

LLL/Ajax

「トリプル・エル/エイジャックス」のご紹介



LLL/Ajaxとは

- 当社のRAD(Rapid Application Development)ツール LLLシリーズの最新製品です。
- Ajax(Asynchronous JavaScript + XML)アプリケーションを効率良く開発できます。
- 各種OS上のHTML5+JavaScript対応の各種ブラウザ上であれば動作できるので、WindowsのデスクトップPCはもちろん、それ以外のタブレットやスマートフォンなどモバイル環境でも使えるアプリケーションの開発が簡単に行えるツールです。



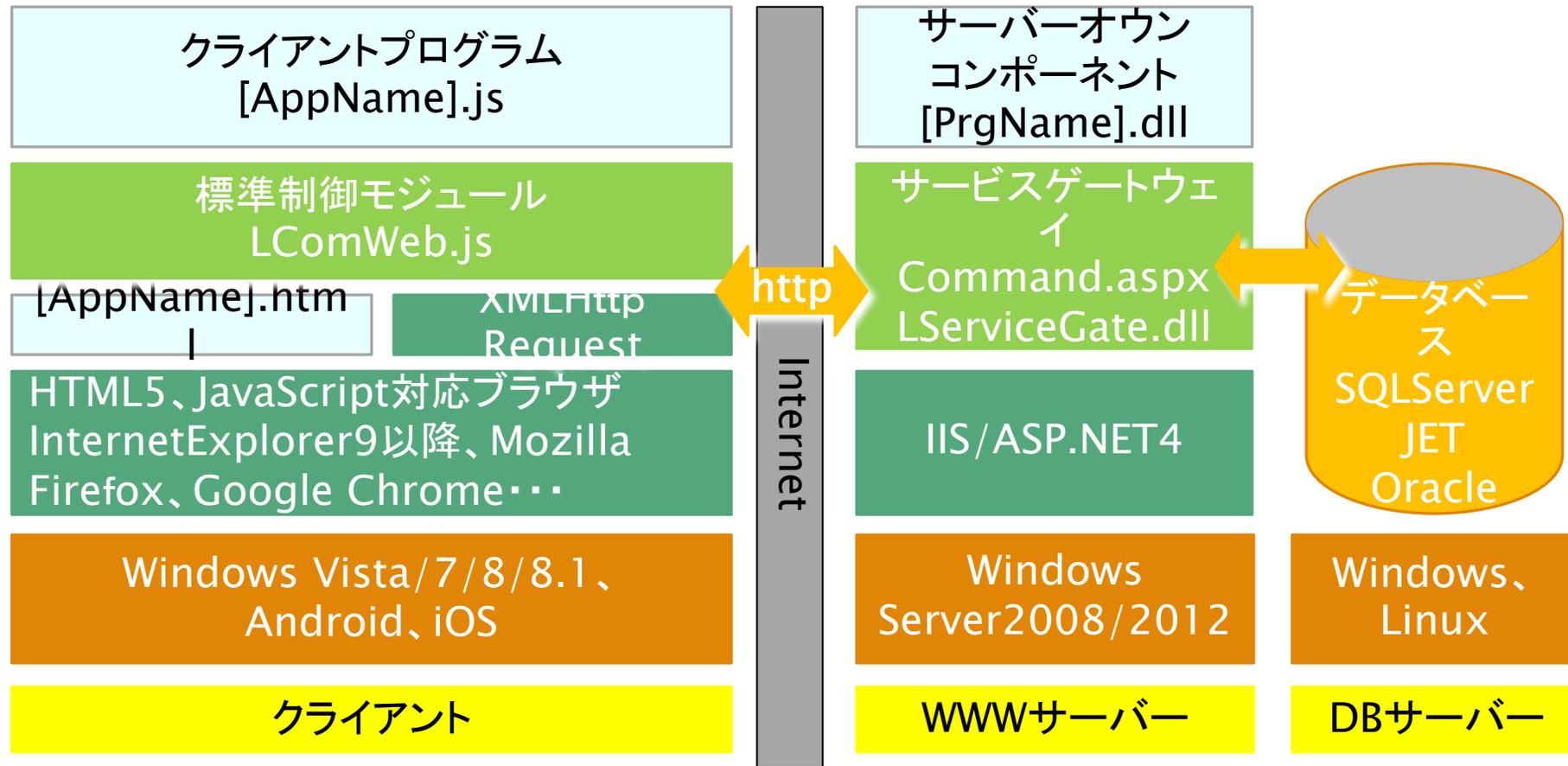
LLL/Ajaxが実現する優れた操作性

- Webブラウザ上で動作するアプリでありながら、ページ単位でまとめて入力後に送信 (Submit) し、結果画面に遷移するのではなく、項目単位で入力のつどデータはサーバーに送信され、画面遷移を伴わずに項目ごとに直ちにクライアントの画面上にレスポンスを描画することができます。
- 例えば明細行で商品コードを入力すると直ちに該当する商品名や単価がその横に表示されるという具合です。
- この動作は DynamicHTML と JavaScript の組み合わせで実現します。

LLL/Ajaxがターゲットとするアプリ

- 優れたユーザーインターフェース(操作性)の実現が必要なアプリ
- デスクトップPCだけでなくインターネットのモバイル環境でも使用できるアプリ
- 企業、学校など各種法人や官公庁等で使用する各種事務用アプリケーション
- クライアントの入力画面からサーバー上のRDB(SQL-Server、ORACLE、Jet)のテーブルに対する登録、参照、修正、削除などを行うアプリケーション
- マスターメンテナンスのように1つのレコード(テーブル)を対象とする入力だけでなく、多くのテーブルを参照したり、1つのヘッダーレコードに対して複数の明細レコードを持つ伝票形式のようなデータをひとつのトランザクションとして更新するようなケースにも簡単に対応できます。

LLL/Ajaxアプリの動作構造図



LLL/Ajaxアプリの動作の仕組み

- Webサーバー上のLLL/Ajaxアプリは、クライアントのブラウザからInternet経由のリクエストをサービスゲートウェイ(Command.aspx/LServiceGate.dll)で受けて、クライアントプログラム([AppName].js)、標準制御モジュール(LComWeb.js)、画面ファイル([AppName].html)をクライアントに配信し、デプロイメントを完了します。
- クライアントでは読み込まれた画面ファイル([AppName].html)がブラウザに表示され、クライアントプログラム([AppName].js)とLLL/Ajaxの標準制御モジュール(LComWeb.js)の記述にしたがいアプリケーションとして動作します。
- 必要に応じてWebサーバー上にもサーバーオウンコンポーネント([PrgName].dll)としてロジックを配置しておくことができます。この機能はクライアントで撮影した写真をアップロードさせて更新するなどの特殊なDB処理でも威力を発揮します。

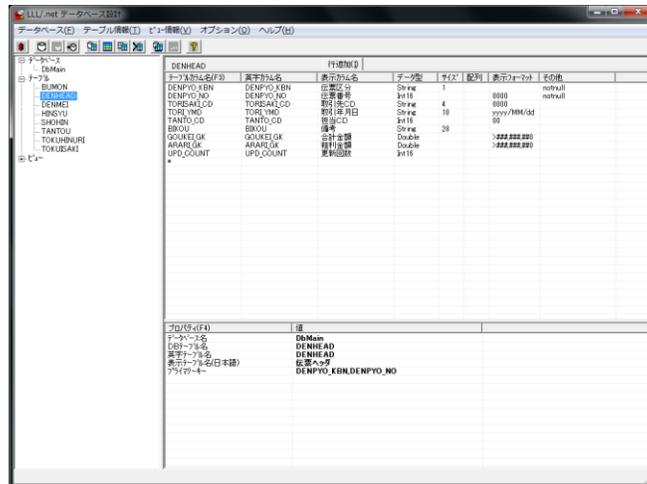
LLL/Ajax が提供する開発/実行機能

製品としての LLL/Ajax は下記の3つの機能から構成されています

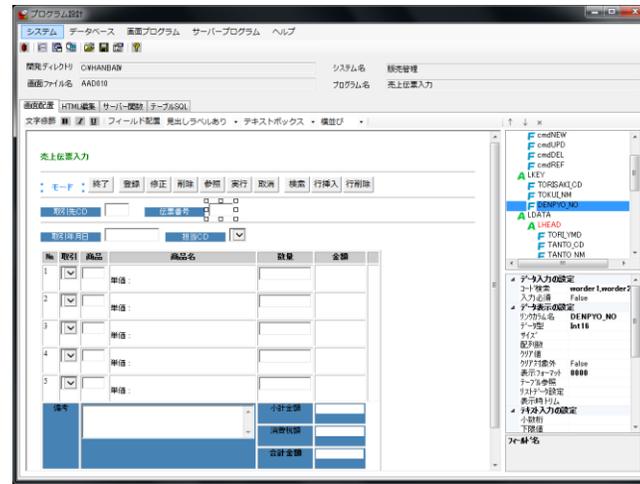
- 標準コンポーネント(ランタイムルーチン)
 1. LComWeb.js (クライアント制御モジュール)
 2. Command.aspx (ASP.NET Webサービス)
 3. LServiceGate.dll (サーバー制御モジュール)
- プログラムテンプレート(処理の流れとフォームの雛形)
「伝票形式」、「単票形式」、「一覧表示」、「フリーフォーマット」
- IDE (LLL/Ajax開発キット本体<統合開発環境>)
「DB設計」、「フォームデザイン」、「プログラム設計」

LLL/Ajax アプリ開発手順図

開発の流れ



データベース設計



画面設計



テスト実行

LLL/Ajax開発手順の説明

- LLL/Ajax開発キット本体が提供するIDE (Integrated Development Environment) = 統合開発環境の「データベース設計」機能を使い、データベースのテーブル情報を設定します。
- 作成しようとするアプリケーションのテンプレート(雛形)を選び使用するテーブルを指定すると、IDEは必要に応じて再編集可能なテーブルSQL(テーブルアクセス用SQL文の雛形)を自動生成し、画面の初期デザインを用意します。
- テーブルSQLによって使用可能となったテーブルカラムを画面上のエリアにドラッグ & ドロップして配置しフォームのデザインを行っていきます。
- 参照テーブルの設定や、必要に応じてJavaScript によるOWNコードの追加を行いプログラム全体を完成させます。
- この時点で、直ちにテスト実行が可能になります。
- 少人数のチームでプロタイプ作成と評価を反復して繰り返し、スパイラル的に完成品に近づけていくアプリケーション開発手法であるRADに適した製品と言えます。

OWNコーディング例

明細行の数量が入力されたら単価 × 数量を計算して金額欄に表示させるというコーディングの例

この例ではテンプレートに用意された明細フィールドのフォーカスロスト時に呼び出される関数 `_app.ControlBlur` 内にOWNコードを記述します。ここで使用している `_app._WebForm` で始まる関数の実体は `LComWeb.js` (クライアント制御モジュール) に実装されたオブジェクトです。

```
_app.ControlBlur = function(field,row){}; //テンプレートには予め中身が空の状態関数が用意されています
```

この関数内にアプリに必要な独自のコードすなわちOWNコード (Javascript) を下記のように追加記述します。

//画面上の明細行番号 row の field がフォーカスを失ったら実行される関数

```
_app.ControlBlur = function(field,row){  
    if(field==="SURYO"){ //フォーカスを失った field が "SURYO" だったら下記を実行  
        var tanka=parseInt(_app._WebForm.GetData("TANKA",row),10); // "SURYO"と同じ明細行の "TANKA"の値を変数tankaに代入  
        var suryo=parseInt(_app._WebForm.GetData("SURYO",row),10); // "SURYO"の値を変数suryoに代入  
        _app._WebForm.PutData("KINGK",row,tanka*suryo); //tanka * suryo の値を "SURYO"と同じ明細行の "KINGK" に表示  
    }  
};
```

以上です。これで明細行の数量欄に値が入力されると即時自動的に金額を計算して表示(更新)される機能がアプリに付加されました。

LLL/.NET VS LLL/Ajax

LLL/.net 実行画面例
(Windows Formアプリケーション)

The screenshot shows a Windows Form application titled "売上伝票入力" (Sales Invoice Entry). It features a menu bar with options like "登録" (Register), "修正" (Edit), "削除" (Delete), "参照" (Reference), and "終了" (End). Below the menu, there are input fields for "伝票番号" (Invoice No.) set to 0001, "取引先CD" (Customer Code) set to 0001 (大阪産業), "取引年月日" (Invoice Date) set to 2003年 4月 1日, and "担当CD" (Staff Code) set to 大田 雄二x. A table lists items with columns for No., 取引 (Transaction), 商品CD (Product Code), 商品名 (Product Name), 単価 (Unit Price), 数量 (Quantity), and 金額 (Amount). The items are: 1. アーモンドカレー 中辛 (300.00 x 2 = 600), 2. カップラーメン 味噌醤油 (100.00 x 3 = 300), 3. ソーセージ M3本束 (200.00 x 2 = 400), 4. えびシューマイ 20個入り (500.00 x 2 = 1,000), 5. シャンプー ポンプ式 (1,000.00 x 2 = 2,000). A "備考" (Remarks) field contains "配送先: 東京都千代田区神田2". Summary fields at the bottom right show: 小計金額 (Subtotal) 15,600, 消費税 (Consumption Tax) 780, and 合計金額 (Total Amount) 16,380. The status bar at the bottom indicates "取引先CD: 検索フォーム (F8)".

No.	取引	商品CD	商品名	単価	数量	金額
1	売上	A001	アーモンドカレー 中辛	300.00	2	600
2	売上	A002	カップラーメン 味噌醤油	100.00	3	300
3	売上	A003	ソーセージ M3本束	200.00	2	400
4	売上	A004	えびシューマイ 20個入り	500.00	2	1,000
5	売上	B001	シャンプー ポンプ式	1,000.00	2	2,000

備考 配送先: 東京都千代田区神田2

小計金額 15,600
消費税 780
合計金額 16,380



LLL/Ajax 実行画面例
(Webブラウザアプリケーション)

The screenshot shows a web browser application titled "売上伝票入力" (Sales Invoice Entry). It features a menu bar with options like "ファイル(E)", "編集(E)", "表示(V)", "お気に入り(A)", "ツール(I)", and "ヘルプ(H)". Below the menu, there are input fields for "取引先CD" (Customer Code) set to 0001 (大阪産業), "伝票番号" (Invoice No.) set to 0001, "取引年月日" (Invoice Date) set to 2003/04/01, and "担当CD" (Staff Code) set to 大田 雄二x. A table lists items with columns for No., 取引 (Transaction), 商品 (Product), 商品名 (Product Name), 数量 (Quantity), and 金額 (Amount). The items are: 1. アーモンドカレー (300.00 x 2 = 600), 2. カップラーメン (100.00 x 3 = 300), 3. ソーセージ (200.00 x 2 = 400), 4. えびシューマイ (500.00 x 2 = 1,000), 5. シャンプー (1,000.00 x 2 = 2,000). A "備考" (Remarks) field contains "配送先: 東京都千代田区神田 2". Summary fields at the bottom right show: 小計金額 (Subtotal), 消費税 (Consumption Tax), and 合計金額 (Total Amount) 16,380. The status bar at the bottom indicates "100%".

No.	取引	商品	商品名	数量	金額
1	売上	A001	アーモンドカレー 単価: 300.00	2	600
2	売上	A002	カップラーメン 単価: 100.00	3	300
3	売上	A003	ソーセージ 単価: 200.00	2	400
4	売上	A004	えびシューマイ 単価: 500.00	2	1,000
5	売上	B001	シャンプー 単価: 1,000.00	2	2,000

備考 配送先: 東京都千代田区神田 2

小計金額
消費税
合計金額 16,380

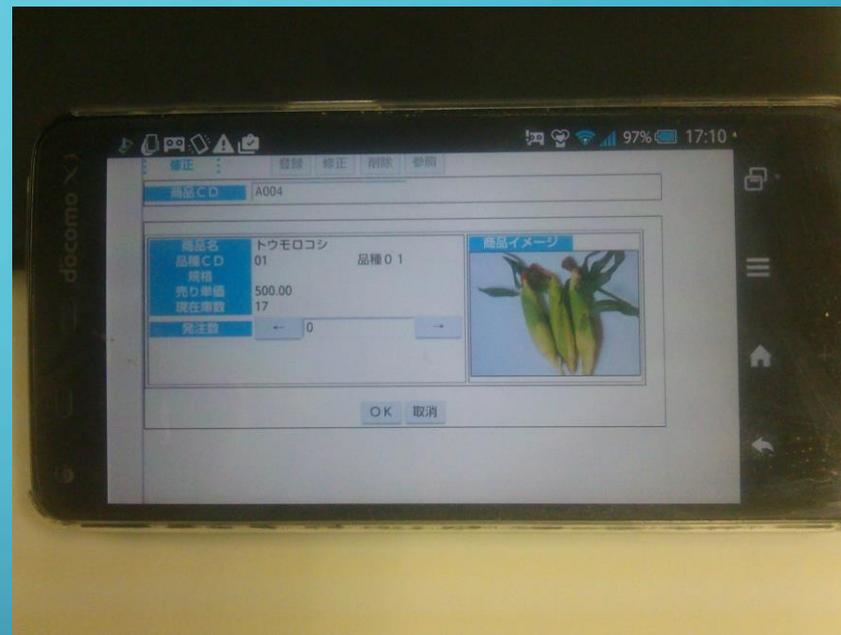
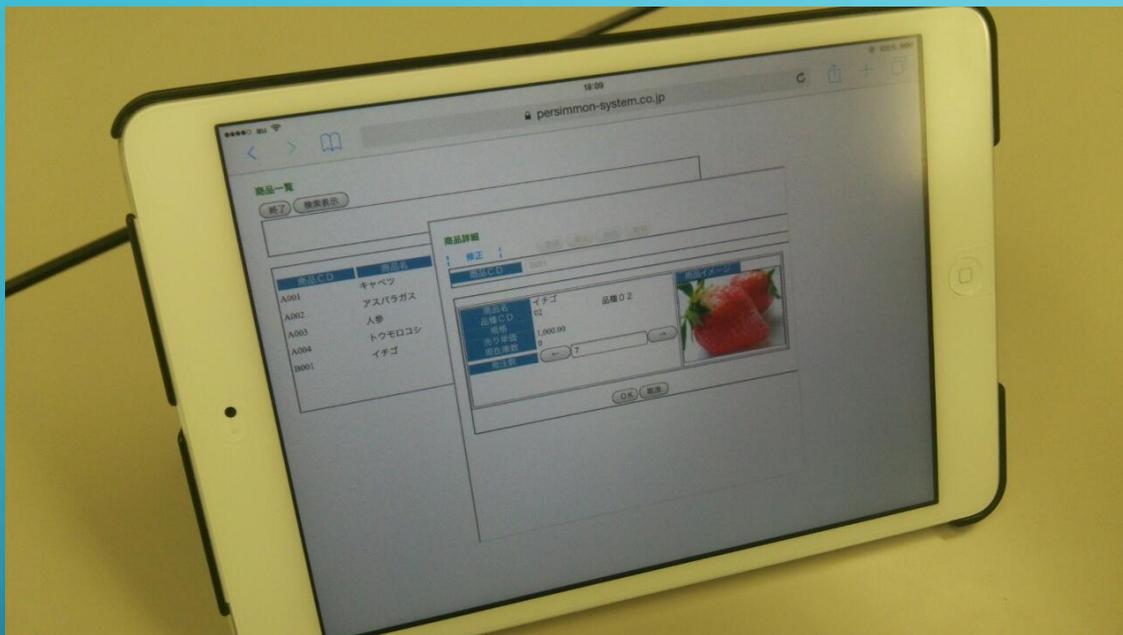
LLL/.NET と LLL/Ajax の違い

- LLLシリーズの現在の主力製品は、.NetFrameworkのスマートクライアントとしてアプリを開発するLLL/.netですが、このことはLLL/Ajax登場後も変わりません。
- LLL/Ajaxは、Webブラウザ上で優れたユーザーインターフェース(操作性)を実現しますが、LLL/.netはデスクトップPC上でさらに優れた操作性を有します。これは.NetFrameworkのWindowsフォームを使用しブラウザ上ではなくネイティブなWindowsデスクトップ上で動作するアプリケーションだからです。
- LLL/Ajaxの優れた操作性とはあくまでもブラウザ上で動作するアプリとして優れているということですが、それによりWindowsデスクトップPC以外のスマートフォンやタブレットなどのモバイル環境をはじめHTML5+JavaScript対応に対応したブラウザ搭載環境であれば、どこでも使用できるアプリケーションを実現しています。
- LLL/Ajaxは、LLL/.netに取って代わる後継品ではなく、LLL/.netと併せてお使いいただくことでアプリの動作可能環境を拡張する補完関係にある製品です。

LLL/Ajax の適用分野

- 既存システムへのスマートフォンやタブレットなどを使用するモバイルアクセス機能の追加
- インターネット利用をした多拠点（自宅、出張先等）接続機能の追加
- 既存システムへの新たなデータ抽出条件によるRDB参照機能の追加
- スマートフォンのカメラ機能やアルバムデータとの連携の追加
- 既存基幹システムのユーザーインターフェース刷新
- もちろん既存基幹システムの全面改訂、再構築、新規開発にもご利用いただけます

LLL/Ajax アプリケーションの モバイル環境での動作実行例



モバイル環境から操作可能な商品マスターメンテアプリのサンプル。(※デスクトップPCからも使えます)
スマートフォンやタブレット搭載のカメラで撮影した写真をブラウザ上のLLL/Ajaxアプリ画面からサーバーにアップロードして更新を行えます。
(※ ブラウザによっては写真連動機能は動作できない場合があります。事前に実機によるテスト確認が必要です。)
このサンプルPGは当社Webサイトで実物を公開中です。(<http://www.persimmon-system.co.jp/products/lllajax/sample/index.html>)

デメリット(注意すべき制約事項)

主にブラウザ上で動作するWebアプリなのでという理由からですが、当社のLLL/.netなどと比較して下記の制約があります。ご採用時にはご注意ください。

1. 開発したアプリは、オフラインでは動作できません。
(※スタンドアロンWeb環境実装時を除く)
2. クライアント上のローカル資源(プログラムやファイル)へのアクセスは、
利用するブラウザの仕様(アプリケーション連携の有無など)に依存します。
3. 帳票の開発機能やプリンターへの帳票出力機能は持っていません。

動作環境

1. 開発環境 < WindowsPC >

WindowsVista/7/8/8.1/10

.NetFramework4/4.5/4.5.2

VisualStudio2010以降(サーバーオウンコンポーネント作成時必要)

InternetExplorer9以降

2. 実行環境 < クライアント (PC、タブレット、スマートフォン、etc.) >

InternetExplorer9以降他、HTML5及びJavaScriptをサポートするWebブラウザ

3. 実行環境 < Webサーバー >

IIS7以降(要ASP.NET4)が搭載されているWindowsServer

.NetFramework4/4.5/4.5.2

4. データベース (上記Webサーバーから呼び出し可能な下記RDBの何れか)

Microsoft Jetデータベースエンジン、Microsoft SQL Server、Oracle

※ 上記環境以外でも例えばサーバーでは mono などの互換環境でも動作できる可能性があります。実環境でテストの上ご検討ください。

用語解説

- **LLL(トリプル・エル)** パーシモンシステムの登録商標で、パーシモンシステム社製開発ツールの中でRAD用製品を指すシリーズ名称。「Libraly Linkage for Lan&wan system」の頭文字が由来。
- **RAD(ラッド)** 「Rapid Application Development」 プロトタイプ(試作品)を繰り返し製作し、評価・改良し完成度を高めるソフトウェアの開発手法。
- **Ajax(エイジャックスまたはアジャックス)** Asynchronous JavaScript + XML Webブラウザに実装されているJavaScriptのHTTP通信機能(XMLHttpRequest)を使って、Webページのリロードを伴わずにサーバとXML形式のデータのやり取りを行いダイナミックHTMLで動的にページの一部を書き換えながら処理を進めていく対話型Webアプリケーションの実装形態。
- **JavaScript(ジャバ スクリプト)** 元はネットスケープコミュニケーションズのブレンダン・アイクによって開発された言語であり、現在はEcmaインターナショナルでECMAScript (ECMA-262) として標準化されている。狭義ではこれのMozilla Foundation(モジラ・ファウンデーション)における実装名。名前が似ているがjavaとは異なるオブジェクト指向のスクリプト言語であり、実行環境が主にWebブラウザに実装され、動的なウェブサイト構築や、リッチインターネットアプリケーションなど高度なユーザインタフェースの開発に用いられる。
- **SQL(エスキューエルまたはシーケル)** リレーショナルデータベース管理システム (RDBMS) において、データの操作や定義を行うためのデータベース言語(問い合わせ言語)である。IBM製品の世界では SQLは、Structured Query Languageと紹介されることもあるが、ISO等の国際標準ではSQLは何かの略号ではないとされている。ただしシーケルと読まれることがあるのは、IBM社が開発したRDBMSの実験実装である System R の操作言語「**SEQUEL (Structured English Query Language)**」こそが現在のSQLの大元であることに由来している。



<http://www.persimmon-system.co.jp/index.html>

COPYRIGHT © 2015-2016 PERSIMMON SYSTEM CORPORATION ALL RIGHTS RESERVED.

[2016年10月31日
版]