

PHPアドバンス 最終課題 ショッピングサイト構築

仕様書を基にしたコーディング

システム開発をする際には、開発するシステムの詳細な仕組みや内容が記述された「仕様書」という書類に基づいて、プログラマはコーディングをしていきます。

特に、大規模なシステムになると、現場ではチームで分担しコーディングを行いますので、それぞれが好きな勝手にコーディングしていくわけにはいきません。全員が開発するシステムの仕組みを共有し、共通のルールで作成するために仕様書が不可欠になります。

プログラマはこの仕様書に書かれている内容を理解し、システムの仕組みを把握し、それに従い正確に仕様を満たすプログラムを書いていくことが求められます。

今回は、この仕様書を解読してのコーディングに挑戦しましょう。

機能概要

商品検索・商品選択

このリンクはダミーです

The screenshot shows the Latte STATION website. The header includes the Latte STATION logo and a navigation bar with links: Home, ショップガイド, よくある質問, 会社案内, お問い合わせ. Below the header is a banner with the text '欲しいものが必ず見つかる' and the Latte STATION logo. A search bar is located below the banner, with a dropdown menu for '分類' (Category) set to '書籍' (Books) and a '検索' (Search) button. Below the search bar is a section titled '現在選択されている商品' (Currently selected items), which shows '選択商品 0 個' (Selected items 0 items). Below this is a section titled '選択カテゴリ商品' (Selected category items), which says '商品を検索してください。' (Please search for products). At the bottom of the page is a footer with the text '©Copyright 2012 ThinkethBank, Inc. All rights reserved.'

この画面では商品の検索をさせます。

分類には、書籍、CD、GAMEの3種類があります。キーワードを入れずに「検索」ボタンを押すと、選択したカテゴリーの商品が全て表示されます。下図は分類のプルダウンメニューから、「書籍」を選択したときのイメージです。

選択カテゴリー商品				
商品	商品名	メーカーなど	価格	
	ホリー・ポッター	ケン・ジャイシー	580円	
	AGE of vase	ダンレッド	410円	
	龍玉60年の歴史	島山明	600円	
	メサニョキ	さくらむむこ	1050円	

キーワードの欄に、商品の名称もしくはメーカー名などの一部を入れて検索ボタンを押すことで、該当する商品のみを表示させます。

カテゴリーを選択して検索して下さい。

分類

書籍 ▼

キーワード

ホリー

検索

選択カテゴリー商品				
商品	商品名	メーカーなど	価格	
	ホリー・ポッター	ケン・ジャイシー	580円	🛒 カートに追加

該当する商品がなかった場合は下図のようにメッセージを表示させます。

カテゴリーを選択して検索して下さい。

分類

書籍 ▼

キーワード

AAA

検索

選択カテゴリー商品	
該当する商品はありません。	

検索された商品の「カートに追加」を押すたびに現在選択されている商品数を増やします。

現在選択されている商品

選択商品 2 個

[🛒 カートの中身を見る](#)

「カートの中身を見る」を押すと、カート内の商品を表示させます。

選択商品表示・取り消し

同じ商品が選択されている場合、その分だけ表示させます。

カート内の商品

現在買い物カゴには以下の商品が入っています。

商品名	メーカーなど	価格	
ホリー・ポッター	ケン・ジャイシー	580円	取り消し
ホリー・ポッター	ケン・ジャイシー	580円	取り消し

 上記内容で注文する

 検索ページに戻る

「取り消し」が押された場合、選択した商品をカートから削除します。

カート内の商品

現在買い物カゴには以下の商品が入っています。

商品名	メーカーなど	価格	
ホリー・ポッター	ケン・ジャイシー	580円	取り消し


 上記内容で注文する

 検索ページに戻る

カートの中に一つも商品がない場合、下図のようにメッセージを表示させます。

カート内の商品

カートに商品はありません。

 検索ページに戻る

注文者情報入力

カート内の商品の価格の合計を表示させ、注文者情報の入力フォームを表示させます。

合計金額	
現在買い物カゴの合計金額は 580円（税込み） です。	
注文者情報	
全項目が必須入力です。	
お名前	<input type="text"/> (例) 山田 太郎
フリガナ	<input type="text"/> (例) ヤマダ タロウ
郵便番号	<input type="text"/> (例) 153-0052 ※郵便番号検索は こちら
住所	<input type="text"/> (例) 東京都目黒区目黒2-2-2 ○○ビル30F 総務部 ※ マンション・アパート名は必ず入力してください。 ※ 住所が勤務先の場合は「部署名」なども入力してください。
電話番号	<input type="text"/> (例) 03-0000-0000
メールアドレス	<input type="text"/> (例) hoge@kenschool.jp
上記の内容で問題なければ、注文を確定します。	
<input type="button" value="注文する"/>	
ショッピングカートに戻る 検索ページに戻る	

未入力がある状態で「注文する」が押された場合、下図のようにメッセージを表示させます。

注文者情報	
未入力項目がございます。ご確認ください。	
お名前	<input type="text"/> (例) 山田 太郎

注文完了確認

すべて入力された状態で「注文する」が押された場合、注文完了画面に遷移し、下図のようにメッセージを表示させます。

注文完了

ご注文ありがとうございます。

24時間以内に当サイト管理者よりメールがまいりますので、よろしくお願い致します。

在庫状況により商品の到着が遅れる場合がございますので、あらかじめご了承ください。

■ [検索ページに戻る](#)

ユーザーの情報をデータベースの customer テーブルに追加します。

+ オプション								
← T →		customer_id	name	furigana	zipcode	address	phone_number	email
<input type="checkbox"/>	編集 コピー 削除	100	山田太郎	ヤマダタロウ	155-0031	東京都世田谷区北沢1-1-1KENコーポ203	03-0000-0000	hoge@kenschool.jp

注文情報をデータベースの order_history テーブルに追加します。

+ オプション									
← T →				order_history_id		customer_id		item_id	
<input type="checkbox"/>		編集		コピー		削除	200	100	1

データベース

データベース情報

サーバー名	localhost
データベース名	latte
ユーザー名	root
パスワード	(なし)
文字コード	UTF-8

テーブル一覧

category	商品の分類情報
customer	注文者情報
item	商品情報
order_history	注文内容

categoryテーブル情報

列名	種別	その他	内容
category	int(8)	AUTO_INCREMENT	カテゴリーID
name	varchar(128)		カテゴリー名

customerテーブル情報

列名	種別	その他	内容
customer_id	int(8)	AUTO_INCREMENT	顧客ID
name	varchar(128)		名前
furigana	varchar(128)		フリガナ
zipcode	char(8)		郵便番号
address	varchar(256)		住所
phone_number	char(20)		電話番号
email	varchar(128)		メールアドレス

itemテーブル情報

列名	種別	その他	内容
item_id	int(8)	AUTO_INCREMENT	商品ID
name	varchar(128)		名前
maker	varchar(128)		メーカー
price	int(8)		価格
image	varchar(128)		画像
category_id	int(8)		カテゴリーID

order_historyテーブル情報

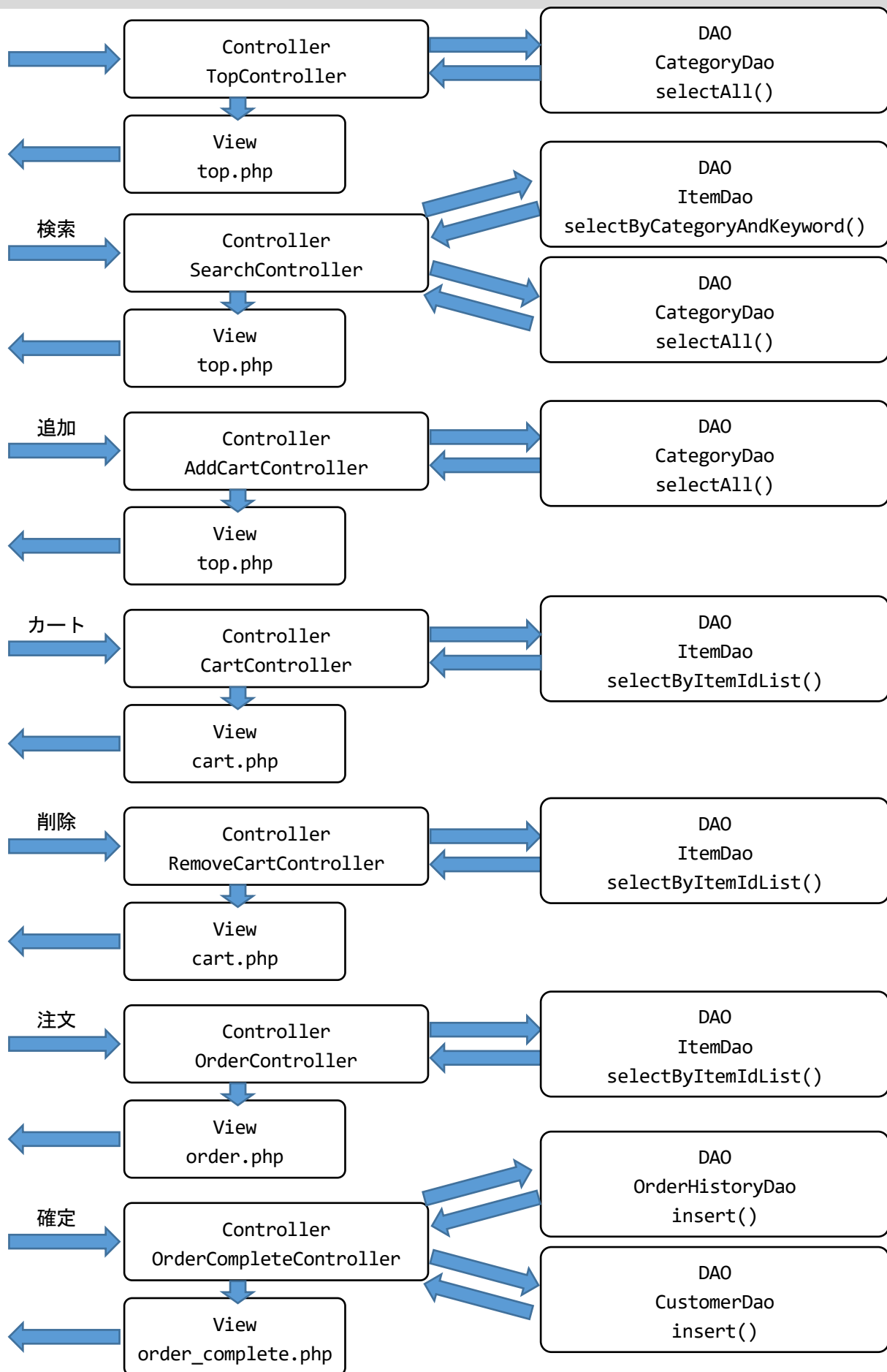
列名	種別	その他	内容
order_history_id	int(8)	AUTO_INCREMENT	注文履歴ID
customer_id	int(8)		顧客ID
item_id	int(8)		商品ID

フォルダ構成の確認

ファイル一覧

```
latte
├── index.php ..... 配布（完成）
├── app
│   ├── controller
│   │   ├── AddCartController.php ..... 新規作成
│   │   ├── CartController.php ..... 新規作成
│   │   ├── Controller.php ..... 配布（完成）
│   │   ├── ErrorController.php ..... 新規作成
│   │   ├── OrderCompleteController.php ..... 新規作成
│   │   ├── OrderController.php ..... 新規作成
│   │   ├── RemoveCartController.php ..... 新規作成
│   │   ├── SearchController.php ..... 新規作成
│   │   └── TopController.php ..... 新規作成
│   ├── dao
│   │   ├── CategoryDao.php ..... 新規作成
│   │   ├── CustomerDao.php ..... 新規作成
│   │   ├── Dao.php ..... 配布（完成）
│   │   ├── ItemDao.php ..... 新規作成
│   │   └── OrderHistoryDao.php ..... 新規作成
│   ├── model
│   │   └── Model.php ..... 配布（完成）
│   ├── utilities
│   │   └── functions.php ..... 配布（完成）
│   └── view
│       ├── cart.php ..... 新規作成
│       ├── error.php ..... 配布（完成）
│       ├── footer.php ..... 配布（完成）
│       ├── header.php ..... 配布（完成）
│       ├── order.php ..... 新規作成
│       ├── order_complete.php ..... 配布（完成）
│       └── top.php ..... 新規作成
├── css
│   └── style.css ..... 配布（完成）
└── img
    └── 【省略】 ..... 配布（完成）
```


処理遷移図



ショッピングサイト Latte 仕様書

ファイル情報																			
ファイル名	index.php																		
ディレクトリ	/latte/																		
処理内容																			
<ol style="list-style-type: none">1. 自作関数を利用するため"app/utilities/functions.php"を読み込む2. セッションを開始する3. Modelオブジェクト生成のため"app/model/Model.php"を読み込む4. Modelオブジェクトの生成し、変数\$modelに代入する5. Modelオブジェクトの生成し、変数\$paramsに代入する6. リクエストパラメータを変数\$paramsに代入する7. スーパーグローバル変数\$_SERVER["REQUEST_METHOD"]が、GETメソッドの場合は①の処理を行い、POSTメソッドの場合は②の処理を行う<ol style="list-style-type: none">① 変数\$paramsにスーパーグローバル変数\$_GETのプロパティをセットする② 変数\$paramsにスーパーグローバル変数\$_POSTのプロパティをセットする8. コントローラクラスをまとめた連想配列を生成し、変数\$controllersに代入する（連想配列の内容は下表参照） <table><tr><th>キー</th><th>値</th></tr><tr><td>top</td><td>TopController</td></tr><tr><td>search</td><td>SearchController</td></tr><tr><td>cart</td><td>CartController</td></tr><tr><td>addCart</td><td>AddCartController</td></tr><tr><td>removeCart</td><td>RemoveCartController</td></tr><tr><td>order</td><td>OrderController</td></tr><tr><td>orderComplete</td><td>OrderCompleteController</td></tr><tr><td>error</td><td>ErrorController</td></tr></table> <ol style="list-style-type: none">9. 変数\$paramsのactionの値が設定されていない場合は①の処理を行い 変数\$paramsのactionの値が変数\$controllersに存在しない場合は②の処理を行う<ol style="list-style-type: none">① 変数\$paramsのactionに文字列"top"をセットする② 変数\$paramsのactionに文字列"error"をセットする 変数\$modelのmessageに「不正なアクセスです」をセットする10. 変数\$controllers（連想配列）から、変数\$paramsのactionの値に対応するクラス名を取得し、それを引数としてfunction.php内のcreateController()関数を呼び出すことで、生成済みのコントローラを取得し、変数\$controllerに格納する11. 次の処理を例外対応する<ol style="list-style-type: none">① 変数\$paramsと変数\$modelを引数として変数\$controllerのaction()メソッドを呼び出し、変数\$viewに格納する12. 例外(Exception)が発生した場合は次の処理を行う<ol style="list-style-type: none">① 変数\$modelのmessageに例外オブジェクト内のメッセージを格納する② 変数\$viewにエラーページ用のアドレス"app/view/error.php"を格納する13. 変数\$viewと変数\$modelを引数にして、変数\$controllerのview()メソッドを呼び出す		キー	値	top	TopController	search	SearchController	cart	CartController	addCart	AddCartController	removeCart	RemoveCartController	order	OrderController	orderComplete	OrderCompleteController	error	ErrorController
キー	値																		
top	TopController																		
search	SearchController																		
cart	CartController																		
addCart	AddCartController																		
removeCart	RemoveCartController																		
order	OrderController																		
orderComplete	OrderCompleteController																		
error	ErrorController																		

ファイル情報	
ファイル名	Controller.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	Controller
継承	なし
修飾子	Abstract
概要	各Controllerの親クラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public abstract
処理概要	リクエストを処理するメソッド。リクエストパラメータとModelオブジェクトを受け取り、Viewのパスを返す抽象メソッドの定義
メソッド名	view
引数	\$viewPath : Viewのパス(PHPファイルのパス) \$model : Modelオブジェクト
修飾子	public
処理概要	Viewを表示する

メソッド詳細			
クラス名	Controller	メソッド名	View
処理内容	<p>1. 第一引数で受け取ったパスに存在するPHPファイルをrequire_once()関数で取り込む</p>		

ファイル情報	
ファイル名	AddCartController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	AddCartController
継承	Controller
修飾子	なし
概要	ショッピングカートへ商品を追加する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	ショッピングカートへ商品を追加する 表示用にカテゴリー情報を取得し、トップページのパスを返す

メソッド詳細			
クラス名	AddCartController	メソッド名	Action
処理内容	<ol style="list-style-type: none"> 変数\$paramsのitemIdが空の場合、次の処理を行う <ol style="list-style-type: none"> 「カートに商品を追加できませんでした。」をエラーメッセージとして持つ例外オブジェクトを生成し、例外を発生させる スーパーグローバル変数\$_SESSION["cart"]に値が設定されていない場合、次の処理を行う <ol style="list-style-type: none"> スーパーグローバル変数\$_SESSION["cart"]に空の配列を格納する スーパーグローバル変数\$_SESSION["cart"]に変数\$paramsのitemIdを新しい要素として追加する 文字列"CategoryDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$categoryDaoに格納する 変数\$categoryDaoのselectAll()メソッドを実行して得られたList情報を変数\$modelのcategoryListに格納する トップページ"app/view/top.php"のパスを返す 		

ファイル情報	
ファイル名	CartController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	CartController
継承	Controller
修飾子	なし
概要	カートページへ遷移する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	カートにある商品の情報を取得し、カートページのパスを返す

メソッド詳細			
クラス名	CartController	メソッド名	Action
処理内容	<p>1. スーパーグローバル変数\$_SESSION["cart"]に値が設定されている場合、次の処理を行う</p> <ul style="list-style-type: none"> ① 文字列"ItemDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$itemDaoに格納する ② スーパーグローバル変数\$_SESSION["cart"]を引数にして、変数\$itemDaoのselectByItemIdList()メソッドを呼び出してカートの情報を取得し、変数\$modelのcartItemsに格納する <p>2. カートページ"app/view/cart.php"のパスを返す</p>		

ファイル情報	
ファイル名	ErrorController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	ErrorController
継承	Controller
修飾子	なし
概要	エラーページへ遷移する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	エラーページのパスを返す

メソッド詳細			
クラス名	CartController	メソッド名	Action
処理内容	1. エラーページ"app/view/error.php"のパスを返す		

ファイル情報	
ファイル名	OrderCompleteController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	OrderCompleteController
継承	Controller
修飾子	なし
概要	注文完了ページへ遷移する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	未入力の項目がある場合、注文ページを返す 未入力の項目がない場合、顧客情報を登録し、その顧客情報と商品IDを元に購入商品情報を登録し、注文完了ページのパスを返す
メソッド名	inputCheck
引数	\$params : リクエストパラメータ
修飾子	public
処理概要	リクエストパラメータの入力内容をチェックし、未入力なしの場合はtrueを返し、未入力ありの場合はfalseを返す

メソッド詳細			
クラス名	OrderCompleteController	メソッド名	Action
処理内容	<ol style="list-style-type: none"> 変数\$paramsを引数として、当該クラスのinputCheck()メソッドを利用した結果、未入力項目がある場合は次の処理を行う <ol style="list-style-type: none"> 変数\$paramsの各プロパティを変数\$modelのプロパティにセットする 変数\$modelのmessageにエラーメッセージ「未入力項目があります。ご確認ください。」をセットする 注文ページ"app/view/order.php"のパスを返す 文字列"CustomerDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$customerDaoに格納する 変数\$paramsの名前、フリガナ、郵便番号、住所、電話番号、Eメールの6つのデータを配列とし、変数\$valuesに格納する 変数\$valuesを引数として、変数\$customerDaoのinsert()メソッドを実行して顧客IDを取得し、変数\$newCustomerIdに格納する 文字列"OrderHistoryDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$orderHistoryDaoに格納する 変数\$multiValuesに空の配列を格納する スーパーグローバル変数\$_SESSION["cart"]内にある要素を一つずつ変数\$itemIdに取得し、要素の分だけ次の処理を繰り返す <ol style="list-style-type: none"> 変数\$newCustomerIdと変数\$itemIdの2つの値を配列とし、変数\$multiValuesへ新しい要素として追加する 変数\$multiValuesを引数として、変数\$orderHistoryDaoのinsert()メソッドを実行する セッションを破棄する 注文完了ページ"app/view/order_complete.php"のパスを返す 		

メソッド詳細			
クラス名	OrderCompleteController	メソッド名	inputCheck
処理内容	<ol style="list-style-type: none"> 引数で受け取ったリクエストパラメータの名前が空である場合はfalseを返す 引数で受け取ったリクエストパラメータのフリガナが空である場合はfalseを返す 引数で受け取ったリクエストパラメータの郵便番号が空である場合はfalseを返す 引数で受け取ったリクエストパラメータの住所が空である場合はfalseを返す 引数で受け取ったリクエストパラメータの電話番号が空である場合はfalseを返す 引数で受け取ったリクエストパラメータのメールアドレスが空である場合はfalseを返す 上記のすべての条件に当てはまらなかった場合、trueを返す 		

ファイル情報	
ファイル名	OrderController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	OrderController
継承	Controller
修飾子	なし
概要	注文ページへ遷移する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	カートの情報がない場合は例外を発生させる カートにある商品の情報を取得し、注文ページを返す

メソッド詳細			
クラス名	OrderController	メソッド名	action
処理内容	<ol style="list-style-type: none"> スーパーグローバル変数\$_SESSION["cart"]に値が設定されていない場合、次の処理を行う <ol style="list-style-type: none"> 「カートに商品がないため、注文ページへ遷移できませんでした。」をエラーメッセージとして持つ例外オブジェクトを生成し、例外を発生させる スーパーグローバル変数\$_SESSION["cart"]に値が設定されている場合、次の処理を行う <ol style="list-style-type: none"> 文字列"ItemDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$itemDaoに格納する スーパーグローバル変数\$_SESSION["cart"]を引数にして、変数\$itemDaoのselectByItemIdList()メソッドを呼び出してカートの情報を取得し、変数\$cartに格納する 変数\$totalに0を代入する 変数\$cart内にある要素を一つずつ変数\$itemに取得し、要素の分だけ次の処理を繰り返す <ol style="list-style-type: none"> 変数\$item内の"price"要素を\$totalへ加算する スーパーグローバル変数\$_SESSION["sumPrice"]へ\$totalを代入する 変数\$modelのmessageに「全項目が必須入力です。」をセットする 注文ページ"app/view/order.php"のパスを返す 		

ファイル情報	
ファイル名	RemoveCartController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	RemoveCartController
継承	Controller
修飾子	なし
概要	ショッピングサイトから商品を削除する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	カート情報が空の場合は例外を発生させる 指定された商品をカートから削除し、カートページのパスを返す

メソッド詳細			
クラス名	RemoveCartController	メソッド名	action
処理内容	<ol style="list-style-type: none"> スーパーグローバル変数\$_SESSION["cart"]に値が設定されていない場合、次の処理を行う <ol style="list-style-type: none"> 「カートから商品を削除できませんでした。」をエラーメッセージとして持つ例外オブジェクトを生成し、例外を発生させる スーパーグローバル変数\$_SESSION["cart"]に格納されている情報と、変数\$paramsのindexと、数値の1を引数にして、array_splice()関数を呼び出すことで、該当する要素を削除する 文字列"ItemDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$itemDaoに格納する スーパーグローバル変数\$_SESSION["cart"]を引数にして、変数\$itemDaoのselectByItemIdList()メソッドを呼び出してカートの情報を取得し、変数\$modelのcartItemsに格納する カートページ"app/view/cart.php"のパスを返す 		

ファイル情報	
ファイル名	SearchController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	SearchController
継承	Controller
修飾子	なし
概要	商品検索をする処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	検索内容であるcategoryとkeywordおよび検索結果をセッションに格納する 表示用にカテゴリー情報を取得し、トップページのパスを返す

メソッド詳細			
クラス名	SearchController	メソッド名	action
処理内容	<ol style="list-style-type: none"> 1. 文字列"ItemDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$itemDaoに格納する 2. 変数\$ paramsのcategoryと変数\$ paramsのkeywordを引数にして変数\$itemDaoのselectByCategoryAndKeyword()メソッドを呼び出すことで検索結果を取得し、スーパーグローバル変数\$_SESSION["searchResults"]へ登録する 3. 変数\$ paramsのcategoryをスーパーグローバル変数\$_SESSION["category"]へ登録する 4. 変数\$ paramsのkeywordをスーパーグローバル変数\$_SESSION["keyword"]へ登録する 5. 文字列"CategoryDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$categoryDaoに格納する 6. 変数\$categoryDaoのselectAll()メソッドを実行して得られたList情報を変数\$modelのcategoryListに格納する 7. トップページ"app/view/top.php"のパスを返す 		

ファイル情報	
ファイル名	TopController.php
ディレクトリ	/latte/app/controller/
クラス情報	
クラス名	TopController
継承	Controller
修飾子	なし
概要	トップページへ遷移する処理を管理するクラス
メソッド	
メソッド名	action
引数	\$params : リクエストパラメータ \$model : Modelオブジェクト
修飾子	public
処理概要	セッション内のcategoryおよびkeywordについて、情報がセットされていなければそれぞれ初期化する 表示用にカテゴリー情報を取得し、トップページのパスを返す

メソッド詳細			
クラス名	TopController	メソッド名	action
処理内容	<ol style="list-style-type: none"> スーパーグローバル変数\$_SESSION["category"]に値が設定されていない場合、次の処理を行う <ol style="list-style-type: none"> スーパーグローバル変数\$_SESSION["category"]にnullをセットする スーパーグローバル変数\$_SESSION["keyword"]に値が設定されていない場合、次の処理を行う <ol style="list-style-type: none"> スーパーグローバル変数\$_SESSION["keyword"]に空文字をセットする 文字列"CategoryDao"を引数としてfunction.php内のcreateDao()関数を呼び出すことでDAOオブジェクトを取得し、変数\$categoryDaoに格納する 変数\$categoryDaoのselectAll()メソッドを実行することで得られたList情報を変数\$modelのcategoryListに格納する トップページ"app/view/top.php"のパスを返す 		

ファイル情報	
ファイル名	Dao.php
ディレクトリ	/latte/app/dao/
クラス情報	
クラス名	Dao
継承	なし
修飾子	なし
概要	各Daoクラスの親クラス
プロパティ	
修飾子	private
用途	PDOオブジェクトを格納するための変数
メソッド	
メソッド名	open
引数	なし
修飾子	public
処理概要	PDOオブジェクトを取得し、フィールド変数へセットする
メソッド名	close
引数	なし
修飾子	public
処理概要	データベースとの通信を切断する
メソッド名	prepare
引数	\$sql : SQL文
修飾子	public
処理概要	PDOStatementオブジェクトを返す
メソッド名	lastInsertId
引数	なし
修飾子	public
処理概要	最後に挿入された行IDを返す
メソッド名	beginTransaction
引数	なし
修飾子	public
処理概要	トランザクションを開始する
メソッド名	commit
引数	なし
修飾子	public
処理概要	トランザクションをコミットする
メソッド名	rollback
引数	なし
修飾子	public
処理概要	トランザクションをロールバックする

メソッド詳細							
クラス名	Dao						
メソッド名	open						
処理内容	<p>1. データベースへ接続するために必要な情報を格納する</p> <table border="1"> <tr> <td>変数\$dsn</td><td>"mysql:dbname=latte;host=localhost;charset=utf8"</td></tr> <tr> <td>変数\$user</td><td>"root"</td></tr> <tr> <td>変数\$password</td><td>""</td></tr> </table> <p>2. 変数\$dsn,変数\$user,変数\$passwordを引数としてPDOオブジェクトを生成し、プロパティ\$dbへ格納する</p> <p>3. 「PDO::ATTR_ERRMODE」と「PDO::ERRMODE_EXCEPTION」を引数に指定し、プロパティ\$dbのsetAttribute()メソッドを呼び出すことで、PDOオブジェクトの設定を行う</p> <p>※ データベースへ接続するためのユーザー名およびパスワードは環境によって異なりますので、接続できない場合はご確認ください。</p>	変数\$dsn	"mysql:dbname=latte;host=localhost;charset=utf8"	変数\$user	"root"	変数\$password	""
変数\$dsn	"mysql:dbname=latte;host=localhost;charset=utf8"						
変数\$user	"root"						
変数\$password	""						

メソッド詳細	
クラス名	Dao
メソッド名	close
処理内容	<p>1. プロパティ\$dbにnullをセットすることでデータベースとの通信を切断する</p>

メソッド詳細	
クラス名	Dao
メソッド名	prepare
処理内容	<p>1. 変数\$sqlを引数としてフィールド変数\$dbのprepare()メソッドを呼び出すことで、PDOStatementオブジェクトを取得し、返す</p>

メソッド詳細	
クラス名	Dao
メソッド名	lastInsertId
処理内容	<p>1. フィールド変数\$dbのlastInsertId()メソッドを呼び出すことで、最後に挿入された行IDを取得し、返す</p>

メソッド詳細	
クラス名	Dao
メソッド名	beginTransaction
処理内容	<ol style="list-style-type: none"> フィールド変数\$dbのbeginTransaction()メソッドを呼び出すことで、トランザクションを開始し、その結果を返す

メソッド詳細	
クラス名	Dao
メソッド名	commit
処理内容	<ol style="list-style-type: none"> フィールド変数\$dbのcommit()メソッドを呼び出すことで、トランザクションをコミットし、その結果を返す

メソッド詳細	
クラス名	Dao
メソッド名	rollback
処理内容	<ol style="list-style-type: none"> フィールド変数\$dbのrollback()メソッドを呼び出すことで、トランザクションをロールバックし、その結果を返す

ファイル情報	
ファイル名	CategoryDao.php
ディレクトリ	/latte/app/dao/
クラス情報	
クラス名	CategoryDao
継承	Dao
修飾子	なし
概要	category表のデータを扱うメソッドを管理するクラス
メソッド	
メソッド名	selectAll
引数	なし
修飾子	public
処理概要	category表のすべてのデータを取得する

メソッド詳細	
クラス名	CategoryDao
メソッド名	selectAll
処理内容	<ol style="list-style-type: none"> 1. \$rowsに空の配列を格納する 2. 次の処理を例外対応する <ol style="list-style-type: none"> ① open()メソッドを実行することで、データベースへ接続する ② 下記の文字列（SQL）を変数\$sqlへ格納する "SELECT * FROM category ORDER BY category_id" ③ 変数\$sqlを引数としてprepare()メソッドを呼び出すことでPDOStatementオブジェクトを取得し、変数\$statementへ格納する ④ 変数\$statementのexecute()メソッドを呼び出す ⑤ 「PDO::FETCH_ASSOC」を引数として変数\$statementのfetchAll()メソッドを呼び出すことで検索結果を取得し、変数\$rowsへ格納する 3. 例外(PDOException)が発生した場合は次の処理を行う <ol style="list-style-type: none"> ① 下記のようなエラーメッセージを持つ例外を発生させる 「データベースエラー：ファイル名(行番号)」 4. 例外発生の有無に関わらず、次の処理を行う <ol style="list-style-type: none"> ① 変数\$statementへnullを代入する ② データベースから切断する 5. 変数\$rowsを返す

ファイル情報	
ファイル名	CustomerDao.php
ディレクトリ	/latte/app/dao/
クラス情報	
クラス名	CustomerDao
継承	Dao
修飾子	なし
概要	customer表のデータを扱うメソッドを管理するクラス
メソッド	
メソッド名	insert
引数	\$values
修飾子	public
処理概要	顧客情報を追加し、最後に挿入された行IDを返す

メソッド詳細	
クラス名	CustomerDao
メソッド名	insert
処理内容	<ol style="list-style-type: none"> 変数\$insertIdへnullを代入する 次の処理を例外対応する <ol style="list-style-type: none"> open()メソッドを実行することで、データベースへ接続する 下記の文字列（SQL）を変数\$sqlへ格納する "INSERT INTO customer(name,furigana,zipcode,address,phone_number,email) VALUES (?,?,?,?,?,?)" 変数\$sqlを引数としてprepare()メソッドを呼び出すことでPDOStatementオブジェクトを取得し、変数\$statementへ格納する 変数\$statementのexecute()メソッドを呼び出す 自オブジェクトのlastInsertId()メソッドを呼び出すことで最後に挿入された行IDを取得し、変数\$insertIdへ格納する 例外(PDOException)が発生した場合は次の処理を行う <ol style="list-style-type: none"> 下記のようなエラーメッセージを持つ例外を発生させる 「データベースエラー：ファイル名(行番号)」 例外発生の有無に関わらず、次の処理を行う <ol style="list-style-type: none"> 変数\$statementへnullを代入する データベースから切断する 変数\$insertIdを返す

ファイル情報	
ファイル名	ItemDao.php
ディレクトリ	/latte/app/dao/
クラス情報	
クラス名	ItemDao
継承	Dao
修飾子	なし
概要	item表のデータを扱うメソッドを管理するクラス
メソッド	
メソッド名	selectByCategoryAndKeyword
引数	\$category : カテゴリーID \$keyword : キーワード
修飾子	public
処理概要	item表から、指定のカテゴリーおよびキーワードに該当するデータを取得する
メソッド名	selectByItemIdList
引数	\$itemIdList
修飾子	public
処理概要	item表から、指定のitem_idに該当するデータを取得する

メソッド詳細	
クラス名	ItemDao
メソッド名	selectByCategoryAndKeyword
処理内容	<ol style="list-style-type: none"> 1. 変数\$rowsに空の配列を格納する 2. 次の処理を例外対応する <ol style="list-style-type: none"> ① open()メソッドを実行することで、データベースへ接続する ② 下記の文字列（SQL）を変数\$sqlへ格納する "SELECT * FROM item" ③ 変数\$keywordが空だった場合、次の処理を行う <ol style="list-style-type: none"> a. 変数\$sqlへ文字列" WHERE category_id=? ORDER BY item_id"を追加する b. 変数\$categoryを配列として変数\$paramsへ格納する ④ 引数で受け取ったキーワードが空ではなかった場合、次の処理を行う <ol style="list-style-type: none"> a. 変数\$sqlへ文字列" WHERE category_id=? AND (name LIKE ? OR maker LIKE ?) ORDER BY item_id "を追加する b. 変数\$keywordへ下記内容を格納する "%" . \$keyword . "%" c. 変数\$category, 変数\$keyword, 変数\$keywordの3つの要素を持つ配列を生成し、変数\$paramsへ格納する ⑤ 変数\$sqlを引数としてprepare()メソッドを呼び出すことでPDOStatementオブジェクトを取得し、変数\$statementへ格納する ⑥ 変数\$paramsを引数として、変数\$statementのexecute()メソッドを呼び出す ⑦ 「PDO::FETCH_ASSOC」を引数として変数\$statementのfetchAll()メソッドを呼び出すことで検索結果を取得し、変数\$rowsへ格納する 3. 例外(PDOException)が発生した場合は次の処理を行う <ol style="list-style-type: none"> ① 下記のようなエラーメッセージを持つ例外を発生させる 「データベースエラー：ファイル名(行番号)」 4. 例外発生の有無に関わらず、次の処理を行う <ol style="list-style-type: none"> ① 変数\$statementへnullを代入する ② データベースから切断する 5. 変数\$rowsを返す

メソッド詳細	
クラス名	ItemDao
メソッド名	selectByItemIdList
処理内容	<ol style="list-style-type: none"> 1. 変数\$rowsに空の配列を格納する 2. 次の処理を例外対応する <ol style="list-style-type: none"> ① open()メソッドを実行することで、データベースへ接続する ② 変数\$itemIdListが空ではなかった場合、次の処理を行う <ol style="list-style-type: none"> a. 下記の文字列（SQL）を変数\$sqlへ格納する "SELECT * FROM item WHERE item_id = ?" b. 変数\$sqlを引数としてprepare()メソッドを呼び出すことでPDOStatementオブジェクトを取得し、変数\$statementへ格納する c. 変数\$itemIdList内にある要素を一つずつ変数\$itemIdに取得し、要素の分だけ次の処理を繰り返す <ol style="list-style-type: none"> ア. 変数\$itemIdを要素として持つ配列を作成し、それを引数として変数\$statementのexecute()メソッドを呼び出す イ. 「PDO::FETCH_ASSOC」を引数として変数\$statementのfetch()メソッドを呼び出すことで検索結果を取得し、変数\$rowsへ新しい要素として追加する 3. 例外(PDOException)が発生した場合は次の処理を行う <ol style="list-style-type: none"> ① 下記のようなエラーメッセージを持つ例外を発生させる 「データベースエラー：ファイル名(行番号)」 4. 例外発生の有無に関わらず、次の処理を行う <ol style="list-style-type: none"> ① 変数\$statementへnullを代入する ② データベースから切断する 5. 変数\$rowsを返す

ファイル情報	
ファイル名	OrderHistoryDao.php
ディレクトリ	/latte/app/dao/
クラス情報	
クラス名	OrderHistoryDao
継承	Dao
修飾子	なし
概要	order_history表のデータを扱うメソッドを管理するクラス
メソッド	
メソッド名	insert
引数	\$multiValues
修飾子	public
処理概要	order_history表に対してデータを登録する

メソッド詳細	
クラス名	OrderHistoryDao
メソッド名	insert
処理内容	<ol style="list-style-type: none"> 変数\$insertIdListに空の配列を格納する 次の処理を例外対応する <ol style="list-style-type: none"> open()メソッドを実行することで、データベースへ接続する トランザクションを開始する 下記の文字列（SQL）を変数\$sqlへ格納する "INSERT INTO order_history(customer_id,item_id) VALUES(?,?)" 変数\$sqlを引数としてprepare()メソッドを呼び出すことでPDOStatementオブジェクトを取得し、変数\$statementへ格納する 変数\$multiValues内にある要素を一つずつ変数\$valuesに取得し、要素の分だけ次の処理を繰り返す <ol style="list-style-type: none"> 変数\$valuesを引数として、変数\$statementのexecute()メソッドを呼び出す lastInsertId()メソッドを呼び出すことで最後に挿入された行IDを取得し、変数\$insertIdListに新しい要素として追加する コミットする 例外(PDOException)が発生した場合は次の処理を行う <ol style="list-style-type: none"> 下記のようなエラーメッセージを持つ例外を発生させる 「データベースエラー：ファイル名(行番号)」 例外発生の有無に関わらず、次の処理を行う <ol style="list-style-type: none"> 変数\$statementへnullを代入する データベースから切断する 変数\$insertIdListを返す

ファイル情報	
ファイル名	Model.php
ディレクトリ	/latte/app/model/
クラス情報	
クラス名	Model
継承	なし
修飾子	なし
概要	連想配列の形式で値を管理するクラス
プロパティ	
修飾子	private
プロパティ名	\$data
用途	連想配列を保持するプロパティ
メソッド	
メソッド名	__set
引数	\$name : プロパティ名 \$value : プロパティへ代入する値
修飾子	public
処理概要	連想配列に要素を追加する
メソッド名	__get
引数	\$name : プロパティ名
修飾子	public
処理概要	指定のプロパティ名に対応する連想配列の値を返す
メソッド名	__isset
引数	\$name : プロパティ名
修飾子	public
処理概要	指定のプロパティ名に対応する連想配列の値の存在を確認する
メソッド名	__unset
引数	\$name : プロパティ名
修飾子	public
処理概要	指定のプロパティ名に対応する連想配列の値を削除する
メソッド名	setProperties
引数	\$properties : クラスのプロパティに設定する連想配列
修飾子	public
処理概要	渡された連想配列をクラスのプロパティの連想配列にセットする
メソッド名	getProperties
引数	なし
修飾子	public
処理概要	クラスのプロパティの連想配列を返す

メソッド詳細	
クラス名	Model
メソッド名	__set
処理内容	
1. 第一引数をプロパティとして第二引数のデータを、クラスのプロパティの連想配列に格納する	

メソッド詳細	
クラス名	Model
メソッド名	__get
処理内容	
1. 第一引数の値をプロパティとして持つデータが存在する場合、そのデータを返し、存在しない場合はnullを返す	

メソッド詳細	
クラス名	Model
メソッド名	__isset
処理内容	
1. 第一引数の値をプロパティとして持つデータが存在する場合、そのデータを返し、存在しない場合はnullを返す	

メソッド詳細	
クラス名	Model
メソッド名	__unset
処理内容	
1. 第一引数の値をプロパティとして持つデータを削除する	

メソッド詳細	
クラス名	Model
メソッド名	setProperties
処理内容	
1. 引数に受け取ったデータが配列である場合、そのデータをクラスのプロパティに設定し、配列ではない場合、下記のようなエラーメッセージを持つ例外を発生させる 「Invalid argument is not an array」	

メソッド詳細	
クラス名	Model
メソッド名	getProperties
処理内容	
1. クラスのプロパティの連想配列を返す	

ファイル情報	
ファイル名	functions.php
ディレクトリ	/latte/app/utilities
メソッド詳細	
メソッド名	h
引数	\$string : XSS対策処理を行う文字列
修飾子	なし
処理内容	<p>1. htmlspecialchars()関数を呼び出すことでXSS対策処理後の文字列を取得し、その値を返す。呼び出しの際、第一引数には\$stringを指定し、第二引数にはENT_QUOTESを指定する</p>
メソッド詳細	
メソッド名	session_invalidate
引数	なし
修飾子	なし
処理内容	<p>1. セッションが開始していない場合はセッションを開始する 2. セッション変数を全て解除する 3. セッションクッキーを削除する 4. セッションを破棄する</p>
メソッド詳細	
メソッド名	app_log
引数	\$message : ログの内容
修飾子	なし
処理内容	<p>1. error_log()関数を呼び出すその際、この関数が受け取った引数とこのログの時刻情報を含めた文字列を渡す</p>
メソッド詳細	
メソッド名	createController
引数	\$className : クラス名 (コントローラー)
修飾子	なし
処理内容	<p>1. 引数で指定されたコントローラークラス (ファイル) を読み込む 2. 引数で指定されたコントローラークラスのオブジェクトを生成して返す</p>
メソッド詳細	
メソッド名	createDao
引数	\$className : クラス名 (DAO)
修飾子	なし
処理内容	<p>1. 引数で指定されたDAOクラス (ファイル) を読み込む 2. 引数で指定されたDAOクラスのオブジェクトを生成して返す</p>

ファイル情報	
ファイル名	cart.php
ディレクトリ	/latte/app/view
<pre> <?php header("Content-Type:text/html;charset=utf-8"); ?> <?php \$title = "カートページ" ?> <?php require_once ("header.php"); ?> <div id="content"> <h3>カート内の商品</h3> 変数\$modelのcartItemが空の場合、次の処理を行う <p class="cart_message">カートに商品はありません。</p> 前述の条件を満たしていない場合、次の処理を行う <p class="cart_message">現在買い物カゴには以下の商品が入っています。</p> <table class="cart_items"> <tr> <th>商品名</th> <th>メーカーなど</th> <th>価格</th> <th>&nbsp;</th> </tr> 変数\$indexに0を代入 変数\$modelのcartItemsから、変数\$itemへ要素を取り出し、要素数の分だけ次の処理を繰り返す 返す <tr> <td> 変数\$item内の"name"要素を出力する </td> <td> 変数\$item内の"maker"要素を出力する </td> <td> 変数\$item内の"price"要素を出力する 円</td> <td> <form action="index.php" method="post"> <input type="submit" value="取り消し"> <input type="hidden" name="action" value="removeCart"> <input type="hidden" name="index" value=" 変数\$indexを出力する "> </form> </td> </tr> 変数\$indexをインクリメントする 繰り返す処理内容はここまで </table> <p class="order_button"> </p> 前述の条件に対する処理はここまで <p class="pageback"> 検索ページに戻る </p> </div> <?php require_once ("footer.php"); ?> </pre>	

ファイル情報	
ファイル名	error.php
ディレクトリ	/latte/app/view
<pre> <?php header("Content-Type:text/html;charset=utf-8"); ?> <?php \$title = "エラーページ" ?> <?php require_once ("header.php"); ?> <div id="content"> <h3>エラーページ</h3> <p class="error_message"> エラーが発生しました。
 <?php echo \$model->message; ?> </p> <p class="pageback"> 検索ページに戻る </p> </div> <?php require_once ("footer.php"); ?> </pre>	

ファイル情報	
ファイル名	footer.php
ディレクトリ	/latte/app/view
<pre> <p class="pageup"> </p> </div> <div id="footer"> <small>&copy;Copyright 2012 ThinkethBank,Inc. All rights reserved.</small> </div> </div> </body> </html> </pre>	

ファイル情報	
ファイル名	header.php
ディレクトリ	/latte/app/view
<pre> <!DOCTYPE html> <html lang="ja"> <head> <meta charset="utf-8"> <link rel="stylesheet" type="text/css" href="css/style.css"> <title>ショッピングサイトLatte <?php echo \$title; ?></title> </head> <body> <div id="wrap"> <div id="header"> <h1> </h1> <ul class="navi_list"> <li class="h_home">HOME <li class="h_guide">ショッピングガイド <li class="h_que">よくある質問 <li class="h_com">会社案内 <li class="h_con">お問い合わせ <h2>Latte <?php echo \$title; ?></h2> </div> <div id="content_wrap"> <div id="menu"> <ul class="guide_list"> ご購入方法 お支払方法 配送料 ラッピング 返品・交換 <p class="cart_button"> </p> <p class="notification"> 全国無料配送！
 クレジットカードも
 ご利用になれます。 </p> <address class="com_info"> 販売元
 (株)シンクスバンク </address> </div> </div> </div> </pre>	

ファイル情報	
ファイル名	order_complete.php
ディレクトリ	/latte/app/view
<pre> <?php header("Content-Type:text/html;charset=utf-8"); ?> <?php \$title = "注文完了ページ" ?> <?php require_once ("header.php"); ?> <div id="content"> <h3>注文完了</h3> <p class="complete_message"> ご注文ありがとうございます。
 24時間以内に当サイト管理者よりメールがまいりますので、よろしくお願い致します。
 在庫状況により商品の到着が遅れる場合がございますので、あらかじめご了承ください。 </p> <p class="pageback"> 検索ページに戻る </p> </div> <?php require_once ("footer.php"); ?> </pre>	

ファイル情報	
ファイル名	order.php
ディレクトリ	/latte/app/view
<pre> <?php header("Content-Type:text/html;charset=utf-8"); ?> <?php \$title = "注文ページ" ?> <?php require_once ("header.php"); ?> <div id="content"> <h3>合計金額</h3> <p class="total">現在買い物カゴの合計金額は スーパーグローバル変数\$_SESSION[" sumPrice"]を出力する 円（税込み） です。</p> <h3>注文者情報</h3> <p class="order_message">変数\$modelのmessageを出力する</p> <form action="index.php" method="post"> <table class="customer"> <tr> <th>お名前</th> <td> <input type="text" name="name" size="50" value="変数\$modelのnameを出力する "(">
 （例）山田 太郎 </td> </tr> <tr> <th>フリガナ</th> <td> <input type="text" name="furigana" size="50" value="変数\$modelのfurigana を出力する"(">
 （例）ヤマダ タロウ</td> </tr> <tr> <th>郵便番号</th> <td> <input type="text" name="zipcode" size="8" value="変数\$modelのzipcodeを出 力する"(">
 （例）153-0052
 </pre>	

```

        ※郵便番号検索は<a class="zipcode" href="https://www.post.japanpost.jp/zipcode/index.html" target="_blank">こちら</a>
    </td>
</tr>
<tr>
    <th>住所</th>
    <td>
        <input type="text" name="address" size="60" value="変数$modelのaddressを出力する"><br>
        (例) 東京都目黒区目黒2-2-2 ムビル30F 総務部<br>
        ※ マンション・アパート名は必ず入力してください。<br>
        ※ 住所が勤務先の場合は「部署名」なども入力してください。
    </td>
</tr>
<tr>
    <th>電話番号</th>
    <td>
        <input type="text" name="phoneNumber" size="20" value="変数$modelのphoneNumberを出力する"><br>
        (例) 03-0000-0000
    </td>
</tr>
<tr>
    <th>メールアドレス</th>
    <td>
        <input type="text" name="email" size="50" value="変数$modelのemailを出力する"><br>
        (例) hogehoge@kenschool.jp
    </td>
</tr>
</table>
<p class="order_message">上記の内容で問題なければ、注文を確定します。</p>
<p class="order_button">
    <input name="cart_enter" type="image" src="img/btn_order.gif">
    <input type="hidden" name="action" value="orderComplete">
</p>
</form>
<p class="pageback">
    <a href="index.php?action=cart">ショッピングカートに戻る</a>
    <a href="index.php">検索ページに戻る</a>
</p>
</div>
<?php require_once ("footer.php"); ?>

```

ファイル情報	
ファイル名	top.php
ディレクトリ	/latte/app/view
<pre> <?php header("Content-Type:text/html;charset=utf-8"); ?> <?php \$title = "トップページ" ?> <?php require_once ("header.php"); ?> <div id="main_visual"> </div> <div id="serchbox"> <h3>カテゴリーを選択して検索して下さい。</h3> <form action="index.php" method="post"> <table class="search"> <tr> <th>分類</th> <select name="category"> 変数\$modelのcategoryListから変数\$categoryへ要素を取り出し、要素数の分だけ次の処理を繰り返す 変数\$category内の"category_id"要素がスーパーグローバル変数\$_SESSION["category"]と一致していたら、次の処理を行う <option value="変数\$category内の"category_id"要素を出力する" selecte d> 変数\$category内の"name"要素を出力する </option> 前述の条件を満たしていない場合、次の処理を行う <option value="変数\$category内の"category_id"要素を出力する"> 変数\$category内の"name"要素を出力する </option> 前述の条件に対する処理はここまで 繰り返す処理内容はここまで </select> </td> <th>キーワード</th> <td><input type="text" name="keyword" size="20" value="スーパーグローバル変数\$_SESSION["keyword"]を出力する"></td> <td> <input type="submit" value="検索"> <input type="hidden" name="action" value="search"> </td> </tr> </table> </form> </div> <div id="content"> <div id="select_item"> <h3>現在選択されている商品</h3> <p class="item_count"> 選択商品 スーパーグローバル変数\$_SESSION["cart"]の内容が設定されている場合はその要素数を出力し、内容が設定されていない場合は0を出力する 個 </p> <p class="cart_btn"> </p> </div> <div id="select_category"> <h3>選択カテゴリー商品</h3> </pre>	

```

        スーパーグローバル変数$_SESSION["searchResults"]に情報が登録されている場合は次の処理を行う
        スーパーグローバル変数$_SESSION["searchResults"]が空である場合、次の処理を行う
        <p class="search_message">該当する商品はありません。</p>
        前述の条件を満たしていない場合、次の処理を行う
        <table class="search_items">
            <tr>
                <th>商品</th>
                <th>商品名</th>
                <th>メーカーなど</th>
                <th>価格</th>
                <th>&nbsp;</th>
            </tr>
            スーパーグローバル変数$_SESSION["searchResults"]から変数$resultへ要素を取り出し、要素数の分だけ次の処理を繰り返す
            <tr>
                <td></td>
                <td>変数$result内の"name"要素を出力する</td>
                <td>変数$result内の"maker"要素を出力する</td>
                <td>変数$result内の"price"要素を出力する 円</td>
                <td>
                    <form action="index.php" method="post">
                        <input type="image" src="img/btn_addition.gif">
                        <input type="hidden" name="action" value="addCart">
                        <input type="hidden" name="itemId" value="変数$result内の"item_id"要素を出力する">
                    </form>
                </td>
            </tr>
            繰り返す処理内容はここまで
        </table>
        前述の条件に対する処理はここまで
        前述の条件を満たしていない場合、次の処理を行う
        <p class="search_message">商品を検索してください。</p>
        前述の条件に対する処理はここまで
    </div>
</div>
<?php require_once ("footer.php"); ?>

```

最終更新日 2019年11月25日