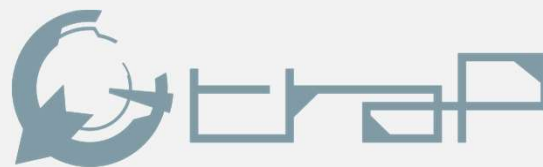




# データベース入門

Webエンジニアになろう講習会 第4回



東京工業大学  
デジタル創作同好会



# 前回のおさらい

---



東京工業大学  
デジタル創作同好会





# HTTPって何だったっけ？

「お願い（リクエスト）」と「お返事（レスポンス）」でやりとりをする仕組み  
リソースを読み取ったり作ったり書き換えたり消したりするやりとりができる

「この動画投稿したいです」「200（投稿しました）」

「このユーザーのデータください」「404（そんな人はいません）」

「このメッセージ消したいです」「403（権限がありません）」





# リクエスト

---

「これください」をもっと詳しく



東京工業大学  
デジタル創作同好会





# リクエストの構造





# レスポンス

---

「これあげます」をもっと詳しく



東京工業大学  
デジタル創作同好会





# レスポンスの構造

バージョン番号

ステータス

HTTP/1.1 200 OK

Age: 445820

Cache-Control: max-age=604800

...

Content-Length: 1256

レスポンス  
ヘッダー

レスポンス  
ボディ

<!doctype html>

<html>

...





# データの伝送

---

HTTPが依存するレイヤー



東京工業大学  
デジタル創作同好会







# TCP

Transmission Control Protocol

伝送（トランスミッション）をコントロールするプロトコル

どんなデータの伝送でも使える

メールの伝送でも使われるし、コンピューターの遠隔制御でも使われる

TCPを扱うプログラムは、送ったデータがどのように扱われるかは気にしない

逆にHTTPを扱うプログラムは、どのようにデータが送られるかを気にしない



# IP

TCPは、データが相手に届くことを前提にしている

自分と相手が直接物理的につながっていなくても、良い感じに機械を経由してデータが届いてほしい

他の機械の80番ポート宛てのデータが届いたりはしない

IPも、宛先などが含まれたリクエストヘッダを前に付けて伝送を管理する

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	バージョン				ヘッダ長				サービス種別								全長															
32	識別子																フラグ		断片位置													
64	生存時間								プロトコル								チェックサム															
96	送信元アドレス																															
128	宛先アドレス																															





# 今日はデータベース

---

Webサービスの”起点”ともいえる存在



東京工業大学  
デジタル創作同好会





# Webサービス概観

(物理) サーバー

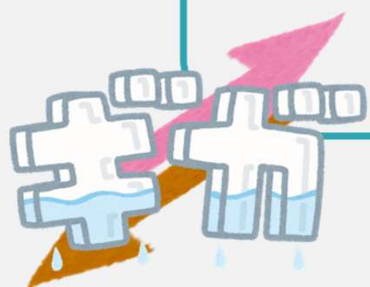


データベース  
データの保存



サーバー  
アプリケーション  
メインのサービス

クライアント  
データの表示・操作



ネットワーク





# Webサービス概観

(物理) サーバー

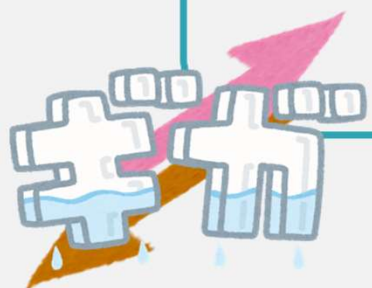


データベース  
データの保存



サーバー  
アプリケーション  
メインのサービス

クライアント  
データの表示・操作



ネットワーク





# DB(DataBase)とは

- ▶ DataBase → データ基地 → データの蓄積場所
- ▶ 図書館もDB
- ▶ 効率よく検索やデータ挿入が可能
  - ▶ “SQL”と呼ばれる言語を用いる(実習編内容)
- ▶ SQLの例→ `SELECT Name, Population FROM city;`
  - ▶ `city`テーブルから`Name`,`Population`の2カラムを選択
    - ▶ テーブル,カラムの意味は後述





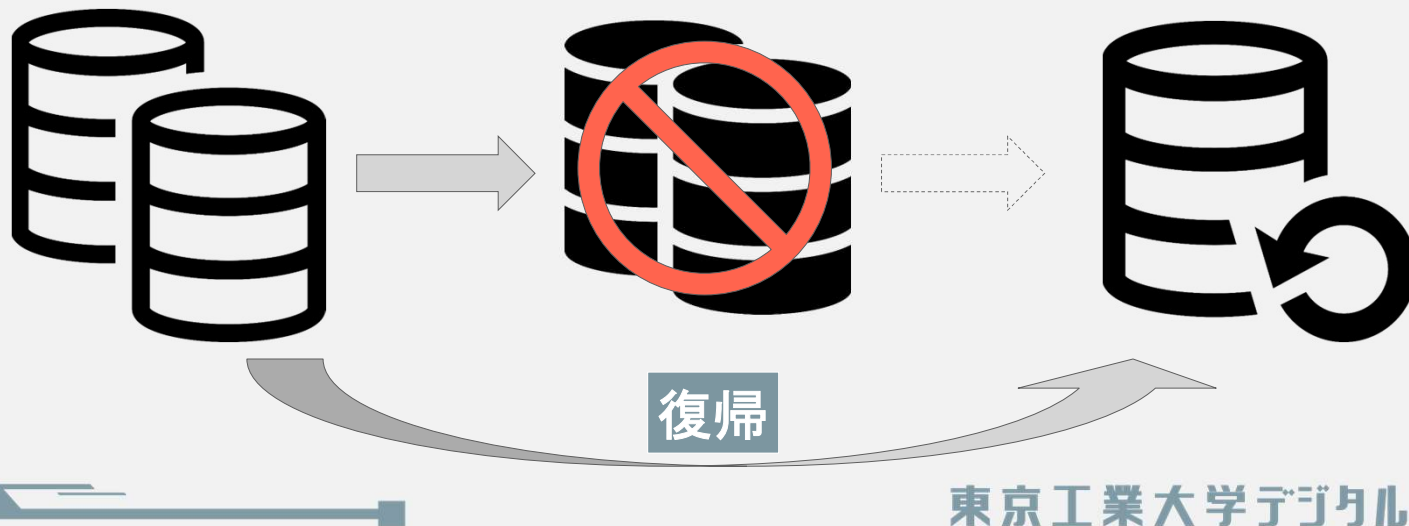
# DBMS(DataBase Management System)

- ▶ データベースの情報管理
  - ▶ “データの整合性”を保つ
- ▶ 図書館でいうなら司書



# データの整合性

- ▶ データが失われない
- ▶ システム障害が起こっても障害前に復帰可能
- ▶ 永続性と呼ばれる





# データの整合性

- ▶ 入力できるデータにルールを決められる
  - ▶ “正の数のみ” といったルールを定められる
  - ▶ 一貫性と呼ばれる
- ▶ 他に不可分性,独立性という性質がある
  - ▶ これら4つの整合性の要素を **ACID特性** という
- ▶ ACID特性を満たすようDBMSが機能している





# RDBMS(Relational DataBase Management System)

- ▶ 表形式でデータを格納(表を`テーブル`という)
- ▶ 管理する情報の種類を列にする (カラム)
- ▶ 1つのデータを1行で管理(レコード)
- ▶ 情報の関連付けがしやすいなど柔軟に扱える

カラム



住民情報テーブル

レコード

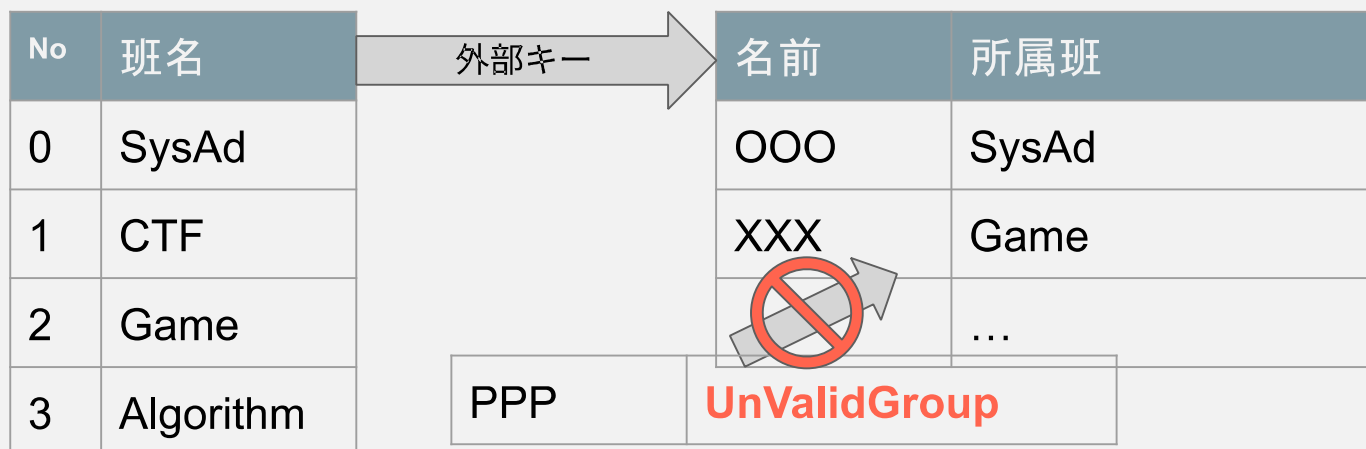


ID	名前	年齢	住所
001	田中	24	東京都大...
002	...		
003			



# RDBMS

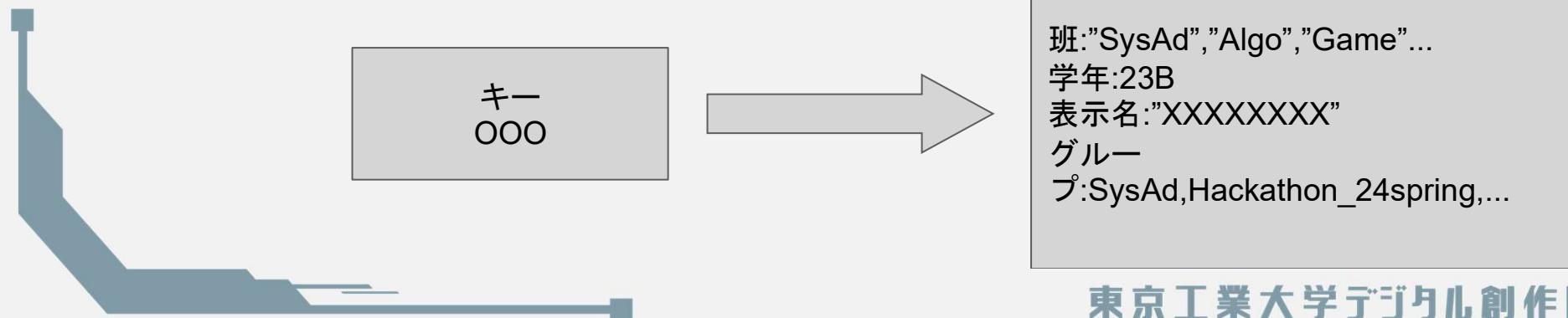
- ▶ 他のテーブルと値を関連させることができる
  - ▶ “あるテーブルの値のみ入力可能な項目”を設定可能
  - ▶ 例) 部員テーブルの”所属班”には”班”テーブルの値しか挿入できない
  - ▶ ここで”班テーブル”は部員テーブルの”外部キー”と呼ばれる






















# NoSQL

- Not Only SQL の略
- 実際にはRDBMS以外のDBMS全般
- 各DBMS毎様々なデータ格納方法で高パフォーマンスを実現
  - データの取り扱いに制限があったり取り出しの工夫が必要

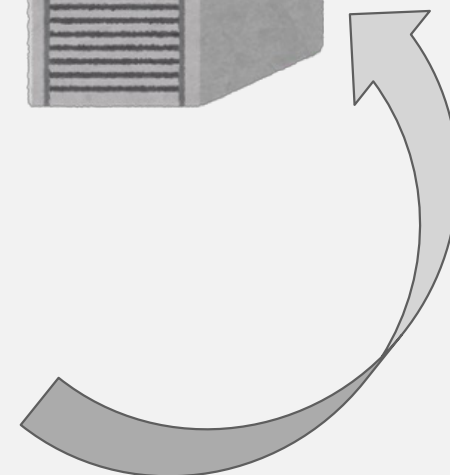


# 様々なDBMS

	SQL	NoSQL
汎用	    	       
特化	 	 

# 環境変数の話

- ▶ 別サーバ(アプリケーション)へのログイン
  - ▶ ユーザ名など様々な認証情報が必要
  - ▶ パスワードなど機密性が必要な情報も
  - ▶ プログラムでログイン:どこに記録するか
- ▶ 環境変数
  - ▶ プロセス間で共有される変数
  - ▶ 機密情報もここに書くことが多い





# 環境変数の話

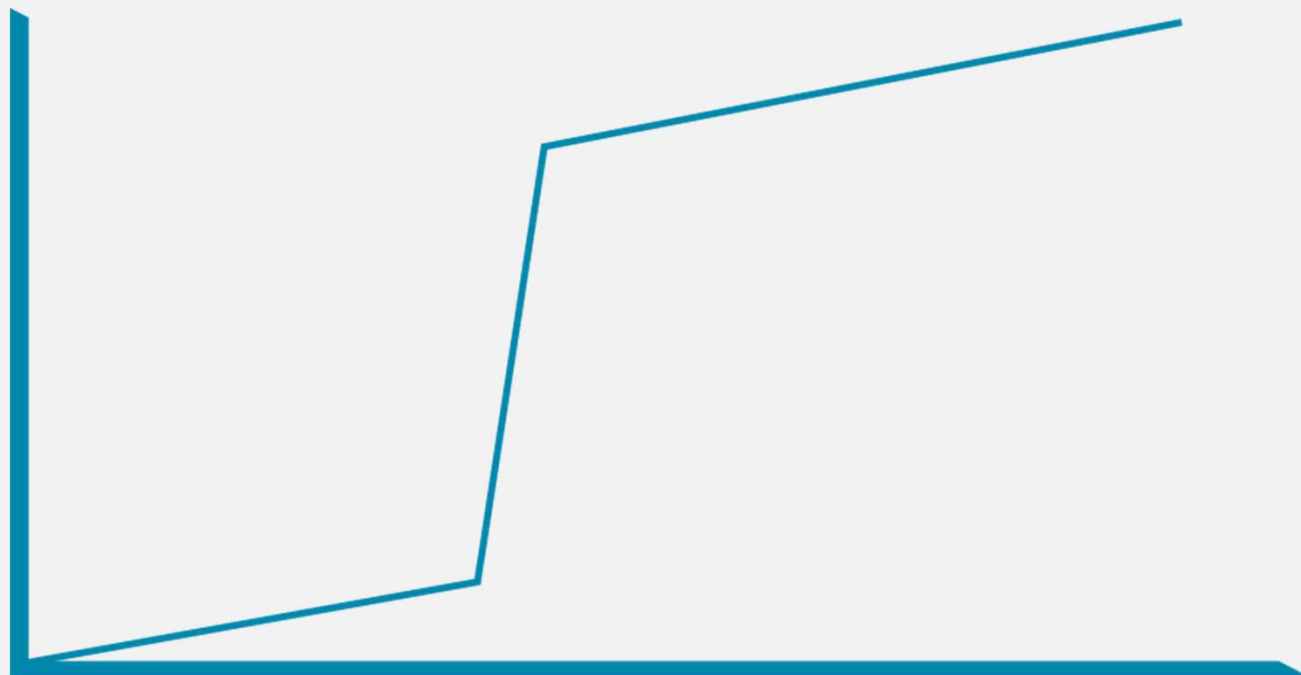
- ▶ パスワードなどの機密情報をプログラムにベタ打ちしない
  - ▶ Gitを用いる時には誤ってpush”されない”ように注意
- ▶ .gitignore
  - ▶ Gitで管理しないファイルを設定できる
  - ▶ 機密情報を記したファイルは設定しておくこと
  - ▶ 開発環境に応じたテンプレートもある





# Webエンジニアになるにあたって

作れるもの



知識・経験

東京工業大学デジタル創作同好会

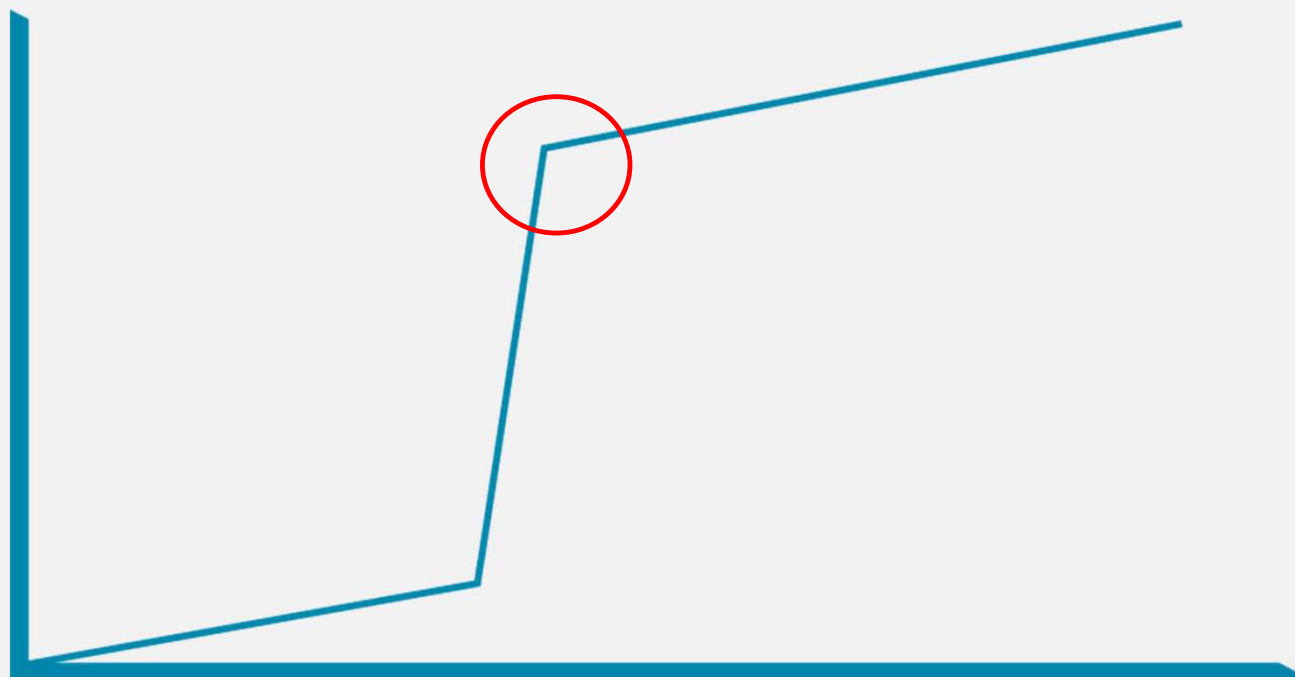






# Webエンジニアになるにあたって

作れるもの



知識・経験

東京工業大学デジタル創作同好会

