

---

# プログラミング演習2 (1)

## イントロダクション

---

中村, 高橋, 小林, 橋本

# まずは

---

- 教員・TAからの自己紹介を

# EP演習の発表会

---

- どうでしたか？
- 恥ずかしがらずに是非自分が作ったものは  
どんどん発信していきましょう！
- 経験上...
  - 完璧なもの・理想的なものができるまで発信しないひとは、いつまでも発信せず、そして成長していきません
  - 十分でなくてもいいので今あるものを発信できるひとは勢いよく成長していきます

# 前にも言いましたが、プログラミング習得

- 習得する方法には主に、2つの道があります

## (1) 作りたい欲にまかせて、自分で調べつつ色々なプログラムを作っていく

- 本当はこちらが上達速度も早くおすすめです
- ただ誰もがができる道ではないとはわかっています

## (2) ひたすら写経していく

- 作りたい欲がない場合、一番手っ取り早いのが勘所を掴むため写経するということになります。ひたすら書いて覚える！
- そのためにプログラミング演習1では `typing.run` を用意していました

# プログラミングの習慣をつけよう

- #dailycoding
  - Twitter に毎日作ったプログラムをアップしていくという活動
  - とにかく、毎日作るということを心がけていくこと、ただそれだけ



takawo shunsuke  
@takawo

...

気がつくと #dailycoding のハッシュタグを使う人が増えている？「続かない」と思っていたり躊躇してる人がいたら、いつ始めても、止めても、再開してもいいのがデイリーコーディングなので、気軽に気負わずに作ったものをシェアしていきましょう🙌🙌

[twitter.com/hashtag/dailyc...](https://twitter.com/hashtag/dailyc...)

# どこでやる？

---

- OpenProcessingを使っていきましょう！
  - <https://openprocessing.org/>
  - そこでプログラムを書けて、かつURLで共有できちゃいますので、Twitterに投稿しよう！
- Processingでも書けますし、p5.jsでも書くことができます！

# p5.js って何？

---

- Processing と同等の表現を JavaScript で記述可能とするライブラリ
  - 気軽にWebのかっこいいインタフェースを作ることができます
- どれくらい違うのか？
  - 変数に型がなくなります（letやvarで定義）
  - void/int/floatがなくなります
  - 関数の定義はfunctionです
  - sizeの代わりにcreateCanvasです
  - それ以外はほぼ同じです
    - [typing.run](#)のp5.js版を参照

# ちなみに

- dailycoding 活動をされている takawo さんがNFTアートとして10000個を販売し、2時間で完売（売上600万ドル以上）
  - ちなみに、売上はProcessingを中心としたコミュニティへ全額寄付
  - <https://generativemasks.on.fleek.co/#/>



# 繰り返しになるけど

---

- 「作りたい欲にまかせて、自分で調べつつ色んなプログラムを作っていく」という入り口にみなさんすでに立ててるとおもいます
- この方法は、上達速度も早くおすすめですので、ぜひこの方法で力を伸ばしていける人が増えてほしいと思っています

# さてさて

---

- とはいえ講義です
- プログラミング演習2もまた、その表現のための基礎を作り、そして応用するためにあります

# プログラミング演習2について

---

- プログラミング演習1と同じところ
  - Processingをやります
- プログラミング演習1との違い
  - 予習より復習を重視して下さい
  - typing.runはありません
  - 中間試験があります
  - 宿題があります
  - グループ制作 / 発表してもらいます
    - 再履修の学生は時間が合わないことも多いためグループ制作か最終課題かを選択可能とします。

# グループ制作 / 発表

---

- 研究室をベースとしてグループ分け
- 他者とプログラムする事の難しさ、楽しさを知り、1人ではできない事を実現する！
  - ぶっちゃけグループ制作は大変です

# 講義の実施形態

---

- 序盤は「クラス（オブジェクト指向）」など座学中心
- 中間試験は対面で実施予定
  - COVID-19の状況次第で実施方法を検討予定
  - 10月末～11月上旬を予定していますが、COVID-19の状況次第で変更するかもしれませんし、他の手立てを検討するかもしれません
- 11～12月は対面で応用実習系を実施予定
  - ネットワーク、ファイル入出力、XMLやJSON、WebAPI、フィジカルコンピューティングなど
- 1月に発表会
  - グループ制作が無理な場合は個人発表またはその他の方法に切り替えるかもしれません（昨年度は動画発表＋各自コメント投稿）

# 講義の流れ（あくまで目安）

---

- 1コマ目：
  - 0～20：宿題の解説
  - 20～50：今日のトピックの解説＋課題の提示
  - 50～100：今日の課題に取り組む（要提出）
- 2コマ目：
  - 0～50：今日の課題に取り組む
  - 50～100：今日の課題の解説（答え合わせ）
  - 次回までの宿題の提示

# 評価：シラバスと違います

---

- 宿題：20点
- 授業中課題：20点（要提出）
- 中間テスト：40点
- 発表会：20点
  
- 注意
  - 5回休み = 評価対象外
  - 課題のチェック状況は開示しませんので、念のため隣や前後のひとなどに一度動作を見てもらってから提出して下さい！

# 要注意！

---

- プログラミング演習は良い答えを提出することが大事なのではなく、良い答えを自分で出せる基礎力を身につけることが目的なので、他者のプログラムをコピーしない＆コピーさせないこと
- コピーなどの不正行為が発覚した場合、その学期の全ての科目登録が取り消され、懲戒処分の対象ともなります
- コピーさせた人も同様の処分になることに注意してください！

# 提供してもらわないように！

---

- みなさんがコピーしてしまうことにより、コピー提供者も迷惑を被ります
- この講義の後も
  - メディアプログラミング実習（2年春）
  - アルゴリズム基礎（2年春）
  - Webプログラミング実習（2年秋）があるため、逃げたらずっと苦しむだけ。勉強にもなりませんのでしんどいなりに立ち向かいましょう！
- これまで英語や数学とかを勉強してきたときどうしています？ どれくらいの時間をかけてきました？

# 提供しないように！

---

- 良かれと思ってやりたくなる気持ちはわかります
- 教えるのが面倒でこれみとけって気分になることもあるでしょう
  - でも、同一レポート（課題）の提出があった場合、提供された側が正直に申告しない場合は提供元・提供先の両方が処罰の対象となります
  - そんなしょうもないことで人生を狂わせてはもったいないので、くれぐれも注意してください
- 他の講義でも問題になりつつあります。絶対にレポートはコピーさせないこと！

# これまで来た質問

---

- 教えたら類似するのでは？
  - 多少は類似するでしょう。ただ、積み木でお城を作るという課題で同じ形、同じ色の積み木を同じように積むことはあり得るでしょうか？
  - 見せなければ一致しない
- 教えることを禁止するのか？
  - ぜひとも教えて下さい、ただコピーが発生するような行為は慎んでください
- 他人のプログラムを見てはだめなのか？
  - 教えるときは見ざるを得ませんので、その時は見て下さい。ただ、見せて教えないで下さい。

# 講義予定（案）

---

- 09/26 第01回 ガイダンス + 前期の復習
- 10/03 第02回 ArrayList とクラス (1)
- 10/10 第03回 クラス (2)
- 10/17 第04回 継承
- 10/24 第05回 復習
- 11/07 第06回 中間テスト
- 11/14 第07回 振り返りとファイル入出力
- 11/21 第08回 ネットワーク
- 11/28 第09回 Web API
- 12/05 第10回 フィジカルコンピューティング
- 12/12 第11回 フィジカルコンピューティング
- 12/19 第12回 フィジカルコンピューティング
- 01/16 第13回 色々
- 01/23 第14回 発表会

# 宿題の提出方法

---

- 宿題の提出期限は次回の講義開始30分前  
(13:00まで)
  - それ以降のものは受理されませんので早め早めに提出をよろしくです！
- Google Driveの各自の宿題フォルダに提出

# ここから

---

- 課題提示
- 宿題の提示
- 課題の提出
- 課題確認できたら解散

# 本日の課題

---

- 参加開始
  - まずは参加開始（出席確認）申請

# プログラミング演習II 課題

## • スケッチ名：basic\_DicePrime

- 1から6までの値をもつサイコロを順に3回振り、1回目を100の位の値、2回目を10の位の値、3回目を1の位の値とする数字を作る。
  - (例) 1回目が5、2回目が3、3回目が4の時、「534」となる。
- この3回サイコロを振ってできる数字が素数となる確率を、サイコロを3回振る試行を1000万回実施することで求め、以下のように出力せよ。\*に数字が入る。

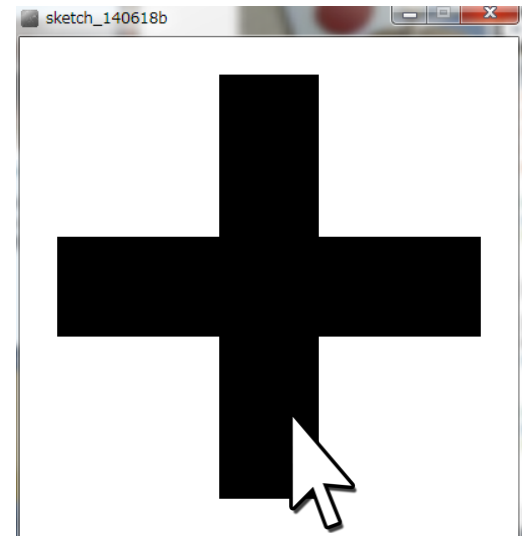
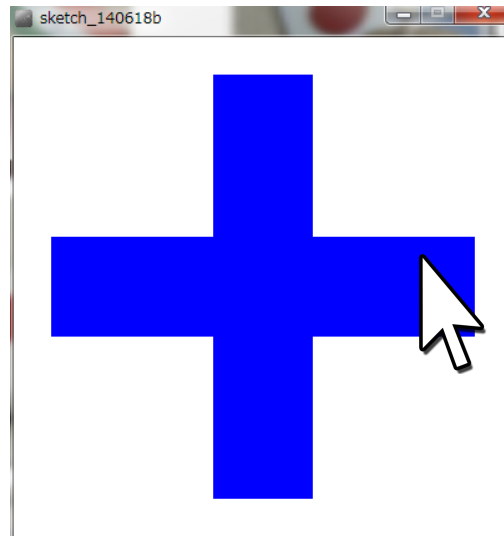
素数となる確率は0.\*\*\*\*\*

- 自身が求めた値があっているかどうかは、数学的にまたはプログラムで全パターンを走査することで検証せよ。

# プログラミング演習II 課題

## • スケッチ名：basic\_Cross

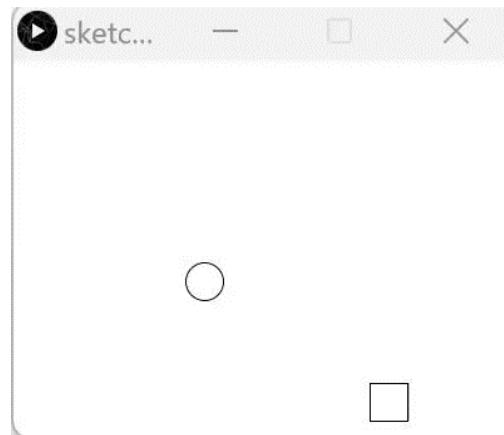
- 十字型の図形に対し、マウスクリックにより当り判定および色が変わるプログラムを書け
- 適当な十字型の図形を描き、図の【内部】をクリックしたら図形の色が赤→青→黒→赤→青→黒と順に変わるようにせよ
- 図形外部をクリックしても反応しないようにすること



# プログラミング演習II 課題

## • スケッチ名：basic\_BallRect

- 400x300のウィンドウ内を動き回る丸（直径30ピクセルの白色の円）と正方形（一辺の長さが30ピクセルの白色の正方形）を実現せよ
- ただし、その初期値はウィンドウ内のランダムな位置とし、またその速度はx、y方向それぞれ-5~5の実数値とせよ。
- また、丸も正方形も壁で跳ね返るようにせよ。跳ね返り方は図形の端でも図形の中央でもよい。



# 宿題1-1: hw\_boundA11

---

- 10個の赤色の○と, 5個の青色の□が画面内を動き回るプログラムを作成せよ
  - ただし、その速度はx、y方向それぞれ-5~5の実数値とせよ
  - また、丸も正方形も壁で跳ね返るようにせよ。跳ね返り方は図形の端でも図形の中央でもよ。

# 宿題1-2: hw\_abababab

---

- 「abababab」のaとbそれぞれに1から9までの数字を当てはめたときにできる8桁の数字について、正の約数の数が最小になるaとbの組み合わせと正の約数の数をプログラムで求め標準出力せよ
- ただし、aとbが同じ数字でも良い
  - 例： a=3, b=5のときは、35353535の約数の数を求めることになる
- 出典：数学検定 準1級の試験より
  - 大学受験数学で解けますので、数学好きな人はチャレンジしてみたら良いですよ！

# 宿題1-3: hw\_dashLine

---

- lineというメソッドは(x1, y1)から(x2, y2)まで線を描画するメソッドですが、表現を広げるため点線を描画できるようにしたい



- 今回描画する点線は5ピクセルの線, 5ピクセルの空白, 5ピクセルの線, 5ピクセルの空白と交互になるようにする
- そのような描画を可能にする  
dashline(x1, y1, x2, y2)を作成し, 描画せよ
  - 色や太さは、その前に設定されたstrokeやstrokeWeightに従うようにせよ (太い場合には間が詰まって良い)
  - 多少うねうねしてもOKとします
  - 微妙にあふれる、または微妙に短いのは良しとする