

モバイル プログラミング2

本日の予定

午前

- PHP開発例
 - BBS
- MySQL基礎
 - MySQLをさわってみる
 - データベース作成
 - データのインポート

午後

- SQL演習
 - データの選択
 - データの挿入
 - データの更新
 - データの削除

前回休んだ人

(°▽°)ｼ

PC借りた人

(°▽°)ｼ

まずは追いつこう

1. GitHubの資料見てね

<http://github.com/katsube/neec>

2. 環境構築

3. 環境構築で困ったらすぐに聞いてください

アンケート (出席カード)

アンケート

1. 提出 = 出席 (授業終了までに限る)
未提出 = 欠席
2. 学籍番号、名前が確認できない場合は**欠席**
3. わからない場合は、どこが理解できなかったか記入

アンケート

1. 「白紙提出」「授業を聞いていたと判断できない」場合は個別にヒアリングを行います。

- よほどのことがなければ呼び出されません
- 大人としての自覚を持って授業に望んで下さい。

2. 一人では解決できないことがある場合、自分から聞きにくるよう。

・返却を希望する場合

モバイルプログラミング 2 出席カード↓

2016/10/31↓

学籍番号 [] 氏名 []↓

☐ 返却を希望する↓

問題 1. PHP の特徴を各項目毎にまとめてみましょう↓

実行方法 ※大きく 2 種類↓

次回～次々回の授業で返却します

アンケート

- 難易度に○をつける

モバイルプログラミング 2 出席カード

2016/11/07

学籍番号 []

氏名 []

難易度 (優しい ・ 最適 ・ 難しい)

☐ 返却を希望する

○をつけてください。

様子を見て難易度を調整します。

前回のアンケートに 答えるコーナー

1. パワポ等に答えを書いてほしい
2. HTMLもGitHubにあげてほしい
3. == と === の違い
4. 部屋が暑い
5. FFXIV楽しいです

PHPの ==, === の違い

10 == "10"

0 == "a"

"1" == "01"

"10" == "1e1"

100 == "1e2"

PHP開発例



GitHubのサンプル参照

- BBS

<https://git.io/vXbcr>

※短縮URLの飛び先は以下

<https://github.com/katsube/neec/tree/master/sample/BBS>

次回の開発デーは 12月19日

- それまでに必ず「チャット」を完成させて授業に挑んでください。
 - この日に提出した物は成績に反映します。
- Chat開発に関して質問があれば、Issue上で受け付けます。

MySQL基礎



※はじめる前に

- DBとは、MySQLとは何か、については後日説明します。
 - まずは触ってもらおうと思います。
- ここではデータの管理を行うサーバの一種だと考えてください。

起動を確認しよう 1

```
$ ps aux
```

```
mysql      1629  0.0 18.2 1119204 185648 ?        S  
17:26    0:00 /usr/sbin/mysqld --daemonize --pid-  
file=/var/run/mysqld/mysqd.pid
```

起動を確認しよう 2

```
$ ps aux | grep mysql
```

```
mysql      1629  0.0 18.2 1119204 185648 ?        Ss  
17:26   0:00 /usr/sbin/mysqld --daemonize --pid-  
file=/var/run/mysqld/mysqd.pid
```

- 検索結果をgrepで絞り込むと楽ちんですね。

起動を確認しよう 3

```
$ ps aux | less
```

- コマンドの実行結果を less で受け取って眺めることもできます。
 - 矢印キー、スペースで操作
 - 「q」キーを押すと終了

起動する

```
$ sudo systemctl start mysqld.service
```

- 起動していなかった場合は、上記コマンドで起動します。
- Webサーバ(Apache)を起動した際とほぼ同じことがわかると思います。

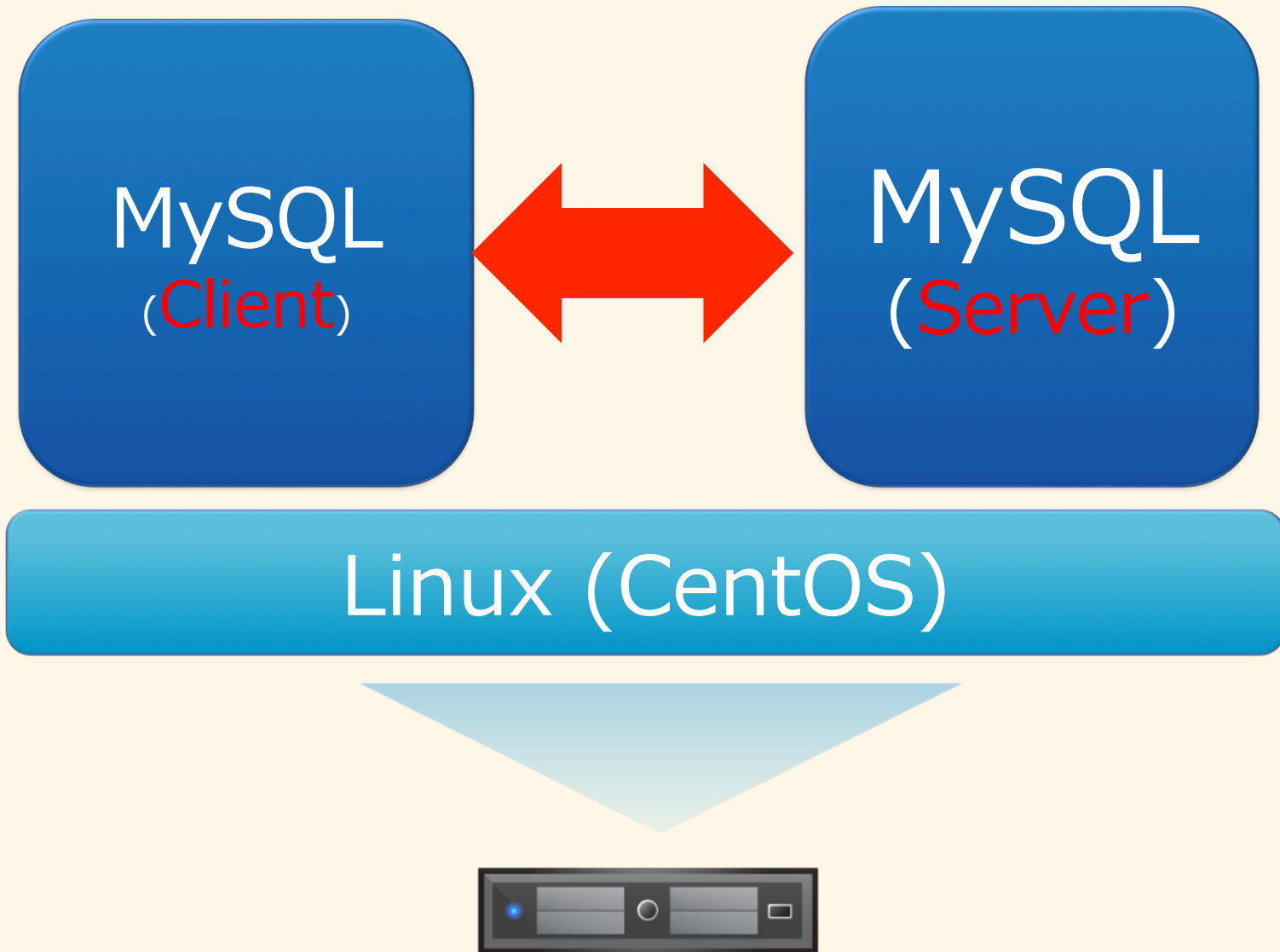
イメージ

MySQL
(Server)

Linux (CentOS)



クライアントから操作する



クライアントを確認

```
$ mysql --version
```

```
mysql Ver 14.14 Distrib 5.7.15, for Linux  
(x86_64) using EditLine wrapper
```

- ハイフンは2つ

MySQLにログイン

```
$ mysql -u root -p
```

Enter password:

- **-u** ログインするユーザー名を指定
- **-p** パスワードを入力する
- パスワードは「**H@chiouji1**」

ここからMySQLの世界

```
mysql>
```

- ここでデータベースへの操作を行います。
- みなさんが今まで使っていたLinuxとは全く別の世界になりました。
- ls や cd などのコマンドは使えません。

イメージ

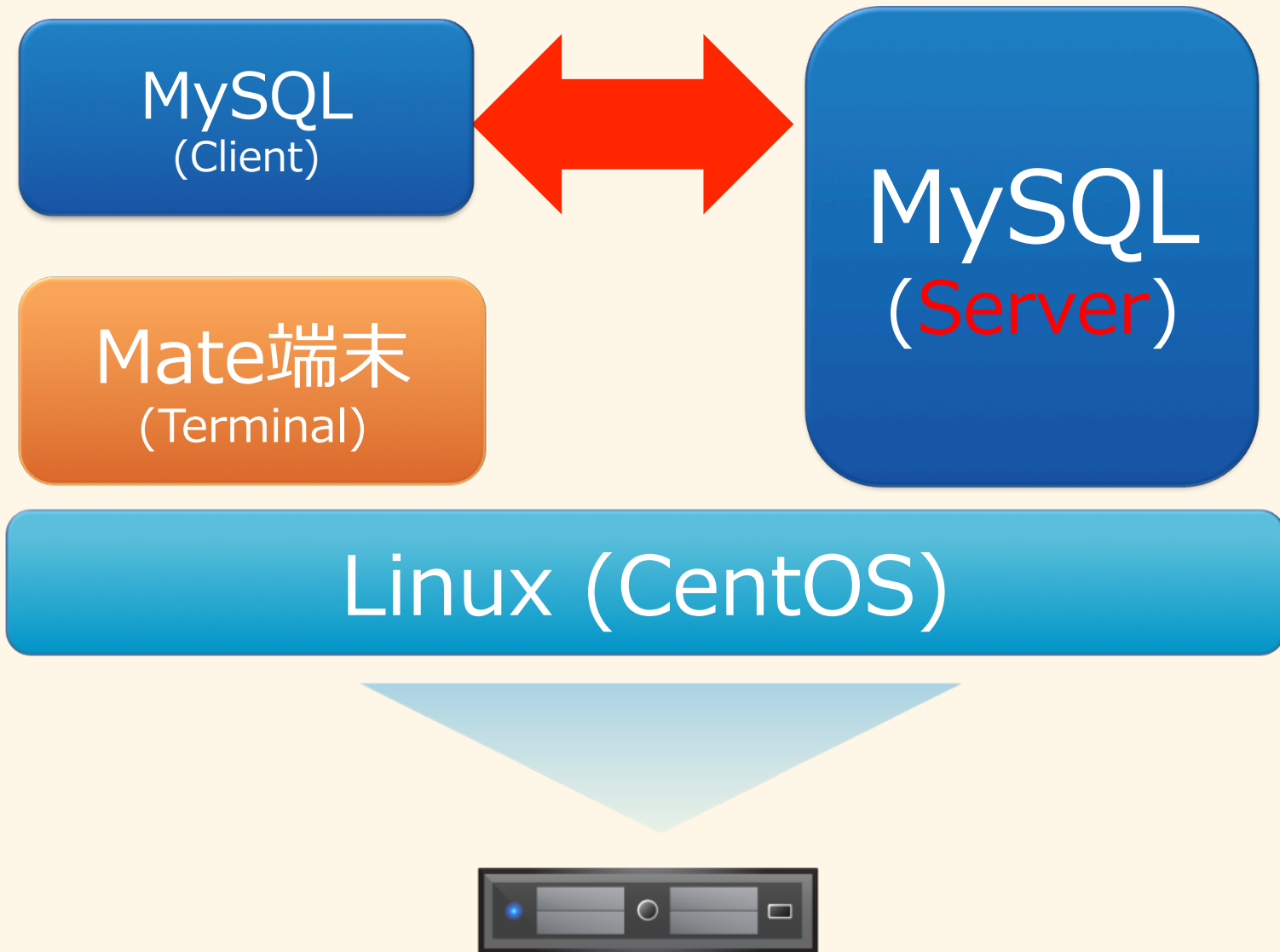
MySQL
(Client)

Mate端末
(Terminal)

Linux (CentOS)



イメージ



データベースを表示する

```
mysql> show databases;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+
```

```
4 rows in set (0.01 sec)
```

- MySQLの機能

データベースを作成する

```
mysql> create database rpgdb;
```

- SQL構文
- `show databases` で作成されたことを確認してみましょう。

使用するデータベースを指定

```
mysql> use rpgdb;
```

```
Database changed
```

- MySQLの機能
- これから操作するデータベースを指定します。

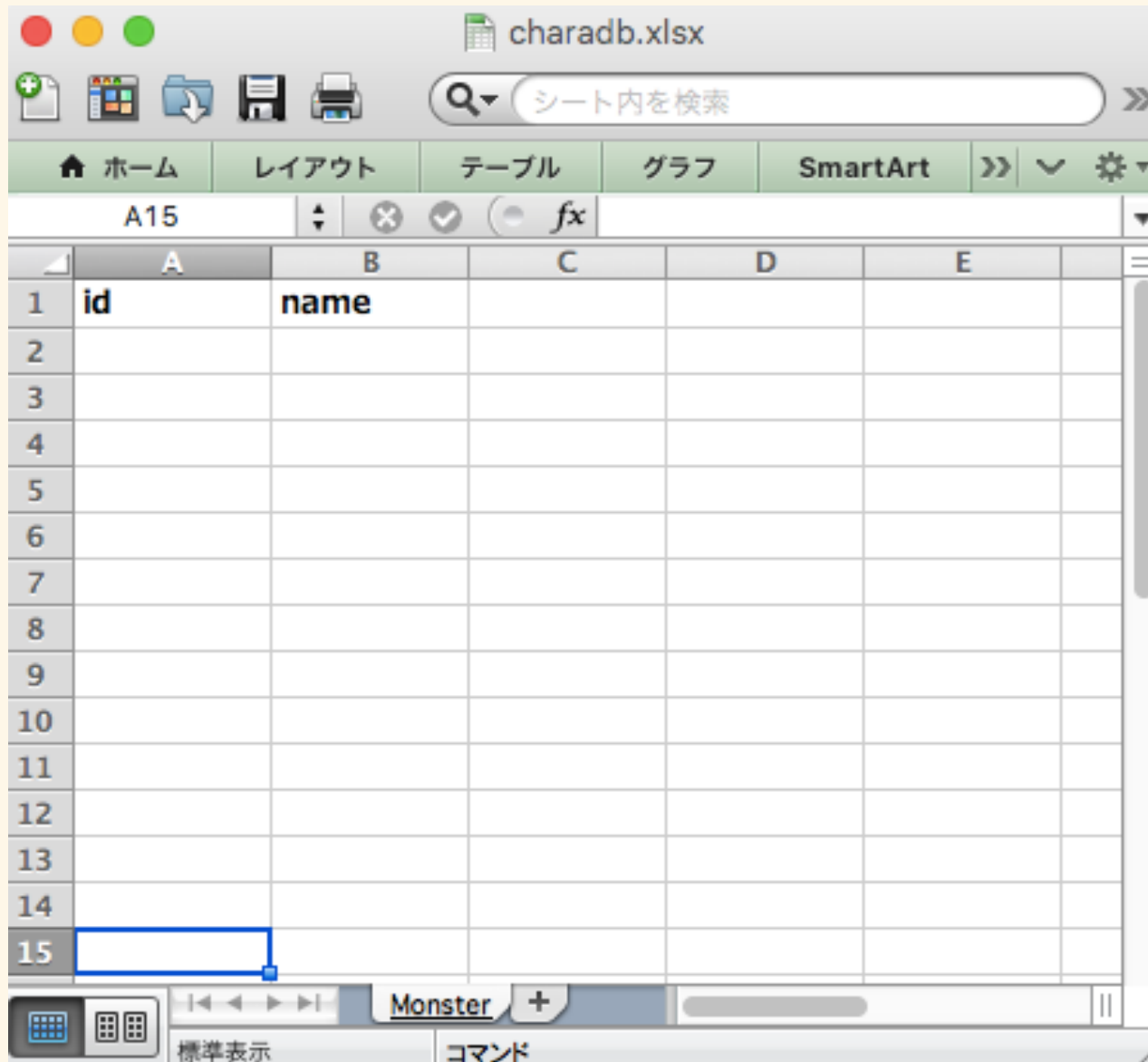
テーブルを作成する

```
mysql> create table Monster(  
-> id int ,  
-> name varchar(8)  
-> );
```

Query OK, 0 rows affected (0.24 sec)

- SQL構文
- id と name の2つの列を持つテーブル「Monster」を作成します。

テーブルってなに？



テーブルってなに？

カラム
(列)

charadb.xlsx データベース

ホーム レイアウト テーブル グラフ SmartArt

	A	B	C	D	E
1	id	name			
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

Monster +

標準表示 コマンド

レコード
(行)

テーブル

テーブルのカラムには型がある

- 数値
 - 整数 (TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT)
 - ※ INT UNSIGNED とすると符号なしにできる
 - 小数点 (FLOAT, DOUBLE, DECIMAL, NUMERIC)
 - ビット (BIT)
- 日付と時刻 (YEAR, DATE, TIME, DATETIME, TIMESTAMP)
- 文字列 (CHAR, VARCHAR, TEXT)
- バイナリ (BINARY, BLOB)

※詳細は公式ドキュメントを参照

<https://dev.mysql.com/doc/refman/5.6/ja/data-types.html>

途中で打ち間違いに気がついたら？

```
mysql> create table Monster(
```

```
->  it int ,  \(^o^)/
```

```
->  ¥c
```

```
mysql>
```

- MySQLの機能
- 「¥c」と打つことで途中で入力をやめて、元の画面に戻ることができます。

テーブルの一覧を表示

```
mysql> show tables;
```

```
+-----+  
| Tables_in_charadb |  
+-----+  
| Monster            |  
+-----+
```

- MySQLの機能

テーブルの構造を表示

```
mysql> desc Monster;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(8)	YES		NULL	

- MySQLの機能

テーブルにレコードを挿入

```
mysql> insert into Monster  
-> values(1, 'dragon');
```

Query OK, 1 row affected (0.00 sec)

- SQL構文
- Monsterテーブルにデータを挿入。

テーブルにレコードを挿入

A15		✖		✔		fx	
	A	B	C				
1	id	name					
2	1	dragon					
3							
4							
5							
6							
7							

定義に反したデータを 挿入すると…？

```
mysql> insert into Monster  
-> values('foo', 'dragon');
```

```
mysql> insert into Monster  
-> values(1, 'fooooooooooooo');
```

- 実際にやってみましょう。

テーブルのデータを表示

```
mysql>select id, name from Monster;
```

```
+-----+-----+  
| id   | name  |  
+-----+-----+  
|  1   | Dragon|  
+-----+-----+
```

```
1 row in set (0.00 sec)
```

- SQL構文
- Monsterテーブルのデータをすべて表示。

テーブルのデータを表示

```
mysql>select * from Monster;
```

```
+-----+-----+  
| id   | name  |  
+-----+-----+  
|  1   | Dragon |  
+-----+-----+  
1 row in set (0.00 sec)
```

- SQL構文
- すべての列を対象とする場合、アスタリスク(*)で代用することができる。

MySQLから抜ける

```
mysql> ¥q
```

```
$
```

- MySQLの機能
- ¥q または **exit** と入力することでMySQLから抜けて、元のLinuxの環境に戻ることができます。

データベースとテーブル

- EXCELをイメージ

- データベース = ファイル
- テーブル = シート
- カラム = 列
- レコード = 行

データベースの操作

- **作成**

- create database [name];

- **削除**

- drop database [name];

- **一覧の表示**

- show databases;

- **使用するデータベースの指定**

- use [name];

テーブルの操作

- **作成**

- create table [name] (実際の定義);

- **削除**

- drop table [name];

- **一覧の表示**

- show tables;

- **構造の表示**

- desc [name];

データの操作

- **挿入**

- insert into [name] (列名)
values(値);

- **表示(抽出)**

- select * from [name];

データを インポートする



テストデータを作成しよう

- ダウンロードした「rpgdb.xlsx」をCentOSで開き、id列11～30の項目を埋めましょう。

アプリケーション 場所 システム 11月13日 (日) 15:30

charadb.xlsx - LibreOffice Calc

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)

VL ゴシック 12

B13

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	name	HP	MP	AT	DF		len				
2	int	varchar(32)	int 1~9999	int 1~255	int 1~255	int 1~255						
3	1	スライム	10	1	2	1		8				
4	2	ドラキー	12	2	3	2		8				
5	3	いっかくウサギ	13	2	6	3		14				
6	4	くざったしたい	30	3	8	5		14				
7	5	まどうし	20	30	3	3		8				
8	6	さまようよろい	50	10	12	10		14				
9	7	キメラ	80	32	26	22		6				
10	8	ばくだんいわ	110	30	28	32		12				
11	9	マドハンド	90	26	23	23		10				
12	10	あくましんかん	120	60	12	16		14				
13	11							0				
14	12							0				
15	13							0				
16	14							0				
17	15							0				
18	16							0				
19	17							0				
20	18							0				
21	19							0				
22	20							0				
23	21							0				
24	22							0				
25	23							0				
26	24							0				
27	25							0				
28	26							0				
29	27							0				
30	28							0				
31	29							0				
32	30							0				

INPUT / EXPORT

Sheet 1 / 2

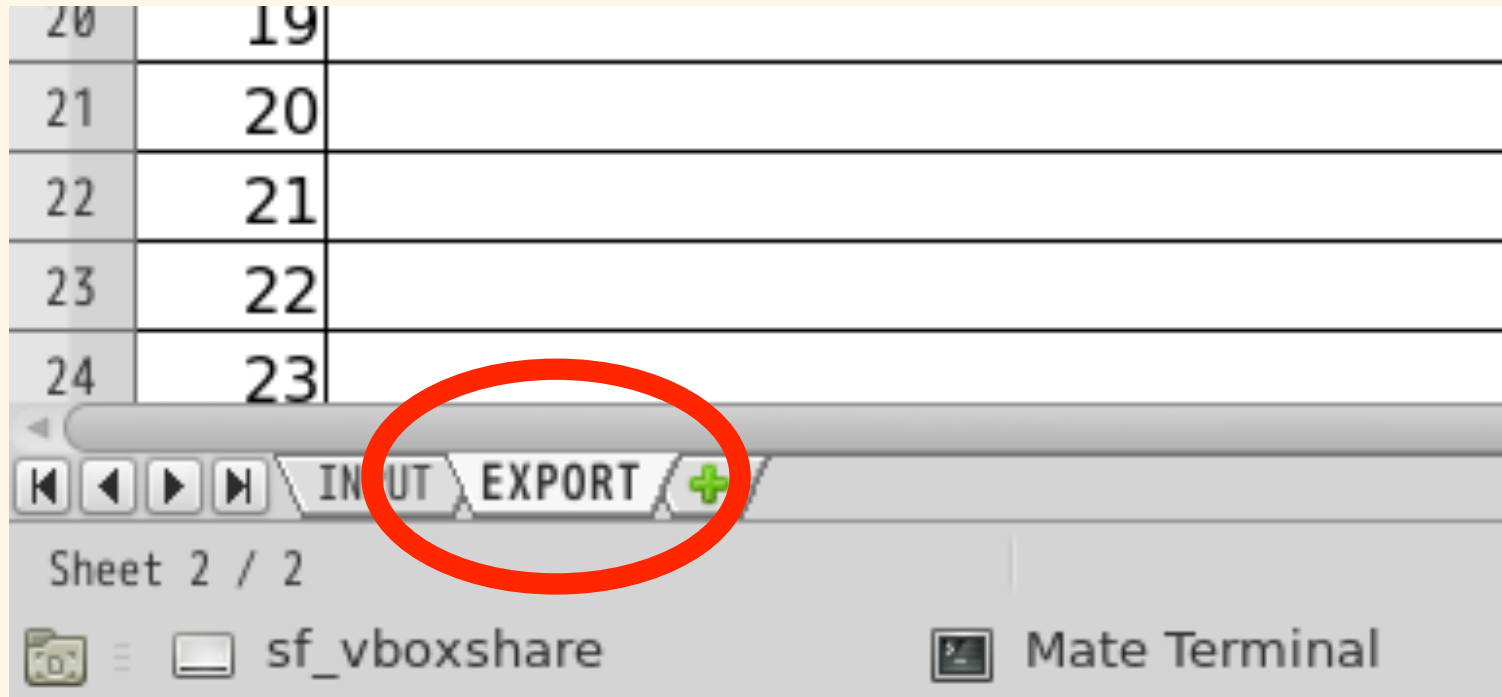
PageStyle_INPUT

合計=0

sf_vboxshare Mate Terminal Mate Terminal charadb.xlsx - LibreOffice Calc

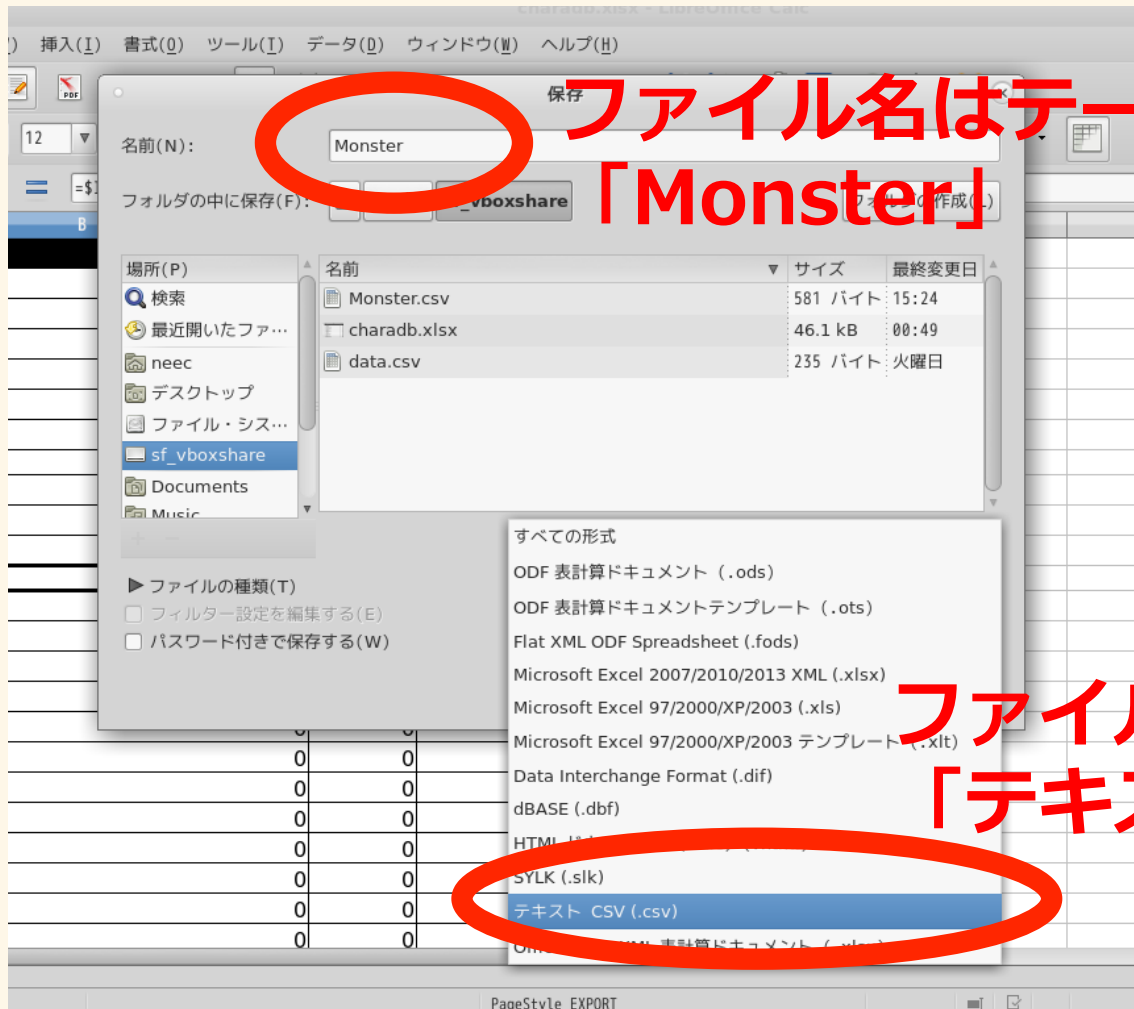
CSV形式でエクスポート 1

- 入力が完了したら、「EXPORT」シートに切り替えます。



CSV形式でエクスポート 2

- メニュー「ファイル」→「名前をつけて保存」を選択。

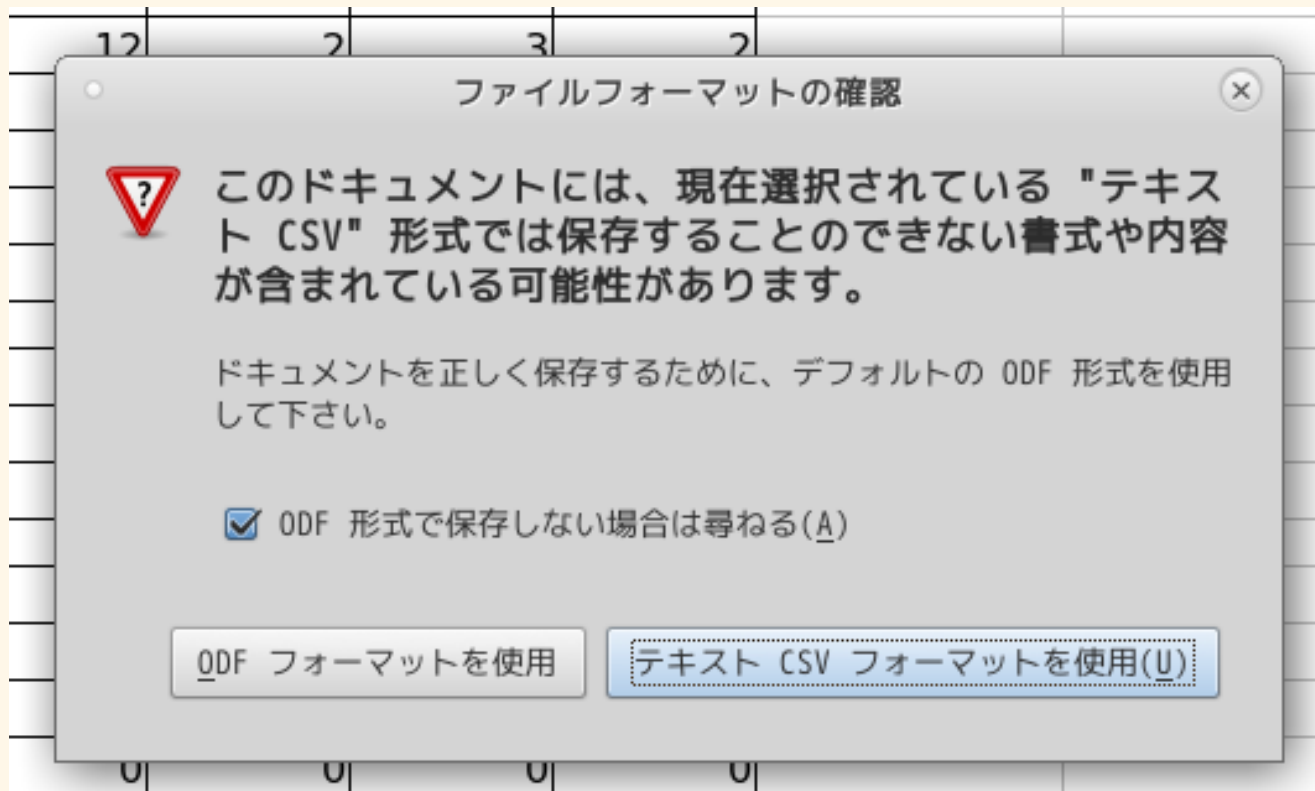


ファイル名はテーブル名と同じ
「Monster」

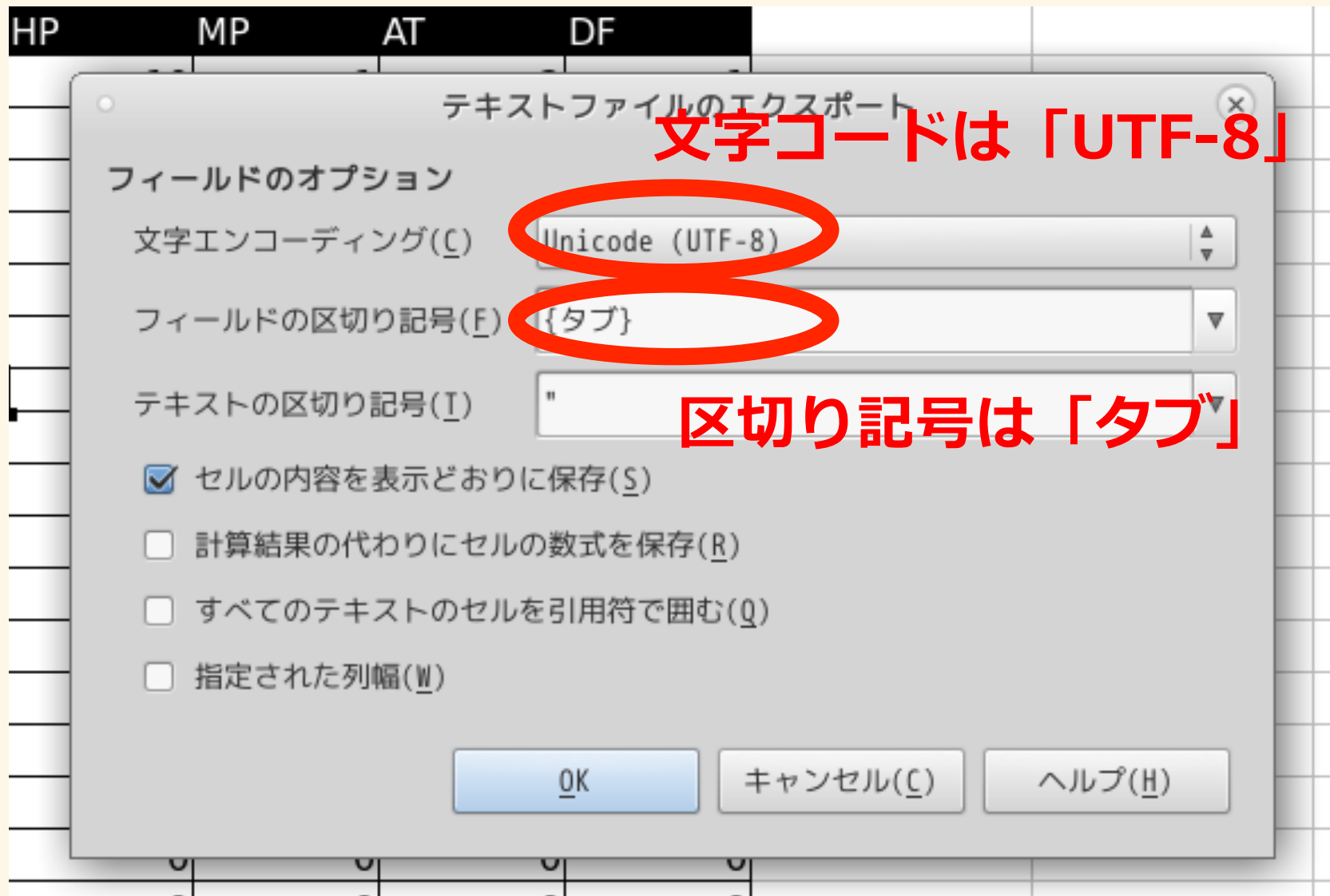
ファイル形式は
「テキスト CSV」を選択

CSV形式でエクスポート 3

- ダイアログは「テキストCSVフォーマットを使用」をクリック



CSV形式でエクスポート 4



CSV形式でエクスポート 5

アプリケーション 場所 システム

Monster.csv (sf_vboxshare /media/sf_vboxshare) - Pluma

ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

開く 保存

Monster.csv x

id	name	HP	MP	AT	DF
2	ドラキー	12	2	3	2
3	いっかくウサギ	13	2	6	3
4	くさったたい	30	3	8	5
5	まどうし	20	30	3	3
6	さまようよろい	50	10	12	10
7	キメラ	80	32	26	22
8	ばくだんいわ	110	30	28	32
9	マドハンド	90	26	23	23
10	あくましんかん	120	60	12	16
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0

なし ▾ タブの幅:: 4 ▾ (26行、22列) [挿入 ...]

エクスポートした
ファイルを開き、
1行目を削除して保存

sf_vboxshare

アイコン表示

Monster.csv

2016-
2016-

discharge.log

テーブルを削除する

```
mysql> drop table Monster;
```

- SQL構文。
- 先程作成したテーブルとは構造が違うのでいったん削除します。

テーブルを作成する

```
mysql> create table Monster (  
-> id int,  
-> name varchar(32),  
-> HP int,  
-> MP int,  
-> AT int,  
-> DF int  
-> );
```

MySQLへインポート 1

```
$ mysqlimport -u root -p --local ¥  
> chardb Monster.csv
```

- MySQLの機能
- 指定したDBへファイル内容をインポートします。
- コマンドが長すぎる場合は、 ¥ を打つと改行ができます。
※ ¥ を打たずにエンターキーを押すと、コマンドが実行されてしまうので注意

MySQLヘインポート 2

```
$ mysql -r root -p
```

Enter password:

```
mysql> use rpgdb;
```

```
mysql> select * from Monster;
```

- 実際にデータがインポートされたかどうか、先程と同じ手順で確認してみましょう。