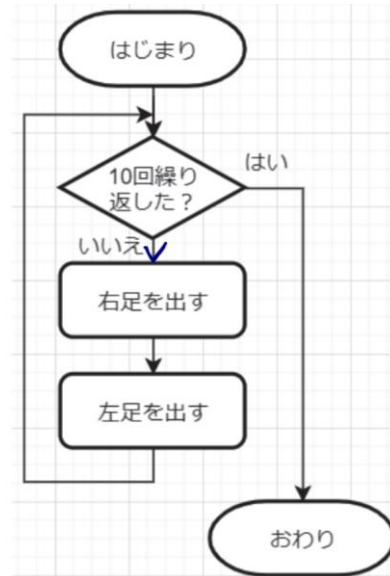
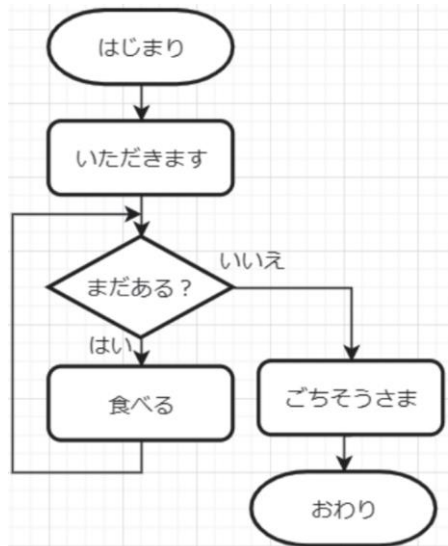


# 自己紹介

- 兼宗 進 (かねむねすすむ)
- 大阪電気通信大学 電子機械工学科教授、副学長
- 専門はプログラミング言語、情報科学教育
  - プログラミング教育の教育法や教材を公開
    - ドリトル、CSアンプラグド、サクセス、ConnectDB

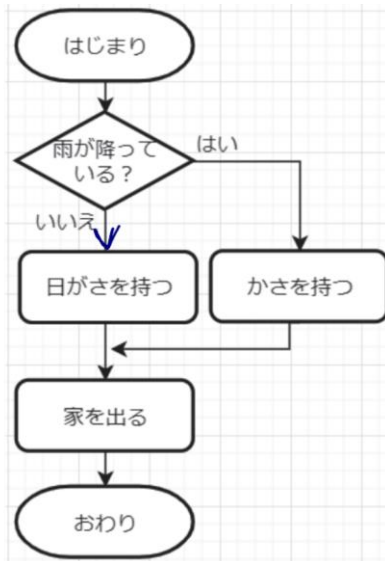
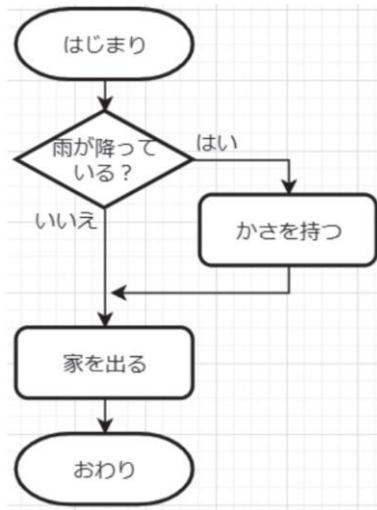
# プログラミングの考え方(反復)

- コンピュータは繰り返しが得意
- 正確に何度でも繰り返す



# プログラミングの考え方(分岐)

- 条件によって処理を分けたい
- 条件分岐はひし形

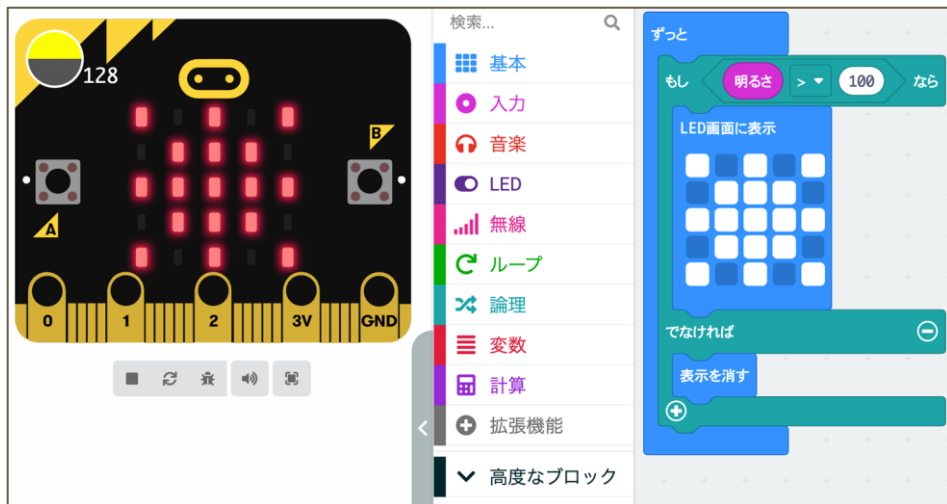


# 小中学校のプログラミング

- 小学校の教科書**17冊**を分析
  - 構文要素：変数(**1**),分岐(**3**),論理(**2**),反復(**6**)
    - 反復は半数が扱っているが、他の要素はほとんどない
  - 組合せ：反復+分岐(**3**),反復+反復(**2**)
    - 「反復+分岐」「反復+反復」とともにほとんどない
- 中学校の教科書**3冊**を分析
  - 構文要素：変数(**1**),分岐(**3**),論理(**2**),反復(**3**),配列(**1**)
  - 組合せ：反復+分岐(**3**),反復+反復(**0**),分岐+分岐(**3**)
    - 変数の扱いが弱い。「反復+分岐」「分岐+分岐」は体験している

# 小中学校の計測制御系のプログラミング

- 「センサ計測→条件分岐→出力（LED/モータ）」を繰り返すことが多い
  - 小学校理科： 光センサでライトをオンオフ
  - 中学校計測制御： 各種センサでライトやモータを制御
- ループの中に**if**がある（純粹な無限ループであり**break**はない）
- センサ値を直接比較する。変数を使わない！



```
loop:
  if センサ値() < 100:
    led(ON)
  else:
    led(OFF)
```

※micro:bitを利用 (<https://makecode.microbit.org/>)

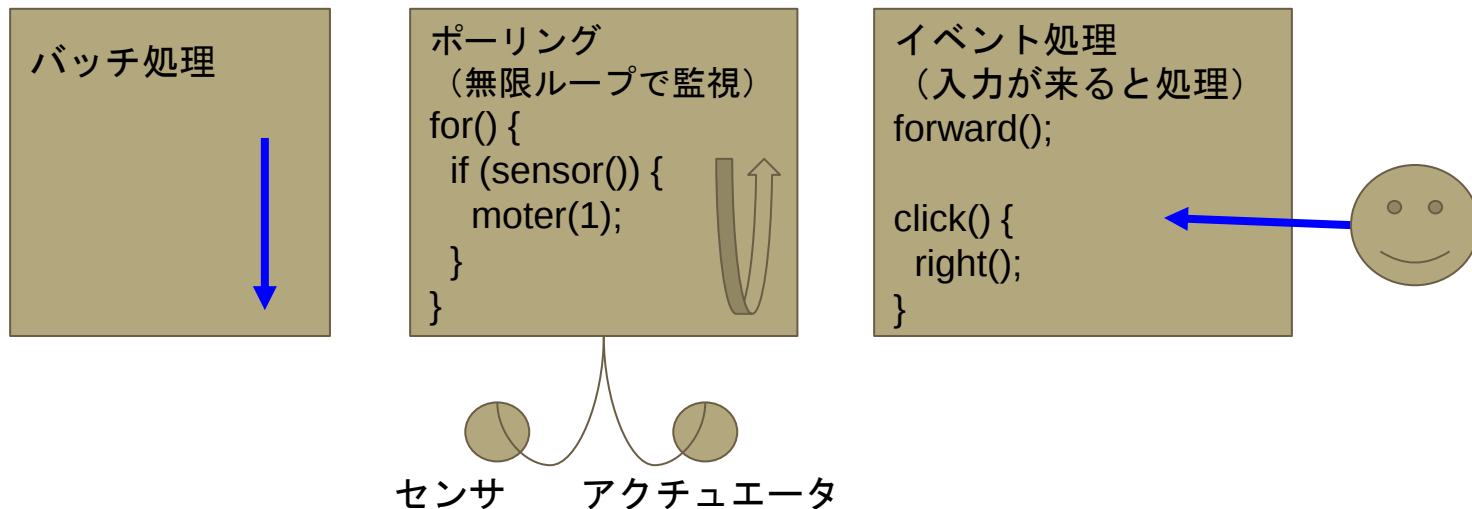
# 無限ループは空気のような存在？

- 「1秒ごとに信号を見て、赤の間は止まり青を見たら渡る、を繰り返す」を  
「信号が青になったら渡りましょう」と教えられるのであれば、
- 「1秒ごとにセンサを見て、明るければ消して、暗ければ点けるを繰り返す」を  
「暗くなったら点けましょう」と教えることも可能？
- 無限ループにより、「反復＋分岐」という複雑な構造をシンプルな形で理解できる（工夫点）



# 外部入力処理

- 外部（人、デバイス、他のコンピュータ）とやり取りするときのモデル
- ポーリングとイベント処理が基本



# 高等学校のプログラミング

- 情報Ⅰ（必修）の教科書**11冊**を分析
  - 構文要素：変数(**11**),分岐(**11**),論理(5),反復(**10**),配列(**10**),関数(**10**)
  - 組合せ：反復+分岐(**9**),反復+反復(**5**),分岐+分岐(**4**)
- 情報Ⅱ（選択）の教科書**3冊**を分析
  - 構文要素：分岐(3),論理(1),反復(2),配列/辞書(**3**),関数(**3**),クラス(2)
  - 組合せ：反復+分岐(**2**),反復+反復(**2**),分岐+分岐(**2**)

## 高等学校「情報Ⅰ」の全国での学習状況

- オンライン学習環境「Bit Arrow<sup>(\*)</sup>」の利用状況
- 情報Ⅰ初年度(2022年度)の秋冬に大きく増加
- 言語はPythonなど

(\*)大阪電気通信大学/東京農工大学/明星大学で共同開発

<https://bitarrow.eplang.jp>

### オンラインプログラミング学習環境「Bit Arrow」

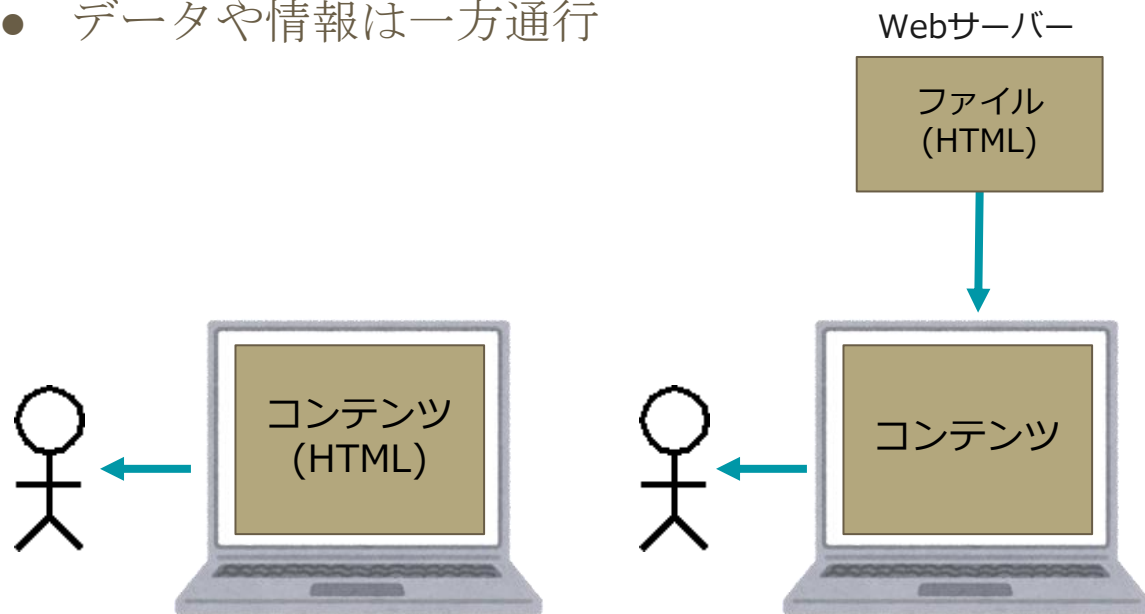
2022年度は年間17万人、11月は週あたり1万人以上が利用



# ネットワークを利用した 双方向性のあるコンテンツのプログラミング

# 従来のコンテンツ(1)

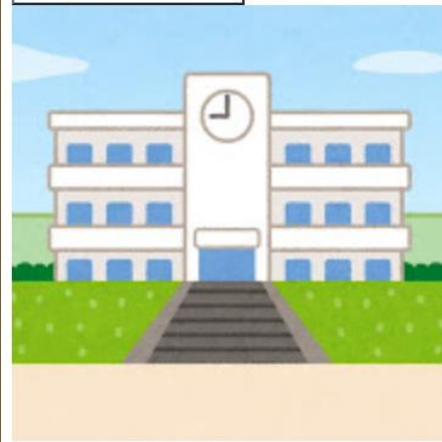
- スライドやWebサイトを作成
- 文字/画像/動画などを組み合わせたマルチメディア作品
- 静的（動きなし）や決まった処理（文字が流れるなど）
- データや情報は一方通行



## 文部中学校

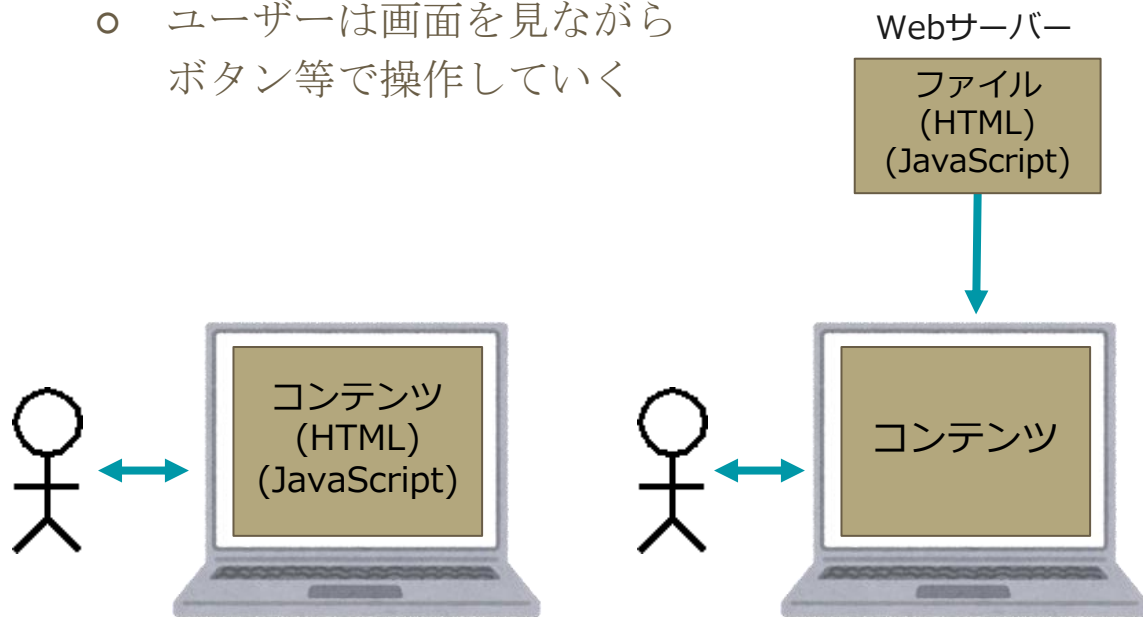
### 学校行事

| 月 | 行事   |
|---|------|
| 4 | 入学式  |
| 5 | 健康診断 |
| 6 | 遠足   |



# 従来のコンテンツ(2)

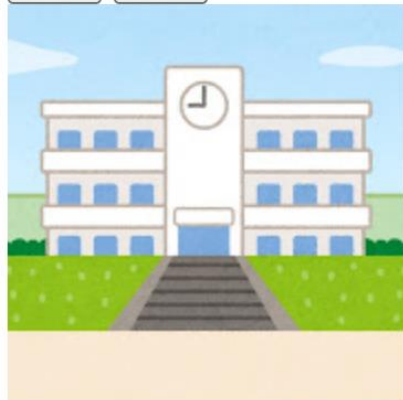
- スライドやWebサイト+プログラム
- ネットワークはダウンロードのみの一方通行
- 人とプログラムは双方向
  - ユーザーは画面を見ながら  
ボタン等で操作していく



## 文部中学校

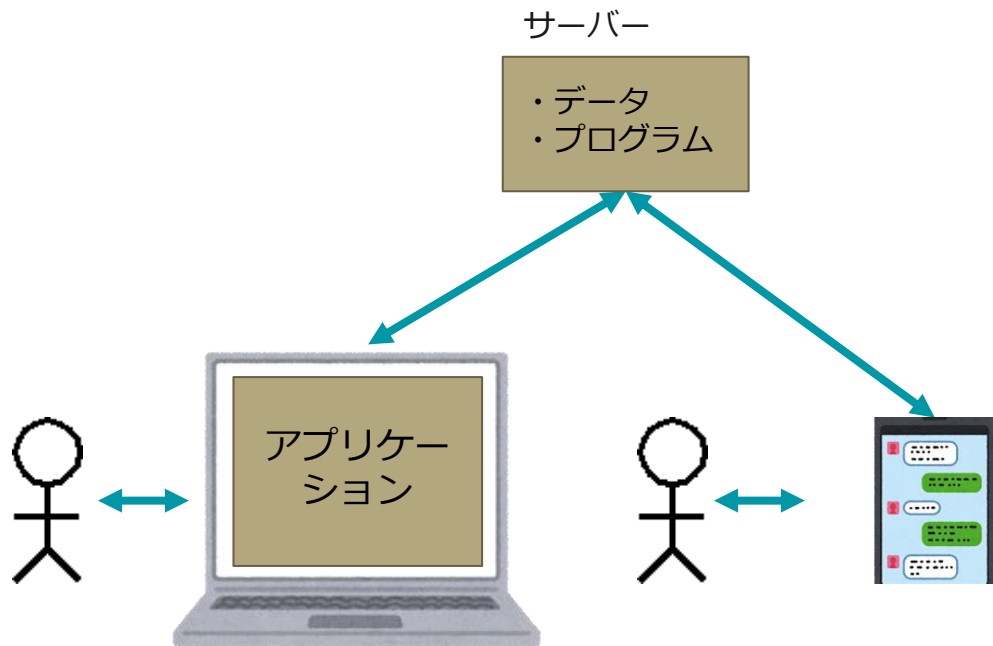
### 学校行事

| 月  | 行事   |
|----|------|
| 4  | 入学式  |
| 5  | 健康診断 |
| 6  | 遠足   |
| 背景 | 写真   |



# 世の中の情報システム

- 最近のアプリは、ネットワークでサーバーと連携して動作
  - 端末（スマホやPC）のプログラムとサーバーのプログラムが双方向に通信
  - メールやメッセージをサーバーに保管など



ネットワークを利用した  
双方向性のあるコンテンツ  
のプログラミング

# 教材の事例

- 兼宗が公開している双方向性の教材を2つ紹介
- (1)ドリトル
  - 中学校技術科の教科書に掲載
  - 生徒PCにインストール（ファイルを置くだけ）
  - サーバーを使わずに、生徒のプログラム同士で双方向通信
- (2)Bit Arrow
  - 高等学校を中心に、**10**万人以上の授業で利用されている
  - インストール不要でブラウザでプログラミング学習
  - 大阪電気通信大学、東京農工大学、明星大学で共同研究
  - さまざまな言語に対応。今回はHTML、JavaScript、Python。  
（その他、ドリトル、Processing、Cなど）

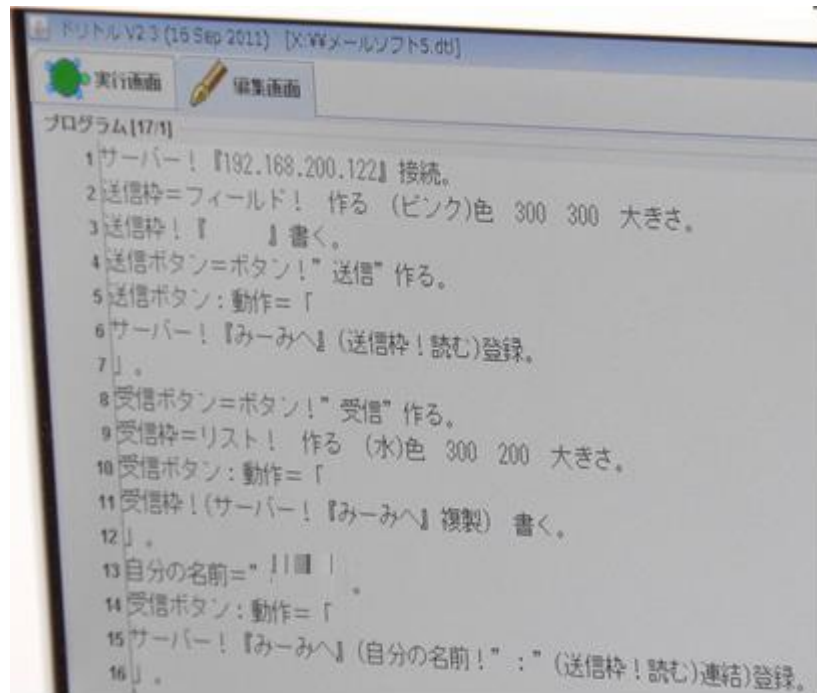
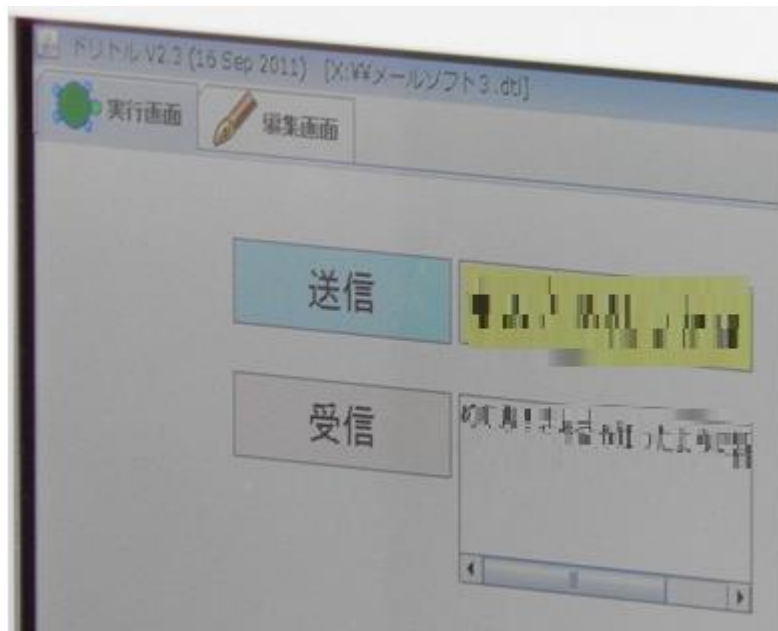
# ドリトルを使った教材例

# ドリトルのプログラム紹介

- インストール版とブラウザ版。インストール版は双方向通信に対応
- バッチ処理（対話的操作なし）
  - グラフィックス： 多角形、複雑な図形、アニメーション
  - 生徒の作品例
- 対話的操作（ボタン等で操作）
  - お絵かきツール、お花拾い、ピンポン
- 双方向性のあるネットワークプログラム
  - 生徒の作品例（中学2年生）： チャット、ピンポン

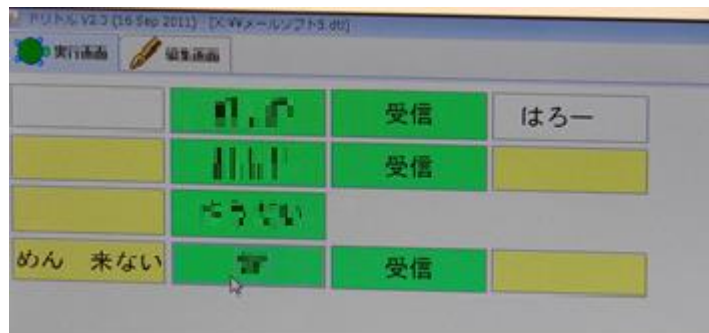
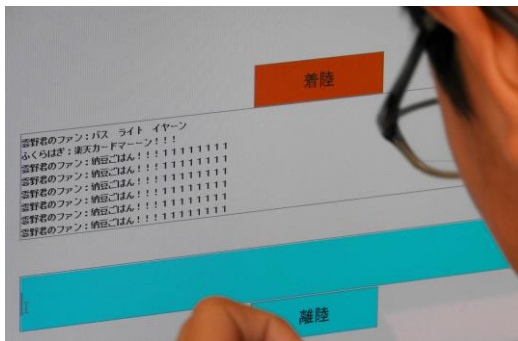
# メッセージ交換の授業例

- サンプルプログラムとして、隣同士の2人組でチャットプログラム
- 左の生徒に送信、右の生徒に受信プログラムのプリントで一方向通信
- 次の時間に生徒同士の教え合いで双方向に改良



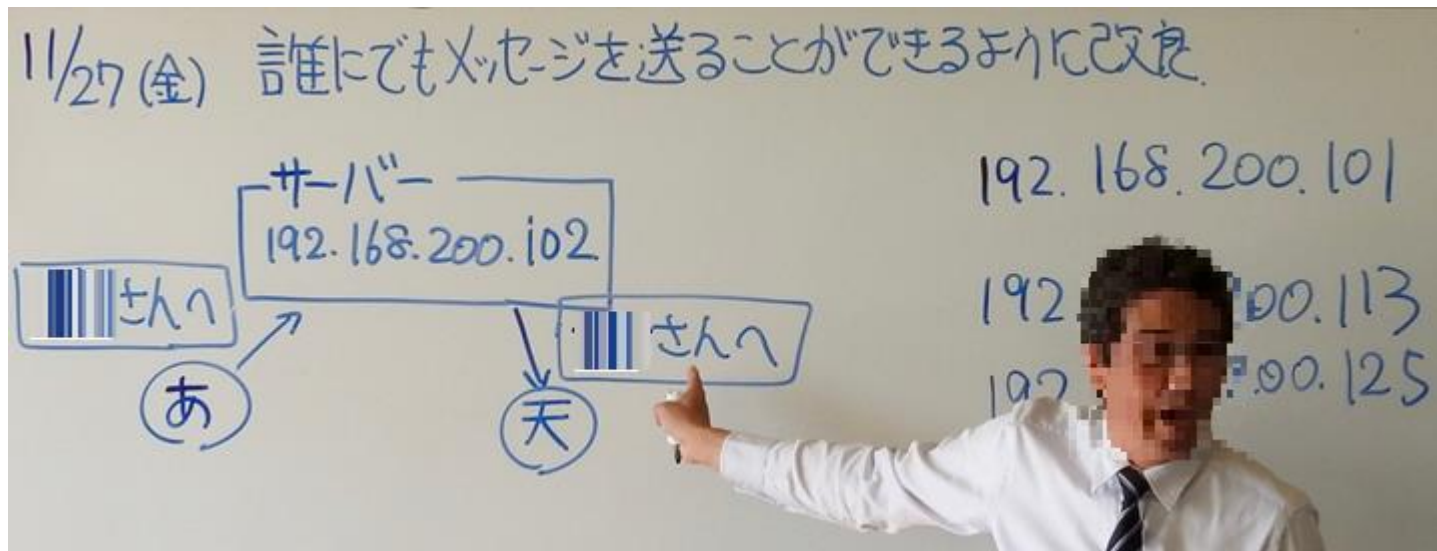
# 授業中の様子

- デザイン変更、複数人と通信する機能、着信通知などを自発的に改良
- 教え合う、アイデアを見せに行く、アクティブラーニングが実現



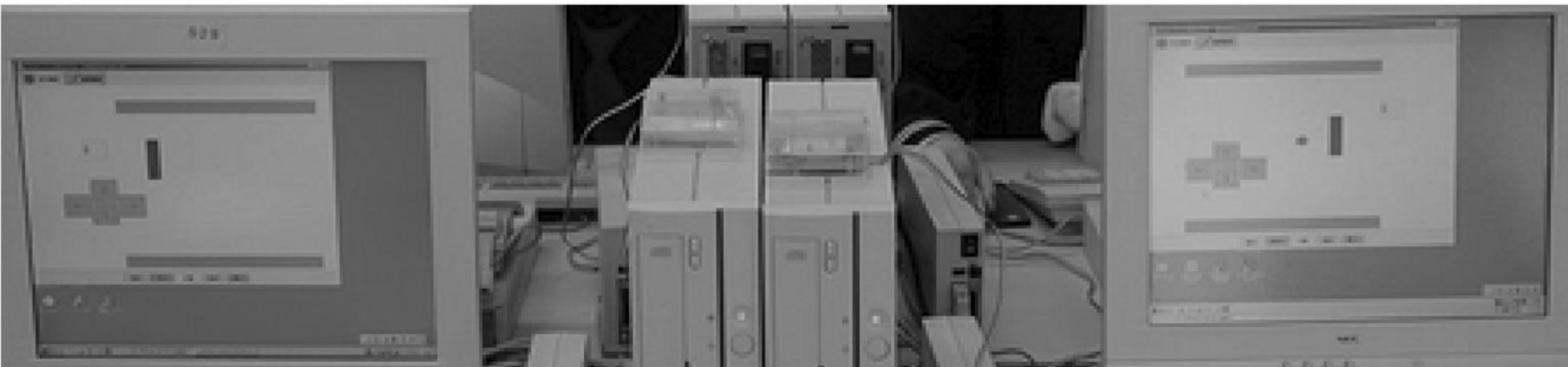
# 授業のまとめの様子

- プログラムを「送信/受信」から「送受信」「自動受信」「着信通知」などに生徒が発展
- サーバを意識させながら、IPアドレスなどネットワークの基礎を説明
- 他人の名前での書き込みを題材に、認証とセキュリティを議論



# 双方向性を持つゲーム例

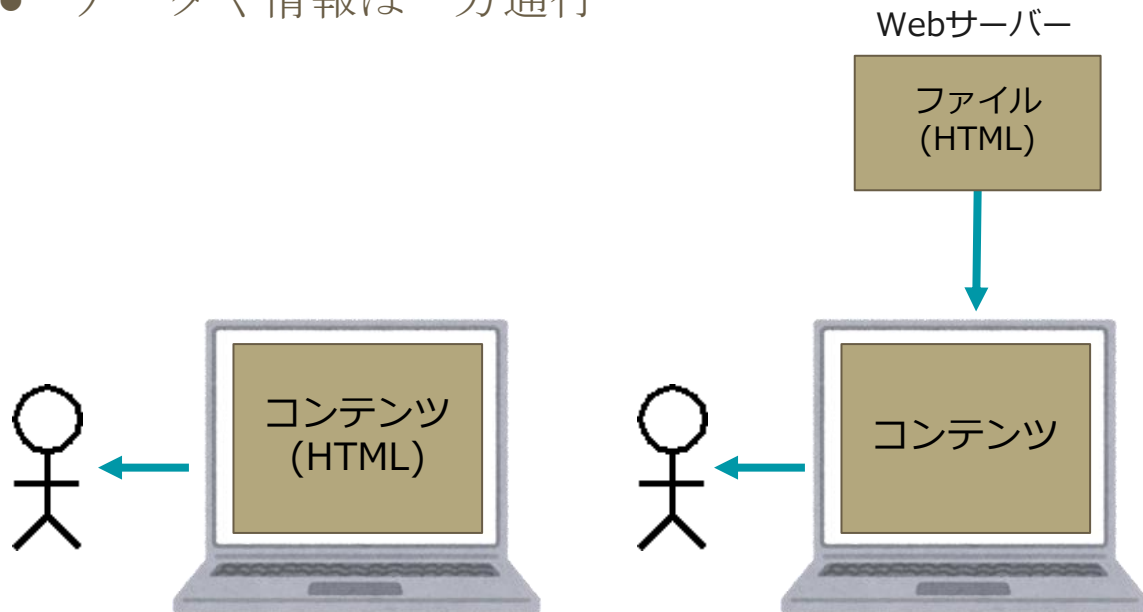
- 1人用ピンポンゲームを改良
- ボールの位置と向き ( $x, y, d$ ) を共有することで対戦ゲーム化
  - マウスによる対話操作 (双方向)
  - ネットワーク通信によるデータ交換 (双方向)
  - ゲームクリア時にアニメーション表示など (コンテンツ)



# Bit Arrowを使った教材例

# 従来のコンテンツ1(実習例)

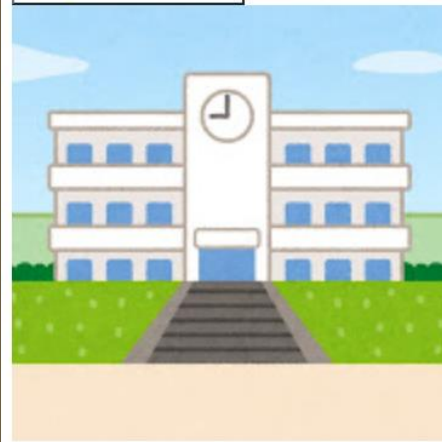
- 学校のWebページをHTMLで記述
- 文字と画像で情報を表現している
- 静的（動きなし）
- データや情報は一方通行



文部中学校

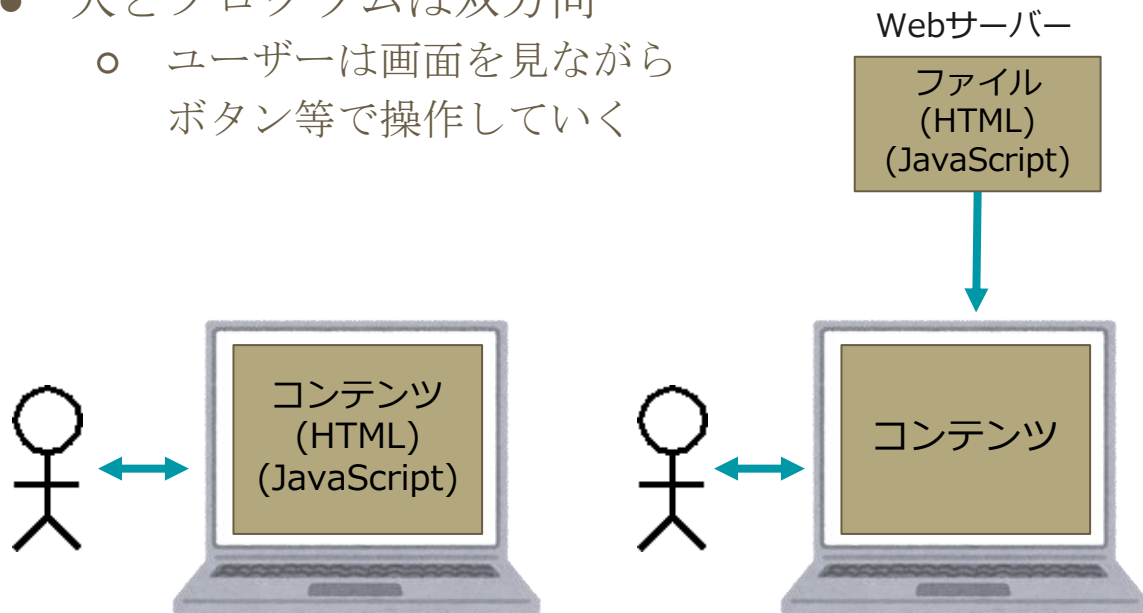
## 学校行事

| 月 | 行事   |
|---|------|
| 4 | 入学式  |
| 5 | 健康診断 |
| 6 | 遠足   |



# 従来のコンテンツ2(実習例)

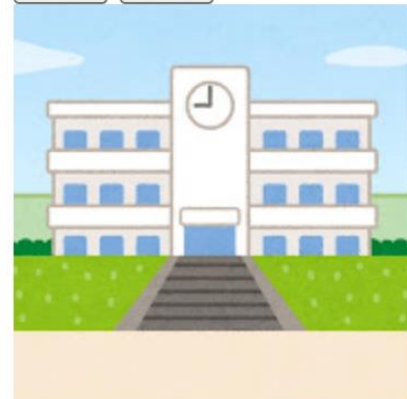
- ボタンを置くことで特定のプログラムを起動できる
- プログラムで背景色や画像などを操作できる
- ネットワークはダウンロードのみの一方通行
- 人とプログラムは双方向
  - ユーザーは画面を見ながら  
ボタン等で操作していく



## 文部中学校

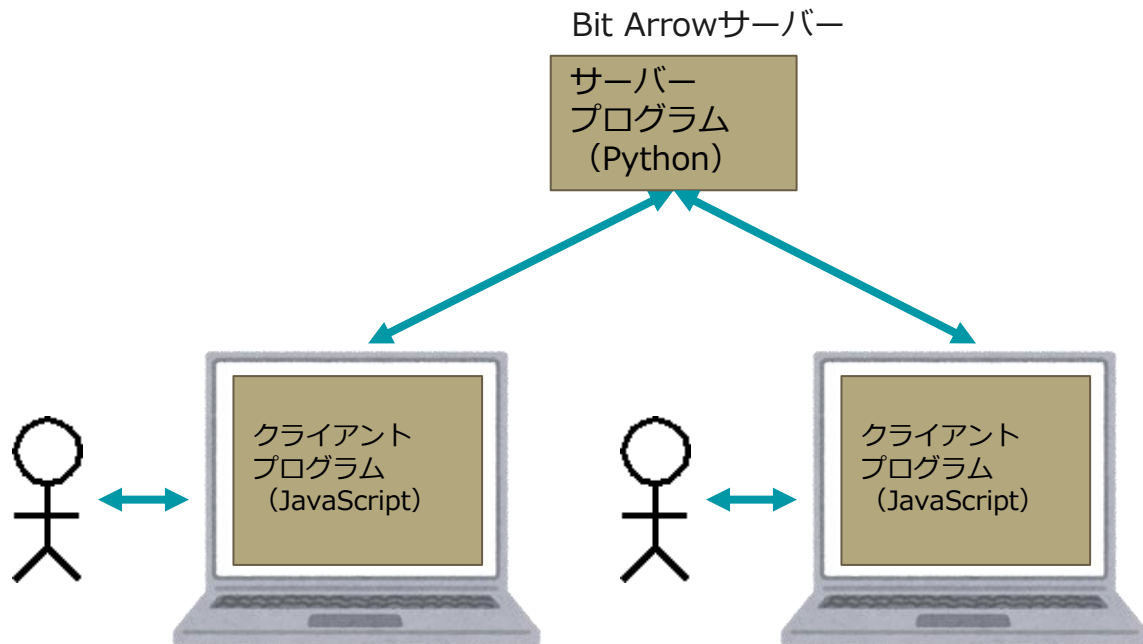
### 学校行事

| 月  | 行事   |
|----|------|
| 4  | 入学式  |
| 5  | 健康診断 |
| 6  | 遠足   |
| 背景 | 写真   |



# Bit Arrowで双方向性プログラミング

- 最近のアプリと同様に、ネットワークでサーバーと連携して動作
  - 端末の生徒プログラムとサーバーの先生プログラムが双方向に通信
  - 商品情報やメッセージをサーバーに保管



ネットワークを利用した  
双方向性のあるコンテンツ  
のプログラミング

# チャットプログラム(サーバー側Python)

- チャットプログラムの例
- クライアント端末からメッセージを受け取り、ファイルに追記
- ファイルを読み込み、クライアントに送信
- どちらも4,5行の短いプログラム。先生が置くことも可能

```
# チャット書き込み
line=input()
f=open("class/chat.txt","a")
f.write(line+"¥n")
f.close()
print("Write end")
```

```
# チャット読み込み
f=open("class/chat.txt","r")
for i in f:
    print(i,end="")
f.close()
```

# チャットプログラム(クライアント側)

- チャットプログラム

```
<html>
<h1>掲示板</h1>
<input name="text"/>
<button
name="write">Write</button><br/>
<div name="result"></div>
<pre name="hist"></pre>
</html>
```

## 掲示板

こんにちは  
はろー  
やっほー

```
native window;
read();
onClick("write", write);
¥read() {

r=callServer("https://run.eplang.jp/.../
read_chat.html");
    setText("hist", r);
}
¥write() {

result=callServer("https://run.eplang.jp/
.../add_chat.html",getText("text"));
    setText("result", result);
    read();
}
```

# ショッピングサイト

- 商品を表示して購入ボタンを押せるサイトの例
- 好きな商品を登録できる機能を作ると盛り上がる

```
<html>
  <style>
    img {max-width: 100px;}
  </style>
  <h1>ショッピングサイト</h1>
  <pre name="status"></pre><br>
  <input id="buy_item" name="buy_item"
type="hidden">
  <p>商品一覧</p><br>
  <pre name="table"></pre>
</html>
```

```
native window;

¥reload() {

r=callServer("https://run.eplang.jp/...
/show_item.html");
  setText("table", r);
  onClick("item_area", buy);
}

¥buy() {
  wait(10);
  buy_id = getText("buy_item");

r=callServer("https://run.eplang.jp/...
/buy_item.html", buy_id);
  window.alert(r);
}

reload();
```

## ショッピングサイト

### 商品一覧



1, イーストスナックス

チキンヌードル<カレー味>, 150円



2, イーストスナックス

坦々ヌードル, 150円

# WebAPIを使った教材例

# 郵便番号検索

- 郵便番号データのダウンロードとAPIによるプログラム検索が可能

```
import requests

url = "http://zipcloud.ibsnet.co.jp/api/search"
param = {"zipcode": "100-0013"}
res = requests.get(url, params=param)
response = res.json()
address = response["results"][0]

print(address["address1"] + address["address2"] +
address["address3"])
```

東京都千代田区霞が関



郵便番号データ配信サービス  
zipcloud

トップ > 郵便番号検索API

### 郵便番号検索API

**郵便番号検索APIとは**

郵便番号検索APIは、日本郵便が公開している郵便番号データを検索する機能をRESTで提供しています。現在使用しているデータは、「2024年7月31日更新分の全国一括データ（加工済バージョン）」です。

» 全国一括データ（加工済バージョン）とは

---

**リクエストURL**

ベースとなるURLは以下になります。

<https://zipcloud.ibsnet.co.jp/api/search>

このURLにリクエストパラメータを加え、リクエストを行います。

※出典：zip cloud (<https://zipcloud.ibsnet.co.jp/doc/api>)

# 図書検索

- ISBNなどから図書の検索が可能
- 書名、著者名、出版社などの情報をプログラムから扱える

```
import requests
url = "https://api.openbd.jp/v1/get"
param = {"isbn": "9784402333409"}
res = requests.get(url, params=param)
response = res.json()
print(response[0]["onix"]["DescriptiveDetail"]["TitleDetail"]["TitleElement"]["TitleText"]["content"])
```

ドリルの王様 5・6年の楽しいプログラミング



※出典：openBD (<https://openbd.jp/>)

# お菓子検索

```
import requests
url = "https://sysbird.jp/toriko/api/"
param = {"apikey": "guest", "keyword": "金平糖"}
res = requests.get(url, params=param)
print(res.text)
```

```
<item>
  <id>11440</id>
  <name>チロル金平糖</name>
  <maker>チロルチョコ</maker>
  <price>38</price>
  <type>chocolate</type>
  <regist>2024年6月28日</regist>
<url>https://sysbird.jp/toriko/2024/06/28/.../</url>
  <tags>
    <tag>チロル</tag>
  </tags>
<image>https://sysbird.jp/toriko/wp-content/.../11440.jpg</image>
  <comment>&lt;p>&lt;チロルチョコのなかにミニ金平糖がたくさん入ってる。
やわらかいチョコと硬い金平糖を同時に口のなかにいれるだなんて、どうなるの
か？考えたこともない組み合わせではないか？&lt;br>&lt;チョコは練乳のような
風味で、金平糖はオレンジや黄色の色とりどり。シロクマアイスのような味がした。
甘いチョコを感じながら金平糖をガリガリ。意外にイケる。&lt;br>&lt;カラフル
な和風のパッケージもいい。&lt;br>&lt;/p>&lt;/comment>
</item>
```

お菓子の虜

・コンビニお菓子の最新情報・

お菓子 お菓子メーカー

2024年6月28日

チロル金平糖



ホワイトチョコのなかにミニ金平糖がたくさん入ってる。やわらかいチョコと硬い金平糖を同時に口のなかにいれるだなんて、どうなるのか？考えたこともない組み合わせではないか？チョコは練乳のような風味で、金平糖はオレンジや黄色の色 [続きを読む](#)

チロルチョコ 38円 0 ★★★★★

検索...

いままでに食べたお菓子

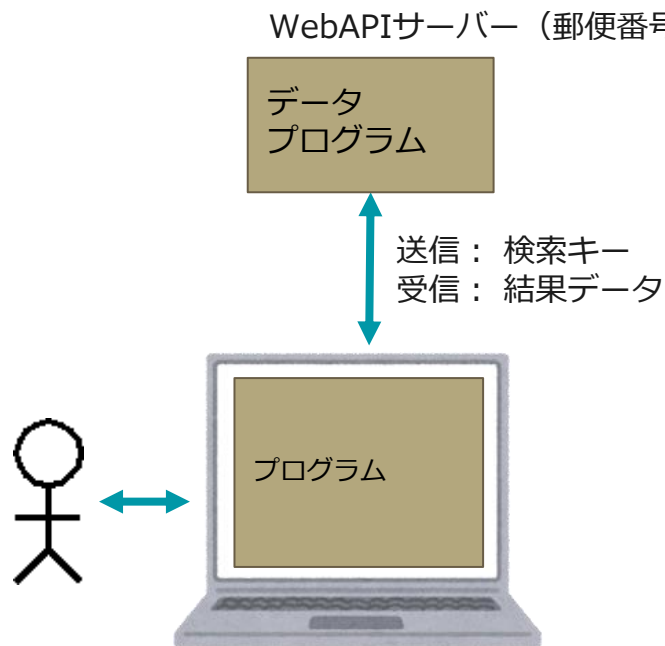
2768 種類

タグ

いちご おっとりとおつまみ おまけ かっぱえびせん さつまいも じゃがりこ のり ばかうけ アゴロ エビ オレンジ カレー カール キットカット キャラクター キヤラメル キヤラメルコーングミ コラ ボ コーヒー チロル チーズ ハッ ビーターン バター パナナ プリッツ ペビースター ポッキー ポ テトチップ マヨネーズ ミント レモン 信州 北海道 大人 季節限定 抹茶 明治太子 暴君ハバネロ 期間限定 糖の種 梅 激辛 濃い味

# WebAPIの利用

- 公開されているWebAPIサーバーはプログラムから利用可能
  - 端末（スマホやPC）のプログラムとサーバーのプログラムが双方向に通信
  - 検索条件などを対話的に指定することで双方向性のある操作が可能

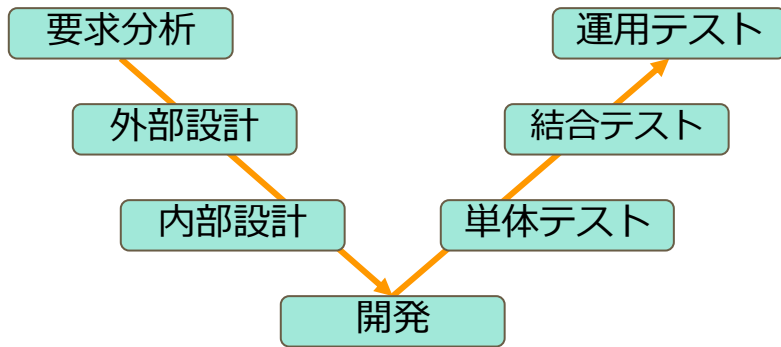


（メモ）  
Webサイトと同様に、いつの間にか閉鎖されることも多い。  
授業で安心して使えるように、授業用のAPIサーバーを用意するのは有意義かもしれない。（検討したいと思います）

# ソフトウェア開発の流れについて

# 情報システム開発の流れ

- システム開発の手順
- 作業の工程としては、このようなものがある
  - 要求分析
  - 設計（外部設計、内部設計）
  - 開発（プログラミング）
  - テスト（単体テスト、結合テスト）
  - 運用・保守

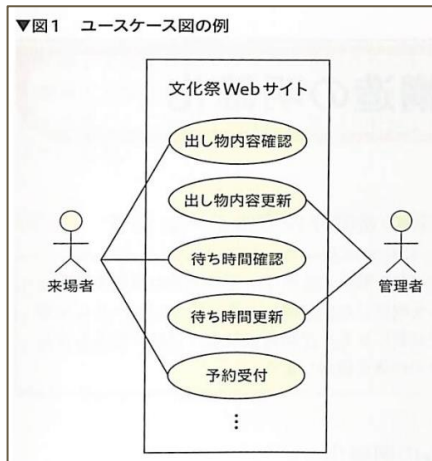


# 要求分析

- 要求分析を間違えると誰も使わないシステムができてしまう
- 要求分析【最も重要】
  - システム開発の目的、実現すべき内容、達成目標などを定義する
  - インタビューでユーザが望んでいるものを明らかにする
- 要件定義
  - 要求分析の結果を元に、システムの具体的な仕様を決める
  - 業務上の要件と、システム上の要件を区別し、明確にする

# 要求分析とシステムの可視化

- 要件定義を作るための要求分析がいちばん重要
- 要求分析なので、「分析」する
- まず、関係者を考える
  - 開発者（本人）
  - 発注者（直接のお客さん）
  - エンドユーザー（見えないお客さん）
- ユースケース図で整理したい
- 誰がどんな機能を使うのか？
- 要求分析を間違えると誰も使わないシステムができてしまう...



# 外部設計

- ユーザから見える部分の設計
- ハードウェア、OS
- 画面構成
- 入力項目
- 表示・印刷する出力項目
- ファイルやデータベースの論理設計

# 内部設計

- 開発者の立場でどのように作るかという詳細部分の設計
- ユーザから見えない。ユーザに理解する必要はない
- モジュール設計
- ファイルやデータベースの詳細設計

# テストと運用

- テスト
  - ソフトウェアは、作っただけではほぼ**100%**動かない
- ホワイトボックステスト
  - ソースコードを見ながらテスト内容を考える
- ブラックボックステスト
  - ソースコードを見ないでテスト内容を考える
- 単体テスト
  - テストコードを書いてからプログラムを書くやり方も（TDD）

# まとめ

# 参考資料 ドリトル

- ドリトルのサイト <http://dolittle.eplang.jp>
  - インストール版。生徒**PC**でファイル展開
  - オンライン版はブラウザで動作。双方向は不可
    - Bit Arrowでも使えます
- オンライン資料
  - 東京書籍「ドリトルによるプログラミング学習」
  - 開隆堂「やってみようプログラミング」
  - 日本文教出版「Scratchとドリトルではじめるプログラミング」

# 参考資料 Bit Arrow

- 授業前の準備（先生）
  - 教員登録し授業クラスを作って利用する
    - クラス名は学校+学年+年度など
  - 生徒は一括で登録可能。パスワードの有無は選べる
- 授業ごと（生徒）
  - 生徒はクラス名と生徒名でログイン
  - 言語ごとのプロジェクトを作って実習する。提出不要は便利
    - 先生がサンプルを一括配布しておくで全員にプロジェクトが作られて便利
- 必要があれば（先生）
  - 授業の様子を一覧
  - 作品ファイルの一括ダウンロード（ZIPファイル）
  - 個別のサポート
  - TTに対応

# 今日の振り返り

- 小中高の情報教育とプログラミングの流れ
- ネットワークを利用した双方向性のあるコンテンツのプログラミング
  - マルチメディアコンテンツに加えて、プログラミングによる双方向の対話性と、ネットワークを利用してコンピュータが協調動作する現代的なアプリの仕組み
- (1) ドリトルの事例： 教室内の**端末同士**での双方向プログラミング
  - 生徒のプログラム作品から双方向の学習まで
- (2) Bit Arrowの例： **サーバーと通信**する双方向プログラミング
  - サーバープログラムを先生が用意
- (補足) 公開されている**WebAPI**を利用
- コンテンツの表現や同級生とのコミュニケーションのある楽しい授業