# 5 ネットワークを利用したプログラム（メッシュ）（第 4 時）

＿＿＿ 年 ＿＿＿ 組 ＿＿＿ 番　氏名 ＿＿＿＿＿＿＿＿＿＿　参加コード ＿＿＿＿＿＿＿＿

**今日のめあて** **メッシュ**拡張機能を使って、**別のコンピューター**と文字をやりとりする。

▶ **レッスンで練習しよう**　「チャットをつくろう」の **レッスン 3** 3-2・3-3 → **レッスン 4「チャットを作ろう」**（ペアで。ひとりなら練習相手）　終わった □

## 1. メッシュでできること

① **メッセージを全員に送れる**――「○○を送る」が**同じグループの全部のコンピューター**に届く。
② **変数を共有できる**――自分の変数の中身を、相手が **「○○ センサーの値」** で読める。

## 2. つなぎ方（ペアでやります）

| やくわり | 手順 | できた |
|---|---|---|
| **ホスト**（1 人） | 「拡張機能を追加」→ **メッシュ** → 画面上の「メッシュ」から **グループを作る** → グループ名を相手に伝える | |
| **メンバー**（もう 1 人） | 同じように**メッシュ**を追加 → **グループに参加する** → 相手のグループ名を選ぶ | |

**うまくつながらないとき**：①同じ Wi-Fi か　②グループ名が合っているか　③ページを読み込み直してグループを作り直す

## 3. 課題⑤：2 台のコンピューターでチャット

**送る側と受ける側で、プログラムが少しちがいます。**ペアで役わりを決めて作りましょう。

**シマラビの役の人の端末**（シマラビが自分、シマネコが相手）




シマラビ＝**自分のメッセージを送る**




シマネコ＝**相手のメッセージを表示する**

**シマネコの役の人の端末**（シマネコが自分、シマラビが相手）




シマネコ＝**自分のメッセージを送る**




シマラビ＝**相手のメッセージを表示する**

**課題⑤**　ペアでメッシュにつなぎ、**2〜3 往復**やりとりする。作品名 **04_チャット基本**。　ペアの相手：＿＿＿＿＿＿／グループ名：＿＿＿＿＿＿

## 4. Ruby では

**自分のメッセージを送る**

```
when_flag_clicked do
  ask("メッセージを入力")
  $シマラビのメッセージ = answer
  broadcast("シマラビのメッセージ")
end
```

**相手のメッセージを表示する**

```
when_receive("シマネコのメッセージ") do
  say(mesh.sensor_value("シマネコのメッセージ"))
end
```

**★ 4 ページ（同じコンピューターの中）と、このページ（別のコンピューター）で、プログラムはどこが変わりましたか。**

……

……

**教員用メモ｜5 ページ（ここが最重要・事前確認を推奨）**
・変わったのは **「変数を言う」→「センサーの値を言う」** の 1 か所だけ。ここを板書で強調すると、ネットワークが「遠くの変数を読むこと」だと理解しやすい。
・**グループには接続の期限（数十分）がある**。切れたら作り直せばよい。授業のはじめに作らせるとよい。
・**先に見本のプログラムを用意して配ってしまってよい**（このページのプログラムをそのまま作ったもの）。0 から組ませると 1 時間で終わらないことがある。「見本を配って動かす → 変数名を自分たちの名前に変える」だけでも狙いは達成できる。
・トラブル時の最短の逃げ道：**ペアを先生の端末に変える／3 人グループにする**。


https://smalruby.app ／ この教材は **CC0 1.0（パブリックドメイン）**です。自由にコピー・改変・配布できます（出典表示は不要）