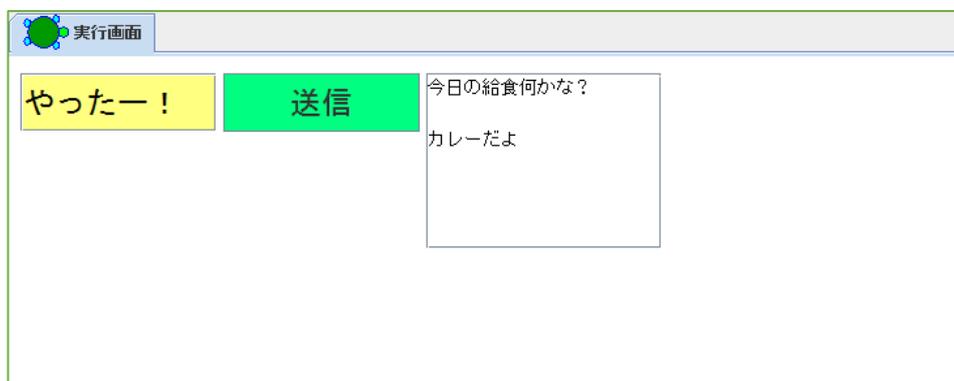
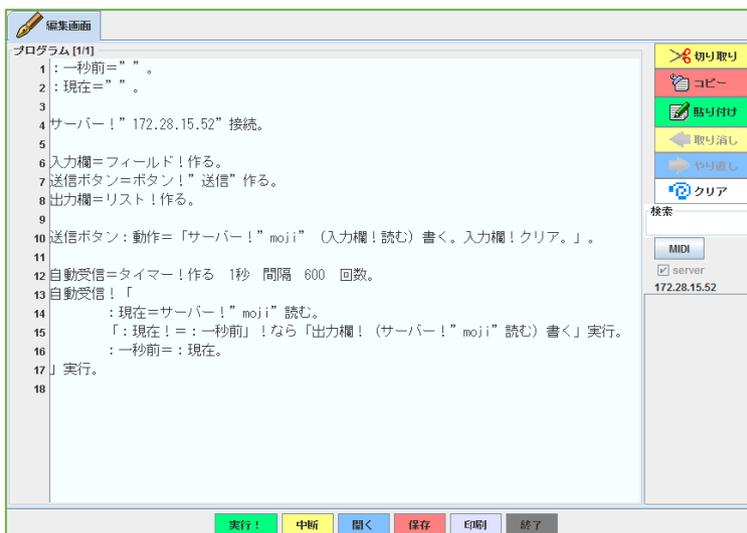


プログラミング言語

「ドリトル」

を使ってみよう

～通信編～

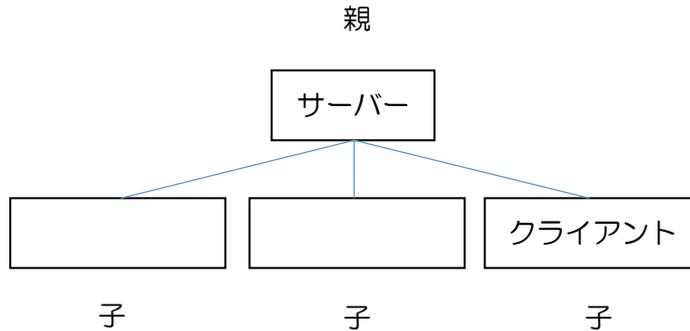


年 組 号 氏名

サーバーの起動

通常、パソコン室では、先生機から生徒画面を見たり、強制的に操作したりすることはできるが、生徒同士で情報をやりとりすることなどはできない。これは、先生機が【親】の立場にあって、【子】である生徒機を管理している状態だからである。この管理下では【子】同士のやり取りが基本的に許されていない。

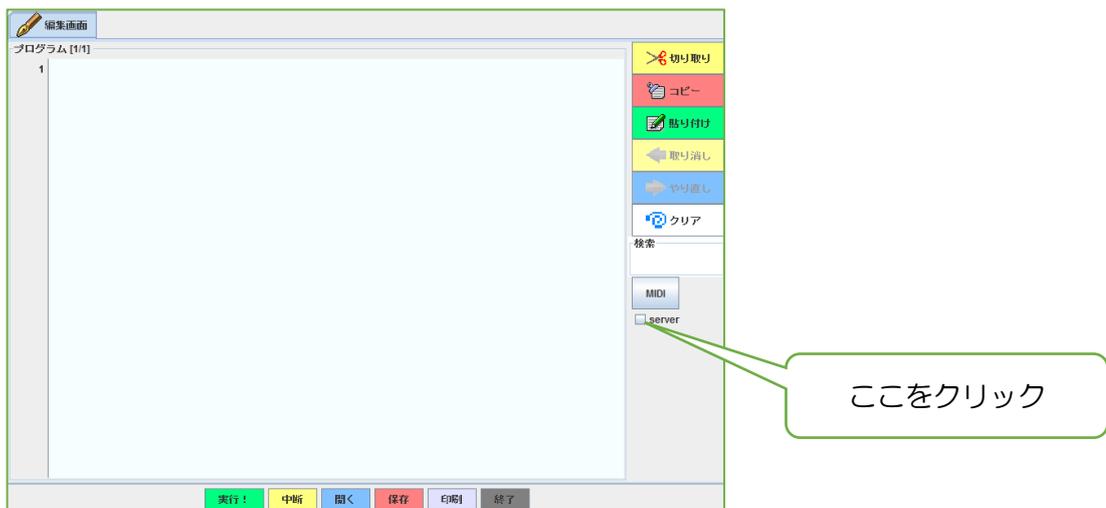
ネットワークで親の立場にあるコンピュータを【サーバー】といい、子の立場であるコンピュータを【クライアント】という。



ここで注意したいのは、上図において、クライアント同士は直接つながっていないということである。つまり、クライアント同士で情報をやり取りするときも、必ずサーバーを通すことになる。

ドリトルをインストールしたパソコンでは、親（サーバー）を指定して、それに子（クライアント）として接続すれば、生徒同士であっても情報のやり取りができる。

サーバーの起動は、編集画面で【server】横のチェックボックスをクリックするだけでよい。

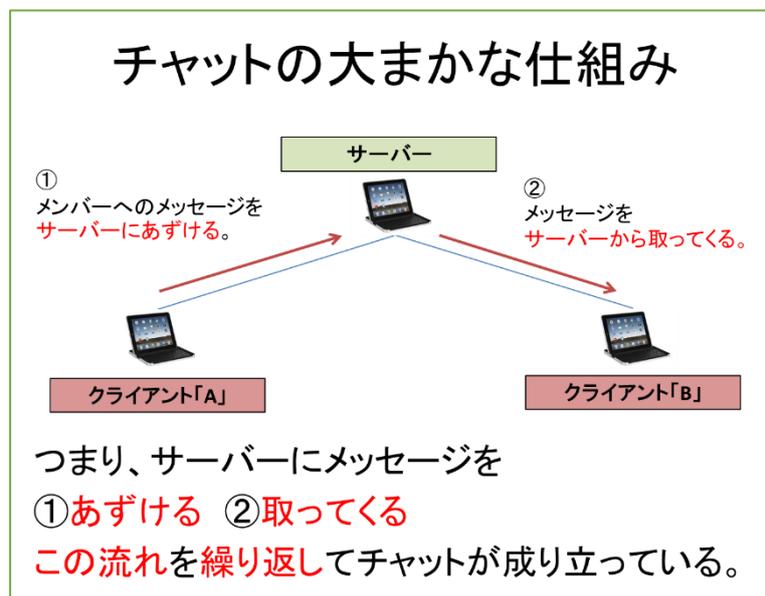


すると、【server】の下に次のような数字が表示される。



この数字を【IP アドレス】といい、ネットワークにおけるこのパソコンの住所（番地）にあたるものである。当然、すべてのパソコンでこの数字は異なる。近くの人と確認しあってみよう。

この授業では、基礎編で学んだことを使いながら、チャットのプログラムを作成する。チャットとは、ネットワークに接続している人どうしでリアルタイムにメッセージのやり取りをおこなうものである。その根本的な仕組みは、現在よく使われている SNS などのやりとりともほぼ変わらない。

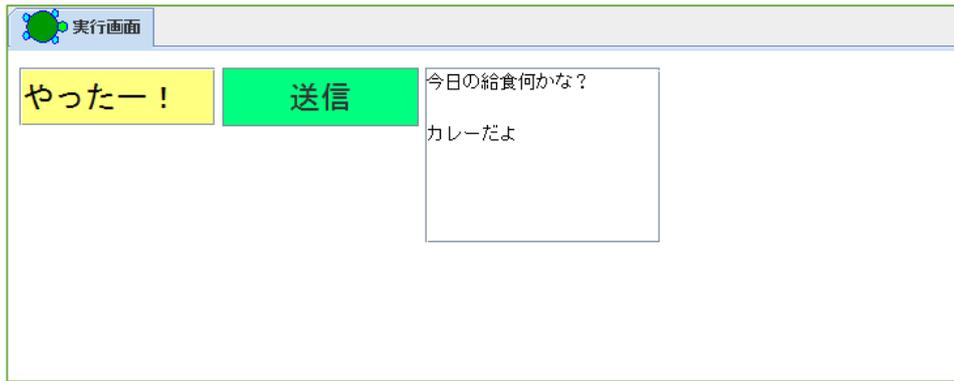


完成作品を使ってみよう ※一旦ドリトルを終了し、新たに起動し直しておく。

ここからは、基本的に 4 人 1 組で学習を進めていく。まず、4 人のうち、誰が親（サーバー）になるかを定める。親になった人は、【server】横のチェックボックスをクリックし、サーバーの IP アドレスを子（クライアント）に伝えておく。

まず、完成作品を転送するので、自由に使ってイメージをつかもう。と、その前に・・・

編集画面を開き、1 行目を見ると・・・サーバー！“localhost” 接続。”
となっているので、【localhost】の部分をサーバーの IP アドレスに変更する。例えば、サーバーの IP アドレスが【172.28.100.67】なら・・・サーバー！“172.28.100.67” 接続。
と打ち変える。そして実行ボタンを押すと・・・



プログラムの概要

メッセージを入力して、送信ボタンを押すと、サーバーにそのメッセージが必ずけられる。他のメンバーは、サーバーに必ずけられたメッセージを自動的に受信して、枠内に表示する。

プログラミングをしよう

おおまかに次のような流れで作業を進めていく。

- ①サーバーに接続する
- ②部品を作る
- ③部品を操作できるようにする
- ④ここまでの問題点を解決する

①サーバーに接続する

(ここでは、サーバーのIPアドレスを 172.28.100.67 として説明しているが、実際は各サーバーのIPアドレスを指定すること。)

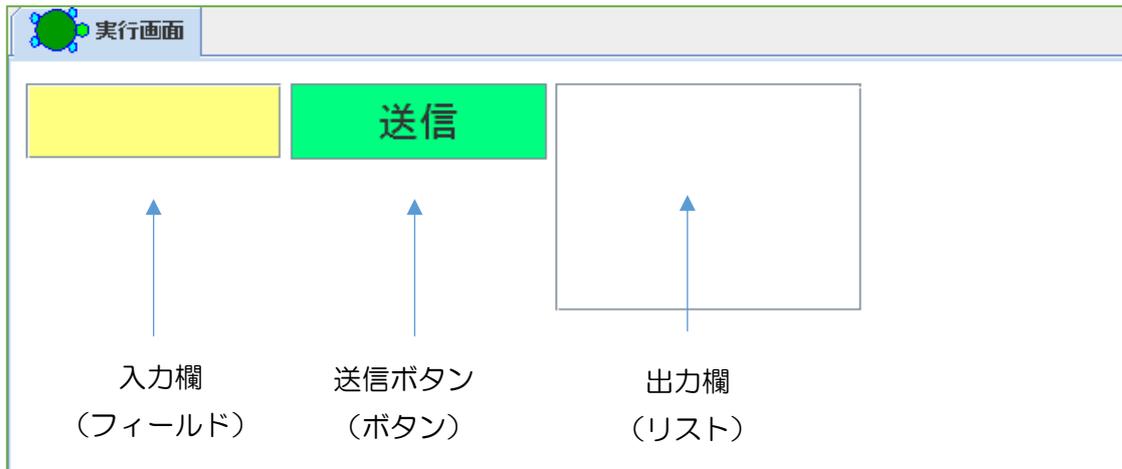
配布したプログラムの1行目だけを残して、あとは全部消す。

1行目 `サーバー!” 172.28.100.67” 接続。`

サーバー役の人は、自分のパソコンに接続するというおかしなことをやっているようだが、これで、そのパソコンがサーバーとして機能するようになった。また、他のクライアント役の人は、これでサーバーと接続でき、計4台(サーバー1台、クライアント3台)の小さなネットワークを作ることができた。

つまり、サーバーと接続するには1行目のように・・・ `サーバー!” サーバーのIPアドレス” 接続。`と書く。

②必要な部品を作る

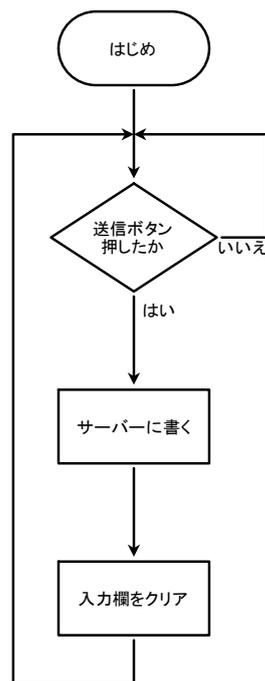


- 2 行目 入力欄=フィールド! 作る。
- 3 行目 送信ボタン=ボタン! (“送信”) 作る。
- 4 行目 出力欄=リスト! 作る。

③部品を操作できるようにする。

送信ボタンを押したときにしなければならないことは・・・

- ① サーバーにメッセージをあずける (書く)。
- ② (次の入力に備えて) 入力欄に入っている文字を消しておく。



まず、サーバーにメッセージをあずける（書く）には・・・ **サーバー！”目印名”（名前！命令）書く。**
とするルールがある。

目印名を“moji”として、入力欄の文字をサーバーにあずけるには・・・

① サーバー！ “moji”（入力欄！読む）書く。

また、次の入力に備えて、入力欄の文字を消しておくには・・・

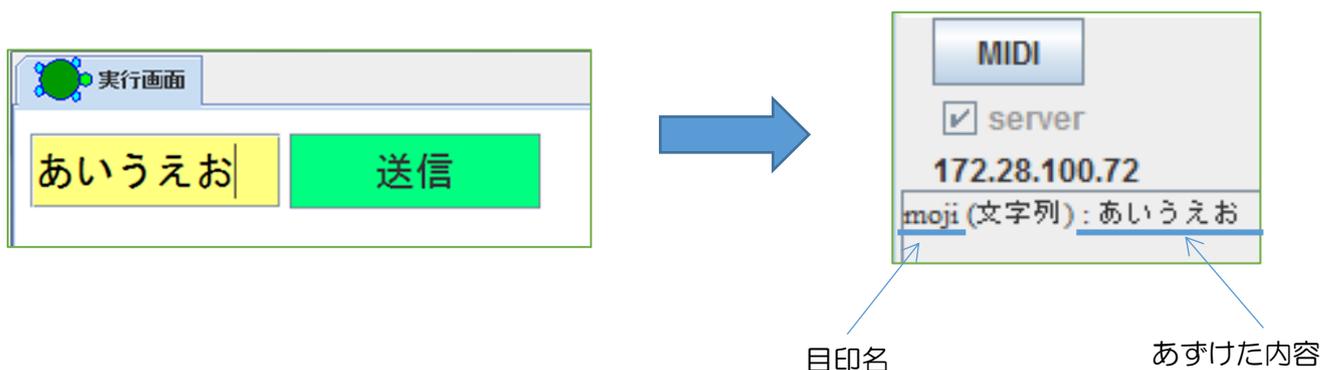
② 入力欄！クリア。 → ※ 基礎編でも出てきた **名前！命令。** の形。

さて、ボタンを操作できるようにするには・・・ **ボタン名：動作＝「名前！命令。」** と書く必要があった。

上の①②を送信ボタンの動作に設定すると・・・ ※2つ以上のことをするから【。】が必要。

5行目 **送信ボタン：動作＝「サーバー！”moji”（入力欄！読む）書く。入力欄！クリア。」**

それではここで、送信ボタンの動作確認を試みよう。クライアントの1人が入力欄に【あいうえお】と書いて送信ボタンを押してみる。そうすると、サーバー役の画面に、目印名を“moji”として、入力欄を読んだ【あいうえお】という内容があずけられていることがわかる。さらに、クライアントの入力欄から【あいうえお】が次の入力に備えて消えている。



送信の場合は、送信したいときにメッセージを入力して送信ボタンを押せばよい。しかし、受信は、新しいメッセージがいつ送られてくるかわからないので、そのチェックを自動化する必要がある、

基礎編でも学習したように、動きを自動化するためにはタイマーを使えばよい。タイマーを作って、【自動受信タイマー】という名前をつけるには・・・

④ここまでの問題点を解決する

作成したプログラムを実行してみて、問題点をあげてみよう。

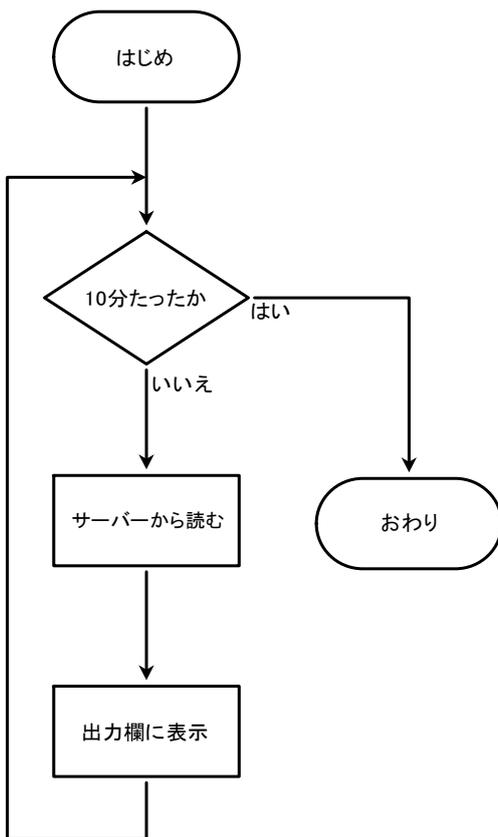
() の解決

フローチャートで考えよう。

現在のフローチャート（受信用）



変更したフローチャート（受信用）



※1 秒ごとに出力欄に表示している。

プログラミングの方針（

）の解決のために・・・

今後の学習（さらに深めよう！！）

★現在残っている問題点は・・・

- （）
- （）

などがある。それらの問題点を解決するためのアイデアを出し合おう。

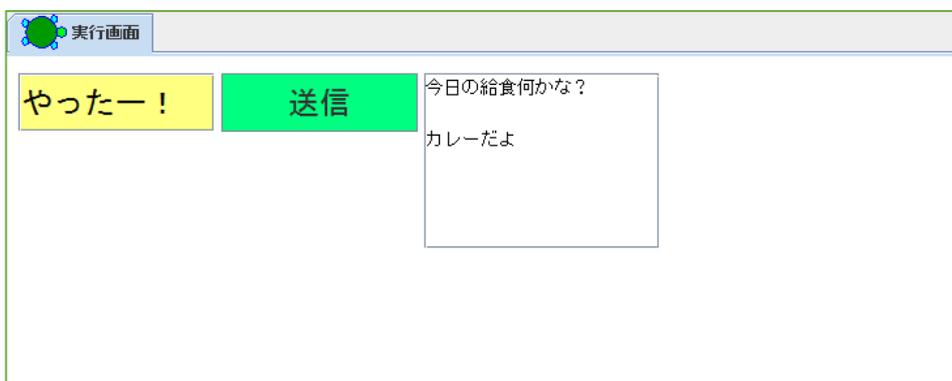
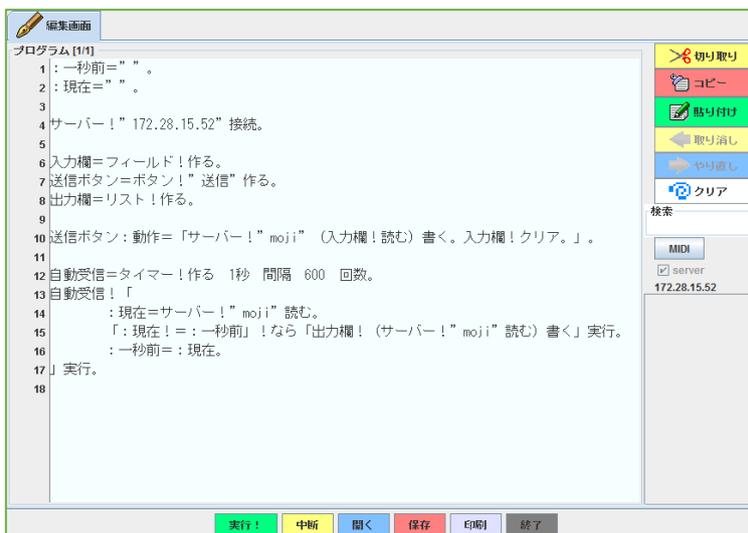
★チャットプログラムからわかる情報セキュリティの問題について、意見を出し合おう。

また、ネットワークを利用するときの注意点をまとめよう。

「ドリトル」

を使ってみよう

～通信編～

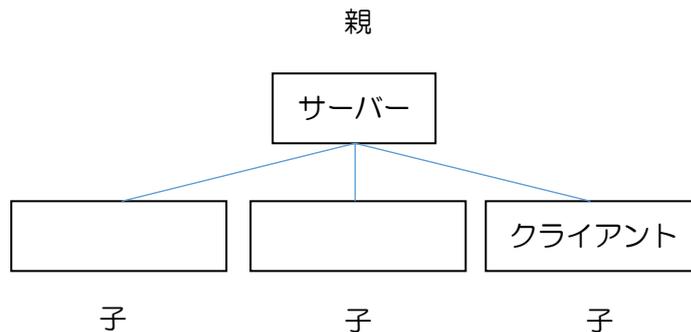


○年 ○組 ○○号 氏名 ○○ ○○

サーバーの起動

通常、パソコン室では、先生機から生徒画面を見たり、強制的に操作したりすることはできるが、生徒同士で情報をやりとりすることなどはできない。これは、先生機が【親】の立場にあって、【子】である生徒機を管理している状態だからである。この管理下では【子】同士のやり取りが基本的に許されていない。

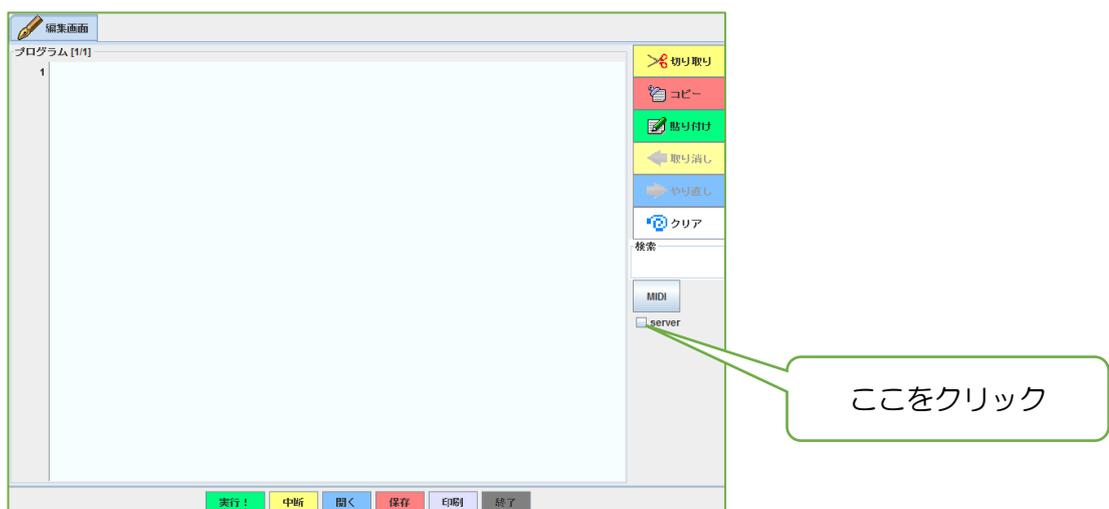
ネットワークで親の立場にあるコンピュータを【サーバー】といい、子の立場であるコンピュータを【クライアント】という。



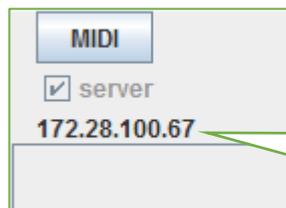
ここで注意したいのは、上図において、クライアント同士は直接つながっていないということである。つまり、クライアント同士で情報をやり取りするときも、必ずサーバーを通すことになる。

ドリトルをインストールしたパソコンでは、親（サーバー）を指定して、それに子（クライアント）として接続すれば、生徒同士であっても情報のやり取りができる。

サーバーの起動は、編集画面で【server】横のチェックボックスをクリックするだけでよい。

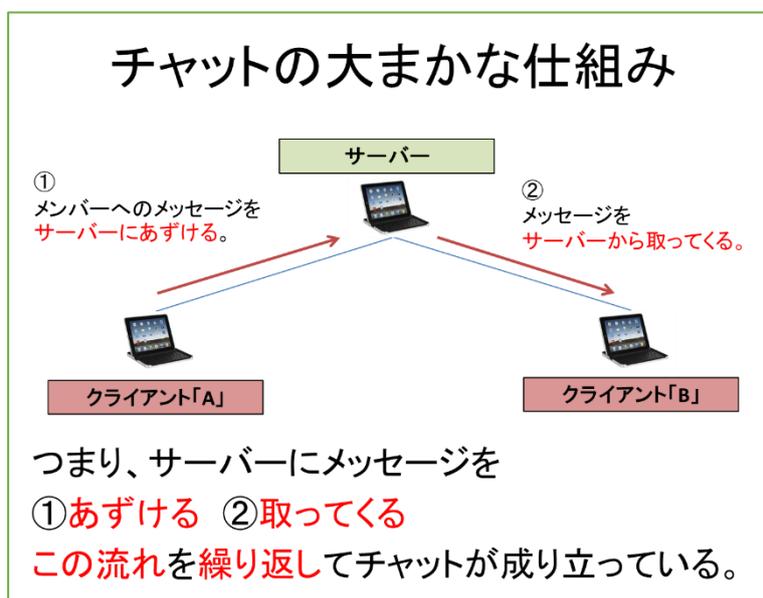


すると、【server】の下に次のような数字が表示される。



この数字を【IP アドレス】といい、ネットワークにおけるこのパソコンの住所（番地）にあたるものである。当然、すべてのパソコンでこの数字は異なる。近くの人と確認しあってみよう。

この授業では、基礎編で学んだことを使いながら、チャットのプログラムを作成する。チャットとは、ネットワークに接続している人どうしでリアルタイムにメッセージのやり取りをおこなうものである。その根本的な仕組みは、現在よく使われている SNS などのやりとりともほぼ変わらない。

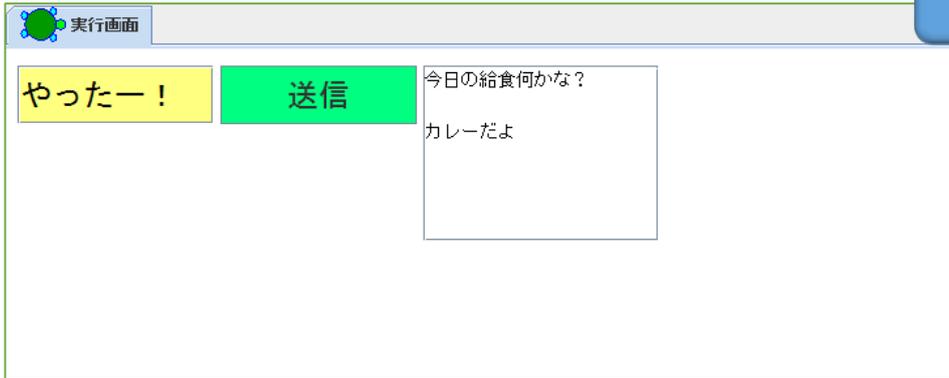


完成作品を使ってみよう ※一旦ドリトルを終了し、新たに起動し直しておく。

ここからは、基本的に 4 人 1 組で学習を進めていく。まず、4 人のうち、誰が親（サーバー）になるかを定める。親になった人は、【server】横のチェックボックスをクリックし、サーバーの IP アドレスを子（クライアント）に伝えておく。

まず、完成作品を転送するので、自由に使ってイメージをつかもう。と、その前に・・・

編集画面を開き、1 行目を見ると・・・サーバー！“localhost” 接続。”
となっているので、【localhost】の部分サーバーの IP アドレスに変更する。例えば、サーバーの IP アドレスが【172.28.100.67】なら・・・サーバー！“172.28.100.67” 接続。
と打ち変える。そして実行ボタンを押すと・・・



プログラムの概要

メッセージを入力して、送信ボタンを押すと、サーバーにそのメッセージが必ずけられる。他のメンバーは、サーバーに必ずけられたメッセージを自動的に受信して、枠内に表示する。

プログラミングをしよう

おおまかに次のような流れで作業を進めていく。

- ①サーバーに接続する
- ②部品を作る
- ③部品を操作できるようにする
- ④ここまでの問題点を解決する

①サーバーに接続する

(ここでは、サーバーのIPアドレスを 172.28.100.67 として説明しているが、実際は各サーバーのIPアドレスを指定すること。)

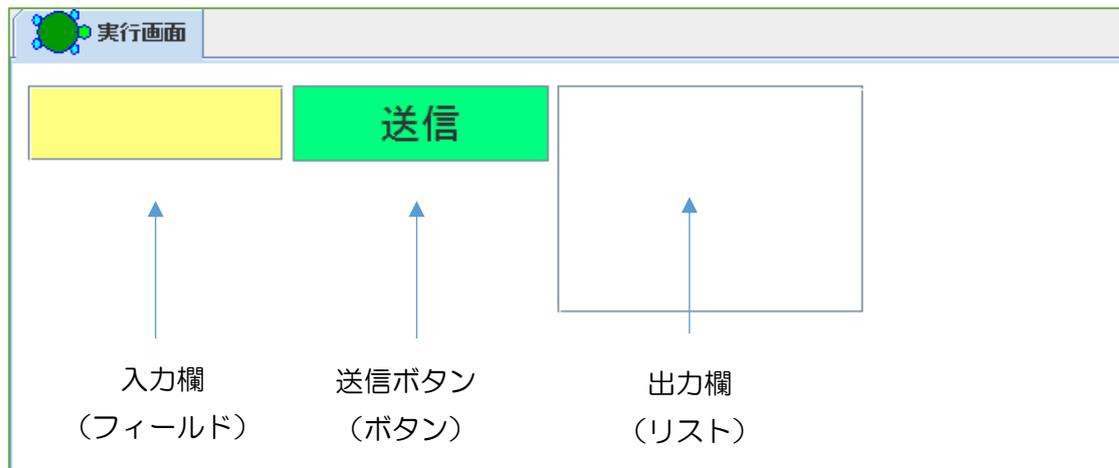
配布したプログラムの1行目だけを残して、あとは全部消す。

1行目 `サーバー!" 172.28.100.67" 接続。`

サーバー役の人は、自分のパソコンに接続するというおかしなことをやっているようだが、これで、そのパソコンがサーバーとして機能するようになった。また、他のクライアント役の人は、これでサーバーと接続でき、計4台(サーバー1台、クライアント3台)の小さなネットワークを作ることができた。

つまり、サーバーと接続するには1行目のように・・・ `サーバー!" サーバーのIPアドレス" 接続。` と書く。

②必要な部品を作る

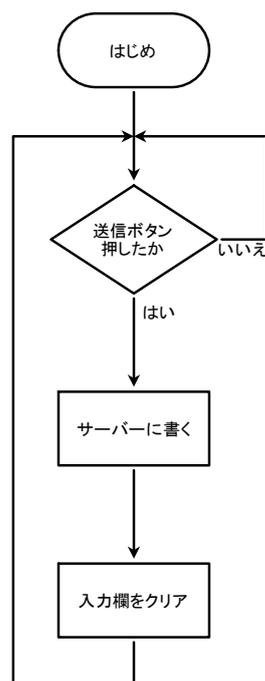


- 2 行目 入力欄=フィールド! 作る。
- 3 行目 送信ボタン=ボタン! (“送信”) 作る。
- 4 行目 出力欄=リスト! 作る。

③部品を操作できるようにする。

送信ボタンを押したときにしなければならないことは・・・

- ① サーバーにメッセージをあずける (書く)。
- ② (次の入力に備えて) 入力欄に入っている文字を消しておく。



まず、サーバーにメッセージをあずける(書く)には・・・とするルールがある。

サーバー！”目印名”（名前！命令）書く。

目印名を“moji”として、入力欄の文字をサーバーにあずけるには・・・

① サーバー！ “moji”（入力欄！読む）書く。

また、次の入力に備えて、入力欄の文字を消しておくには・・・

② 入力欄！クリア。 → ※ 基礎編でも出てきた **名前！命令。** の形。

さて、ボタンを操作できるようにするには・・・あった。

ボタン名：動作＝「名前！命令。」

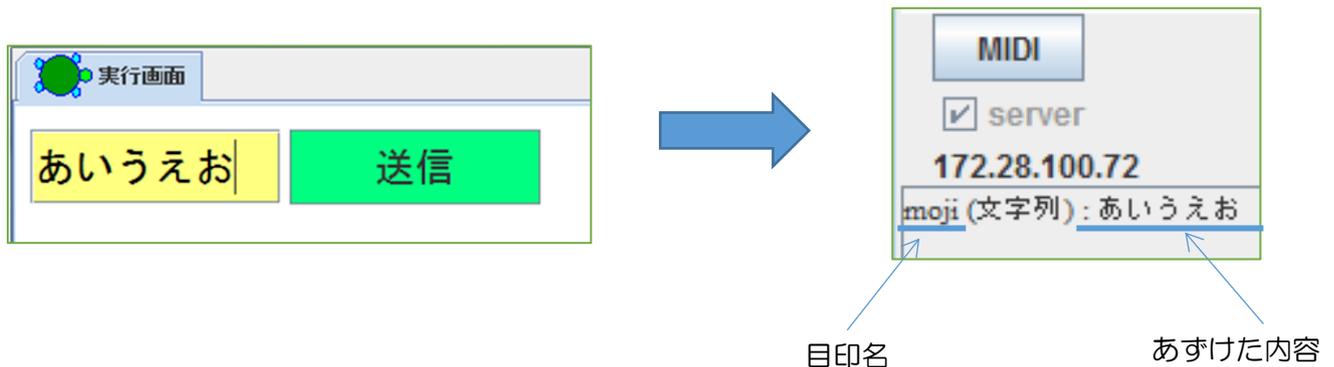
と書く必要が

上の①②を送信ボタンの動作に設定すると・・・

※2つ以上のことをするから【。】が必要。

5行目 **送信ボタン：動作＝「サーバー！” moji”（入力欄！読む）書く。入力欄！クリア。」。**

それではここで、送信ボタンの動作確認を試みよう。クライアントの1人が入力欄に【あいうえお】と書いて送信ボタンを押してみる。そうすると、サーバー役の画面に、目印名を“moji”として、入力欄を読んだ【あいうえお】という内容があずけられていることがわかる。さらに、クライアントの入力欄から【あいうえお】が次の入力に備えて消えている。



送信の場合は、送信したいときにメッセージを入力して送信ボタンを押せばよい。しかし、受信は、新しいメッセージがいつ送られてくるかわからないので、そのチェックを自動化する必要がある、基礎編でも学習したように、動きを自動化するためにはタイマーを使えばよい。タイマーを作って、【自動受信タイマー】という名前をつけるには・・・

6行目 **自動受信タイマー=タイマー！作る。**

※普通のタイマーは、0.1秒ごとに100回（10秒間）繰り返す動きを自動化する。通常、メッセージのやり取りは10秒間では終わらないので、ここでは1秒ごとに600回（10分間）繰り返すように指定しておく。

タイマーで自動化させたい動きは・・・

- ① サーバーからメッセージを取ってくる（読む）。
- ② それを出力欄に表示する。

まず、サーバーからメッセージを取ってくる（読む）には・・・ **サーバー！”目印名”読む。**とするルールがある。

取ってくる内容の目印名は“moji”なので、ルールに従うと・・・

- ①サーバー！“moji”読む。

それを出力欄に表示するには・・・

- ②出力欄！（ ）書く。 → ※ **名前！命令。** の形。必要に応じて（ ）が入る。
↑
表示させたいもの

つまり、①と②を組み合わせると・・・出力欄！（サーバー！“moji”読む）書く。

さて、タイマーを動かすには・・・ **タイマー名！「名前！命令」実行。** と書く必要があった。

よって、タイマーに自動化させたい動きを設定すると・・・

7行目 **自動受信タイマー！「出力欄！（サーバー！“目印”読む）書く」実行。**となる。しかし、実行すると大きな問題が・・・。これを解決していこう。

④ここまでの問題点を解決する

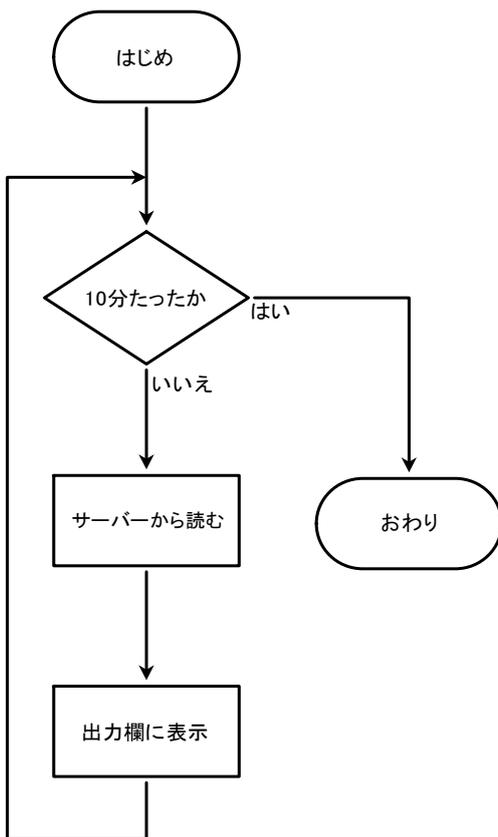
作成したプログラムを実行してみて、問題点をあげてみよう。

- ・入力した文字（同じメッセージ）が何度も表示される
- ・誰のメッセージかわからない
- など

（ 同じメッセージが何度も表示される ）の解決

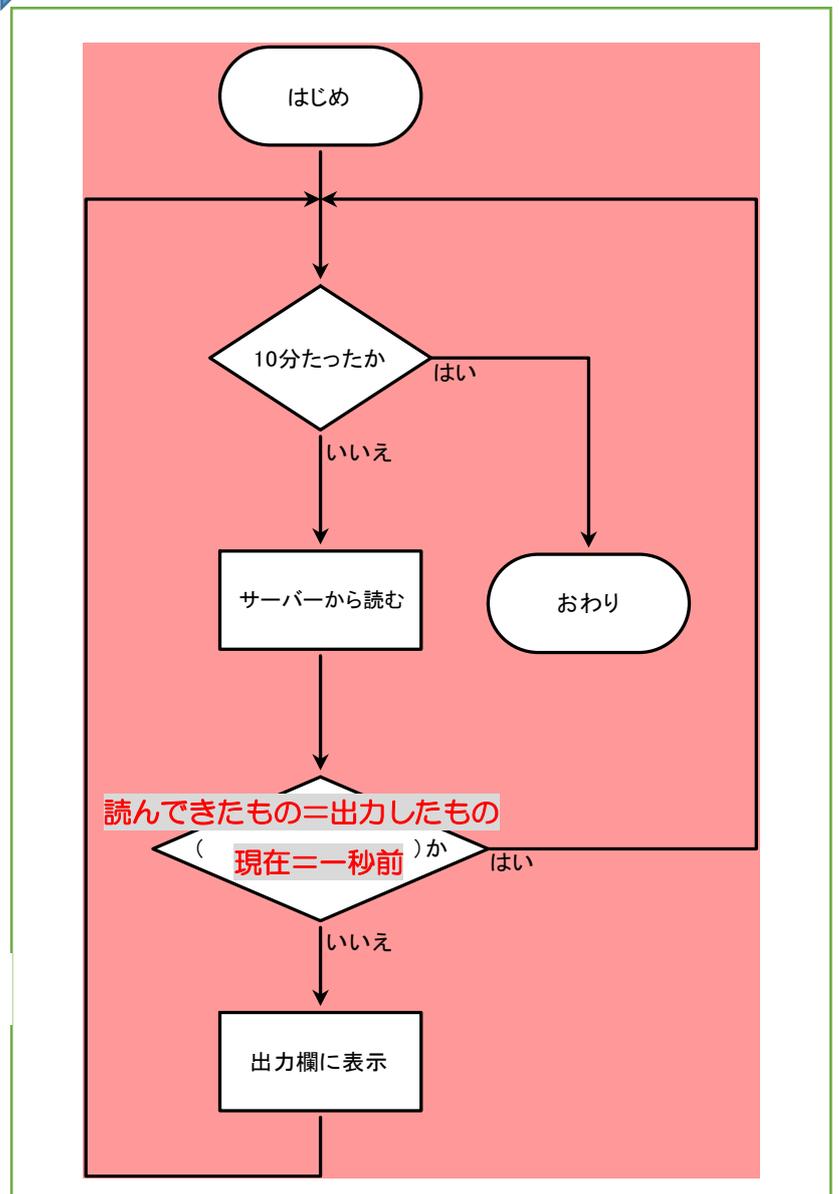
フローチャートで考えよう。

現在のフローチャート（受信用）



※1秒ごとに出力欄に表示している。

変更したフローチャート（受信用）



プログラミングの方針（例：誰のメッセージかわからない）の解決のために・・・

1行目 サーバー！”172.28.100.67”接続。

2行目 入力欄=フィールド！作る。

3行目 送信ボタン=ボタン！”送信”作る。

4行目 出力欄=リスト！作る。

5行目 送信ボタン：動作=「サーバー！”moji”（入力欄！読む）書く。入力欄！クリア。」。

6行目 自動受信タイマー=タイマー！作る 1秒 間隔 600 回数。

7行目 :一秒前=” ”。

8行目 :現在=” ”。

9行目 自動受信タイマー！「

:現在=サーバー！”moji”読む。

10行目 「:現在!=:一秒前」！なら「出力欄！（サーバー！”moji”読む）書く。」実行。

11行目 :一秒前=:現在。

12行目 」実行。

4と5行目の間 名前=”カメ太”。

新5行目 送信ボタン：動作=「サーバー！”moji”（名前！”:”（入力欄！読む）連結）書く。入力欄！クリア。」。

今後の学習（さらに深めよう！！）

★現在残っている問題点は・・・

- ・（例：なりすまし、誹謗中傷やウソ・デマ、過去のメッセージや履歴）
 - ・（例：表示の仕方、見やすくする、時間制限）
- などがある。それらの問題点を解決するためのアイデアを出し合おう。

★チャットプログラムからわかる情報セキュリティの問題について、意見を出し合おう。
また、ネットワークを利用するときの注意点をまとめよう。

※赤文字は解答例で生徒が記入したところです。

※ この冊子を活用した時の授業の流れは、下記の指導計画（6～8時間）で行いました。

題材名 「双方向性のあるコンテンツのプログラミングと情報セキュリティ」

題材の指導計画（全12時間）

時間	学習活動
1～5	<ul style="list-style-type: none"> ・情報を処理する手順を知り、目的に合った手順を考える。 ・プログラミングの基礎を知る。 ・プログラミングを行う。
6～8	<ul style="list-style-type: none"> ・情報通信ネットワークで情報を伝える仕組みを知る。 ・チャットのプログラミングを行う。
9	<ul style="list-style-type: none"> ・ネットワークの危険性と安全に利用するための仕組みを知る。
10	<ul style="list-style-type: none"> ・安心、安全に利用できるチャットプログラムへの工夫・改善と生活の中での情報セキュリティを考える。
11～12	<ul style="list-style-type: none"> ・インターネットの特性を知り、情報モラルと知的財産について考える。

6時目 情報通信ネットワークについて
プログラミング(①サーバーに接続する ②部品を作る)
冊子の1ページ ～ 3ページ

7時目 プログラミング(③部品を操作できるようにする)
冊子の4ページ ～ 6ページ

8時目 プログラミング(④ここまでの問題を解決する)
冊子の7ページ ～ 8ページ

《参考文献》

・佐賀県小城・多久地区技術家庭科部会(技術分野)

『全国大会へ向けての発表資料』

・兼宗 進・久野 靖 『ドリトルで学ぶプログラミング』〔第2版〕 平成23年4月



ネットを安心、安全に利用するためには

どんな情報セキュリティが必要だろうか？

情報セキュリティとは・・・

1 「問い」に対する答えや問題点・危険な点などを自分なりに書いてみよう



2 自分が担当する情報セキュリティの仕組み (A ・ B ・ C ・ D)

・資料のポイントや問いの答えの参考になることを書こう (エキスパート活動)

・何のために(どのような危険に対するしくみか)・何をするのか(どのような働きか)など



5 今日の授業を振り返り、「問い」に対する答えをまとめよう

(これからネットを利用する時、自分はどうかを具体的に書いてみよう)



(クロストーク活動)

4 各グループで「問い」に対する答えや問題点・危険性に対する対策をまとめてみよう



3 各パートからのポイントをメモしよう (ジグソー活動)

A：不正侵入を防ぐ技術

B：コンピュータウイルスに対する技術

C：情報を安全に送受信する技術

D：違法・有害情報に対する技術

A：不正侵入を防ぐ技術
B：コンピュータウイルスに対する技術
C：情報を安全に送受信する技術
D：違法・有害情報に対する技術

安心、安全に利用できるチャットプログラムに工夫・改善しよう

自分がすべき情報セキュリティをまとめよう

今まで解決してきた問題点

- ・ 同じメッセージが何度も繰り返し表示される
 - 条件分岐命令 (同じメッセージでないなら、表示させる)
- ・ 誰のメッセージかわからない
 - 送信者の名前を表示 (送信ボタンの動作に連結命令)

① 制作したチャットプログラムの問題点をあげてみよう

② 問題点を解決するためにはどのようなプログラムにしたらよいだろうか、話し合ってみよう

③ 生活の中でインターネットを利用する際に、自分が行う情報セキュリティをあげてみよう

④ 今日の授業を振り返っての感想を書こう
