



# オンラインをこえて

プログラミング演習科目の  
オンライン実施を経験して

**明治大学 中村聡史**

Mail: [satoshi@snakamura.org](mailto:satoshi@snakamura.org)

Web: <http://nkmr-lab.org>

# 自己紹介

- 中村聡史

- インタラクション
- 平均手書き文字
- 周辺視野と集中促進
- 選択誘導
- ネタバレ防止
- などの研究
- BADUIを集めるのが趣味



人にまつわる、人を中心とした研究

# 中村聡史 研究室 (2020/07)

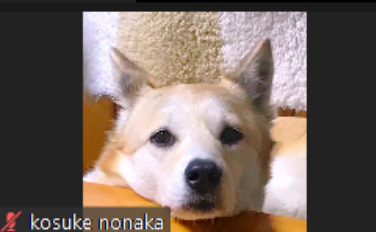
STANDBY  
**00:00**  
経過時間 ペル4/5/10  
又吉康綱



Naoto Matsuyama

Hamano Karin

Ryota Nakamura



Kota Yokoyama

Yukino Aoki

Kota Ninomiya

Juran Kuwabara

佐々木美香子

taku takahashi

r  
risa ito



A  
Ami Shimizu

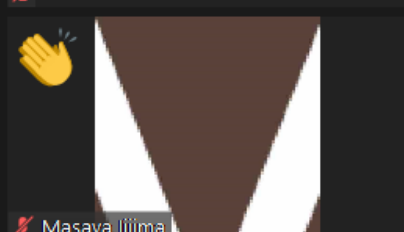
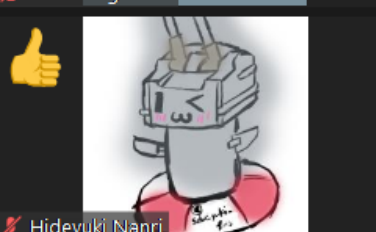
知佳  
Chika Sugimoto

Saeka Furuichi

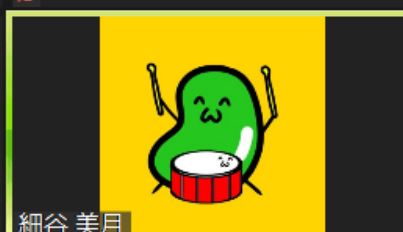
Miho Kajita

yukina

植木里帆



菅野一平



Ikumi Yamazaki

# COVID-19

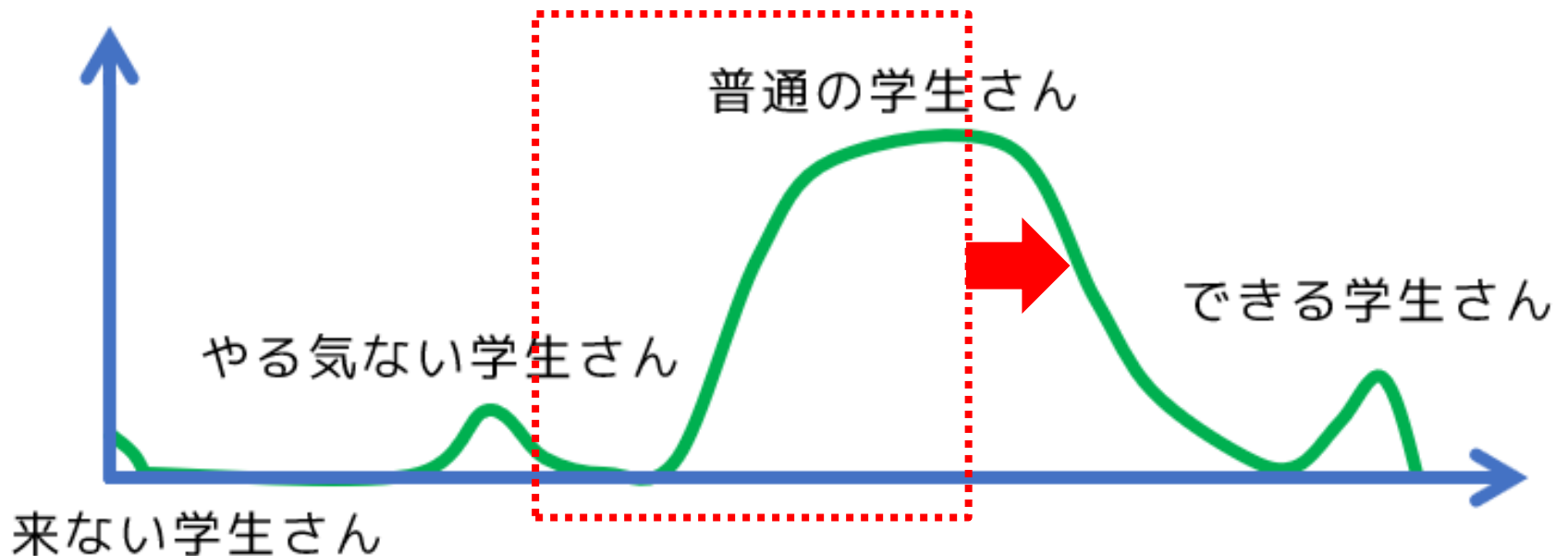
- とりあえずにつくき新型コロナウイルス
  - 家族旅行中止
  - 研究室合宿中止
  - 他研究室との合同研究会をオンライン化
  - 色々な学会・研究会がオンラインまたは中止
    - デンマーク、イタリア、東京、大阪、函館、沖縄
  - 施設での実証実験中止
  - 共同研究が1件取りやめに
  - アドバイザしていた案件が延期
  - もちろん、講義関係も大変

# 担当した講義

- 全体向けの座学講義（1回だけ）
  - 先端メディアサイエンス概論（110名程度）
  - 総合数理概論（300名程度）
- 専門科目（選択）：座学＋演習
  - 情報分析と可視化（60名程度）
- 1年次必修：演習（クォーター、週4コマ）
  - プログラミング演習1（120名程度）
- ゼミ
  - 1年ゼミ、2年ゼミ
  - 研究室のゼミ（3年・4年・大学院）

# 前提：プログラミング演習1

- 1年次必修のプログラミング教育では学生のレベル差が激しいため、上を飽きさせず、下も含め全体的に底上げすることが重要



# 本当に大変ですよね…

- 是非、いろいろと成功事例、失敗事例を教えてください！
- プログラミングに関する必修科目を担当されている各大学の先生方とは、今回の話を踏まえて数年後においしいお酒が飲めそうだなと勝手に思っています

# 講義のデザイン

- 講義時間内に課題を提示し、その課題を講義時間内に解決 & 提出することで習得を目指す
- 課題は、基本課題（全員向け）と、発展課題（上級者向け）を用意
  - 発展課題は加点要素
  - <http://nkmr.io/lecture/> にすべて公開していますのでどうぞ
- 講義時間をできるだけ課題解決に充てるため、予習を必須にして最初に小テストを実施
  - <http://nkmr.io/lecture/> に予習資料などをすべて公開
  - 課題は、研究室との接続性を考慮し、多様に
- 最終成果発表会で、プログラムを自作し発表！



# 興味を持たせる課題のデザイン

- 世の中とつながる課題のデザイン
  - 斜方投射、錯視、GUI、コンプガチャ & ジャンケンのシミュレーション、螺旋描画、リサーチユカーブ、素数1桁、曼荼羅、双子素数、モンテカルロ法、エラトステネスの篩、マクローリン展開、フーリエ級数展開、ラングトンのアリ、ライフゲーム、飛行機座席問題、周波数と音など
- 接続性を考慮したデザイン
  - 数学系課題、ネットワーク・セキュリティ系課題、音・音楽系課題、画像処理課題、フィジカルコンピューティング、ウェブ、HCI基礎的課題

# 色々な問題点

- 小テストが実施できないがどうするか？
  - 予習への強制力を発揮することができない
- 発表会が実施できないがどうするか？
  - モチベーションを上げ、自身で取り組むことができない
- 学生TA比をどうするか？
  - FMS学科は学生120人にTA8~10人のため、1人のTAが10人以上を見なければならない

4月からずっと胃が痛かった

# これまでにもあった問題点

- タイピングが遅くて課題に時間がかかる
  - 入力速度が遅いのはそもそもの問題
- 日本人学生にとって、英単語からなるプログラムに対して抵抗がある
- プログラムに利用される「, ; () []」などの記号と、その記法になかなか慣れない
- 上記の問題を小テストができない問題と絡めて一気に解決を目指す

# システムデザイン

- 理解を促す仕組み
  - コメントを上に表示して視界に入れる
  - プログラムが完成する前に逐次実行し、部分プログラムの意味を何気なく理解する
- 内発的動機づけ
  - 入力速度を自分の過去より早く！
- 外発的動機づけ
  - 他人とランキングを競うことでより上を目指す
  - タイピングを予習課題にする

# プログラミング学習支援 [又吉2020]

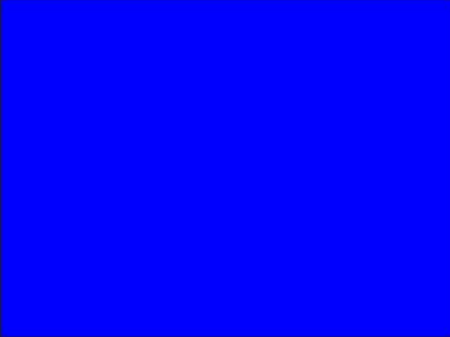
- typing.run ( <https://typing.run> )
  - タイピングとプログラミングの写経を組み合わせた学習システム
  - 写経しながら徐々にプログラムが動いていく

```
typing.run

マウスの位置で領域を2色で塗り分けよう
マウスの位置に応じて赤色と青色の領域が変わるようにするにはどうしたら良いか？

void setup()
{
  size(400,300);
}

void draw()
{
  // 背景を青色(0,0,255)で塗りつぶす
  background(0,0,255);
  // 塗りつぶし色を赤色(255,0,0)に設定する
  fill(255,0,0);
  // (0,0)から横幅mouseX (マウスのX座標) 縦幅300で四角形を描画
  rect(0,0,mouseX,300);
}
```

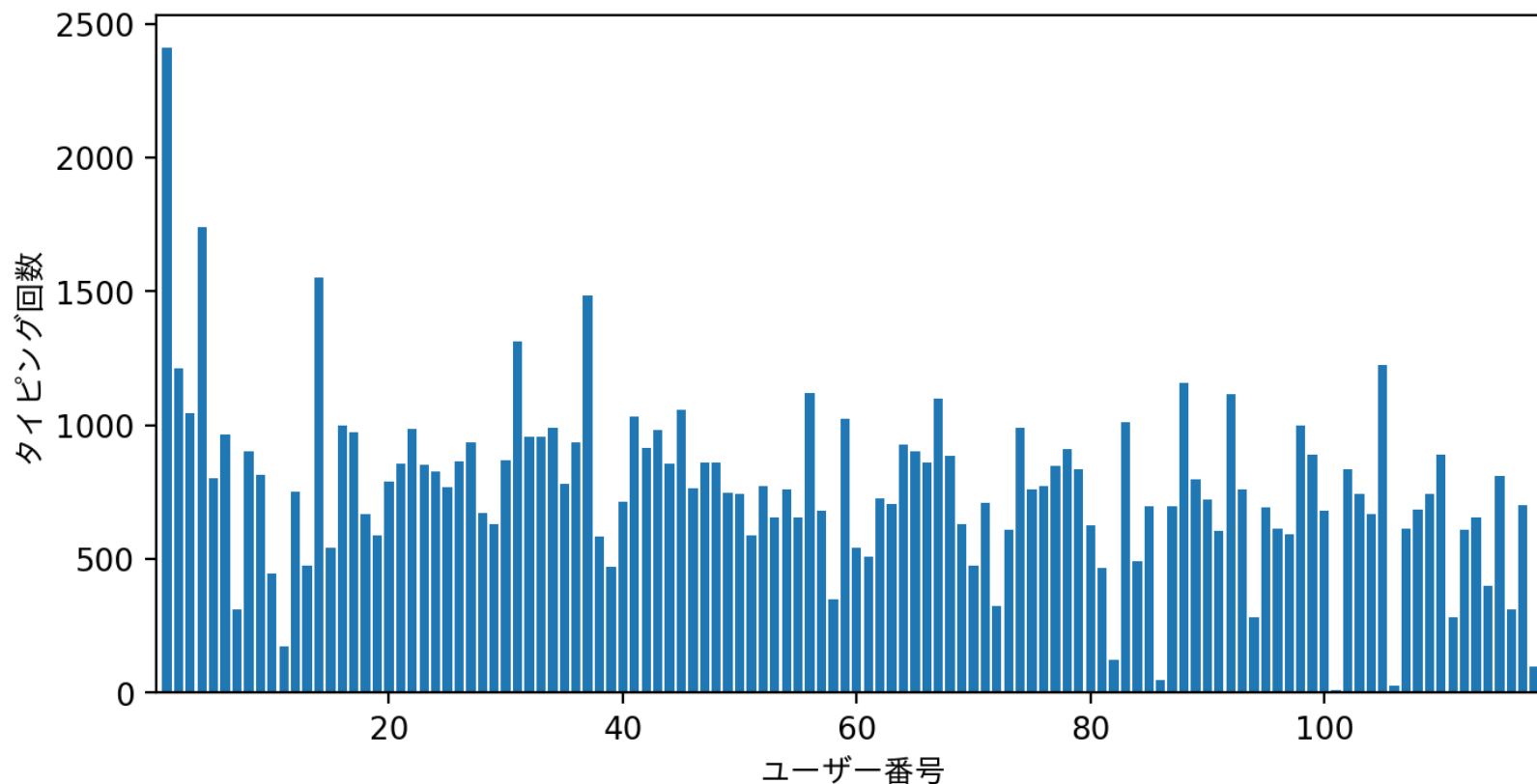


time:31.21 CPM:138  
missType:7 accuracy:82%

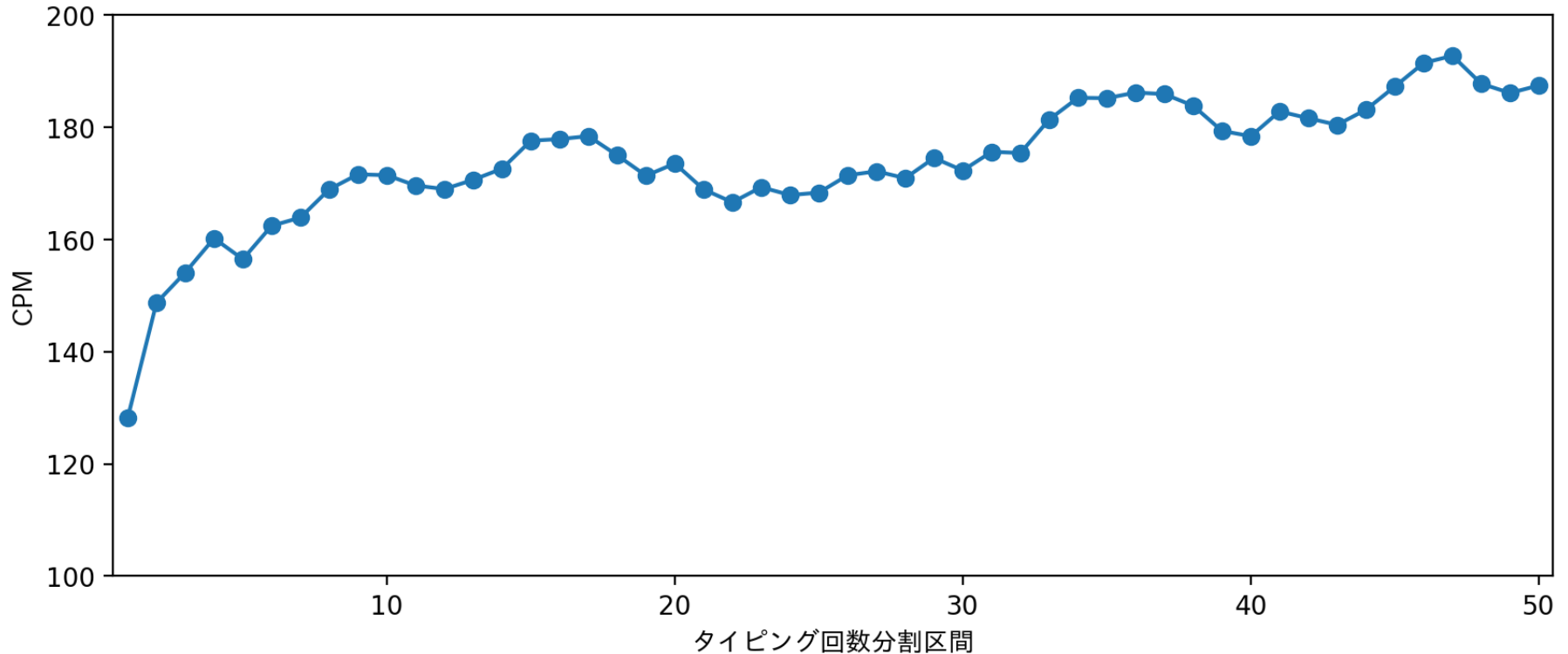
# 運用結果

2020年6月～8月上旬

- 総参加人数：176名（内履修者118名）
- 総タイピング回数：92,289回（内履修者88,055回）



# 運用結果：区間ごとの平均CPM



個人ごとに時系列順のスコアを50分割したCPMの平均

後半につれて右肩上がりに上昇をしている

→ 全体として期間後半になるほどCPM平均が上昇している

# 効果があったのでオススメです！

- タイピング速度がかなり上がっていた
- タイプミスによる基礎的なミスが少なかった（printlnをprintln、printInなど）
- 例年に比べ、レベルが高かった（単純比較は難しいが）
- タイピングできて、プログラミングできる気になっていた学生もいた

学生を管理して利用したい  
場合はお金をいただければ！



# 最終成果発表会ができない問題

- 発表会用のプログラムを用意するのではなく、条件に応じた課題とそのプログラムを作ってもらふことを課題に
  - 課題を作れるというのは、それなりにレベルがあるということ

- 以下の課題を作成せよ（各15点）
  1. 「条件分岐と繰り返し」までを習得した学生向けの課題を作成せよ（配列と関数を利用せずに解答できる課題を作成せよ）
  2. 「配列」を利用した課題を作成せよ（他の条件は自由）
    - 課題は図などを用いつつ、誤解のない記述を心掛け、いつも提示されている課題のように、スライドとして作成し、PDF形式で提出せよ（PowerPointまたはGoogle Slide等で作成せよ）

正の相関（0.56）あり

– 書籍など参考にした情報があれば引用すること

# 課題の提出をどうするか？

- 例年

- 大学の共有フォルダに提出

- 学生は自身のフォルダしかみえず、他者のフォルダは見えない

- 提出されたものについて、チェックし、そのチェック状況を学生に提示



A screenshot of a spreadsheet showing submission status for various dates. The spreadsheet has columns for dates and rows for submission status. The status is indicated by 'O' (submitted) and 'X' (not submitted). The spreadsheet is partially obscured by a blue box at the bottom.

3-26	3-24	3-32	3-11	3-03	3-02	3-50	3-48	2-3-59	3-22
3-43	3-06	3-25	3-20	74 OOX	3-17	10 OOX	3-37	3-14	3-30
XOX	3-47	01 OOX	07 XOO	3-08	3-13	3-12	29 OOX	3-49	21 XXXX
XXXX	45 XXXX	27 XXXX	44 XXXX	16 XXXX					

できない！

# Google Form/Spreadsheetの活用



## 提出報告・採点状況の確認

### 提出後できたらGoogleフォームで報告

- 学年組番号, 名前, どの課題を提出したか
- 課題の個数分だけフォームに回答

### 1. 提出したらフォームに入力して待機

フォームに登録されるのでその順に採点します

### 2. シートで提出・採点状況確認

フォームに連動した「回答状況」シートを共有  
「現在どこまで採点したか」がわかるようにします

学生順に並べた「採点状況」シートも共有  
自分の課題についての採点状況はこちらで確認

### 3. NGがいたら再提出 次スライド

6月24日（火） 課題チェック申請フォーム

Googleドライブに課題をアップロードしたらこちらから報告をお願いします。  
申請後にチェックしていく予定ですが、数分お休みは可能です。

\*必須

学年\*

回答を入力

組\*

回答を入力

番号\*  
学籍番号ではありません

回答を入力

(例) 6/24 のフォーム

1	タイムスタンプ	学年	組	番号	氏名	課題名
2	2020/06/23 17:18:21	0	0	0	めいじろう	basic_Progress
3	2020/06/23 17:18:42	0	0	0	めいじろう	basic_DrawCharacter
4	2020/06/23 17:18:54	0	0	0	めいじろう	advanced_SyncIllusion
5	2020/06/24 11:18:59	1	3	1	takahashi	basic_Test
6	2020/06/24 11:19:51	1	4	1	takahashi	basic_Test
7	2020/06/24 11:22:50	1	3	1	takahashi	basic_Progress
8	2020/06/24 11:23:23	1	3	1	takahashi	advanced_SyncIllusion
9	2020/06/24 11:26:09	1	3			

フォームの回答状況

1	学年	組	番号	basic_Test	basic_Progress	basic_DrawCharacter		
2	0	0	0	めいじろう	-	OK	NG	再提出をお願いします
3	1	3	1	青木	OK	-	-	-
4	1	3	2	池谷	-	-	-	-
5	1	3	3	石川	-	-	-	-
6	1	3	4	石橋	-	-	-	-
7	1	3	5	石山	-	-	-	-
8	1	3	6	今川	-	-	-	-
9	1	3	7	七	-	-	-	-

採点状況

Issue ID	Category	Status	Created	Resolution	Comments
basic_LineBoard		最新	2-4-24	OK	合っていますが、何か描画するときはdrawごとにbackground()を入れるようにしてください
advanced_Eratosthenes		最新	1-4-51		発展問題はフィードバックしません（提出を確認したら）
basic_MouseTrace	キャラクターの数を調整しました	最新	1-4-44	OK	
basic_CompleteGacha		最新	1-3-19	OK	
basic_LineBoard		最新	1-3-24	OK	
basic_MouseTrace	再提出したので確認よろしくお願	最新	1-4-24	OK	
basic_CompleteGacha					
basic_CompleteGacha		最新	1-4-51	OK	
basic_LineBoard		最新	1-3-22	NG	プログラム起動時の色を設定してください
basic_MouseTrace		最新	2-4-23	OK	
basic_CompleteGacha	10回カチャになおしました。	最新	1-3-43	OK	
basic_MouseTrace		最新	2-4-53	NG	円のグラデーションが不適切です（マウスに近いものが赤、遠いものが白になるようにしましょう）
basic_LineBoard		最新	1-4-30	OK	
basic_CompleteGacha		最新	1-4-10	OK	合っていますが、棒グラフの変化が小さすぎて分かりにくいです。
basic_MouseTrace		最新	2-4-55	NG	新しい円を手前に書くようにしてください
basic_CompleteGacha		最新	2-4-9	OK	
basic_LineBoard		最新	1-3-12	OK	
basic_MouseTrace		最新	1-3-43	OK	
basic_LineBoard		最新	1-3-45	NG	初めの色が設定されていません。
basic_CompleteGacha		最新	2-4-43	OK	
basic_MouseTrace		最新	1-3-11	NG	
basic_LineBoard		最新	1-4-32		
basic_LineBoard		最新	1-4-45		
basic_LineBoard		最新	2-4-49		
check		最新	2-3-24	OK	
basic_MouseTrace		最新	2-4-49		
basic_LineBoard		最新	2-3-24		
basic_LineBoard		最新	1-3-32		
basic_CompleteGacha		最新	2-4-49		
basic_MouseTrace		最新	1-4-27		
basic_MouseTrace		最新	1-4-29		
basic_MouseTrace	色を変えました。	最新	2-4-11		
basic_MouseTrace		最新	1-3-9		
basic_MouseTrace		最新	1-4-8		
basic_CompleteGacha		最新	1-4-5		
basic_CompleteGacha		最新	1-4-47		
basic_MouseTrace		最新	1-3-7		
basic_MouseTrace		最新	1-3-45		

# むしろ良かった！

- チェック状況が分かってよい！
  - 自分の順番待ちがどの程度か
  - 何がダメだったのか
- むしろフィードバックを返せる＆残せることがよかった
  - 教員・TA間の採点基準の共有化
  - 他の学生がどういった理由で不可とされているか判断可能
  - 次年度も採用予定！

# ちなみに

- ここに至るまでの失敗談もあります…それはご興味がありましたらこちらを

– <https://note.com/nkmr/n/na57e5612eed3>

さて、このような感じで色々と時間をかけて準備を進めてきましたが、実際のところ1回目の演習講義については失敗に終わりました。例年であれば初回は全員が時間内に課題の提出が終わり、最後に時間が余るはずなのに時間内に課題の提出が終わらない学生さんが多発したこと。また、こちらも例年であれば余裕に終わっていた時間内に課題のチェックが終わらなかったこと。そして、いくつかの問題についてTwitterで学生さんたちから不満が出ていたことなどです。

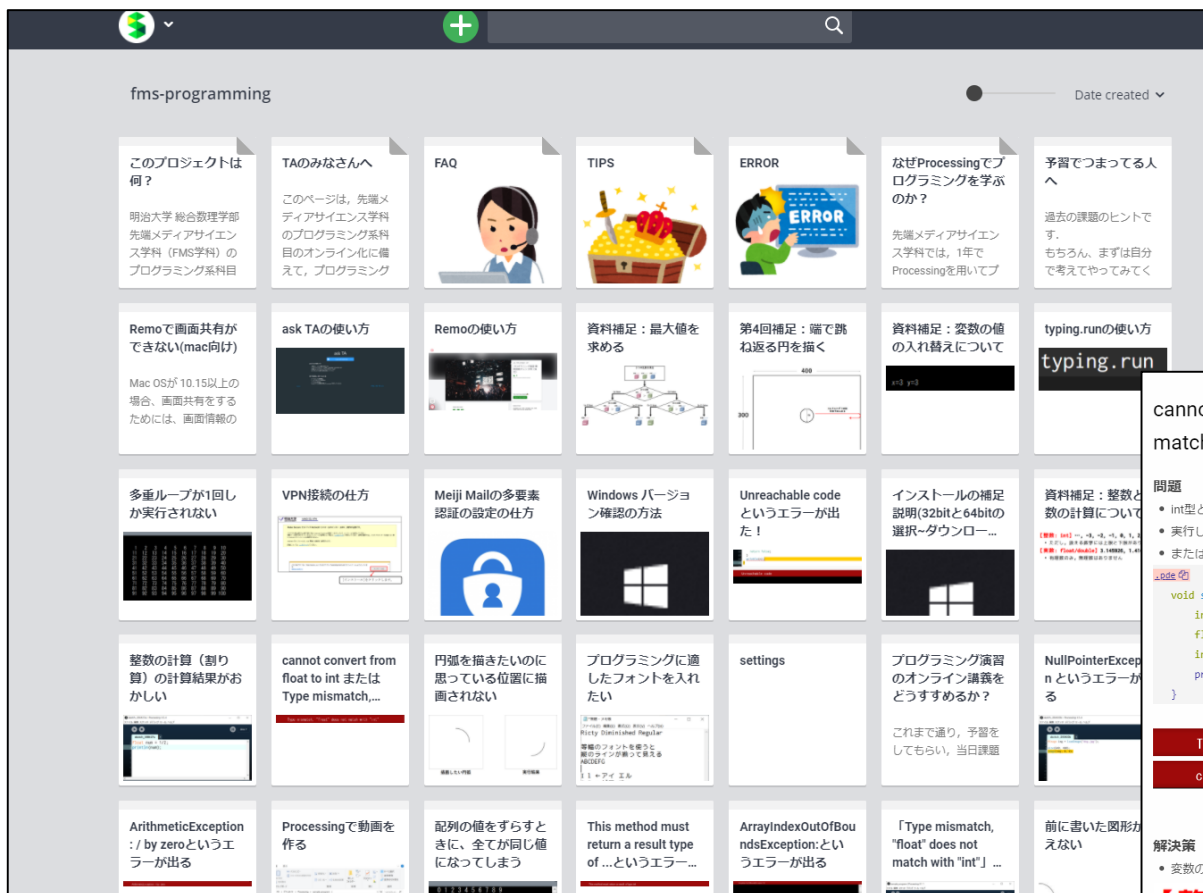
# 学生・TA比をどうする？

- よくある質問はサポートページで対応する
- 簡単な質問は学生同士で解決し、解決しない問題をTAに質問してもらう
- 順番待ちをうまく処理する



# よくある質問はページを用意

- Scrapboxを教員・TAで編集して用意
  - <https://scrapbox.io/fms-programming/>



多重ループが1回しか実行されない

問題

- 以下のような出力を目指す。

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	32	33	34	35	36	37	38	39	40

cannot convert from float to int または Type mismatch, "float" does not match with "int" というエラーが出る

問題

- int型とfloat型の変数を計算しようとした
- 実行したら「cannot convert from float to int」というエラーが出た
- または書いている途中に「Type mismatch, "float" does not match with "int"」というエラーが出ている

```
code
void setup(){
  int x=3;
  float y=0.6;
  int z=x*y; //例では+を使っているが、-,*,/,% の場合でも起こり得る
  print(z);
}
```

Type mismatch, "float" does not match with "int"

cannot convert from float to int

解決策

- 変数の型を一致させる

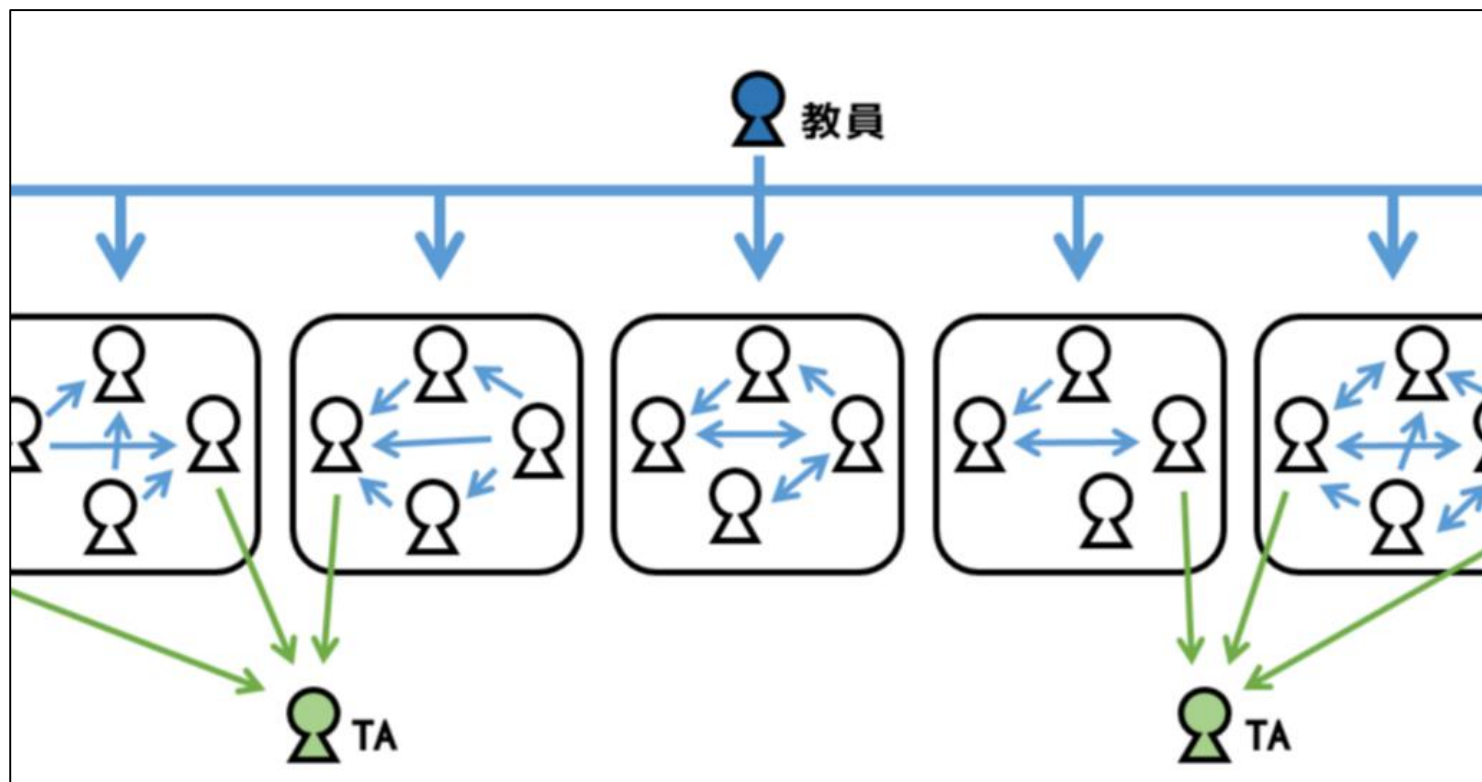
**【整数: int】** ..., -3, -2, -1, 0, 1, 2, 3, 4, ...

ただし、極大値・極小値には上限・下限があります



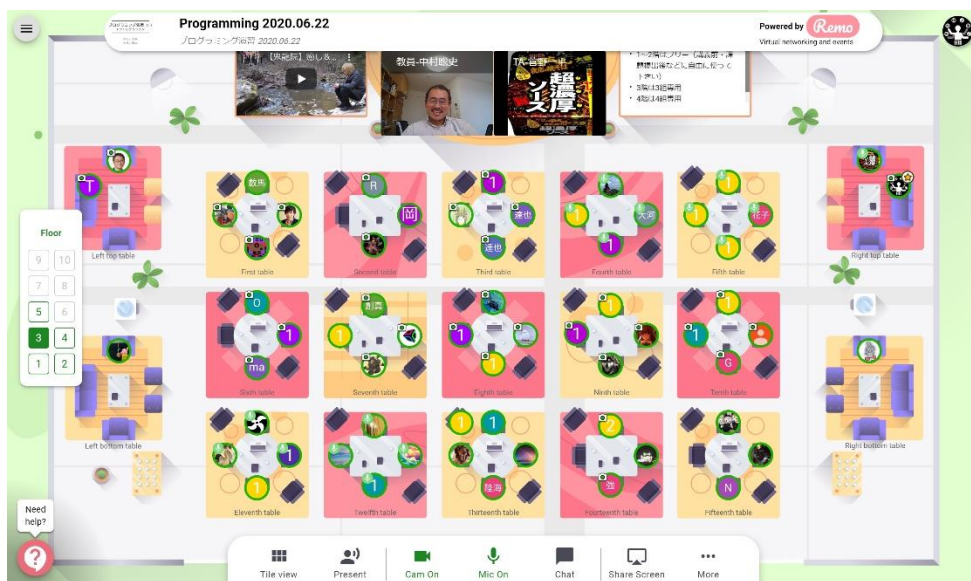
# 学生・TA比をどうする？

- 簡単な質問は学生同士で解決し、解決しない問題をTAに質問してもらう
  - 会話可能な小グループの重要性



# Remo Conference

- 全体向けの講義ができ、またテーブルごとに会話可能なウェブサービス



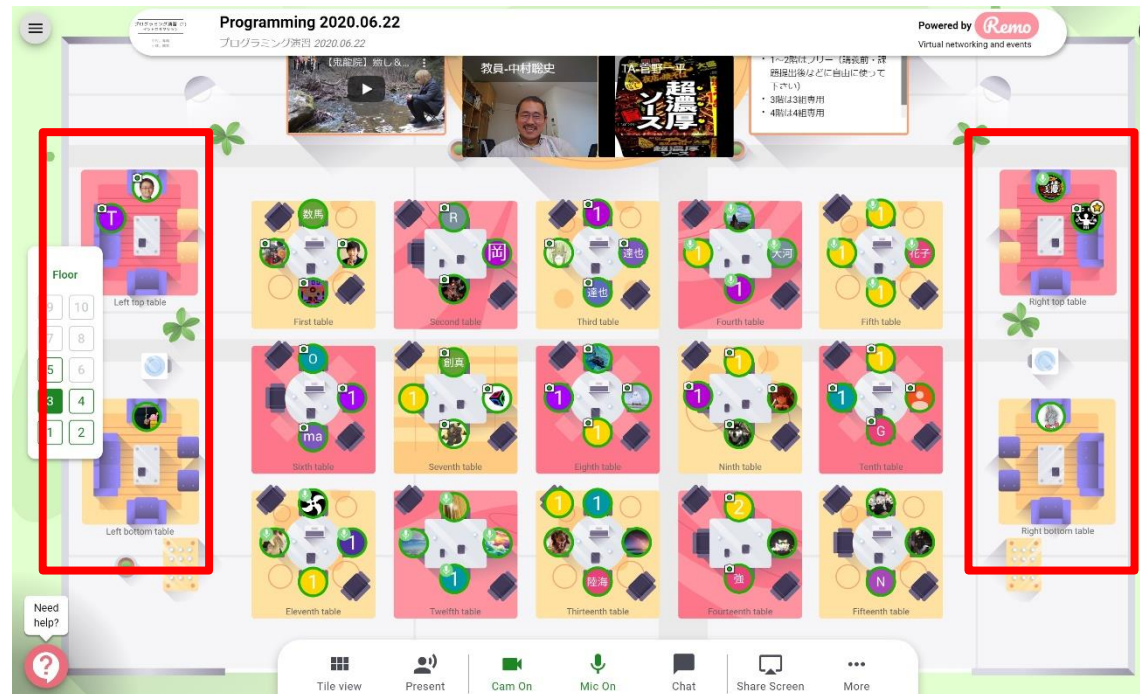
# 小グループの会話を可能に

- テーブルに4人ずつ配置し、その学生同士では画面を共有したり、会話をしたりして解決可能とする
- テーブルはこちらから指定
- ほとんどのテーブルはそれでうまくいっていたが、一部のテーブルは会話がまったくできないなどの問題もあったらしい

レベルなどを考慮して  
定期的にメンバー変更が重要

# 順番待ちの処理のために

- 当初はRemoの両側をTA部屋にして順番待ち
  - 一部TAへの質問の集中
  - TA部屋から出ていかない学生がいる
  - プログラムに介入できなくて不便などの問題発生



# askTA：順番待ちの処理のために

- 手軽に申請
  - 何に対する質問で現在のプログラムはどんな感じかを申請
- 手軽に対応
  - 質問とプログラムを事前に見て対応を考え、対応可能に
  - 音声により対話しながら対応
- 質問してもすぐに対応されず待つ必要あり
  - 入店音で呼び出しを可能に
  - 質問の取り下げも可能に

- 質問UI

## 質問投稿フォーム

組(floor)

組を選択してください

質問をしたい課題

質問をしたい課題を選択してください

質問の種類

質問の種類を選択してください

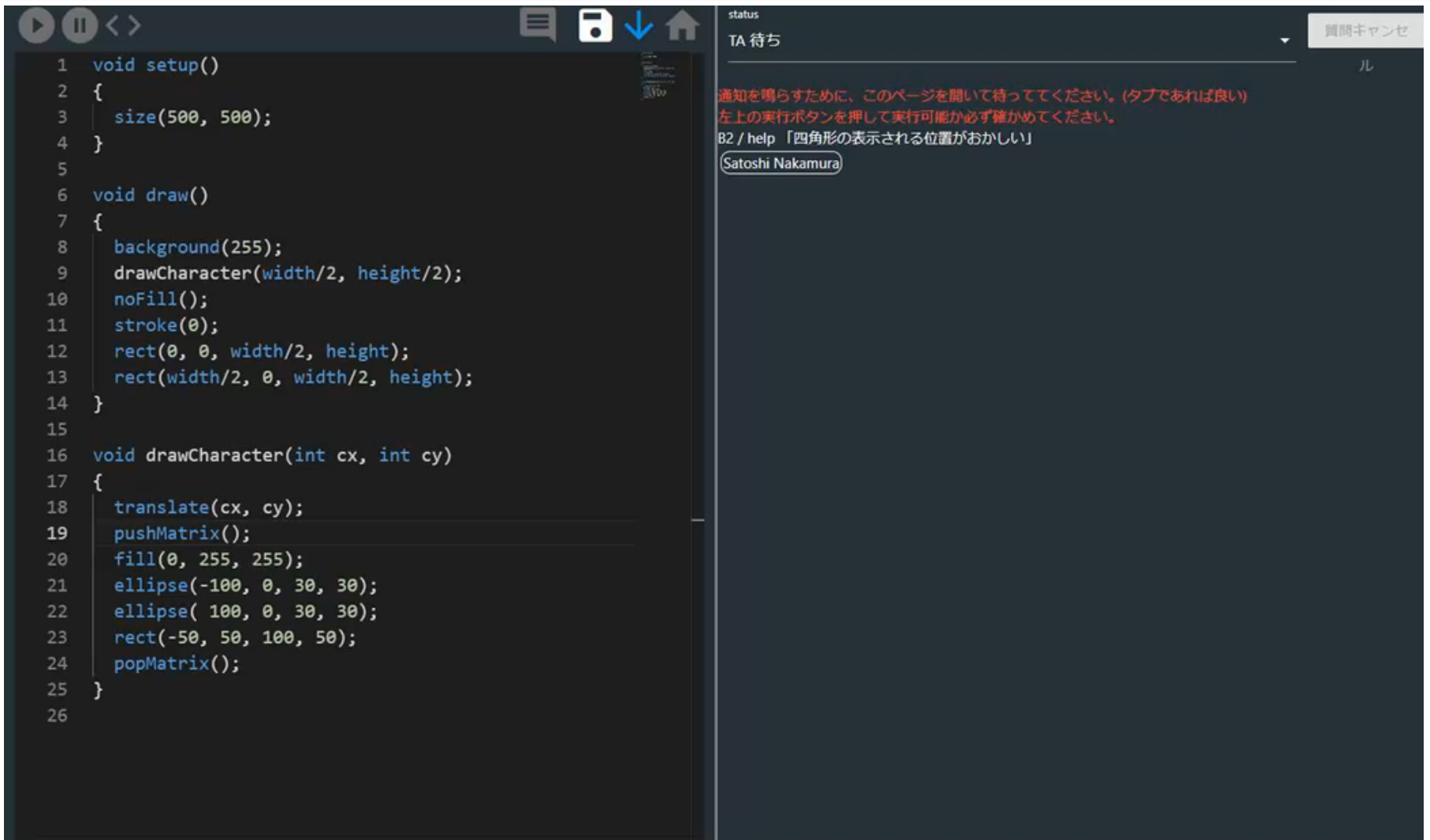
質問の概要(1行程度)

例：画面になにも表示されなくなった。

ソースコード(できたところまでで良い/省略可)

```
void setup()  
{
```

キャンセル 質問を送信



The screenshot displays the askTA interface, which is split into two main sections: a code editor on the left and a chat window on the right.

**Code Editor (Left):**

```
1 void setup()
2 {
3   size(500, 500);
4 }
5
6 void draw()
7 {
8   background(255);
9   drawCharacter(width/2, height/2);
10  noFill();
11  stroke(0);
12  rect(0, 0, width/2, height);
13  rect(width/2, 0, width/2, height);
14 }
15
16 void drawCharacter(int cx, int cy)
17 {
18   translate(cx, cy);
19   pushMatrix();
20   fill(0, 255, 255);
21   ellipse(-100, 0, 30, 30);
22   ellipse( 100, 0, 30, 30);
23   rect(-50, 50, 100, 50);
24   popMatrix();
25 }
26
```

**Chat Window (Right):**

The chat window is titled "status" and "TA 待ち" (Waiting for TA). It contains the following text:

通知を鳴らすために、このページを開いて待ってください。(タブであれば良い)  
左上の実行ボタンを押して実行可能か必ず確かめてください。  
B2 / help 「四角形の表示される位置がおかしい」  
Satoshi Nakamura

At the top right of the chat window, there is a button labeled "質問キャンセル" (Cancel Question).

# ということで

- 小テストが実施できないがどうするか？
  - typing.runにより解決！
- 発表会が実施できないがどうするか？
  - 課題を作成するという課題で解決！
- 学生TA比をどうするか？
  - Scrapbox、小グループ制導入、askTAで対処

乗り切った！



# オンライン演習講義に重要なこと

- 学生の受講環境をどうするか？
  - Processingが動かないと問題外
- サービスが落ちたときどうするか？
  - ウェブ系サービスは落ちると被害甚大
- 学生の様子をどう拾い上げるか？
  - 手が止まっている、悩んでいるが見えない
- 学生に自身の状況をどう提示するか？
  - 受理されたか？ また待ち行列の数はどの程度か？
- いかにして質問のハードルを下げるか？
  - 質問への抵抗感を徹底的に下げる必要がある！
- いかにして手間を減らすか？
  - オペレーションが多数発生すると死ぬ

# 受講環境をどうするか？

- 受講環境に関する調査と大学からパソコンの貸与を実施
  - 教員による電話攻勢で4月中下旬には完了
- 半期通してみると、多少の問題があったがなんとか95%程度の学生は問題なくクリア
  - 一部学生には問題があったようにも見受けられたが、そもそも成績を考えると本気だったのかも不明

# サービスが落ちたときどうする？

- 徹底した多重化により対処
  - 主たる講義： Remo Conference
    - 実際に講義で落ちた
  - Remoが落ちたとき用： Zoom
  - テキストチャット： Slack
  - ポータル： Oh-o! Meiji
  - QAシステム： askTA
  - 説明・解説動画アーカイブ： YouTube



一度落ちたけど  
無事乗り切った！

# 学生の様子をどう把握する？

- 課題の提出・チェック状況を提示して把握
  - とはいえ、これは不完全だった

1	学年	組	番号	basic_Test	basic_Progress	basic_DrawCharacter
4	1			OK	OK	OK
5	1			OK	OK	OK
6	1			OK	OK	OK
7	1			OK	OK	OK
8	1			OK	OK	OK
9	1			OK	OK	OK
10	1			OK	OK	OK
11	1			OK	OK	OK
12	1			OK	OK	OK
13	1			OK	OK	OK
14	1			OK	OK	OK
15	1			OK	OK	OK
16	1			OK	OK	OK
17	1			OK	OK	OK
18	1			OK	OK	OK
19	1			OK	OK	OK
20	1			OK	OK	OK
21	1			OK	OK	OK
22	1			OK	OK	OK
23	1			OK	OK	OK
24	1			OK	OK	OK
25	1			OK	OK	OK
26	1			OK		
27	1			OK	OK	OK
28	1			OK	OK	OK
29	1			OK	OK	OK
30	1			OK	NG	提出されていません。該当フォルダに提出してください
31	1			OK	OK	OK

# 学生に自身の状況をどう提示？

- Google Spreadsheet  
- 採点状況などを可視化

作業名	更新日時	状況	コメント
basic_LineBoard	2-24 OK	合格	合っていますが、何か描画するときbackground色を入れるようにしてください
advanced_Eratosthenes	1-4-61	合格	発表時間はフィードバックしません。(提出を確認したら内容に賛否なくO)
basic_MouseTrace	1-4-44 OK	合格	キャラクターの色を変更しました
basic_CompleteGacha	1-3-19 OK	合格	
basic_LineBoard	1-3-24 OK	合格	
basic_MouseTrace	1-4-24 OK	合格	再提出したので確認よろしくお願
basic_CompleteGacha	1-4-24 OK	合格	
basic_LineBoard	1-4-61 OK	合格	
basic_LineBoard	1-3-27 NG	不合格	プログラム実行時の色を設定してください
basic_MouseTrace	2-4-23 OK	合格	
basic_CompleteGacha	1-3-23 OK	合格	
basic_MouseTrace	2-4-63 NG	不合格	円のグラデーションが不適切です。(マウスに近いものが赤、遠いものが白になるようにしよう)
basic_LineBoard	1-4-30 OK	合格	
f-basic_CompleteGacha	1-4-10 OK	合格	合っていますが、棒グラフの幅が小さすぎて分かりにくいです。
f-basic_MouseTrace	2-4-6 OK	合格	新しい色を事前に書くようにしてください
basic_CompleteGacha	2-4-8 OK	合格	
basic_LineBoard	1-3-12 OK	合格	
basic_MouseTrace	1-3-43 OK	合格	
basic_LineBoard	1-3-45 OK	合格	
basic_CompleteGacha	1-3-45 OK	合格	初期の色が設定されていません。
basic_LineBoard	2-4-43 OK	合格	
basic_MouseTrace	1-3-11 NG	不合格	
basic_LineBoard	1-4-32	不合格	
basic_LineBoard	1-4-45	不合格	
basic_LineBoard	2-4-49	不合格	
f-check	2-3-24 OK	合格	
basic_MouseTrace	2-4-49	合格	
basic_LineBoard	2-3-24	合格	
basic_LineBoard	1-3-32	合格	
basic_CompleteGacha	2-4-49	合格	
basic_MouseTrace	1-4-27	合格	
f-basic_MouseTrace	1-4-29	合格	
f-basic_MouseTrace	2-4-11	合格	色を変えました。
basic_MouseTrace	1-3-81	合格	
basic_MouseTrace	1-4-81	合格	
basic_CompleteGacha	1-4-61	合格	
basic_LineBoard	1-4-47	合格	
basic_MouseTrace	1-3-71	合格	
basic_LineBoard	1-1-15	合格	

1	学年	組	番号	basic_Test	basic_Progress	basic_DrawCharacter
4	1	1	1	OK	OK	OK
5	1	1	2	OK	OK	OK
6	1	1	3	OK	OK	OK
7	1	1	4	OK	OK	OK
8	1	1	5	OK	OK	OK
9	1	1	6	OK	OK	OK
10	1	1	7	OK	OK	OK
11	1	1	8	OK	OK	OK
12	1	1	9	OK	OK	OK
13	1	1	10	OK	OK	OK
14	1	1	11	OK	OK	OK
15	1	1	12	OK	OK	OK
16	1	1	13	OK	OK	OK
17	1	1	14	OK	OK	OK
18	1	1	15	OK	OK	OK
19	1	1	16	OK	OK	OK
20	1	1	17	OK	OK	OK
21	1	1	18	OK	OK	OK
22	1	1	19	OK	OK	OK
23	1	1	20	OK	OK	OK
24	1	1	21	OK	OK	OK
25	1	1	22	OK	OK	OK
26	1	1	23	OK	OK	OK
27	1	1	24	OK	OK	OK
28	1	1	25	OK	OK	OK
29	1	1	26	OK	OK	OK
30	1	1	27	OK	OK	OK
31	1	1	28	OK	NG	提出されていません。該当フォルダに提出してください
32	1	1	29	OK	OK	OK

- askTA  
- TA待ちを可視化

3組のTAとの質問状況

TA待ち: 質問内容: ヒント: ...

TA回答: 回答内容: ...

学生キャンセル: ...

4組のTAとの質問状況

TA待ち: 質問内容: ヒント: ...

TA回答: 回答内容: ...

学生キャンセル: ...

# 質問のハードルをどう下げる？

- 対面で実施しても質問へハードルを感じる
  - 小グループで会話してある程度問題を解決
  - TAに直接質問するのではなく、システムに質問を投げ、TAがそれを拾い呼び出すことで、質問へのハードルを下げる（消極性デザイン）

## 3組のTAとの質問状況

### TA待ち

v686mybjsq 基本1: ヒント: `rect(100, 100, 200, 200)`: 何をすればいいのかわからない

e77bypowwy 基本1: ヒント: `onClick()`: クリックするたびにプログラムが書けない

### TA対応中

ojwp8ji5b 基本3: ヒント: `color(255, 0, 0)`: 色の変化が三色ではなく二色になってしまう

4q3rismown 基本2: ヒント: `var x = 100`: 変数が変わらない、画面に変化が出ない

### TA終了

t6czjbmnu 基本2: ヘルプ: `rect(100, 100, 200, 200)`: rectの描画する領域がおかしい

trebxt4tlz 基本1: ヘルプ: `onClick()`: クリックしても実行されない。

### TA保留

### 学生キャンセル

mw18csd2hz 基本2: ヘルプ: `unexpected tokenの処理`: キャラクタのプログラムをどこに書いてよいかわからない

# いかにして手間を減らすか？

- Zoomだとホストがブレイクアウトルームの制御をする必要あり
- Remo Conferenceによりテーブルの移動などを簡易化

# そんなこんなで

- 色々反省点はあるものの、何とか乗り切りました
- 成績も例年より良かった！
- 学生からもかなり高評価（一部紹介）

プログラミング初心者だったので、最初のころは課題ができるか不安でした。予習資料やtyping.run、askTAなどの学習しやすい環境を整えていただいて、本当に助かりました。毎週日曜日と火曜日に友達と予習会をして、とても充実した講義でした。ありがとうございました。

完全な初心者だったが、プログラミングの一から学ぶことができ、回を追うごとに自分の成長が感じられて楽しかった。また、Remoを使ったシステムも、授業を成立させる分には十分な環境だと感じた。scrapboxはTAさんに聞くまでではない簡単な質問を解決することのできるとても便利なものだと感じた。

タイピング演習がとても良いと思います。来学期も似たような演習があるとありがたいです。そして、わからない所をTAさんに気軽に聞けることがとても助けになりました。でも、TAさんと話した後わからない所が時々あります。もう一度TAさんに聞くのも少し躊躇いがありました。そういった場合はどうすればいいのかよくわかりませんでした。もちろん、TAさん達にはとても感謝しています。ありがとうございました。先生たちのヒントや解説もありがとうございました。来学期はもっとaskTAを利用するつもりです。よろしくお願いします。

とてもわかりやすい授業だった。

配列からいきなり難しくなってよく分からなくなったため、夏休みに復習したい。  
授業前の解説をオンデマンドにするとより予習やタイピングランの理解が深まると思った。

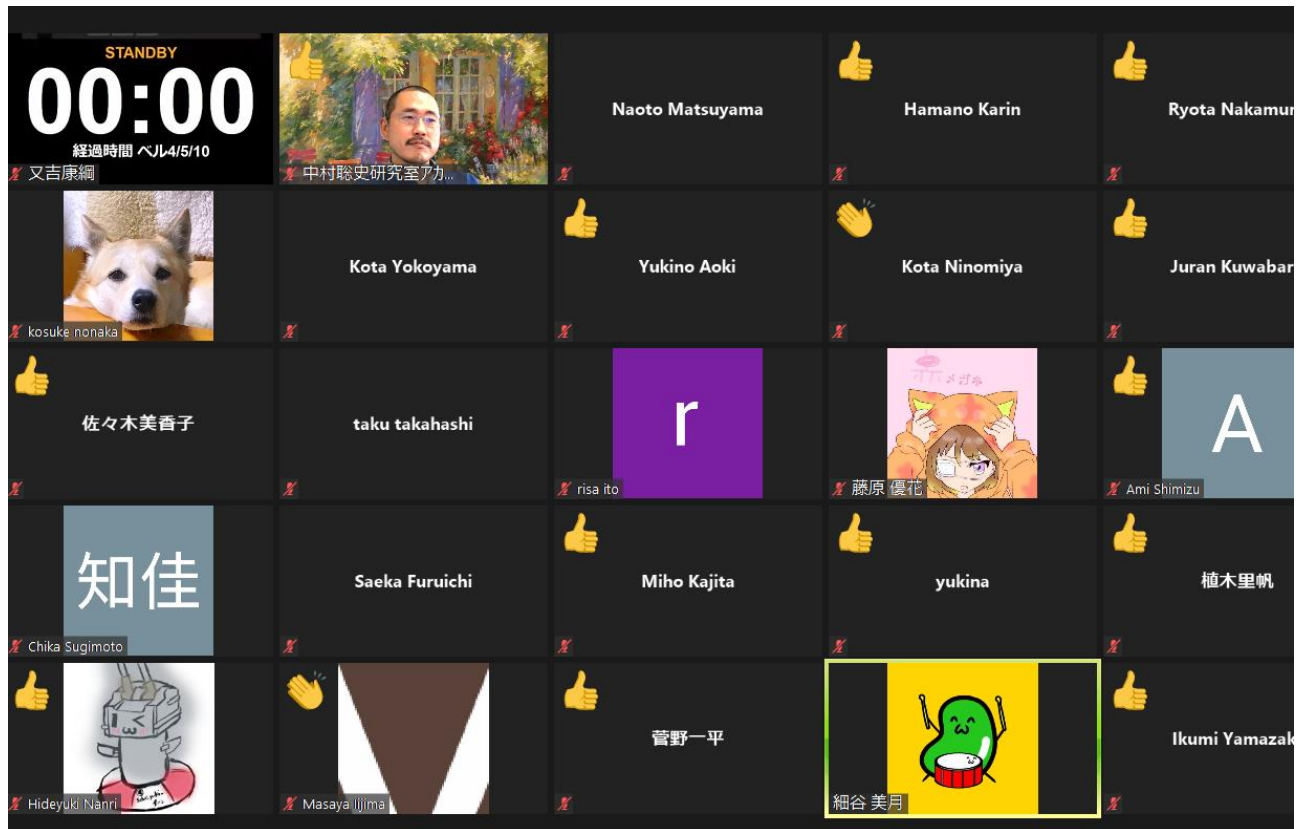


# 遠隔講義のテクニック

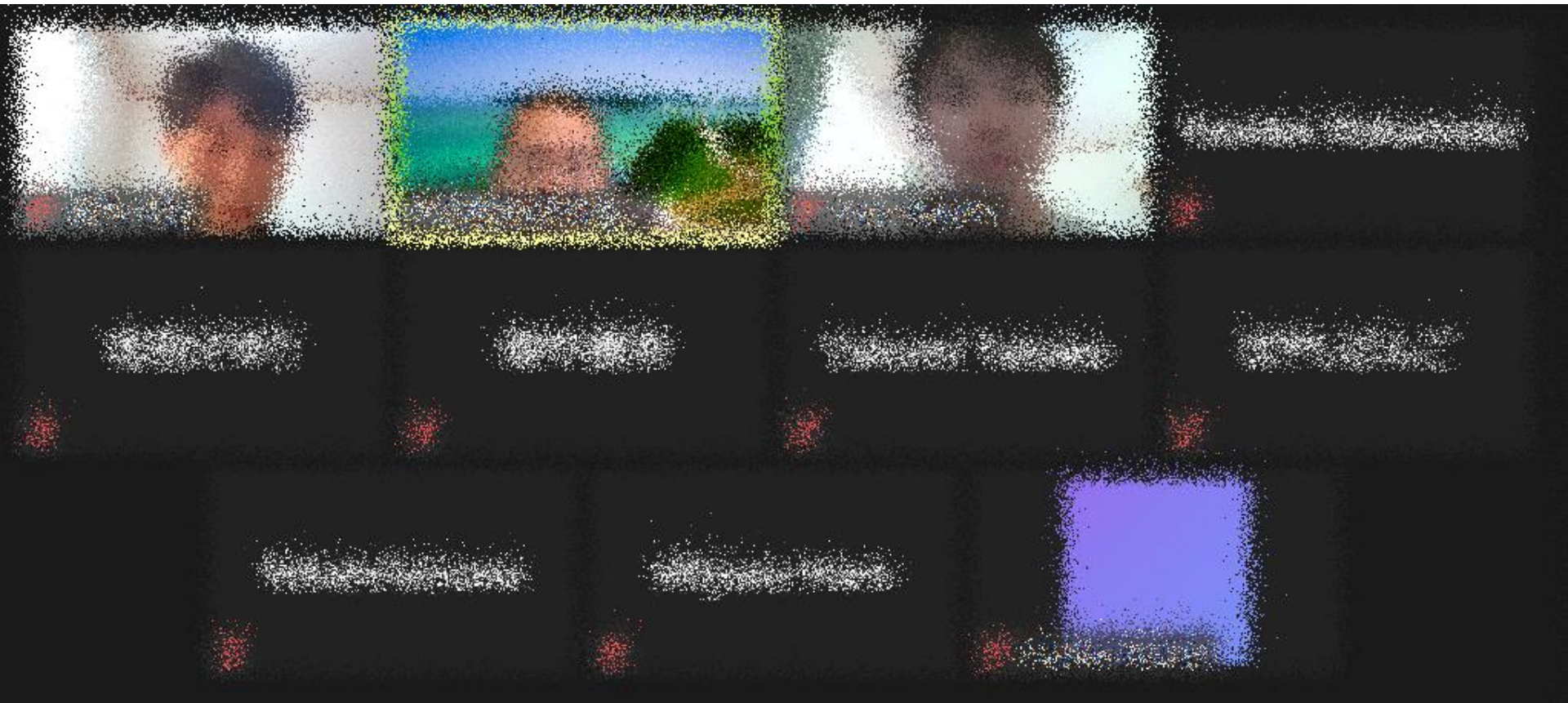
- CommentScreenによる対話的講義
- カメラON/OFFを活用した進捗チェック
- Scrapbox/Google Docsなどを併用した対話的講義
- Discordでの多段配信による助け合う受講

# リアクションはわかりにくい

- リアクションは時間により消えてしまう
- 挙手は下げ忘れが頻発する



# 学生の様子を見えるように



# 学生の様子を見えるように

- カメラON/OFFを活用した進捗チェック
  - ZoomのカメラをOFFにすることで、できたことを伝達してもらい、あとどれくらいできていないかを把握
- 副次的効果
  - 徐々にほかの人がいなくなるので、残された人（取り組んでいる人）は焦る
  - 残った人が少なくなると質問しやすい
  - カメラOFFな人は自由な時間ができる

# ということで

- プログラミング演習の遠隔講義をいかにして実現したのかという話を少しでも紹介
- ご興味のある人はこちらもどうぞ
  - <https://note.com/nkmr/m/m04e7640c83cf>
- ご自由にお使いください
  - <https://typing.run/>
- 60～90分で話せるネタありますので、ご希望が有りましたら私まで
  - [satoshi@snakamura.org](mailto:satoshi@snakamura.org)
  - 日本大学とはこだて未来大学でお受けしました