明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

プログラミング演習2
ネットワーク通信

中村，小松，菊池

研究室配属（案）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 研究室への配属人数は等分配
  - 2014年4月の2年次ゼミの研究室は14研究室
    - 阿原，荒川，菊池，小林，小松，嵯峨山，鹿喰，鈴木，中村，橋本，福地，宮下，渡邊
    - 191/14 = 13.6 (人/研究室)
    - 希望者数に応じ人数が13人または14人
  - 2015年4月の3年次ゼミの研究室は15研究室
    - 阿原，荒川，五十嵐，菊池，小林，小松，嵯峨山，鹿喰，鈴木，中村，橋本，福地，宮下，渡邊
    - 191/15 = 12.7 (人/研究室)
    - 希望者数に応じ人数が12人または13人

研究室配属（案）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 2年次は希望を重視
  - 学生は希望研究室順に1～15を付与（2014は1～14）
  - 希望順位に応じて，枠が一杯になるまで研究室に配属
  - 希望順位が同じで，研究室の人数があふれた場合は，1年次と所属が違う学生を優先．さらにあふれた場合は成績順で優先順位を決定（前後期のGPAを予定）
- 3年次は学生の希望と，教員評価点を考慮
  - 学生は希望研究室順に1～15を付与
  - 教員は各自の優先する何らかの指標（数学重視，プログラミング重視，英語重視など）に基づき学生に評価点を付与
  - 安定結婚のためのGSアルゴリズムで研究室に配属
- 4年次は，3年次の研究室にそのまま所属
  - 3年次から4年次で研究室変更は無し

研究室配属（案）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 日程（案）
  - 詳細は12月下旬～1月上旬に正式開示
  - 研究室で何をやるかという説明会も12月下旬～1月上旬に実施予定
  - 1月試験前に希望調査〆切
  - 4月1日までには配属決定通知

使ってますよね？

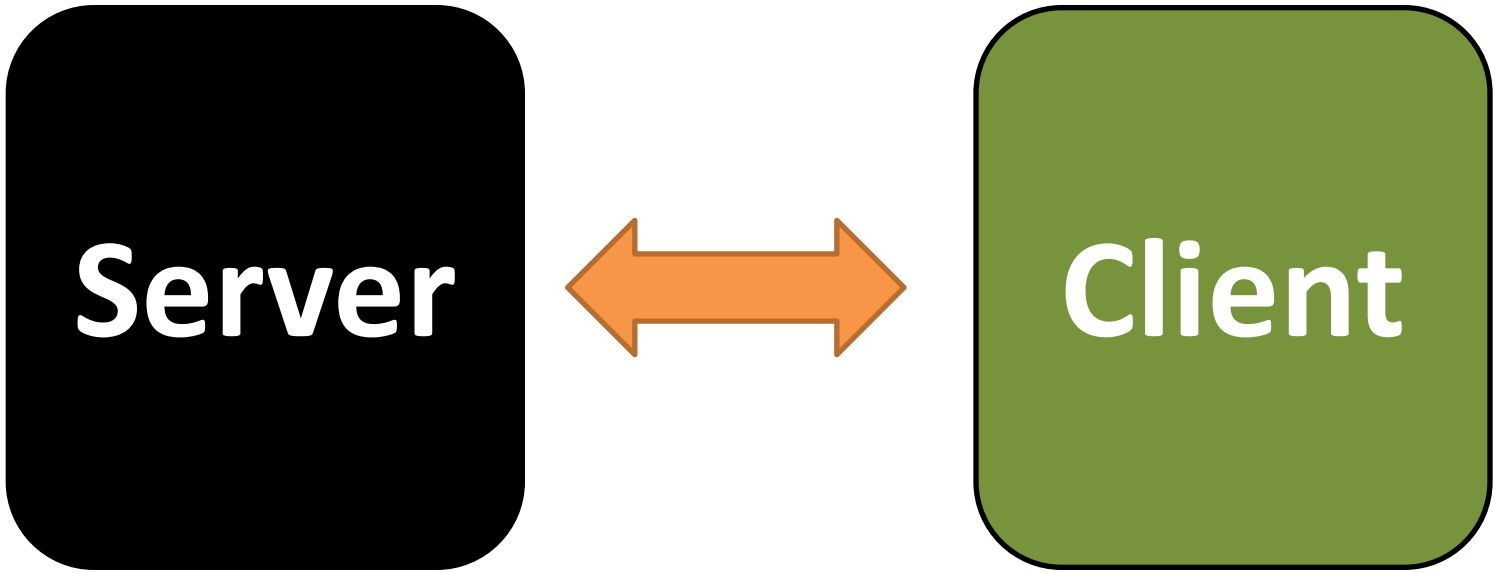
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- LINE
- Twitter
- Facebook
- Skype
- FaceTime

本日の内容

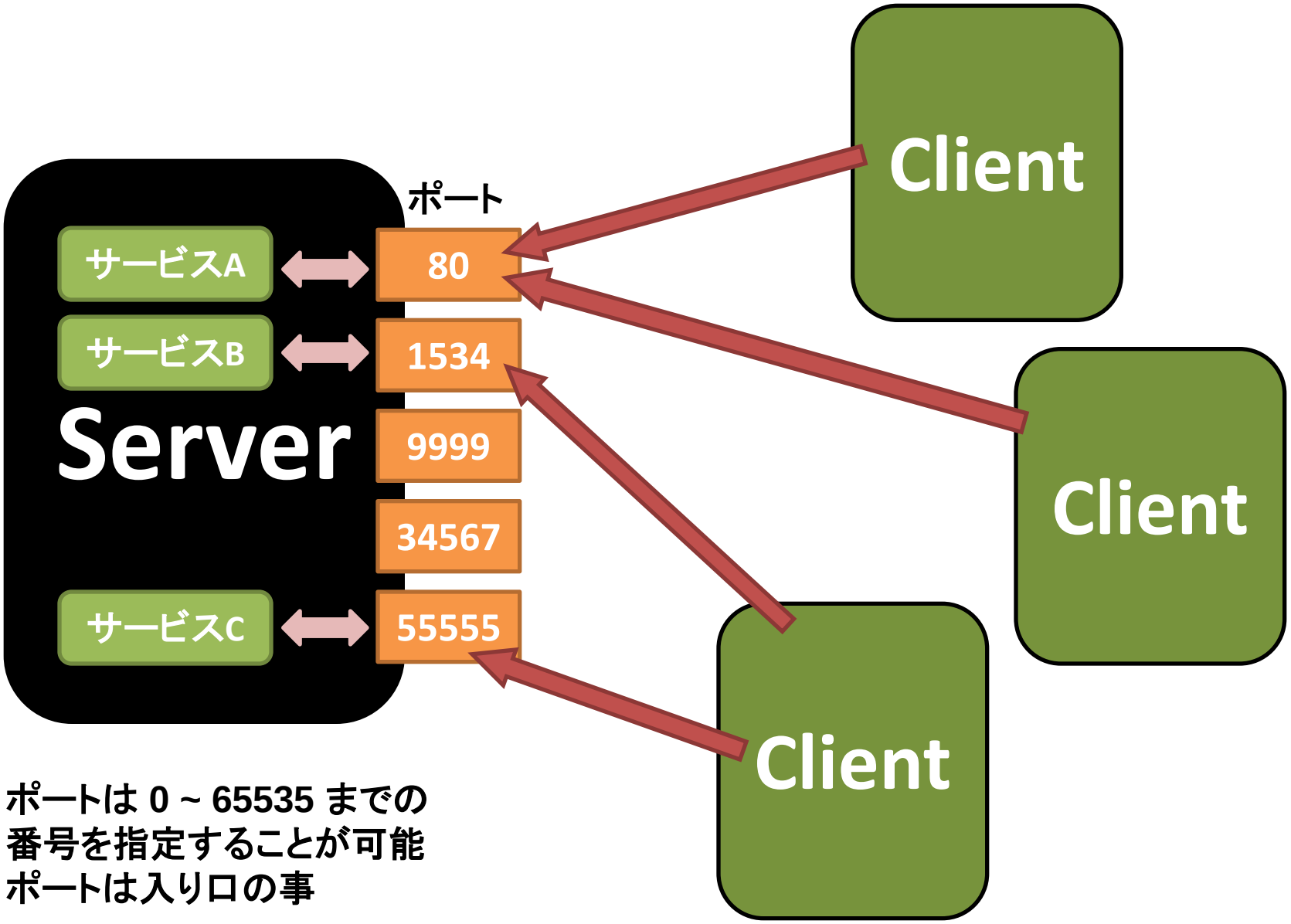
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ネットワーク通信で情報のやりとりをしよう



通信の基本

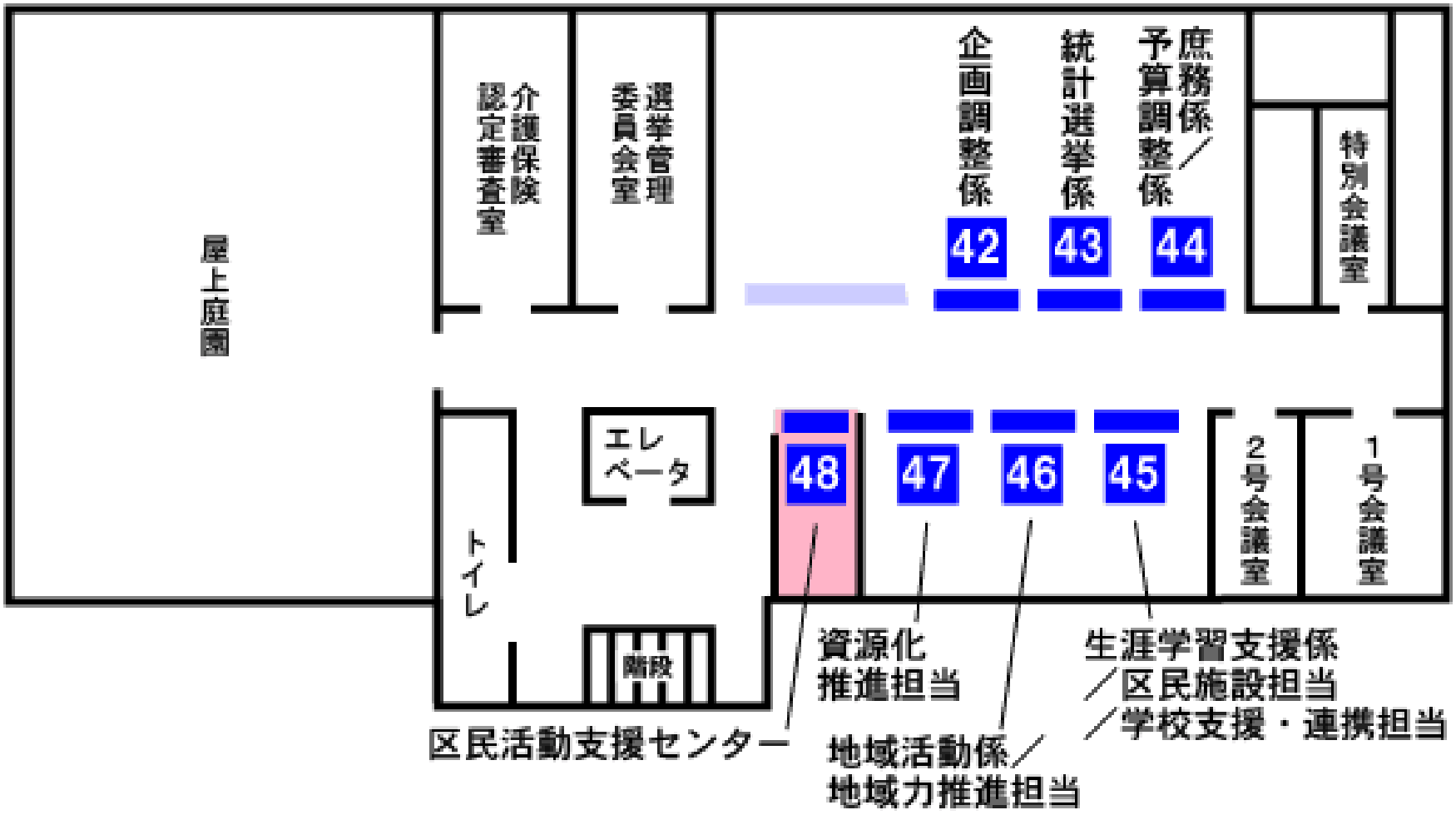
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



ポートは 0 ~ 65535 までの
番号を指定することが可能
ポートは入り口の事

窓口とサービスの対応関係

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



http://www.city.yokohama.lg.jp/kohoku/kyoutuu/mado/mado-48.html

市役所で考えてみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

1. 役所の係の人が窓口を開けて準備する
2. 市民が役所（の住所）に行き，窓口に並ぶ
3. 窓口の人が最初に並んでいる市民を呼ぶ
4. 窓口で，役所の人と，市民が対話して何らかの処理を行う

サーバ・クライアント

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

1. サーバがポートを開けて準備する
2. クライアントがサーバのアドレスとポート番号を指定してサーバに接続要求を行う
3. サーバがクライアントに対して接続許可を出すことで，情報交換のための経路が確立される
4. サーバとクライアント間でやりとりが行われる

注意点

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ファイル名やフォルダ名にServerやClientという文字列を含まない事
- 下記をプログラムに入れること

```
void stop(){
    myClient.stop();
}
```

```
void stop(){
    myServer.stop();
}
```

- 問題が解決しない場合は再起動すること
  - ファイアウォールに関する問題が出てきたら、再起動しないとダメな模様

とりあえず書いてみよう

明治大学総合数理学部

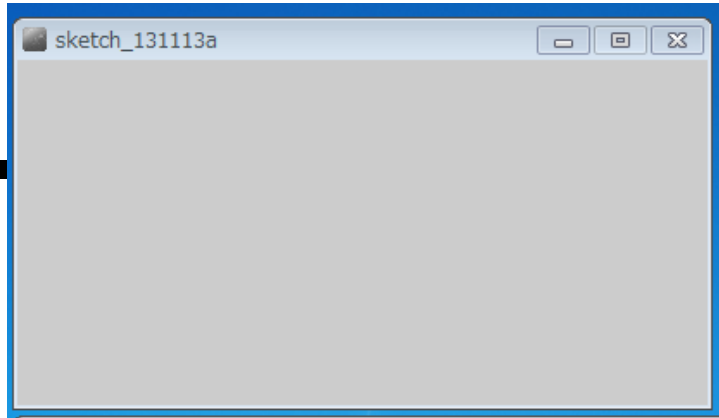
サーバ（Server）

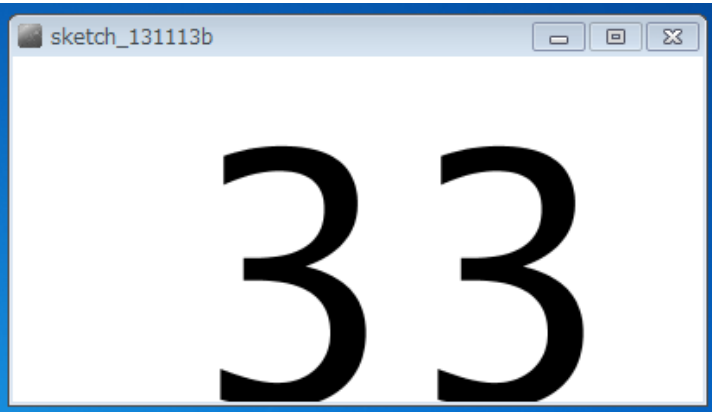
```
import processing.net.*;
Server myServer
                = new Server(this, 12345);
int count = 0;

void setup() {
   size(400, 200);
}
void draw() {
}
void stop(){
   myServer.stop();
}
void mousePressed() {
   count++;
   myServer.write(count);
}
```

クライアント（Client）

```
import processing.net.*;
Client myClient = new Client( this,
                        "127.0.0.1", 12345 );
int recvInt;
void setup() {
   size(400, 200);
   textSize( 200 );
   fill( 0 );
}
void stop(){
   myClient.stop();
}
void draw() {
   background( 255 );
   if (myClient.available() > 0) {
      recvInt = myClient.read();
   }
   text( recvInt, 100, 200 );
}
```

127.0.0.1 は
自分という意味





サーバ（Server）

クライアント（Client）

サーバ（Server）から先に起動すること！
（窓口が準備できていないとだめ！）

サーバ側でクリック連打してみよう！！

とはいえ

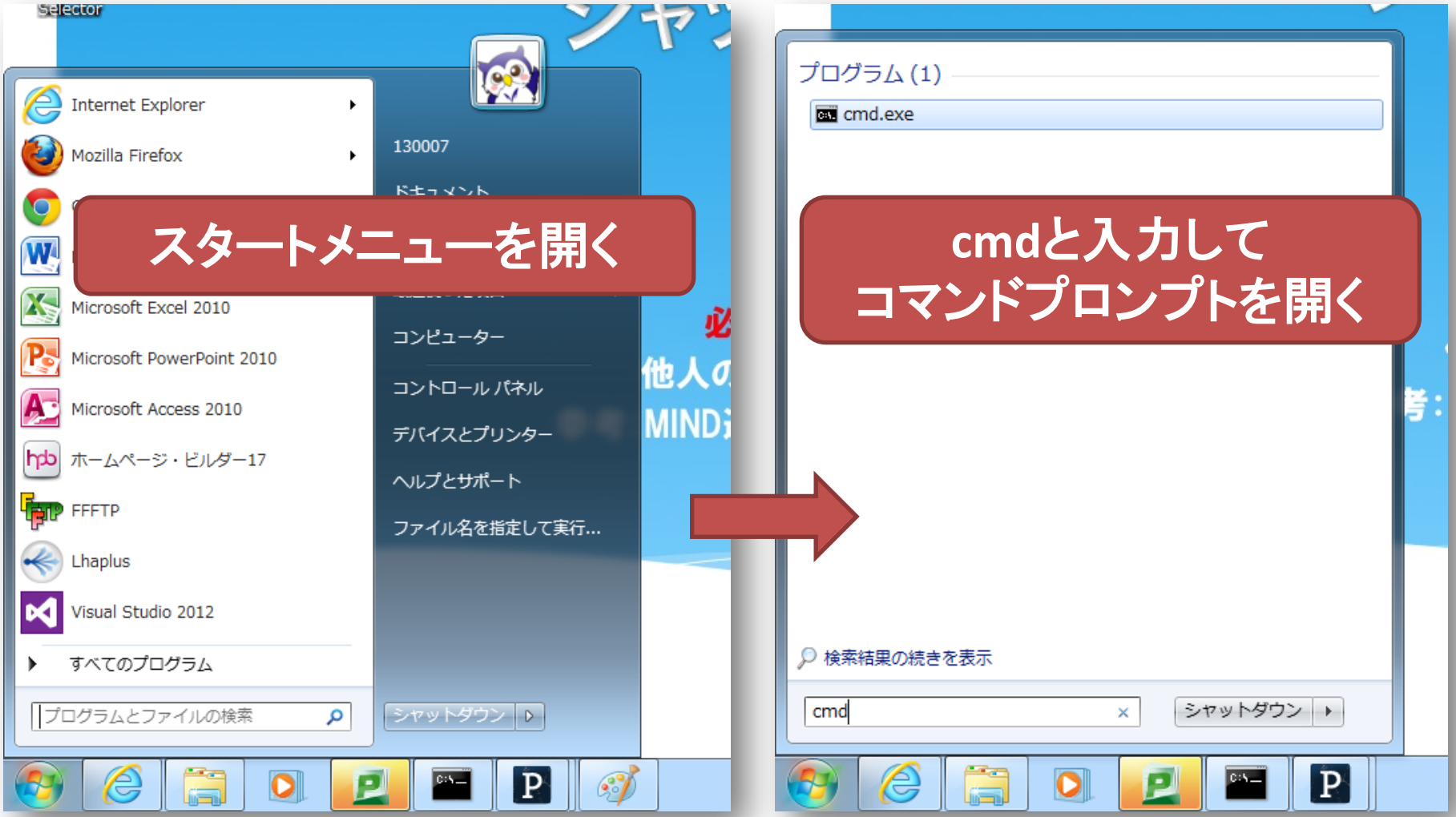
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- なんだか，自分の中でServerとかClientとか言ってもよくわからない．

- 隣の人とペアを作って接続してみよう！
  - 隣の人がいない場合は，前後の余っている人同士でやってみる！
  - 一方はサーバ役，他方はクライアント役
    - 交互にやりましょう
  - まずは，隣の人のアドレス（住所）を聞く！
    - 実際の住所ではなく，ネットワーク上の住所

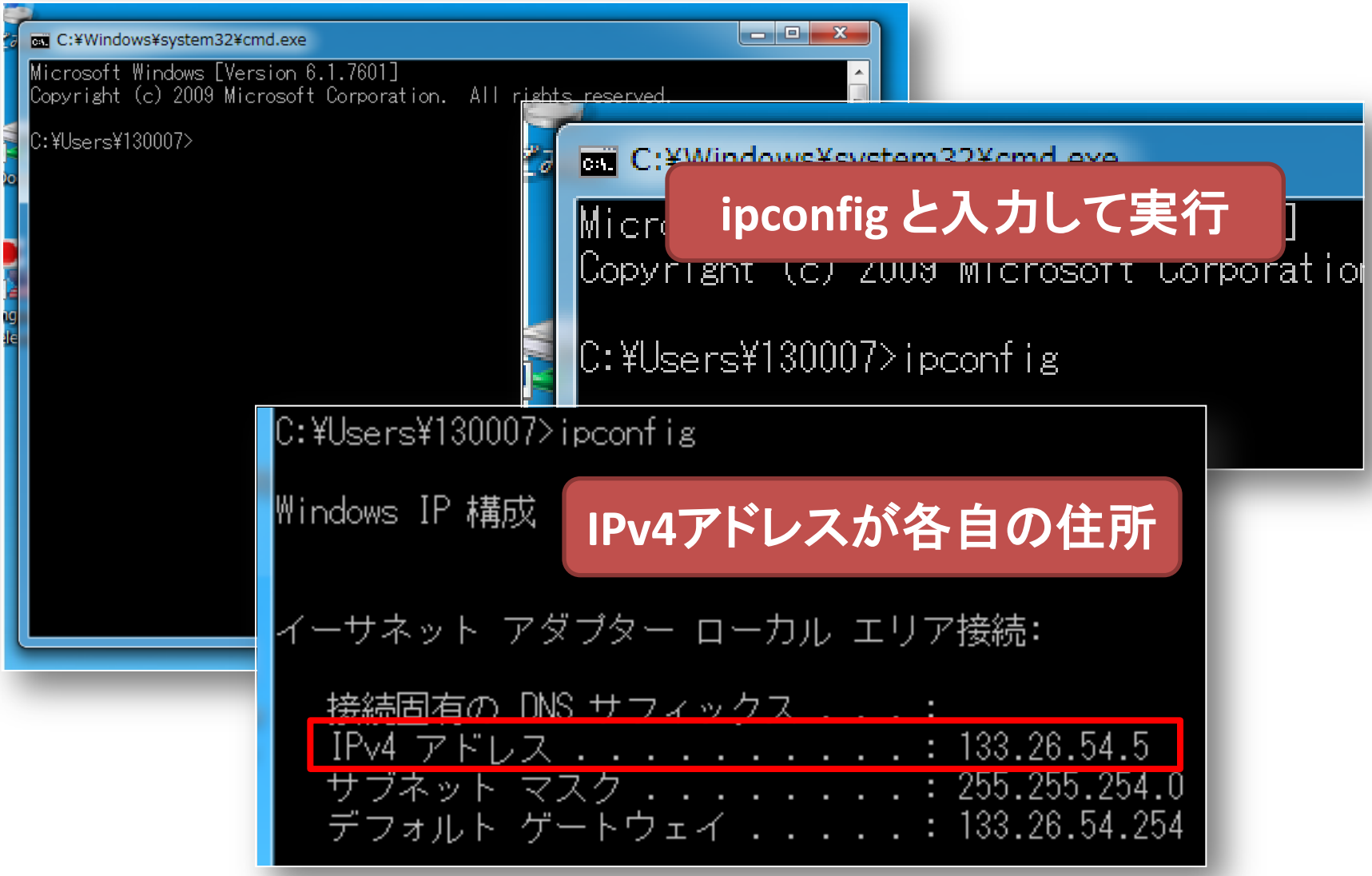
隣に接続してみよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



隣に接続してみよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



IPv4アドレスを隣の人に伝えよう

とりあえず書いてみよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
import processing.net.*;
Client myClient  = new Client( this, "133.26.54.5", 12345 );
int dataIn;

void setup() {
    size(400, 200);
    textSize( 200 );
    fill( 0 );
}
void stop(){
    myClient.stop();
}
void draw() {
    background( 255 );
    if (myClient.available() > 0) {
        dataIn = myClient.read();
    }
    text( dataIn, 100, 200 );
}
```

Server役の人の
IPアドレスが133.26.54.5の場合

クライアント（Client）

準備と相互接続

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サーバ･クライアント用のライブラリの準備

```
import  processing.net.*;
```

- サーバの準備＋接続待ち

```
Server  myServer = new Server( this,
                                接続待ちポート番号 );
```

- クライアントの準備＋接続

```
Client  myClient = new Client( this,
        "接続先アドレス", 接続先ポート番号 );
```

メッセージの送信

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サーバからクライアントへのメッセージの送信（整数または文字列を送信）

```
myServer . write( 100 ) ;
myServer . write( "hogehoge" ) ;
```

- クライアントからサーバへのメッセージの送信（整数または文字列を送信）

```
myClient . write( 100 ) ;
myClient . write( "hogehoge" ) ;
```

クライアントでのデータ受信

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- クライアントでのデータ受信（整数）

```
int  recvInt = myClient . read() ;
```

- クライアントでのデータ受信（文字列）

```
String  recvStr = myClient . readString() ;
```

- クライアントでのデータ受信量取得

```
int  size = myClient . available() ;
```

サーバでのデータ受信

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- データ受信待ちしているクライアントの取得

```
Client  nextClient = myServer . available() ;
```

- あるクライアントからのデータ受信（整数）

```
int  recvInt = myClient . read() ;
```

- あるクライアントからのデータ受信（文字列）

```
String  recvStr = myClient . readString() ;
```

通信終了

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サーバからの切断

```
myClient . stop() ;
```

- 特定のクライアントの切断

```
myServer . disconnect( thisClient ) ;
```

- サーバのサービスの終了

```
myServer . stop() ;
```

解説

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

133.26.54.5 さん

ライブラリ準備

```
import processing.net.*;
Server myServer
                = new Server(this, 12345);
int count = 0;

void setup() {
   size(400, 200);
}
void draw() {
}
void stop(){
   myServer.stop();
}
void mousePressed() {
   count++;
   myServer.write(count);
}
```

12345番ポートで準備

クリック時にcountを増やし
クライアントにcountを送信！

133.26.54.6 さん

ライブラリ準備

```
import processing.net.*;
Client myClient = new Client( this,
                    "133.26.54.5", 12345 );
int recvInt;
void setup() {
   size(400, 200);
   textSize( 200 );
   fill( 0 );
}
void stop(){
   myClient.stop();
}
void draw() {
   background( 255 );
   if (myClient.available() > 0) {
      recvInt = myClient.read();
   }
   text( recvInt, 100, 200 );
}
```

133.26.54.5さんの
12345番ポートに接続

avialable() で受信
しているかチェック
0より大きければ受信

read() で値を受信して表示！

送受信の時に・・・

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- どのようなフォーマット（形式）でデータを送受信するかを考える
- クリック数は1つの数字だけだが，ある座標を送受信しようと思うと，2つの値が必要となるため困ったことに

```
myServer . write( mouseX ) ;
myServer . write( mouseY ) ;
```

```
myServer . write( mouseX + "," + mouseY ) ;
```

送受信の時に・・・

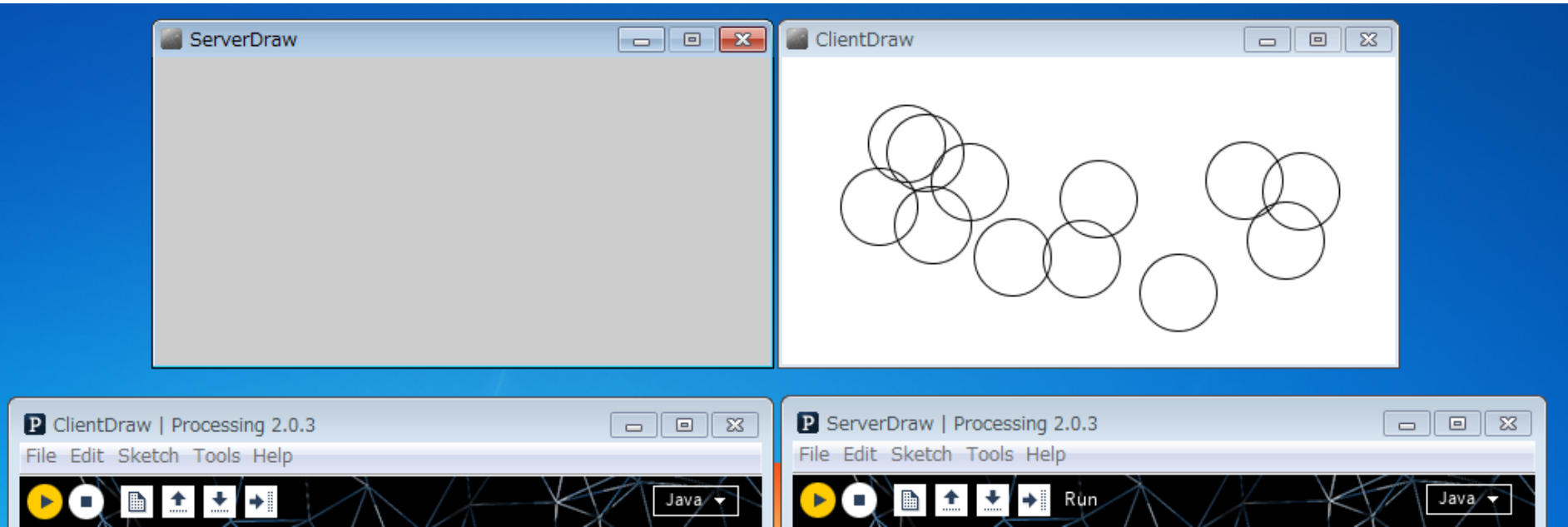
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

1. カンマ区切りで文字列として送信（他でもOK）
   - "x座標,y座標"（例）100,230
   - XXXX **. write(mouseX+","+mouseY);**
2. 文字列として受信
   - String recvStr = YYYY **. readString();**
3. 受信した文字列（カンマ区切り）を，splitを利用してカンマで分割（"100,230" ➔ "100" と "230" に）
   - String [] data = **split(recvStr, ',');**
4. 分割された文字列を整数に戻す
   - int x = **int(data[0]);**
   - int y = **int(data[1]);**

クライアントに描画する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サーバで操作することで，クライアント上に描画してみる
  - サーバとクライアントで相互接続
  - サーバ上でクリックされた位置をクライアントに送信
  - クライアントで送信されてきた座標に円を描画



サーバから描画してみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

サーバ（Server）

```
// ライブラリを読み込む
import processing.net.*;
// サーバを2222ポートでスタート
Server myServer = new Server( this, 2222 );

void setup() {
  size(400, 200);
  noFill();
}

void mousePressed() {
  // クリック座標をカンマ区切りで送信
  myServer.write(mouseX+","+mouseY);
}

void stop(){
   myServer.stop();
}
```

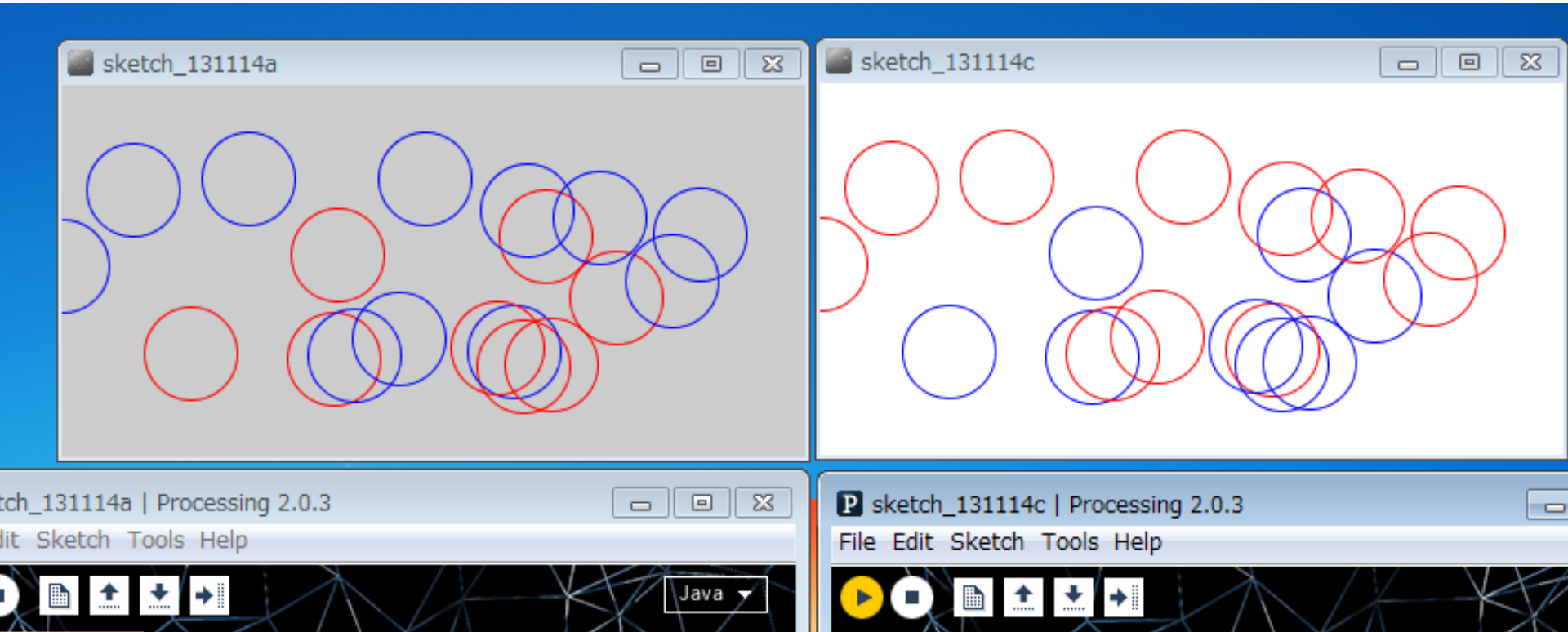
クライアント（Client）

```
// ライブラリを読み込む
import processing.net.*;
// 133.26.54.5 の 2222 番に接続
Client myClient = new Client(this, "133.26.54.5", 2222);
void setup() {
  size(400, 200);
  noFill();
  background( 255 );
}
void stop(){
   myClient.stop();
}
void draw() {
  // 何か受信しているか確認
  if (myClient.available() > 0) {
    // 文字列として受信
    String msg = myClient.readString();
    // 文字列をカンマ区切りで分割し整数に戻す
    String [] data = split( msg, ',' );
    ellipse( int(data[0]), int(data[1]), 50, 50 );
  }
}
```

クライアントに描画してみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 一方向だとなんだか面白く無い
- 双方向にしてみよう！
  - サーバでクリックするとクライアントに，クライアントでクリックするとサーバに！



双方向で描画してみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

サーバ（Server）

```
import processing.net.*;
Server myServer = new Server( this, 2222 );
void setup() {
  size(400, 200);
  noFill();
}
void mousePressed() {
  myServer.write(mouseX+","+mouseY);
}
void stop(){
   myServer.stop();
}
void draw(){
  // データ送信しているクライアントの確認
  Client nextClient = myServer.available();
  // != null はクライアントがあるという意味
  if( nextClient != null ){
    // readString() でデータ受信＆分割＆描画
    String recvStr = nextClient.readString();
    String [] data = split( recvStr, ',' );
    ellipse( int(data[0]), int(data[1]), 50, 50 );
  }
}
```

クライアント（Client）

```
import processing.net.*;
Client myClient = new Client(this, "133.26.54.5", 2222);

void setup() {
  size(400, 200);
  noFill();
  background( 255 );
}

void mousePressed(){
  myClient.write( mouseX+","+mouseY );
}

void stop(){
   myClient.stop();
}
void draw() {
  if (myClient.available() > 0) {
    String recvStr = myClient.readString();
    String [] data = split( recvStr, ',' );
    ellipse( int(data[0]), int(data[1]), 50, 50 );
  }
}
```

双方向で描画してみる v.2

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

サーバ（Server）

```
import processing.net.*;
Server myServer = new Server( this, 2222 );
void setup() {
  size(400, 200);
  noFill();
}
void mousePressed() {
  myServer.write(mouseX+","+mouseY);
  stroke( 255, 0, 0 );
  ellipse( mouseX, mouseY, 50, 50 );
}
void stop(){
   myServer.stop();
}
void draw(){
  Client nextClient = myServer.available();
  if( nextClient != null ){
    String recvStr = nextClient.readString();
    String [] data = split( recvStr, ',' );
    stroke( 0, 0, 255 );
    ellipse( int(data[0]), int(data[1]), 50, 50 );
  }
}
```

クライアント（Client）

```
import processing.net.*;
Client myClient = new Client(this, "133.26.54.5", 2222);
void setup() {
  size(400, 200);
  noFill();
  background( 255 );
}
void mousePressed(){
  myClient.write( mouseX+","+mouseY );
  stroke( 255, 0, 0 );
  ellipse( mouseX, mouseY, 50, 50 );
}
void stop(){
   myClient.stop();
}
void draw() {
  if (myClient.available() > 0) {
    String recvStr = myClient.readString();
    String [] data = split( recvStr, ',' );
    stroke( 0, 0, 255 );
    ellipse( int(data[0]), int(data[1]), 50, 50 );
  }
}
```

クライアントから命令！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サーバ役とクライアント役，どちらかがどちらかになって下さい．また，IPアドレスをそれぞれサーバ役の人のにし，サーバを実行し，次にクライアントを実行してみよう！

サーバ

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
import processing.net.*;
import java.awt.*;
import java.awt.event.*;
Server myServer = new Server( this, 3333 );
Robot  robot;

void setup() {
  size(20, 20);
  try {
    robot = new Robot();
  } catch ( Exception e ) {
  }
}

void stop(){
   myServer.stop();
}
```

3333番ポートで準備

```
void draw() {
  Client nextClient = myServer.available();
  if ( nextClient != null ) {
    String recvStr = nextClient.readString();
    String [] data = split( recvStr, ',' );
    if ( data[0].equals( "move" ) == true ) {
      doMove( int( data[1] ), int( data[2] ) );
    } else if (data[0].equals( "click" ) == true ) {
      doClick( int( data[1] ), int( data[2] ) );
    }
  }
}

void doClick( int x, int y ) {
  robot.mousePress( InputEvent.BUTTON1_MASK );
  delay(1);
  robot.mouseRelease( InputEvent.BUTTON1_MASK );
  delay(1);
}

void doMove( int x, int y ) {
  robot.mouseMove( x, y );
}
```

最初のカンマの前が
動作に関すること

クライアント

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
import processing.net.*;
Client myClient = new Client( this, "133.26.54.5", 3333 );

void setup() {
  size( 1200, 800 );
}
void stop(){
   myClient.stop();
}
void draw() {
}
// マウスが押された時
void mousePressed() {
  // "click,100,80" のように送信
  myClient.write( "click,"+mouseX+","+mouseY);
}
// マウスが移動した時
void mouseMoved() {
  // "move,100,80" のように送信
  myClient.write( "move,"+mouseX+","+mouseY );
}
```

133.26.54.5さんの
3333番ポートに接続

write() で
メッセージを送信

やってること

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サーバのマウスを操作してみよう！
  - "move,x,y" でマウスの移動情報を送信
  - "click,x,y" でマウスのクリック情報を送信

- トロイの木馬でやっているのは，これの更に高度なこと

LINEもどきを作ってみよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- スタンプを相手に貼り付けよう！

サーバ


```
import processing.net.*;
Server myServer = new Server( this, 5555 );
PImage [] stamp = new PImage [7];
int [] stampMessage = new int [10];
int [] stampFrom = new int [10];
int lastMessageNum = -1;

void setup() {
  size(300, 800);
  stamp[0] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/40/580x580x42472428b7971cefc004cf3d.jpg" );
  stamp[1] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/38/197x212x553fc9733bf33ba25fabc8de.jpg" );
  stamp[2] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/61/260x260x0860b5b9e2ff67b2d7f02278.jpg" );
  stamp[3] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/22/339x246xdfc75cc1c6b692462c54c6e7.jpg" );
  stamp[4] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/29/399x351x7d40231ed0ad556af902e19d.jpg" );
  stamp[5] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/30/232x222xb26506eb0d3173a63f8070f2.jpg" );
  stamp[6] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/128/228x230x425a49f2bac2e3770f407435.jpg" );
}

void mousePressed() {
  lastMessageNum++;
  if ( lastMessageNum == 8 ) {
    lastMessageNum = 0;
  }

  int id = 1 + (int)(mouseX / (width/7) );
  myServer.write( id );
  stampMessage[lastMessageNum] = id;
  stampFrom[lastMessageNum] = 1;
}
void stop(){
   myServer.stop();
}
void draw() {
  background( 255 );
  Client nextClient = myServer.available();
  if ( nextClient != null ) {
    int id = nextClient.read();
    lastMessageNum++;
    if ( lastMessageNum == 8 ) {
      lastMessageNum = 0;
    }
    stampMessage[lastMessageNum] = id;
    stampFrom[lastMessageNum] = 2;
  }

  for ( int i=0; i<8; i++ ) {
    int id = stampMessage[i];
    if ( stampFrom[i] == 1 ) {
      // 1なら左に表示
      image( stamp[id-1], 180, i*100, 100, 100 );
    }
    else if ( stampFrom[i] == 2 ) {
      // 2なら右に表示
      image( stamp[id-1], 20, i*100, 100, 100 );
    }
  }
}
```

クライアント


```
import processing.net.*;
//Client myClient = new Client(this, "133.26.54.5", 2222);
Client myClient = new Client(this, "127.0.0.1", 5555);
PImage [] stamp = new PImage [7];
int [] stampMessage = new int [10];
int [] stampFrom = new int [10];
int lastMessageNum = -1;

void setup() {
 size(300, 800);
 background( 255 );
 stamp[0] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/40/580x580x42472428b7971cefc004cf3d.jpg" );
 stamp[1] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/38/197x212x553fc9733bf33ba25fabc8de.jpg" );
 stamp[2] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/61/260x260x0860b5b9e2ff67b2d7f02278.jpg" );
 stamp[3] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/22/339x246xdfc75cc1c6b692462c54c6e7.jpg" );
 stamp[4] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/29/399x351x7d40231ed0ad556af902e19d.jpg" );
 stamp[5] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/30/232x222xb26506eb0d3173a63f8070f2.jpg" );
 stamp[6] = loadImage( "http://imgcc.naver.jp/kaze/mission/USER/20121015/78/718648/128/228x230x425a49f2bac2e3770f407435.jpg" );
}

void mousePressed() {
 lastMessageNum++;
 if ( lastMessageNum == 8 ) {
   lastMessageNum = 0;
 }

 int id = 1 + (int)(mouseX / (width/7) );
 myClient.write( id );
 stampMessage[lastMessageNum] = id;
 stampFrom[lastMessageNum] = 1;
}
void stop(){
   myClient.stop();
}
void draw() {
 background( 255 );
 if (myClient.available() > 0) {
   int id = myClient.read();
   lastMessageNum++;
   if ( lastMessageNum == 8 ) {
     lastMessageNum = 0;
   }

   stampMessage[lastMessageNum] = id;
   stampFrom[lastMessageNum] = 2;
 }

 for ( int i=0; i<8; i++ ) {
   int id = stampMessage[i];
   if ( stampFrom[i] == 1 ) {
     // 1なら左に表示
     image( stamp[id-1], 180, i*100, 100, 100 );
   }
   else if ( stampFrom[i] == 2 ) {
     // 2なら右に表示
     image( stamp[id-1], 20, i*100, 100, 100 );
   }
 }
}
```

サーバに情報を送ろう！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- クリック回数を数えた結果をサーバに送信するクライアントプログラムを作成し，クリックの度にサーバに対してクリック数を送信せよ
  - write(count) で整数データを送信せよ
  - ポート番号は 54321 とする

サーバの負荷を減らすのは大変

サーバに情報を送ろう！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

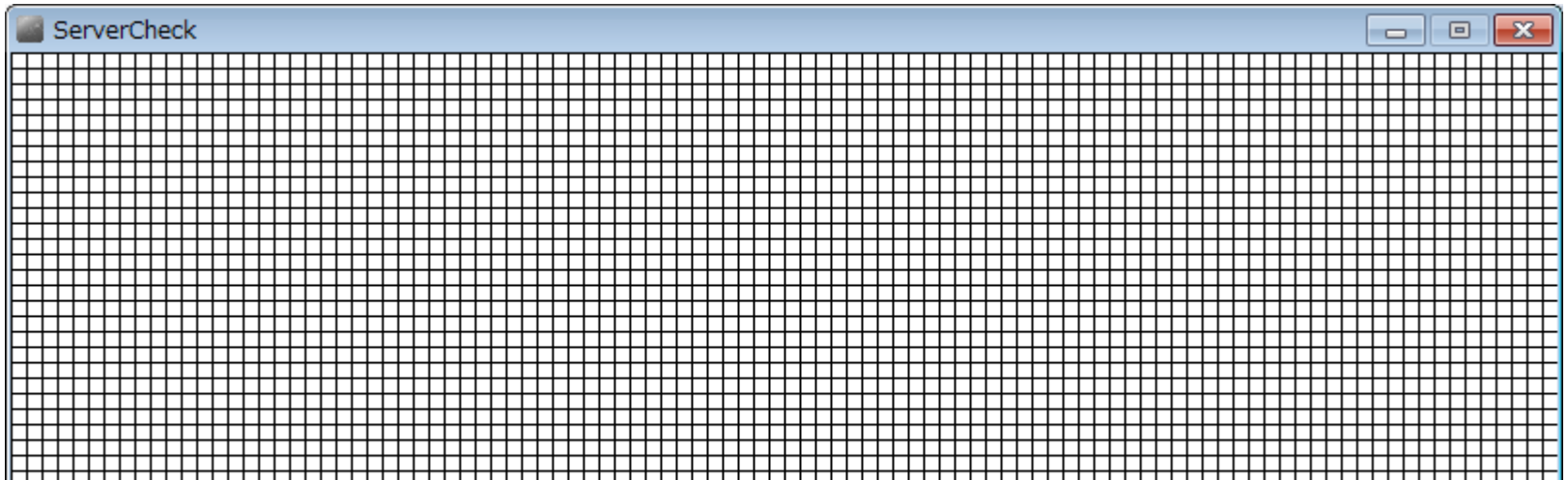

```
import processing.net.*;
Client myClient = new Client(this,                , );
int click = 0;
void setup() {
   size( 100, 100 );
}
void stop(){
   myClient.stop();
}
void draw(){
}

void mousePressed() {
   click++;
   myClient.write(          );
}
```

陣取りゲームを作ってみよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 100x100のマス目のオセロのような陣取りゲーム
- クライアント上には100x100のマス目を表示
- マウスクリックした場所を，サーバ（教卓のIPアドレス）に接続し，サーバに対して自身の組内番号と，マス目のx, y座標（それぞれ0～79）を送信する
- ポート番号 6666，送信フォーマット "5,52,25" のようにカンマ区切りとする

番号　座標




```
import processing.net.*;
Client myClient = new Client( this,                    ,          );
int unitSize = 8;
int ux = 100;
int uy = 100;
int [][] unit;
int myID =          ;

void setup() {
   size( 800, 800 );
   stroke( 0 ) ;
   fill( 255 ) ;
}

void draw() {
   for ( int x=0; x<ux; x++ ) {
      for ( int y=0; y<uy; y++ ) {
         rect( x*unitSize, y*unitSize, unitSize, unitSize );
      }
   }
}

void mousePressed() {
   int x = mouseX / unitSize;
   int y = mouseY / unitSize;
   myClient.write(                    );
}
```

宿題

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- なんでも良いので，サーバクライアントで動作する仕組みを作ってみましょう
  - 提出期限は11月25日の講義の最初の5分まで
  - プログラム名は ServerApp と ClientApp とし，組番号のフォルダにまとめ，宿題フォルダに提出
  - （例）
    - ピンポンゲーム（サーバとクライアントで跳ね返す）
    - 占い（サーバが占い結果を送信する）
    - ランキングシステム（クリック数を集約する）
    - 落書きシステム（授業で扱ったのとは違うものを）
    - などなど