




補足資料その2



フォーム関連

フォームを2回に分けて

お客様情報

会社名	株式会社サンプルサイト		
お名前	必須	山田	太郎
メールアドレス	必須	info@example.com	
お電話番号	000-000-0000		
ご住所	必須	000-0000	都道府県 ▾
		名駅中村区3丁目18-5	
		モンマートビル5F	

ご記入いただいた個人情報はプライバシーポリシーに基づき管理いたします。（カスタムフィールドで自由に書き換えたいエリア）

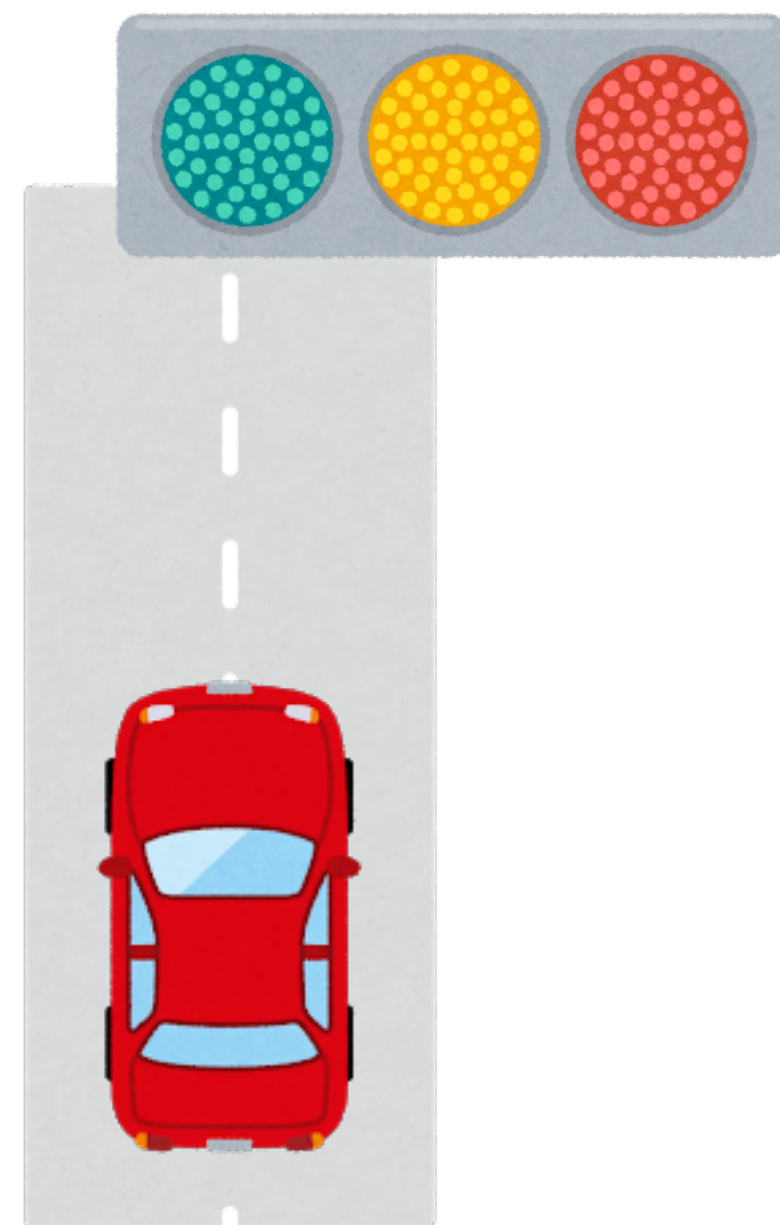
送信内容の確認へ

1回目 ・ ・ 簡易版 ・ ・ まずは概要を
2回目 ・ ・ 詳細版 ・ ・ セキュリティも意識

Web通信 ・ http

フォームには必須のhttp通信

Hyper Text Transfer Protocol (ルール)



クライアント



サーバー



Laravel

httpと https

http, https ・ ・ Web通信の決まり毎。
httpsは暗号化。できるだけhttpsで。

SSL(Secure Socket Layer)

ドメイン毎に認証

レンタルサーバーでは共有の暗号化の仕組みも

<https://www.xserver.ne.jp/>

HTTPリクエストとレスポンス

HTTPリクエスト

- HTTPリクエスト行 (メソッド)
- HTTPヘッダー
- データ本体



HTTPレスポンス

- レスポンス状態行 (状態コード)
- HTTPヘッダー
- データ本体

httpメソッド

GET ・ ・ URLに表示される。(検索条件など) → クエリーストリング

POST ・ ・ 見られてはNGなデータはこっち

[https://qiita.com/7968/items/
4bf4d6f28284146c288f](https://qiita.com/7968/items/4bf4d6f28284146c288f)


フォームの種類

input type=text, radio, checkbox,
menu, textarea, submit, etc...

<https://webliker.info/39533/>

<http://www.htmq.com/html5/input.shtml>

[https://developer.mozilla.org/ja/docs/Web/
HTML/Element/Input](https://developer.mozilla.org/ja/docs/Web/HTML/Element/Input)



フォームの セキュリティ

代表的な攻撃と対策



XSS (Cross-Site Scripting)

クリックジャッキング

CSRF (Cross-Site Request Forgeries)

SQLインジェクション->DB時

etc...

対策：サニタイズ (消毒)、バリデーション
(検証)

CSRF対策

\$_GET
\$_POST



1回きり



\$_SESSION



残る



\$_SESSIONを使ったトークンを発行

バリデーション

HTML5側とサーバー側 両方で

文字・・・未入力、文字数。

メールアドレス・・・未入力、1つだけか。

性別などの選択項目・・・未入力。

郵便番号・電話番号・カナ etc..



おまけ Bootstrap4

Bootstrap4

よく使われるレイアウト、デザインが
セットになったCSSフレームワーク。
CSS + jQuery(JavaScript)

グリッドシステムが特徴
画面を12分割で考え、
画面幅によって表示を変える。



ベーシック認証

認証の種類・・・フレームワーク推奨

ベーシック認証(Basic認証) SSL/TSL推奨
ダイジェスト認証
セッション認証
データベースを使った認証
JWT(JSON Web Token)認証
OAuth2.0 認証 (SNS認証)
2段階認証(多要素認証)

ベーシック認証の設定ファイル

.htaccess ファイルで指定

- ・ ・ サーバー(Apache)の設定ファイル
ディレクトリ毎に動作を制御できる
- ・ リダイレクト
- ・ アクセス切り替え(PC版とスマホ版など)
- ・ 特定IPアドレス・プロバイダからアクセス制限
- ・ ベーシック認証 etc...



ファイル操作

データを保存する方法

ファイル(テキストファイル)
->手軽・データのやり取り

データベース(MySQL, MariaDB)
->大量のデータを保管

ファイル操作の方法

- ファイル名型 (ファイル丸ごと)
file_get_contents, file_put_contents
- ストリーム型 (1行ごと)
fopen, fclose, fgets, fwrite
- オブジェクト型 (オブジェクトとして)
SplFileObject


<https://qiita.com/tadsan/items/0955b3de7dc58490ddaf>

ファイル操作の流れ(ストリーム型)

1. 開く `fopen (r, w, a)`
2. 排他ロック `flock`
3. 読込/書込/追記 `fgets/fwrite`
4. 閉じる `fclose` (ロック解除)

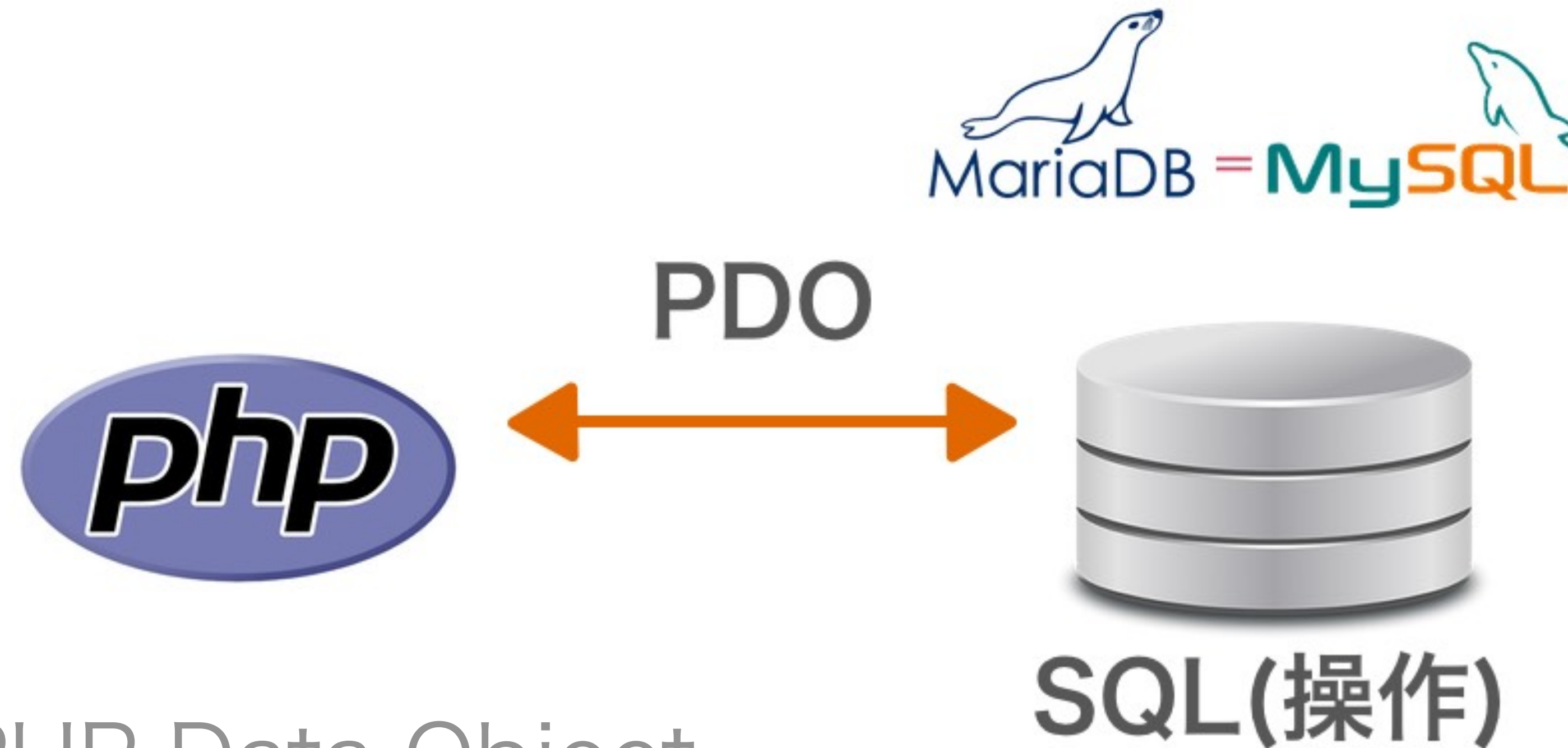
<https://wepicks.net/phpsample-file-create/>

<https://www.flatflag.nir87.com/fgets-810>



データベース (MySQL, MariaDB) と接続

データベースと接続



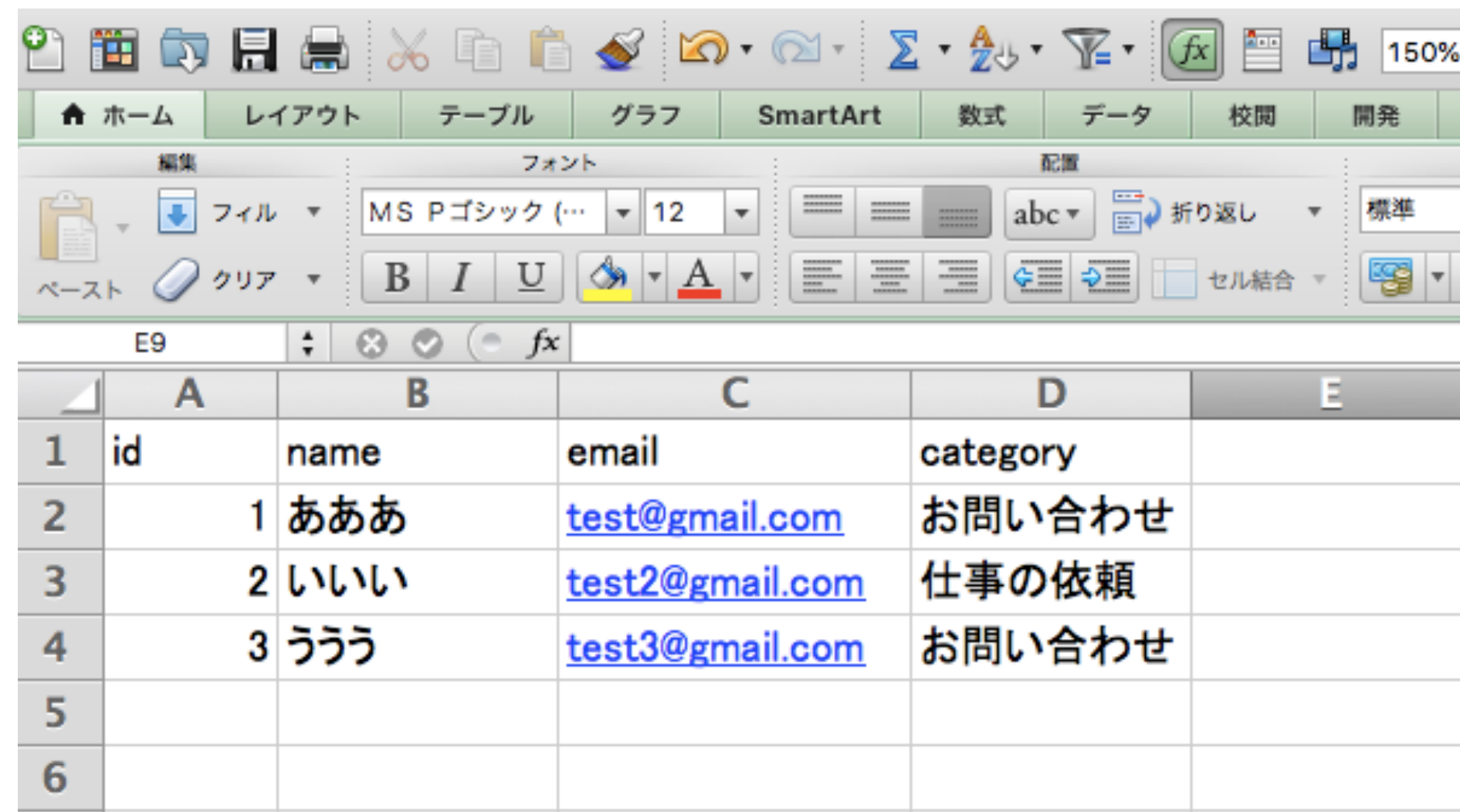
PDO ・ ・ PHP Data Object

ORM, OR/マッパー ・ ・ SQLをラップ(SQLをPHPで書ける)

エクセルに例えてみると



データベース・・・エクセルファイル
テーブル・・・シート

A screenshot of the Microsoft Excel application window. The ribbon at the top shows tabs for 'ホーム' (Home), 'レイアウト' (Layout), 'テーブル' (Table), 'グラフ' (Chart), 'SmartArt', '数式' (Formulas), 'データ' (Data), '校閲' (Review), and '開発' (Developer). The 'ホーム' tab is active, showing options for '編集' (Edit), 'フォント' (Font), and '配置' (Alignment). The font settings are set to 'MS Pゴシック' and size '12'. The alignment settings show '標準' (Default) for text alignment and '折り返し' (Wrap) for text wrapping. The formula bar shows 'E9'. The worksheet grid displays data in columns A through E and rows 1 through 6. The data is as follows:

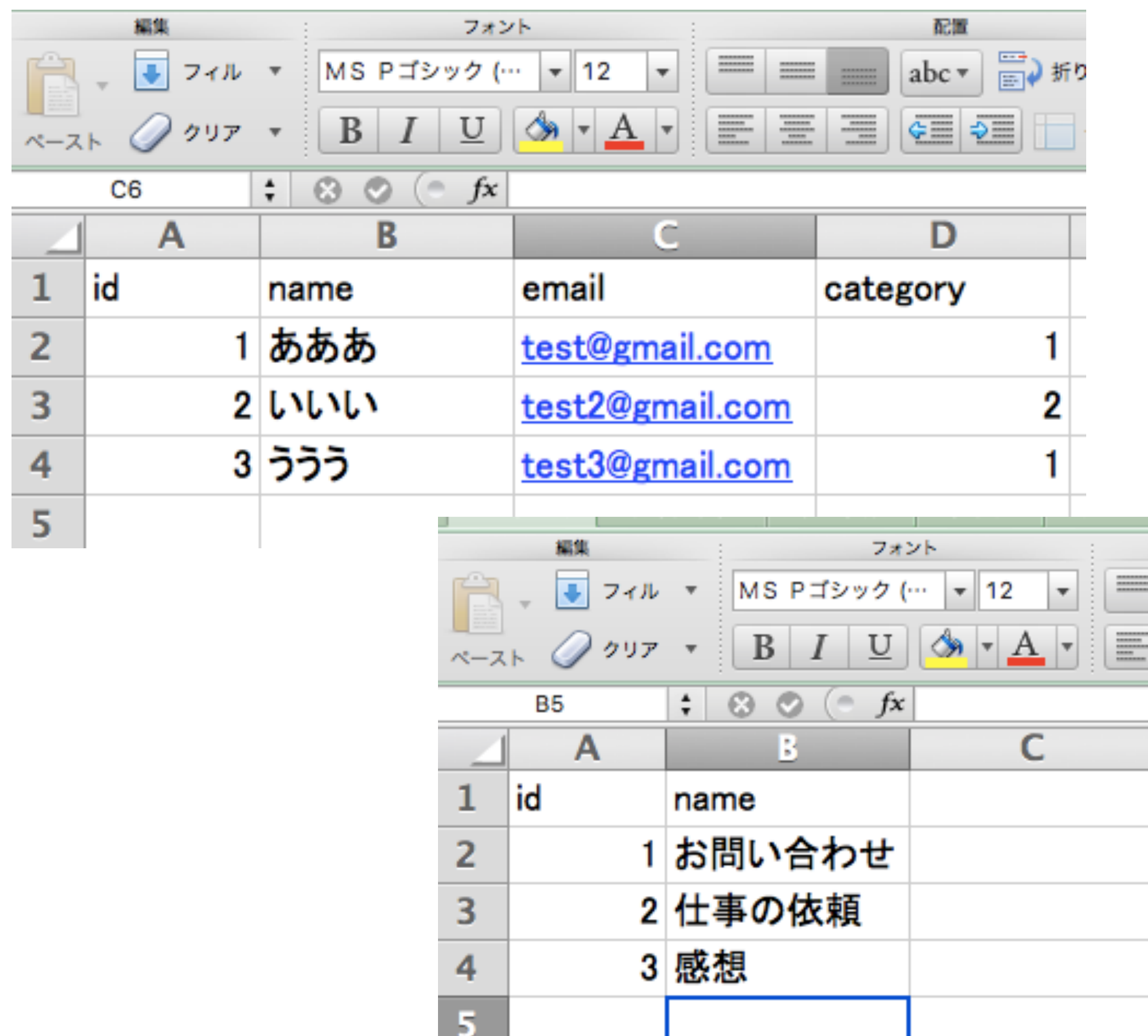
	A	B	C	D	E
1	id	name	email	category	
2	1	あああ	test@gmail.com	お問い合わせ	
3	2	いいい	test2@gmail.com	仕事の依頼	
4	3	ううう	test3@gmail.com	お問い合わせ	
5					
6					

レコード(ロウ)・・・行 (横)

カラム・・・列 (縦)

フィールド・・・セル(1つ1つ)

DBの種類と特徴



	A	B	C	D
1	id	name	email	category
2	1	あああ	test@gmail.com	1
3	2	いいい	test2@gmail.com	2
4	3	ううう	test3@gmail.com	1
5				

	A	B	C
1	id	name	
2	1	お問い合わせ	
3	2	仕事の依頼	
4	3	感想	
5			

膨大なデータを管理

・ 数百万、数千万でも。

RDB ・ ・

リレーショナルデータベース

複数のテーブルを紐づける事ができる

NoSQL ・ ・ 高速処理

DB操作の基本 CRUD

Create 新規作成 insert

Read 表示 select

Update 更新 update (上書き)

->履歴を残すか、完全に上書きするか

Delete 削除 delete

->完全に消すか、非表示にするか

データ量が膨大になるなら

パーティション(分割)、インデックス(索引)

レプリケーション(ミラーリング)も要検討

テーブル作成時に抑えたい事

ストレージエンジン・・・基本はInnoDB

照合順序・MySQL 5.5.3 (2010年)以降はutf8mb4推奨

データ型・・・

数字(整数(int) 少数(float)、金額なら整数かdecimal)

文字列(varchar)、boolean(真偽)、Date など

想定される文字数に合わせる

Nullを許容するか

DB操作の基本 Select

Select 表示させたい項目

From table名

Where 検索条件

Group by グループ化 Order by 表示順序

Join テーブル結合(inner/outer/left)

少し高度・サブクエリ、Window関数



クラスの説明(PDO)

クラスの考え方 (ひとまとめ)

変数/定数
関数

クラス(class)

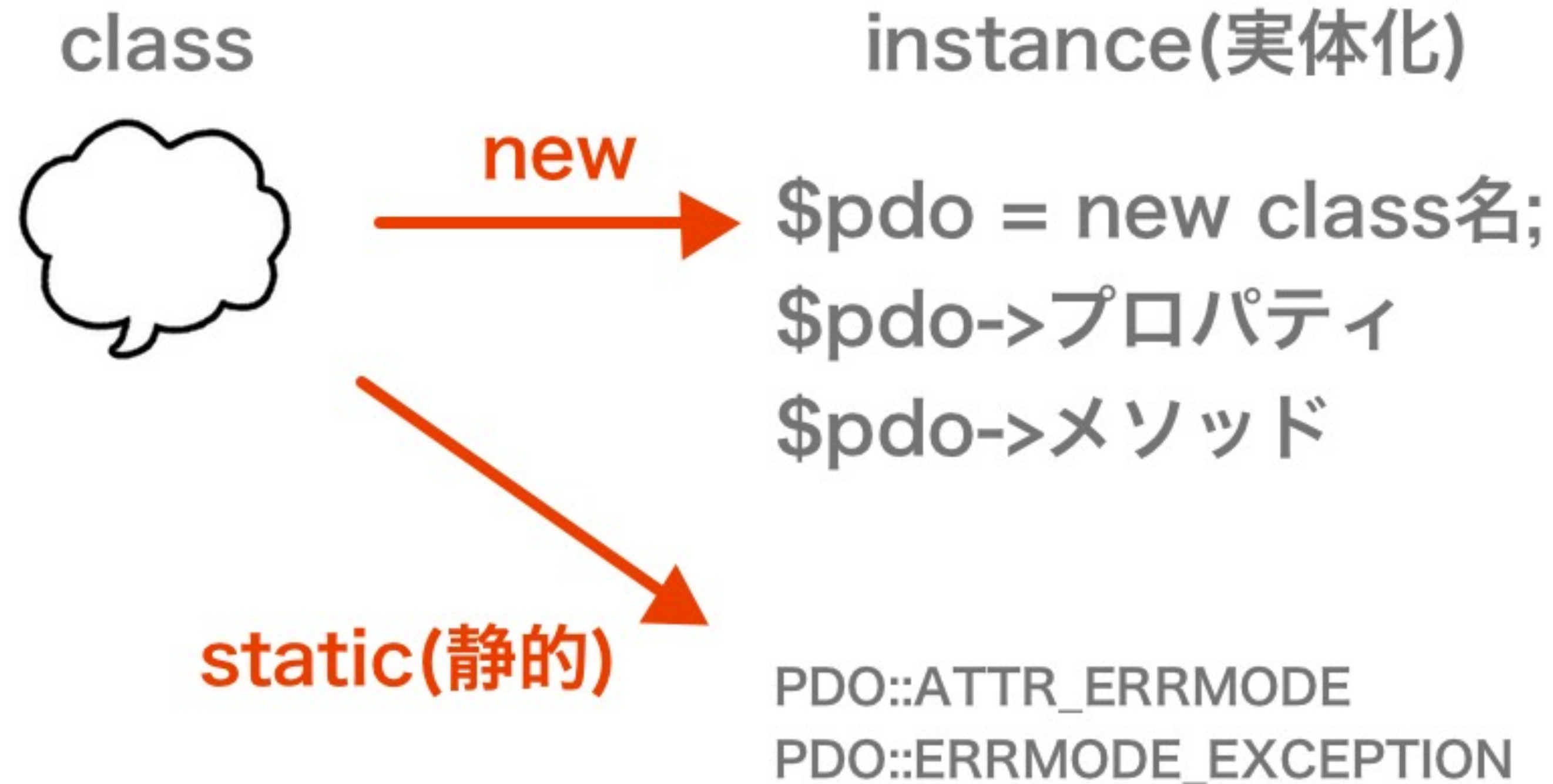
変数/定数
関数

変数/定数->プロパティ

関数->メソッド

関連する機能をひとまとめに(責務の分離)

クラスの使い方(2パターン)



動的と静的
アロー演算子とスコープ演算子

データベース接続 用語

プリペアドステートメント(予約文)

プレースホルダ(仮の情報)

バインド(紐づける)

トランザクション(排他ロック)

正規化(第3正規化)

SQLインジェクション(セキュリティ対策)



クッキーと セッション

クッキーとセッション

\$_COOKIE



\$_SESSION

パスワード保存はNG

セッション認証

近年GDPRなどで

利用が制限されつつある

セッション認証のサンプル

[https://tadworks.jp/archives/
1147#accounts](https://tadworks.jp/archives/1147#accounts)

これまでの知識で読めるようになっていていると思います。



関数あれこれ

関数あれこれ

引数にデフォルト値(初期値)設定可能
タイプhint(型を明示できる)
可変引数 (ドット3つ)

無名関数、クロージャ、コールバック関数
引数にインスタンス(メソッドチェーン)



クラスと オブジェクト指向

オブジェクト指向



オブジェクト指向とは-> 役割分担

会社・・部門毎に分ける

・・・営業、総務、開発、経営、PRなど

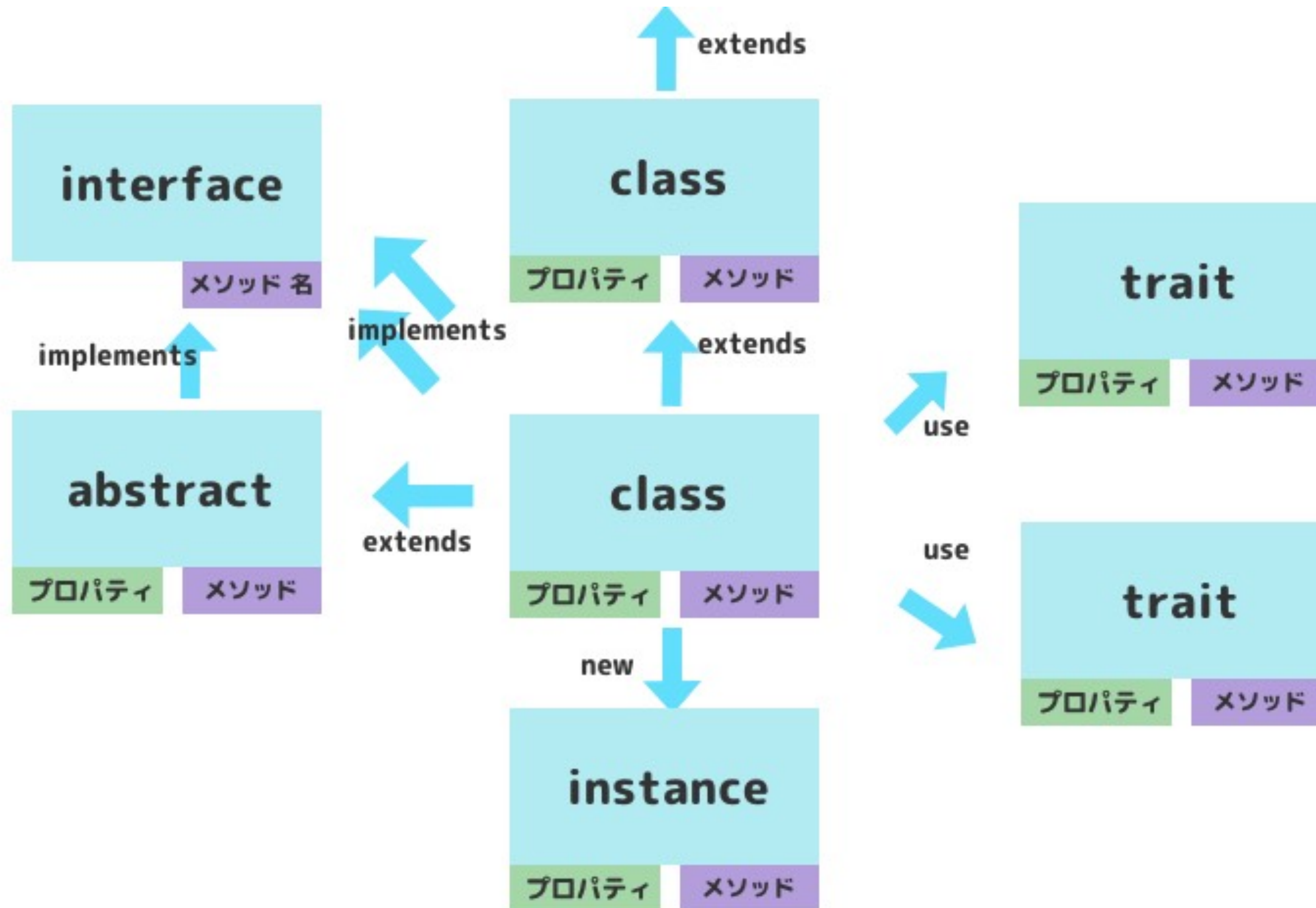
オブジェクト指向の3要素

- ・カプセル化

- ・継承

- ・ポリモーフィズム(多様化)

クラスの関係図



それぞれの特徴(ざっくり)

タイプ	宣言	インスタンス化	利用	アクセス修飾子	特徴
具象クラス	class	○	extends	pub, pro, pri	単一継承
抽象クラス	abstract class	×	extends	pub, pro, pri	継承前提
インターフェース	interface	×	implements	Pub	メソッド名のみ
トレイト	trait	×	use	pub, pro, pri	複数use

オブジェクト指向のコツ

1. 用語・使い方を知る(なんとなく)
2. ソース読む(Laravel、各種ライブラリ)
3. デザインパターンを知る

デザインパターン



[http://shimooka.hateblo.jp/archive/
category/phpdp](http://shimooka.hateblo.jp/archive/category/phpdp)

[https://github.com/shimooka/
PhpDesignPattern](https://github.com/shimooka/PhpDesignPattern)



モダンPHP

モダンPHP(最近のPHP)

PSR-1,2,4(PHPコーディング規約)

namespace (名前空間)

autoload (自動クラス読み込み)

composer

(PHPライブラリ管理+オートロード)

Composer

composer init (初期設定)

composer.json 設定ファイル

composer.lock 設定ファイル(バージョン固定)

composer update //設定ファイル更新

composer dump-autoload //オートロード更新