳票管理

10

Webアプリケーションの開発事例

- 公開系
 - 一般顧客向けに公開しているWebページに、顧客に対するサービスとしてWebサービスを追加したもの
 - BtoC (Business to Consumer)アプリケーション
 - 旅行の予約、航空券の予約、各種保険料の見積もり
 - 自動車購入の見積もり、不動産物件の検索
 - オンラインバンキング、オンライントレーディング
 - 掲示板サービス

11

Webアプリケーションの開発事例

- 公開系
 - 一般顧客向けに公開しているWebページに、顧客に対するサービスとしてWebサービスを追加したもの
 - ポータルアプリケーション
 - サーチエンジン
 - Webメールサービス
 - 辞書、翻訳サービス
 - リアルタイム速報(株価、円、スポーツニュースなど)

12

Webサーバ

- クライアントからの要求に応じて、Webページなどの応答をする
 - 画像(静止画、動画)、音声、プログラムやデータ、テキスト、PDFなど
- Apache, IIS(Internet Information Server), Java Web Serverなど

13

Webブラウザ

- サーバに要求を出し、サーバから返された応答を受け付けて処理
- URL (Uniform Resource Locator)を入力すると、サーバにその要求を送る
- サーバが要求に対する応答を返すと、応答に応じた処理をおこなう
- Internet Explorer, Opera, Firefox, Chrome, Safariなど

14

HTTP (HyperText Transfer Protocol)

- Webサーバとブラウザの間で、要求と応答の受け渡しをおこなうための通信規約(プロトコル)。
HyperTextのやりとりを実現するプロトコル
- HyperText
 - 文書と文書、文書とマルチメディア情報(音声、画像など)を関連づけることができるコンテンツ
- ブラウザからの一つの要求に対して、Webサーバは一つの応答を返す
- TCP/IPにおけるアプリケーション層に位置する

15

リクエスト(要求)メッセージ

- WebブラウザからWebサーバに対して、データを要求する時送るメッセージ
- 形式
 - リクエストライン
 - リクエストメソッド URI プロトコルバージョン
 - ヘッダ(必要に応じてブラウザにより指定)
 - フィールド名: フィールド値 {[;パラメータ名=パラメータ値]}
 - ボディ
 - (空行)

16

リクエストメソッド一覧

- GET
 - URIで指定される資源の内容を取得する
- HEAD
 - URIで指定される資源に関する情報だけを取得し、内容自体は取得しない
- POST
 - URIで指定される資源にデータを送り、またその資源の内容を取得する

17

リクエスト(要求)メッセージの例

- `http://www.hoge.ac.jp/index.html`
- メッセージの形式
 - GET /index.html HTTP/1.1
 - HOST: www.hoge.ac.jp
 - (空行)

18

レスポンス(応答)メッセージ

- リクエストメッセージに対するWebサーバの応答メッセージ
 - メッセージの形式
 - ステータスライン
 - HTTPバージョン ステータスコード 結果フレーズ
 - ヘッダ
 - フィールド名: フィールド値 {[;パラメータ名=パラメータ値]}
 - ボディ
 - HTML文本体

19

ステータスコードの例

- 1XX (カテゴリー)
 - インフォメーション (カテゴリーの意味)
- 2XX
 - 正常終了
- 3XX
 - リダイレクト(処理の変更通知)
- 4XX
 - クライアントエラー
- 5XX
 - サーバエラー

20

レスポンスメッセージの例

```
HTTP/1.1 200 OK
Data: Mon, 21 Apr 2003 09:15:00 GMT
Server: Apache/1.3.26(Unix) PHP/4.2.2
Last-Modified: Fri, 04 Apr 2003 04:32:48 GMT
Etag: "10028-7fbf-3e8d0af0"
Content-Length: 32703
Content-Type: text/html
```

```
<HTML>
<HEAD>
<TITLE> TUIS Education Domain</TITLE>
</HEAD>
.....
<BODY> ..... </BODY>
</HTML>
```