

サーバー開発言語

さらに一歩進んで、さまざまな情報をWebサーバーに問い合わせるようなWebサイトを作ろうとしたら、別のプログラミング言語が必要になります。「問い合わせる」というのは、例えば掲示板を作ったり、オンラインショップを作ったりすると、必要な情報をサーバーに問い合わせるような処理をしないといけなくなるんです。こういうときに、Webサーバーでデータの処理をするような言語を使わないといけないんですね。

PHPを使う。

◆HTML

HTML：インターネット上のさまざまな情報をつないでいくためのもの。Web ページをレイアウトするための言語ではない。情報を構造的に記すための言語。

CSS（スタイルシート）：実際の画面の表示（デザイン）を担当。

ハイパーテキスト：クリックすると他のところへジャンプするテキストのこと。リンクのこと。

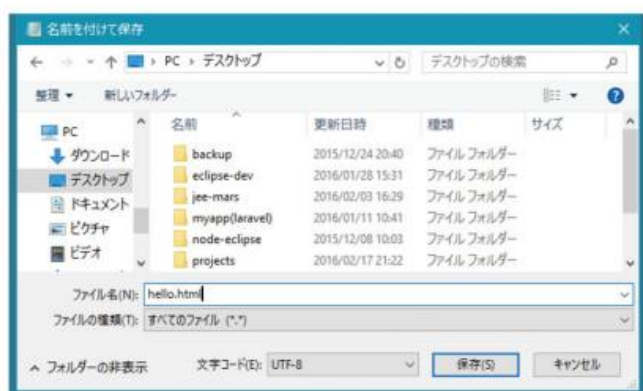


図1-5：保存ダイアログでファイルの種類、文字コード、ファイル名を設定する。

◆スタイルシート



図1-18：styleの値は、スタイル名と値がセットになったものを必要なだけつないで記述する、という形になっている。

ブロック要素	1つの独立した段落のように扱われるもの。その前後の要素から切り離され、少し間隔を開けて表示される。
インライン要素	別の要素の中に埋め込まれるように扱われるもの。その前後から表示場所などが切り離されることなく、つながった状態で表示される。

ブロック要素は「段落」です。<div>はブロック要素ですから、その部分だけが別の段落のように表示されてしまったのです。

テキストの一部分だけスタイル設定したいなら、インライン要素のタグを使わないといけません。これには、というタグが利用されます。

◆JavaScript

Web ブラウザの中に、「JavaScript エンジン」というプログラムが組み込まれている。

この<script>というのが、プログラミング言語によるスクリプトを示すタグです。この開始タグから終了タグまでの間に、JavaScriptのスクリプトを記述します。

ここでは、「document.write」というものが使われていますね。これは、その場にHTMLのテキストを書き出すもので、こんな具合に使います。

Memo

<script>のタグ内に書かれたテキストは、画面には表示されません。スクリプトとしてWebブラウザの内部で実行されます。

```
document.write( 書き出すテキスト );
```

この節のポイント

- ifとswitchを使って処理の分岐ができるようになろう。
- 繰り返し構文のwhileとforの働きの違いをよく理解しよう。
- 配列の考え方と、そのためのforの使い方を覚えよう。

条件演算子

| | |
|---------------|-----------|
| A == B | AとBは等しい |
| A != B | AとBは等しくない |

シンプルな繰り返し「while」

ifやswitchは、「分岐」と呼ばれる構文です。状況によって実行する処理を変更するためのものですね。制御構文は、この分岐の他にもう1つあります。それは「繰り返し」です。

繰り返しは文字通り、**決まった処理を何度も繰り返し実行する**ためのものです。これも、いくつかの構文が用意されています。もっともシンプルなのは、「while」という構文でしょう。こういうものです。

whileの基本形

```
while ( 条件 ){  
    ……繰り返し処理……  
}
```



図 2-12 : whileは、条件をチェックし、trueならば処理を実行し、また元に戻る。falseになったら繰り返しを抜ける。

複雑な繰り返し「for」

繰り返しは、もう1つあります。それは「for」というものです。これは、whileに比べるともっといろいろな要素があって複雑です。そのかわり、覚えるとかなり便利です。

このforは、以下のような形を指定します。

forの基本形

```
for ( 初期化処理 ; 条件 ; 後処理 ){  
    ……繰り返す処理……  
}
```

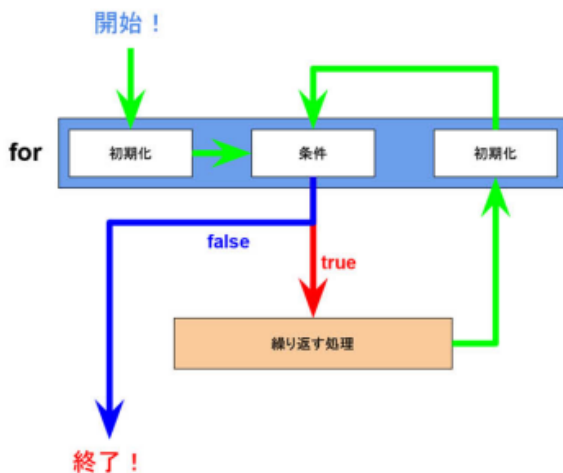


図2-14:forに進むと、まず初期化の処理を実行する。それから条件をチェックし、trueなら繰り返し処理を実行する。実行後、後処理を行ってから再び条件のチェックに戻る。

◆DOM(Document Object Model)

けれど、それではWebブラウザの中で面白いことを実現しようとしてもできません。そこで、描いて完成させたWebページの中身を後から操作できるような方法を考えることにしたのです。そうして誕生したのが「DOM(ドム)」というものです(図3-1)。

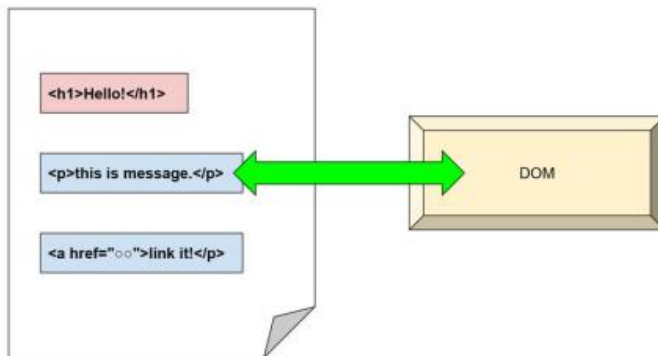


図3-1:DOMは、HTMLに書かれているすべてのタグごとに作成される。このDOMのオブジェクトを操作すれば、組み込んでいるHTMLのオブジェクトを操作できる。

イベントというのは、さまざまな操作や状況などに応じて発生する信号のようなものです。Webブラウザでは、例えば「マウスでどこかをクリックしたとき」「キーを押したとき」「ウインドウをリサイズしたとき」

というように、ユーザーの操作やWebページの状態などに応じたイベントが用意されています。そして、それらの操作が行われると指定のイベントが発生し、そのイベント用に用意された属性の値が実行されるようになっています(図3-6)。

このイベントのおかげで、ユーザの操作に応じてインタラクティブに動くようなWebページが作れるようになっているのです。

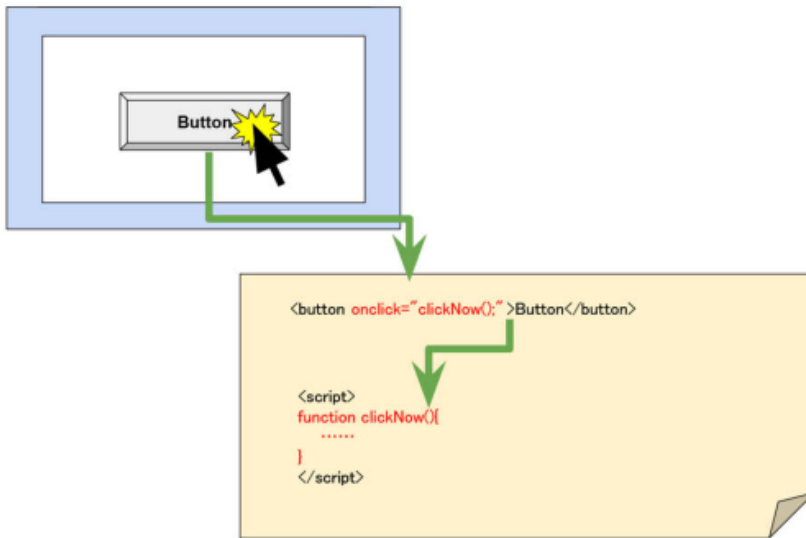


図3-6：ユーザーが操作すると、そのイベント用の属性に書かれている関数が呼び出され実行される。

◆Canvas

この節のポイント

- Canvasとグラフィックコンテキストの基本をしっかり覚えよう。
- fillとstrokeの描画の基本について理解しよう。
- パスを使った図形の描画の仕組みを知ろう。

アニメーション



図4-20：アクセスすると、イメージが左から右へとアニメーションし、停止する。

◆PHP(Hypertext Preprocessor)

この節のポイント

- XAMPPをインストールし、PHPを使えるようにしよう。
- PHPの基本文法をJavaScriptと比べながら覚えていこう。
- PHPのスクリプトを埋め込み利用する書き方に慣れよう。

・変数は必ず\$マークから始まります。

例 \$hensuu

- ・処理の最後尾には必ずセミコロン「;」が必要

言語仕様が簡単

PHPは、非常にシンプルな文法になっています。専用のプログラムファイルなどを用意する必要がなく、JavaScriptなどと同じように、HTMLの中にタグを使ってプログラムを書き込めます。その扱いの手軽さから、プログラミングの経験がない人にも広く利用されています。

XSS 攻撃

JavaScriptのようにWebブラウザの中で動くものと違い、サーバープログラムというのはどんな人間がアクセスしてくるかわかりません。悪意ある人間がアクセスしてくることだってあるのです。

フォームのようにユーザーが入力した情報をそのままWebページ内に表示するようなプログラムの場合、「XSS攻撃」と呼ばれる攻撃を受けることがあります。XSSとは「クロスサイトスクリプティング」の略で、要するにJavaScriptのスク립トを書き込んで送信することでサイトに攻撃を仕掛けるテクニックです。

送信された内容をそのまま画面に表示するようになっていいると、HTMLのタグを書いて送信すれば、そのタグがそのまま出力され、HTMLのタグとして扱われるようになります(図5-19)。<script>タグを使ってJavaScriptのスク립トを書いて送信すれば、それがそのまま画面に書き出され、スク립トが実行されてしまいます。攻撃者が自分のやりたいことを実行するスク립トをフォームから送信することで、サイトにアクセスしてきた人間に攻撃を仕掛けることができるようになってしまうのです。

では、XSS攻撃への対応を行いましょう。これはとても簡単です。PHPには、HTMLで使われている各種の記号類を、すべてエスケープ文字(HTML内で文字として利用できる特殊な書き方)に変換する関数があります。それは「htmlspecialchars」というものです。

Ajax(エイジャックス)

Ajaxは、「非同期」で通信をします。通常、Webブラウザからサーバーにアクセスしたりするのは「同期」通信です。同期というのは、通信が終わるまで待って次に進む、というものです。

これに対して、非同期というのは、通信を開始したら終わるまで待たずにどんどん先へ進む、というものです。通信をまだやってるのにどんどん処理を進めていきますから、通信が終わったときにどうするかは別に考えないといけません。



図5-23: ボタンを押すと、Ajax通信でサーバーから現在の日時のテキストを受け取り、画面に表示する。

これで完成です。できあがったら、Webブラウザからhello.htmlにアクセスしてみましょう。そしてボタンを押すと、サーバーから現在の日時を受け取り画面に表示します。

このやり方だと、フォーム送信のようにリロードされることなく、そのままの表示でサーバーから受け取った情報を使い、表示の一部分だけを更新することができます。

ID 入力→商品情報表示

サーバーから単純にデータを受け取るのはこれでできました。でも、これではあまり実用にはなりません。本格的に使えるようになるためには、サーバーに必要な情報を送り、それに基づいて必要なデータを送り返すような処理でしょう。例えばID番号を送信すると、その番号の商品情報を送り返してくる、みたいな処理ですね。これは、先ほどのAjax通信の基本に、「必要な情報をサーバーに送る」という処理を追加する必要があります。では、修正していきましょう。

オブジェクトの設計がキモ!

実際に動くゲームを作ってみると、本当に難しいのは、1つ1つの処理を組み立てることではなくて、「オブジェクトの設計」であることがわかってくるでしょう。プログラム全体を整理し、どういうオブジェクトを用意して、それにどんな機能を持たせるか。例えば、キャラクタやアイテムなどはイメージしやすいですが、背景や効果の表示などは独立したオブジェクトにしたほうがいいか、それとも他のオブジェクトに「機能」として組み込んだほうがいいか。

そうしたことは、「こうすればOK」という正解があるわけではありません。それぞれが自分なりに「こうオブジェクトを整理するのが自分にはやりやすい」というものを探っていくといけな

2D 疑似アドベンチャーゲーム

今回、サンプルで作ったのは、キャラクタを操作してアイテムを集めて回るゲーム「Adventurer」です(図6-8)。これは、ロールプレイング型のアドベンチャーの最も基本となる「アイテム集め」に特化したミニゲームです。

アクセスすると、Web ページにマップとキャラクタが表示されます。マップの部分をクリックして選択すると、上下左右の矢印キーでキャラクタを動かせるようになります。



図6-8：キャラクタを動かしてアイテムを集める。

JSON

では、このJSONというのはいったいなんなのでしょうか？

——これは、JavaScriptで使われるオブジェクトをテキストの形で記述するためのフォーマットなのです。

JavaScriptでは、オブジェクトをテキストとしてやりとりできるように、このJSONというフォーマットを使います。これを利用することで、オブジェクトをファイルに保存したり、通信で送ったりできるようになるのです。

ここではPHPでjson_encodeを使って、JSON型式で配列を書き出します。Webブラウザ側では、JavaScriptのAjaxを使ってそれを受け取ります。受け取ると、自動的にそのデータはオブジェクトに変換され、そのままJavaScriptのオブジェクトとして利用できるようになるのです。実に便利ですね！

◆スマホアプリ

この節のポイント

- Monacaの基本的な操作を理解しよう。
- スマートフォンでデバッグアプリを使えるようにしよう。
- Webの移植の手順を覚えよう。

ネイティブ・アプリ

スマートフォンのメーカーが推奨する、もっともスタンダードな開発方式で作られるアプリです。AndroidならばJava、iPhoneならばObjective-C(オブジェクトブシー)またはSwift(スウィフト)というプログラミング言語を使って開発します。それぞれ、JavaScriptなどよりはかなり本格的なプログラミング言語であり、これらの言語を使ってスマートフォンに用意されているAPI(スマートフォンの機能を利用するためのライブラリ)を呼び出してプログラムを作成していきます。

「スマホアプリの開発は難しい」というのは、このネイティブ・アプリの開発を言っているのだ、と考えてよいでしょう。

Monaca(モナカ)

Webブラウザさえあればどこでも使える

最大の特徴は、これでしょう。Monacaは、パソコンにソフトをインストールして使うわけではありません。Webブラウザからアクセスして使います。ということは、インターネットにアクセスできる環境ならば、どこでも使うことができるわけです。極端な話、パソコンでなくともかまわないのです。スマートフォンはさすがに画面が小さすぎて難しいでしょうが、iPadなどのタブレットからアクセスして利用するならば十分使えるでしょう。

これ1冊でゼロから学べる

Webプログラミング 超入門

—HTML, CSS, JavaScript, PHPをまるごとマスター

2016年7月27日 初版第1刷発行・電子版ver1.00