

クレジット:

UTokyo Online Education Education コンピュータシステム概論 2018 小林克志

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



# コンピュータシステム概論

## 第9回

小林克志

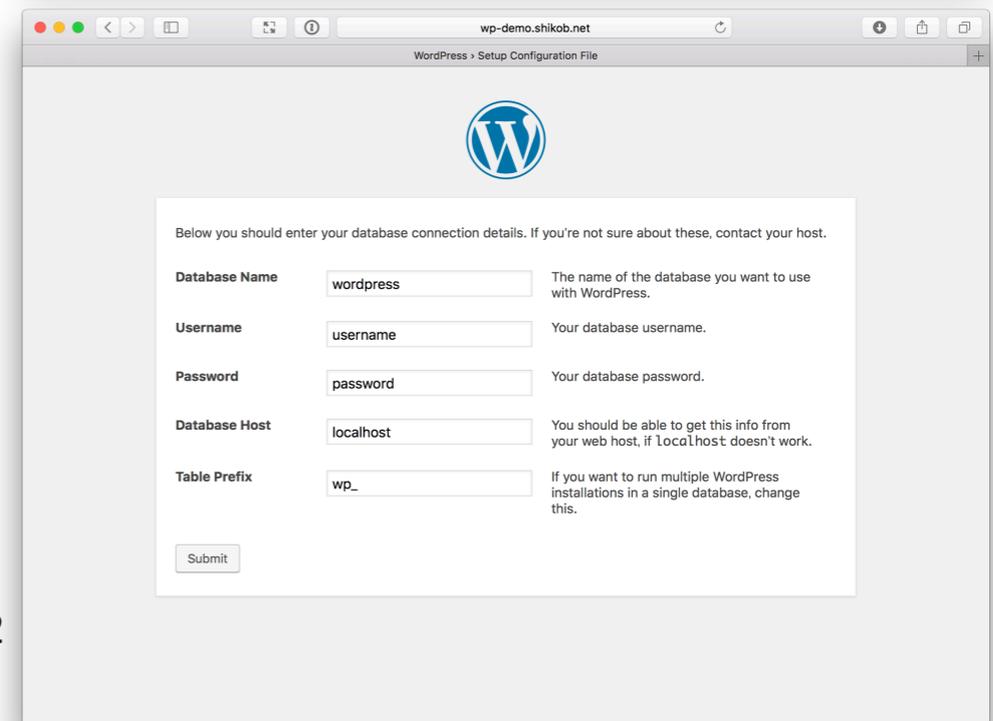
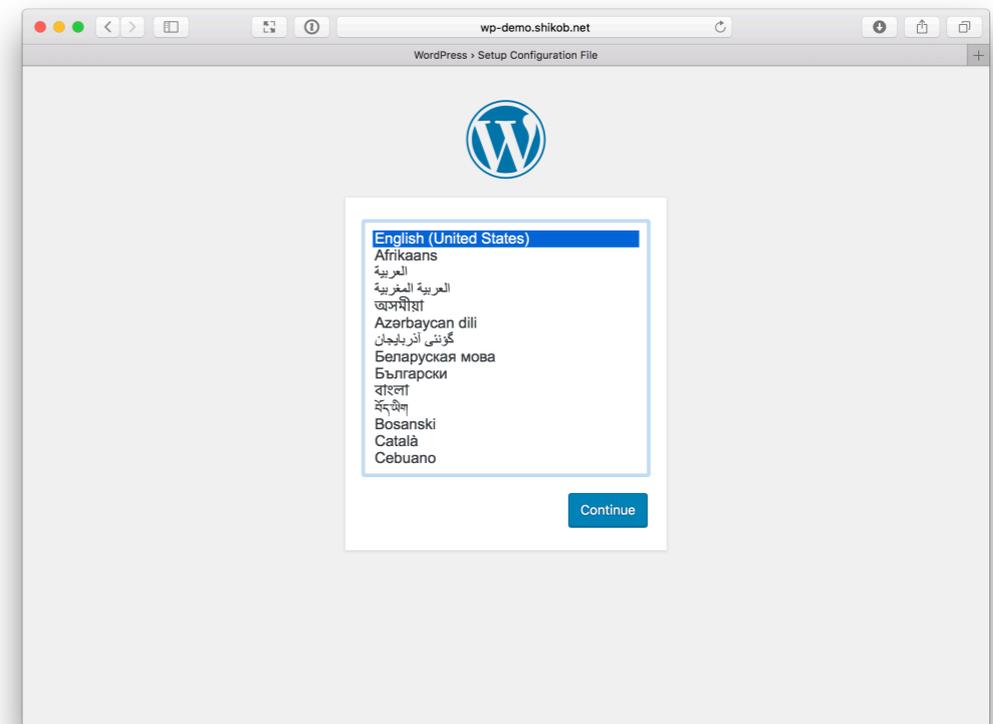
- 事務連絡
- 先週の課題、レビュー（振り返り）
- まくら
- データベース(1)
- 演習：Wordpress データベースを覗く
- データベース(2)

# WP の install にハマっている人向け エディタを使わない WP のインストール方法

- 設定ファイルを消去、あるいはファイル名変更、する。
  - さらに、chown で /var/www/html の owner を apache に変更する。

```
$ sudo rm /var/www/html/wp-config.php
$ sudo chown -R apache.apache /var/www/html
```

- Web でアクセス、言語を適当に選んで Continue をクリック
- 説明を読んで、Let's go! をクリック
- 右下のメニューに設定を書き込んで、Submit する。
- wordpress データベースが中途半端に install されている場合は失敗する。このときは相談すること。



The WordPress software, GPLv2

The screenshot shows a web browser window with the URL [www.city.osaka.lg.jp/](http://www.city.osaka.lg.jp/). The page title is "大阪市【軽量版】". The main navigation bar includes "大阪市" and links for "本文へ", "English", "中文簡体", "한글", "Other Languages", "Living Info", "ふりがな", "文字サイズ", "拡大", and "標準". A red-bordered box contains the text "災害時用（軽量版）のトップページを表示しています". Below this is a red header for "緊急情報". The main content area lists seven items, each with a date of 2018年6月19日 and a link: "大阪府北部を震源とする地震による被害及び対応状況", "各区の避難所一覧（おおさか防災ネット）", "地震にかかる夜間時間帯の注意喚起", "地震によりガスが使用できない場合があります", "地震に関わるこころのケア電話相談", "地震で被災された中小企業を支援します", and "大阪市防災アプリをご活用ください". A red button at the bottom right says "安全・安心に関するお知らせ".

大阪市 [本文へ](#) [English](#) [中文簡体](#) [한글](#) [Other Languages](#) [Living Info](#) [ふりがな](#) [文字サイズ](#) [拡大](#) [標準](#)

災害時用（軽量版）のトップページを表示しています

### 緊急情報

- 2018年6月19日 [大阪府北部を震源とする地震による被害及び対応状況](#)
- 2018年6月19日 [各区の避難所一覧（おおさか防災ネット）](#)
- 2018年6月19日 [地震にかかる夜間時間帯の注意喚起](#)
- 2018年6月19日 [地震によりガスが使用できない場合があります](#)
- 2018年6月19日 [地震に関わるこころのケア電話相談](#)
- 2018年6月19日 [地震で被災された中小企業を支援します](#)
- 2018年6月19日 [大阪市防災アプリをご活用ください](#)

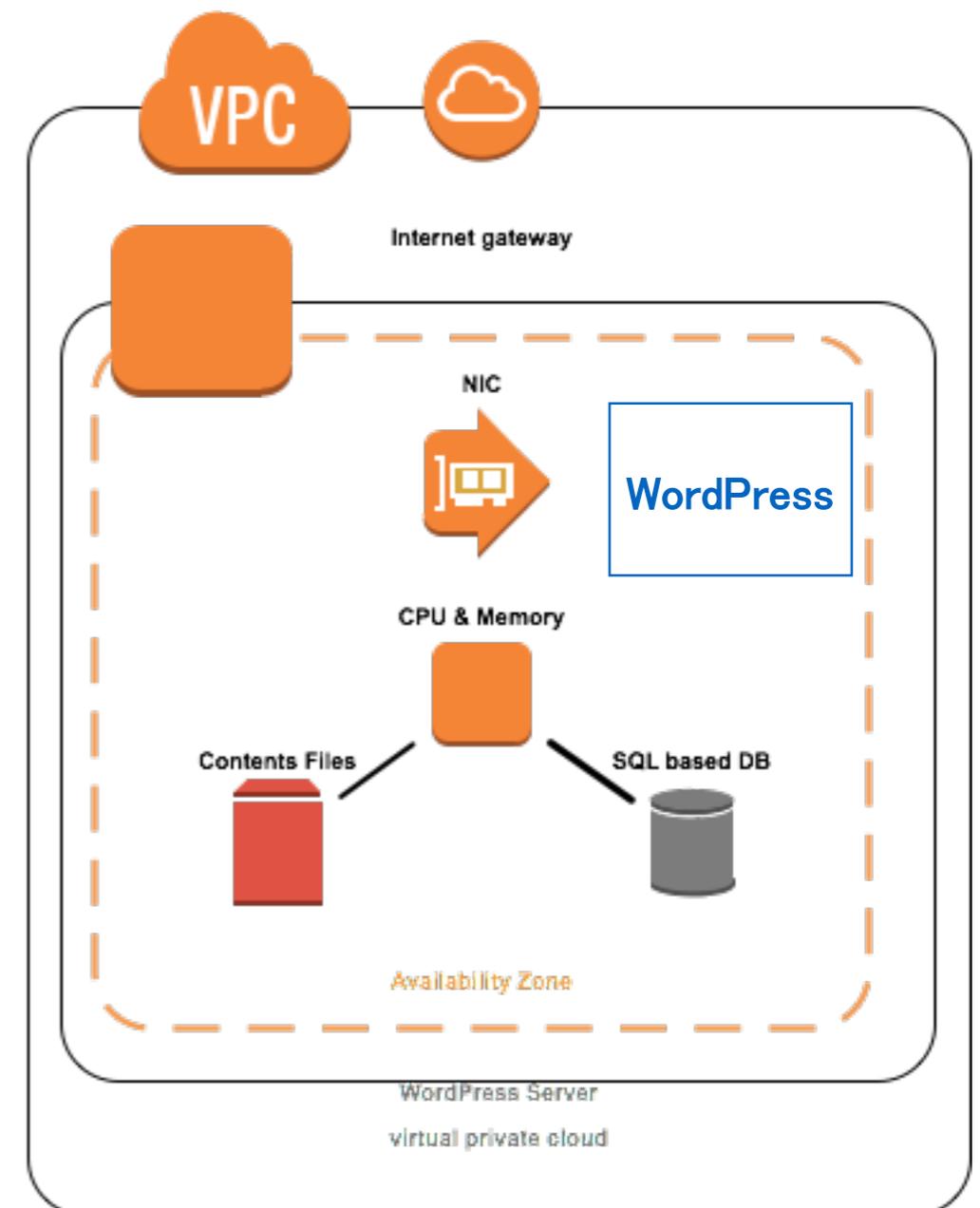
安全・安心に関するお知らせ

# Web サービスにおける トラフィック

- 爆発的なアクセスは常に発生している:
  - 2,68B USD on-line sales on Cyber Monday 2014 (Fundivo)
  - 7M VoD streams at Barack Obama's victory speech in 2009 (Akamai)
  - スポーツイベント
- アクセスに耐えられずサービス不可となれば利用者の満足度は下がる。
  - Loyalty・信頼性の低下 / 機会損失

# 演習：Wordpress インストール (再掲)

- 前提：前回の演習で、index.php が正常に表示されていること。
- Wordpress：コンテンツ管理システム (Contents Management System: CMS) で、動的な Web ページの提供に広く利用されている。
  - 30% の Web サーバで利用されている。(WP 調べ)
- ページ生成：PHP (PHP: Hyper Text Processor)
- コンテンツ管理：RDBMS + ファイル



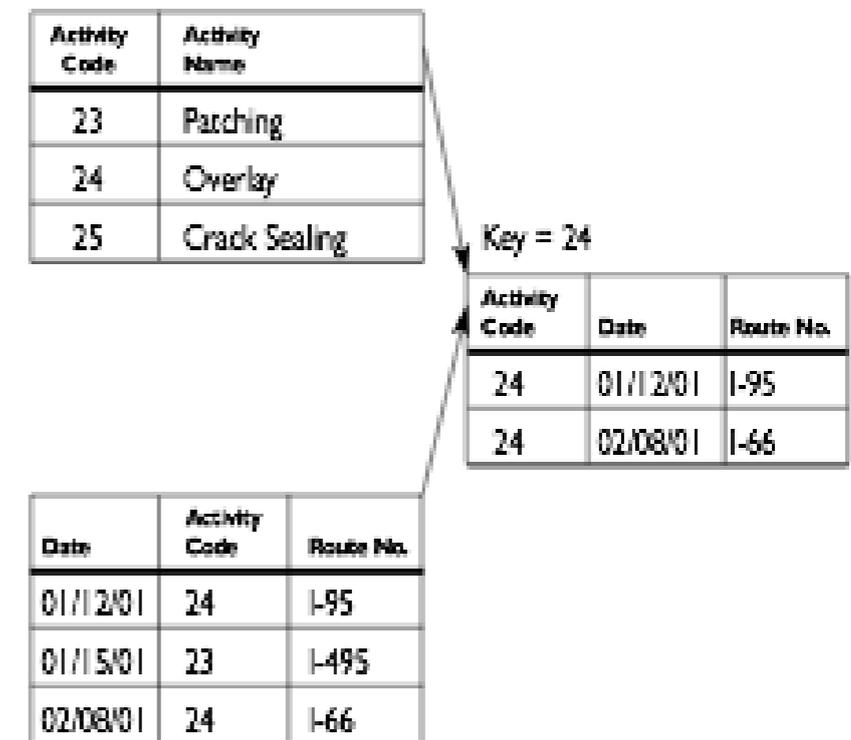
# コンピュータの記憶装置

- 揮発性 (Volatile)
  - DRAM などの主記憶が該当、電源を切ると操作結果が保存されない。
  - 永続性記憶と比較して高速だが高価。
- 永続性 (Persistent)
  - HDD, Flash メモリなどが該当、電源にかかわらず結果が保存される。
  - 速度・コスト・信頼性に多様な選択肢がある。
  - これへの基本操作 CREATE / READ / UPDATE / DELETE をまとめて、CRUD と呼ぶ。

# 関係データベース管理システム (RDBMS)

- 多くの情報システムのデータストア(\*)バックエンドとして広く利用されている。e.g., MySQL, Oracle Database, Postgress, DB2
  - 永続的なデータ保存・蓄積システム
- 関係 (Relational) モデル:  
“Data items organized as a set of formally described tables from which data can be accessed or reassembled in many ways without having to reorganize the database tables. Each table (sometimes called a relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns.”
- アプリケーションは SQL (Structured Query Language) 等からアクセス
- CRUD に加え、トランザクション機能も提供する

Relational Model



Data Integration Glossary,  
US: Department of Transportation, August 2001

# 関係データベース管理システム (RDBMS) (続き)

	データ格納場所・アクセス	複数ユーザ、アプリケーションによる共有	操作結果	初期化
データベース	DBMS が管理、列 (Column)-値(Value) による検索を始め、多様なアクセス	DBMS が管理	永続性	列名・型が定義されたテーブルが必要
ファイルシステム	ユーザが管理、検索はシーケンシャルが基本	排他機構が必要	永続性	不要
主記憶	ユーザが管理	排他機構が必要	揮発性	ゼロクリアは必要

# 演習: Wordpress データベースを覗く ログイン

- mysql コマンドで MySQL にログイン、データベースを表示する。  
行頭の mysql> は mysql の入力待ちプロンプト、SQL では ; が末尾に必要。

```
$ mysql -u root -p
Enter password: <以前設定した MySQL の root パスワード>
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      | <- mysql 設定など
| performance_schema |
| wordpress  | <- wordpress のデータ
+-----+
mysql>
```

- MySQL に登録した wordpress データベースを確認する。

```
mysql> use mysql;           <- mysql データベースに接続
mysql> show tables;        <- Mysql データベース内のテーブルを表示
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| db            |
+-----+
mysql>
+-----+
| user          | <- wordpress のデータ
+-----+
mysql>
```

# 演習: Wordpress データベースを覗く MySQL ユーザの確認

```
mysql> USE mysql;          <- mysql データベースに接続
mysql> SHOW tables;       <- Mysql データベースのテーブルを表示
+-----+
| Tables_in_mysql |
+-----+
| columns_priv |
| db |
+-----+
| user |          <- user テーブルが存在する
+-----+

mysql> DESC user;         <- user テーブルの列情報を表示
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| Host | char(60) | NO | PRI | | |
| User | char(16) | NO | PRI | | |          <- ユーザ名
| Password | char(41) | NO | | | |          <- パスワード
+-----+

mysql> SELECT Host,User,Password from user;
+-----+
| Host | User | Password |
+-----+
| localhost | root | ***** |
| 127.0.0.1 | root | ***** |
| ::1 | root | ***** |
| localhost | wordpress | ***** |          <- wordpress ユーザ
+-----+

mysql> SHOW GRANTS for 'wordpress'@'localhost';
+-----+
| Grants for wordpress@localhost |
+-----+
| GRANT USAGE ON *.* TO 'wordpress'@'localhost' IDENTIFIED BY PASSWORD '*****' |
| GRANT ALL PRIVILEGES ON `wordpress`.* TO 'wordpress'@'localhost' |          <- wordpress ユーザは wordpress データベースに対する権限が与えられている
+-----+

mysql>
```

# 演習: Wordpress データベースを覗く

## Wordpress データベース / ユーザテーブル

```
mysql> USE wordpress;      <- wordpress データベースに接続
mysql> SHOW tables;       <- wordpress データベースのテーブルを表示
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta      |
| wp_comments         |
| wp_links            |
| wp_options          |
| wp_postmeta         |
| wp_posts            |
| wp_term_relationships |
| wp_term_taxonomy   |
| wp_termmeta         |
| wp_terms            |
| wp_usermeta         |
| wp_users            |
+-----+

mysql>DESC wp_users;      <- wordpress ユーザテーブルの列情報を表示
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default        | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID         | bigint(20) unsigned | NO   | PRI | NULL           | auto_increment |
| user_login | varchar(60)         | NO   | MUL |                |                |
| user_pass  | varchar(255)        | NO   |     |                |                |
| user_nicename | varchar(50)        | NO   | MUL |                |                |
| user_email | varchar(100)        | NO   | MUL |                |                |
| user_url   | varchar(100)        | NO   |     |                |                |
| user_registered | datetime           | NO   |     | 0000-00-00 00:00:00 |                |
| user_activation_key | varchar(255)      | NO   |     |                |                |
| user_status | int(11)             | NO   |     | 0               |                |
| display_name | varchar(250)        | NO   |     |                |                |
+-----+-----+-----+-----+-----+-----+

mysql> select * from wp_users; <- wordpress ユーザテーブルの値を表示、ここで * は全ての列
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_registered | user_activation_key | user_status | display_name |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | ikob      | ***** | ikob         | foo@example.com |          | 2018-06-19 01:19:24 |                    | 0          | ikob         |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

mysql>
```

# 演習: Wordpress データベースを覗く

## Wordpress データベース / 投稿データ

```
mysql>DESC DESC wp_posts;          <- 投稿データベースの列情報を表示
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default        | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID         | bigint(20) unsigned | NO   | PRI | NULL           | auto_increment |
| post_author | bigint(20) unsigned | NO   | MUL | 0               |                |
| post_date  | datetime      | NO   |     | 0000-00-00 00:00:00 |                |
| post_date_gmt | datetime      | NO   |     | 0000-00-00 00:00:00 |                |
| post_content | longtext      | NO   |     | NULL           |                |
| post_title | text          | NO   |     | NULL           |                |
+-----+-----+-----+-----+-----+-----+

mysql> select ID, post_title from wp_posts;  <- ID, タイトル を表示してみる。
+-----+-----+
| ID | post_title |
+-----+-----+
| 5 | My First page |
| 6 | My First page |
+-----+-----+

mysql> select post_content from wp_posts where ID = <自身が作ったページの ID >;
+-----+
| post_content |
+-----+
| This is my first page |
+-----+

mysql> select post_content from wp_posts where post_content LIKE %自身のコンテンツに含まれる文字列%; <- ワイルドカードを使った検索、% は任意の文字列にマッチ
+-----+
| post_content |
+-----+
| This is my first page |
| This is my first page |
+-----+

mysql> exit
Bye
$
```

# 参考：ファイルベースのデータ

- 既定の設定で Linux のユーザ情報は /etc/passwd に以下の形式で書かれている。

```
name:password:UID:GID:GECOS:directory:shell
```

name : ユーザ名

password : パスワード

UID : ユーザ ID (整数)

GID : グループ ID (整数)

GECOS: ユーザに対するコメント(氏名など)

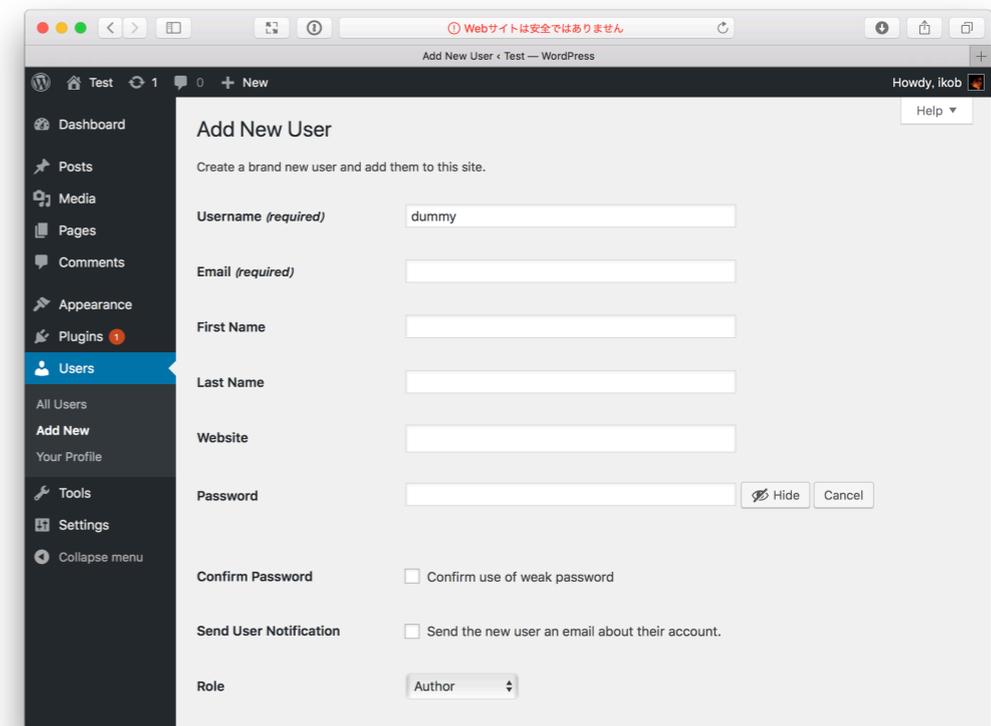
shell: ホームディレクトリ:ログイン時に実行されるコマンド

- cat コマンドでこれを表示すると、以下のように表示される。

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
~~~~~
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
dbus:x:81:81:System message bus:./sbin/nologin
ec2-user:x:500:500:EC2 Default User:/home/ec2-user:/bin/bash
apache:x:48:48:Apache:/var/www:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
$
```

# 課題 1 wordpress ユーザ追加とデータベース確認

- 前ページまでの演習で Wordpress データベースのユーザテーブルの内容を確認
- 以下の URL から Wordpress の管理画面にアクセス  
<http://<自身が install した wordpress>/wp-admin>
- Users→ Add New で近くの席の学生をユーザとして追加する。
  - Role は Author
  - Send User Notification は OFF
- Users→ Add New で近くの席の学生をユーザとして追加する。
- SQL コマンドでユーザが追加できていることを確認する。  
。SQL コマンドの出力結果をレポートする。



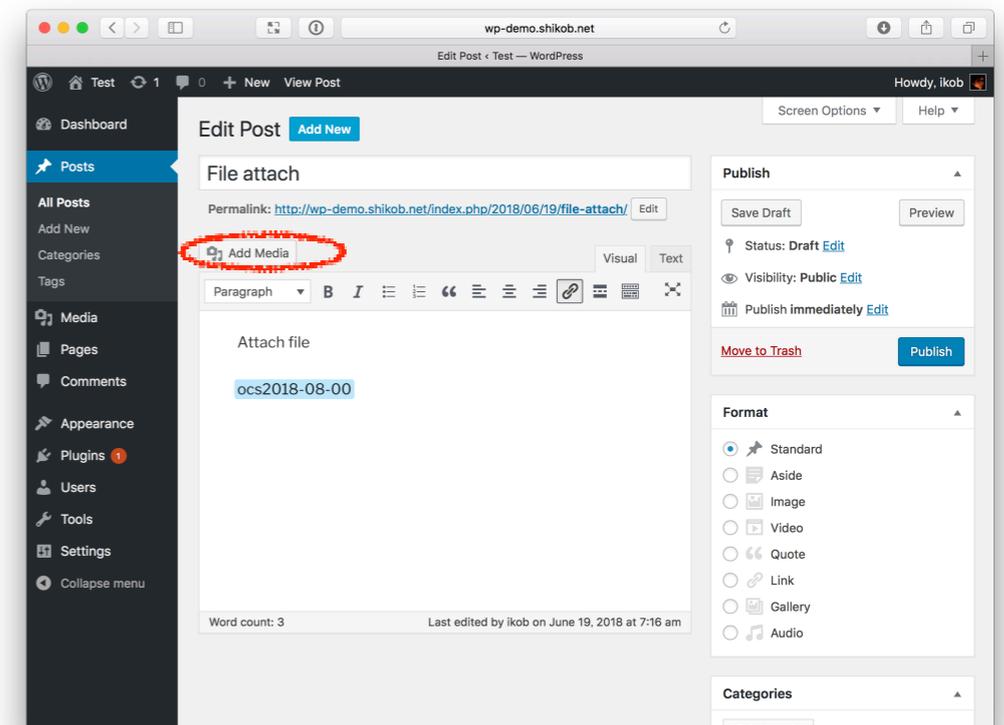
The WordPress software, GPLv2

# 課題 2 WordPress ファイルアップロードとデータベース確認

- chown で WordPress 関連ファイルの権限を変更する

```
$ cd /var/www  
$ sudo chown -R apache.apache *
```

- +New をクリック、新規ページ作成画面に遷移する。
- 作成画面で Add Media をクリック
- 適当な画像 or PDF ファイルをアップロードする。
- ページに Insert するかどうか聞かれるので Insert し、ページを Publish する。
- SQL コマンドで wp\_posts に新しいページ、アップロードされたファイルの URL(guid) が追加されていることを確認する。
- アップロードしたファイルの実体は別にある。WordPress サーバのどこに置かれるか？実態が別にあることの利点・欠点を考察せよ。



The WordPress software, GPLv2

# トランザクション処理

- DB に対する アプリケーションレベル の最小粒度 (atomic) の操作:
  - アプリケーションが求める整合性 (consistency) を達成するには複数の CRUD 操作が必要になることが多い。
    - 例えば、以下が求められる銀行口座においては:
      - 個別の口座残高はマイナスにならない
      - 銀行内送金では全預金の総額は変わらない
    - A から B への送金処理には、下記の処理全体が不可分となる
      - 1.A の残高を確認
      - 2.A の残高を更新(減らす)
      - 3.B の残高を更新(増やす)
    - 一部の操作が実行できない場合は、処理前の状態に戻される。

# Money Transfer from A to B

## 2. money transfer

Consistent

Not consistent

Consistent

Customer	Balance
A	4000

Customer	Balance
A	4000

Customer	Balance
A	5000

B	3000
---	------

B	2000
---	------

B	2000
---	------

2a. deposit to A

2a. withdrawal from B

# トランザクション処理

- 4つの特性(ACID)が求められる
  - Atomic (原子性 / 不可分性)
  - Consistency (整合性 / 一貫性)
  - Independency (独立性)
  - Durability (耐久性)