

2023/06/30  
14:30~16:00

# 静岡県「情報」教育研究会 プログラミング教育研修

日出学園中学校・高等学校  
武善 紀之



# Agenda

1. 自己紹介
2. 情報 I の全体像とプログラミングの位置付け
3. プログラミング・アラカルト
4. Pythonプログラミング授業の流れ
5. 自身の課題
6. 質疑応答(いただいた質問を共有)

# 1. 自己紹介

---

# 自己紹介

## 武善紀之 (たけよし のりゆき)



- 教員10年目 (32歳)
- 千葉県私立学校 教諭
- 中高一貫校  
(日出学園中学校・高等学校)
- パソコン部顧問
- 高校1年生担任
- 筑波大学情報学群出身
- 所有免許  
「情報」「数学」「社会」「公民」「技術」

- 東京書籍教科書編集委員  
「新編情報Ⅰ」「情報Ⅰ」「情報Ⅱ」
- Webページ

[http://high.hinode.ed.jp/share/takeyoshi/n\\_takeyoshi.html](http://high.hinode.ed.jp/share/takeyoshi/n_takeyoshi.html)

## 去年は、「南極」にいました!

- 国立極地研究所「教員派遣プログラム」
- 南極からZoom授業 (情報科)



## 情報科カリキュラム

- 高1 必修2単位「情報Ⅰ」
- 高2 選択2単位「人間と機械」(≒倫理)
- 高3 選択2単位「情報Ⅱ」  
選択2単位「実践情報」(≒演習)

# 学外の仕事1「スタディサプリ」

## スタディサプリ

- <https://studysapuri.jp/>
- 「情報 I」の全単元を満遍なく授業
- 前半を武善、後半を安藤先生が担当
- 有料(月額2,178円,無料体験14日有)

### ▼講義動画（要点整理）のイメージ

The screenshot shows a lecture slide titled "2進法とは" (What is binary). The slide content includes:

- ④ 2進法 「0」と「1」だけで色々な数値を表現する方法(コンピューターが得意!)
- ⑤ 10進法 「0」~「9」の数で数値を表現する方法(人間が得意!)
- 0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → ……
- ↑ 桁を増やせばたくさん表現可能
- 2進法でも同じです。
- 0 → 1 → 10 → 11 → 100 → ……
- ↑ 桁を増やせばたくさん表現可能

The teacher is pointing to the slide content. The logo "スタディサプリ" is visible in the bottom right corner of the video frame.

- 座学授業の参考になると思います。
- ただ、本当に面白いネタは削っています(尺 & 配信都合)。

# 学外の仕事2「文科省解説動画」

高等学校情報科『情報Ⅰ』  
授業・研修用コンテンツ

- <https://www.nttls-edu.jp/joho/>
- 無料
- 「情報Ⅰ」の単元を実習ベースで解説
- 武善は「データの活用」単元を担当



- 実習授業のネタです。
- 他の分野もあります(プログラミングも超おすすめ)
- 生徒に動画を見せて、そのまま取り組ませるのもありかと。

# 学外の仕事3「NHK高校講座」

## NHK 高校講座

- [https://www.nhk.or.jp/kokokoza/jouho\\_u1/](https://www.nhk.or.jp/kokokoza/jouho_u1/)
- 「監修講師」として参加
- 「講師による説明」スタイルを完全排除。体験的に情報Ⅰの学びを深める。
- 本編以外に、「(連載)ゲームを創ろう!」「Watch情報の宇宙(企業等取材)」コーナーあり。

- 授業で部分的に見せるのもありかと思います。
- 僕の回は、まもなく7/7放送。
- (準備が)むちゃくちゃしんどいです。



武善 紀之  
たけよし のりゆき  
日出学園中学校・高等学校教諭

# 学外の仕事4「高校探究プロジェクト」



東京学芸大学  
先端教育人材育成推進機構  
高校探究プロジェクト

- <https://g-tanq.jp/is/4>
- 4名の教員が年間指導計画を説明
- 年間指導計画の作り方やポイント、工夫を紹介

武善紀之Webページ | トップページ | 授業 | 南極 | 雑文 | リンク集

武善 紀之 Takeyoshi Noriyuki



Last Updated: 2023/05/20 Since: 2020/07/10

Profile 発表・執筆 単発取材

学園の情報科のページは[コチラ](#)。ICTのページは[コチラ](#)。

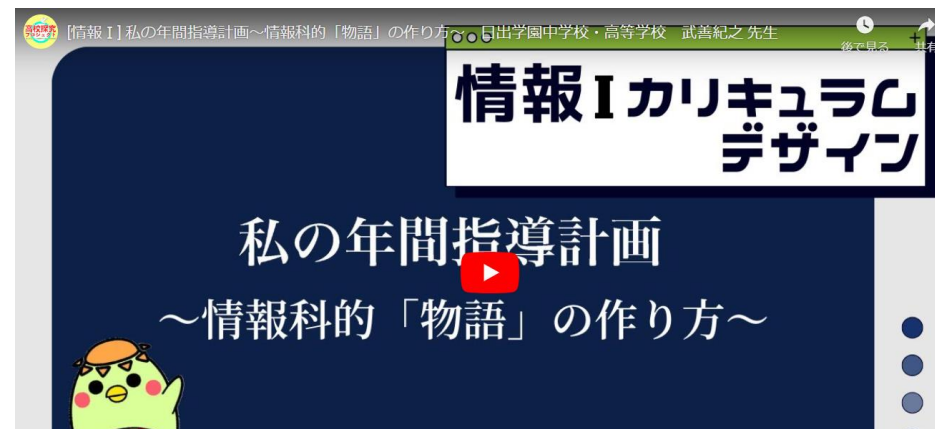
## Profile

### 主な経歴

- 教員10年目。「情報科」に加え、一部「数学科」「公民科」の授業を担当。
- 日出学園中学校・高等学校(2014-)
  - 日出学園法人企画室 ICT推進チームリーダー(2018-2020)
  - パソコン部顧問
  - 2023年度授業: 情報科(情報Ⅰ・社会と情報・情報の科学)、公民科(学校設定科目「人間と機
- 高等学校情報科用教科書「新編情報Ⅰ」「情報Ⅰ Step Forward!」編集委員(東京書籍, 2022-)

その他過去の実践  
等は全て個人サイ  
トに掲載

[http://high.hinode.ed.jp/s\\_hare/takeyoshi/n\\_takeyoshi.html](http://high.hinode.ed.jp/s_hare/takeyoshi/n_takeyoshi.html)



## 「情報Ⅰ」年間指導計画(例)

内容	補記(主にスト	
<1学期>「私と情報」まずはコンピュータを使えるようになる!「自分」から周囲へアンテナを広げる		
1	メディアリテラシー(グラフのトリック・相関と因果・バイアス)	情報科≠PCの
2	コミュニケーションと情報デザイン(抽象化・可視化・構造化)	主に情報の発
3	【自由実習】PC教室オリエン&ピクトグラム作成(PowerPoint)	初コンピュ
4	【自由実習】ピクトグラム作成	単純図形の組
5	【自由実習】相互評価発表&ユニバーサルデザイン	相互評価(匿名
6	知的財産権	自身の作品を
7	デジタル化①(2進法)	現代のコミュ
8	デジタル化②(文字・音と画像のデジタル化)	標本化の精度
9	デジタル化③(DTM実習)	テキスト音楽工
10	デジタル化④(色と動画のデジタル化)	「効率の良い作
11	デジタル化⑤(圧縮技術)	動画ファイルの
12	Webページ作成	テキストの構



- 明後日、静岡に行く予定です。



<https://k-hana-tori.com/>

2.プログラミングで、  
何を・どこを目指すか？

---

# 事前アンケートの結果

- 専門科目でないため、皆様の実践例など色々教えていただきたいです。
- プログラミング自体、自分も全然わかっていないし、まだ授業も行っていないため、基本からじっくり教えてほしい。
- 今年度より校内で初めて情報 I の授業を開講するため、まだ未実施です。
- Python3、GoogleColab、FizzBuzzと配列の初歩まで教えます。ソートは発展課題として一部生徒のみ
- C言語。生徒全員が普通教室にてクロームブックでBYOD回線で「オンラインC」使用。

## 【多様な方が参加されている】

- 様々な教材(無料のものだけ)を、感想と共に紹介する。
- 授業の流れを紹介する(自作プリント,教材と共に)。
- 個別の課題については、質問事項で拾っていく。

プログラミングで、  
何を・どこを目指すか？

(の前に……)

「**情報科**」とは、そもそも何なのか？

# 情報科の歴史は案外長い！

×新教科「情報Ⅰ」では扱う内容が大幅に増加！

○「プログラミング」や「シミュレーション」については、  
今まで「選択」であった内容が「必修」になった。



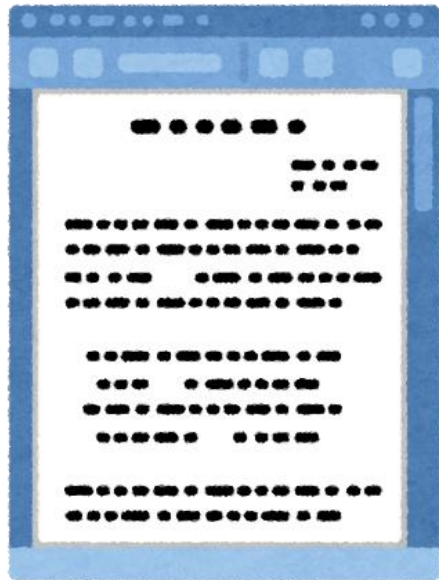
→ 既に豊富な実践例がある！

# しかし実際は…

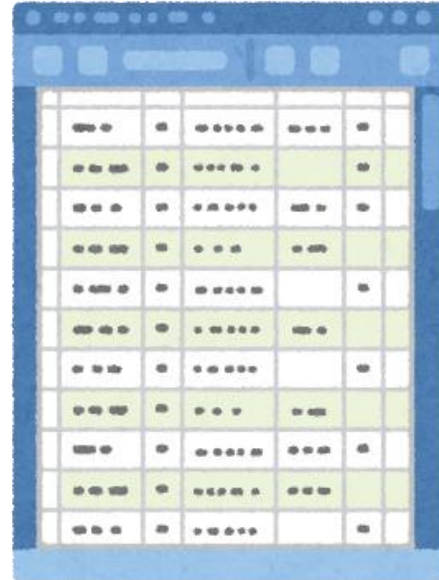
## 高校「情報科」教員, 4割近くが保健体育, 音楽の免許で

毎日新聞 2018年9月4日

<https://mainichi.jp/articles/20180904/k00/00e/040/196000c>



Wordで志願理由書を書こう！



Excelの関数を20個マスターしよう



好きなものをみんなに紹介しよう！

Word, Excel, PowerPoint, タイピング……  
オフィスソフトウェアの操作習得に終始してしまっていた

# 新しい情報科は“仕切り直し”



## 【情報入試】

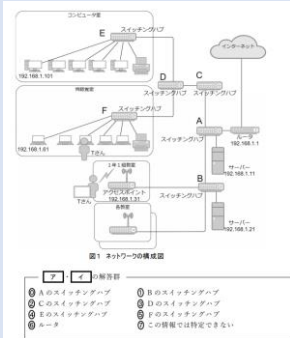


図4 アルファベットの出現頻度を数え上げる配列

0	1	2	3	4	5	6	7	8	...	20	21	22	23	24	25
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

図5 出現頻度を求めるプログラム

```
(01) Angoubun = ["p", "q", "e", "b", ...]
(02) 配列 Hindo のすべての要素に 0 を代入する
(03) i を 0 から 要素数 (Angoubun)-1 まで 1ずつ増やしながら:
(04)   bangou = 差分 [ケ]
(05)   もし bangou != -1 ならば: ケ @Angoubun[i]
(06)   [コ] = [コ] + 1
(07) 表示する (Hindo)
```

【関数の説明】

要素数 (値) ...配列の要素数を返す。  
例: Data=["M", "I", "s", "s", "1", "s", "s", "1", "s", "s", "1", "p", "p", "1"]の時  
要素数 (Data) は 11 を返す

差分 (値) ...アルファベットの「a」との位置の差分を返す  
値がアルファベット以外の文字であれば -1 を返す  
例: 差分 ("e") は 4 を、差分 ("x") は 23 を返す  
差分 ("5") や差分 (" ") は -1 を返す

大学入学共通テストへの「情報」の出題について (情報処理学会)

<https://www.ipsj.or.jp/education/edu202012.html>

## 【小中学校 1人1台整備】

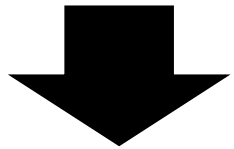
- ・ オフィスソフトには入学時点で、ある程度習熟
- ・ 「プレゼンやったことない」生徒が、今よりも稀に
- ・ アンケート調査等は既に体験済み
- ・ プログラミングも未体験者は相当数減少

# 情報Ⅰ・情報Ⅱへ

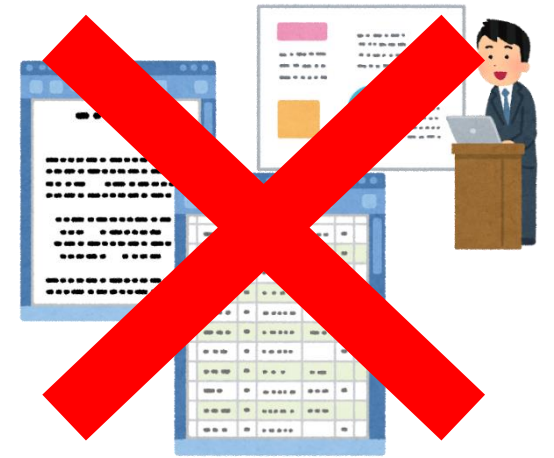
# 新学習指導要領「情報Ⅰ」

## 「情報Ⅰ」のコンセプト

1. コンピュータを活用して、**問題解決**ができるようになること
2. **情報科学(Computer Science)**をきちんと学ぶこと



- ◆第1章 情報社会の問題解決
- ♥第2章 コミュニケーションと情報デザイン
- ♣第3章 コンピュータとプログラミング
- ♠第4章 情報通信ネットワークとデータの活用





# 1.コンピュータを活用して問題解決ができるようになる

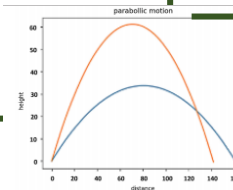
「情報デザイン」「データサイエンス」「プログラミング」は、  
問題解決の三種の神器



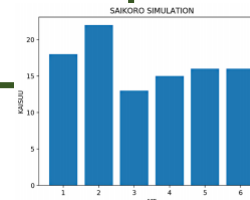
魅力的なポスターを作って集客しよう！  
情報デザイン



幾らで売れば、一番利益が出るだろう？  
プログラミング



過去の来客数を分析して、傾向を判断！  
データサイエンス



# 1.コンピュータを活用して問題解決ができるようになる

「情報デザイン」「データサイエンス」「プログラミング」は、

問題解決の三種の神器


ただし、

ここで要求されている力は  
コンピュータを操作する力ではない！

溢れる「情報」, 処理できている?



# 私たちは、その「情報」を処理できている？



量が多くて  
整理できない！

何が本当か  
わからない！

欲しい情報が  
見つからない！

相手にうまく  
伝わらない！

# 「情報」って何を学ぶ科目なの？

ときにはコンピュータを活用しながら、  
「情報の海」を乗り越こなす力を身に着ける科目

情報の多面的な解釈

効率の良い情報処理

情報の的確な伝え方

情報へアクセスする力

新しい情報の創造

情報を管理する力

# 「情報」って何を学ぶ科目なの？

ときにはコンピュータを活用しながら、  
「情報の海」を乗り越こなす力を身に着ける科目

情報の多面的な解釈

効率の良い情報処理  
プログラミング

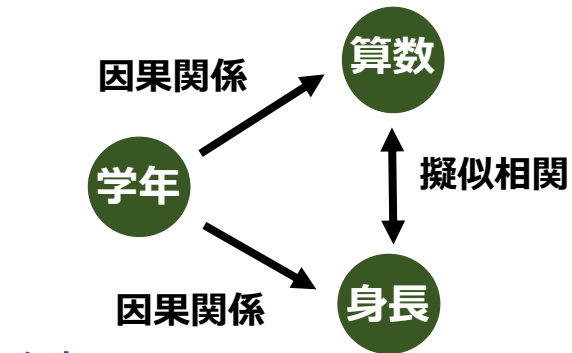
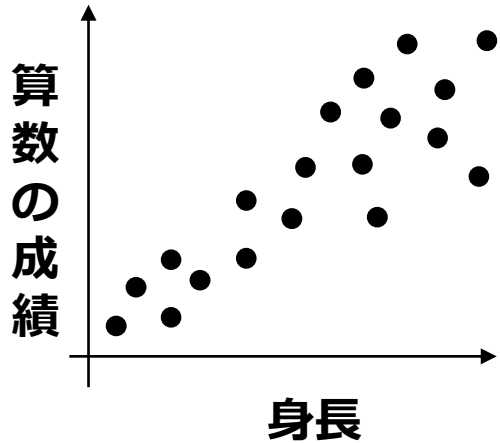
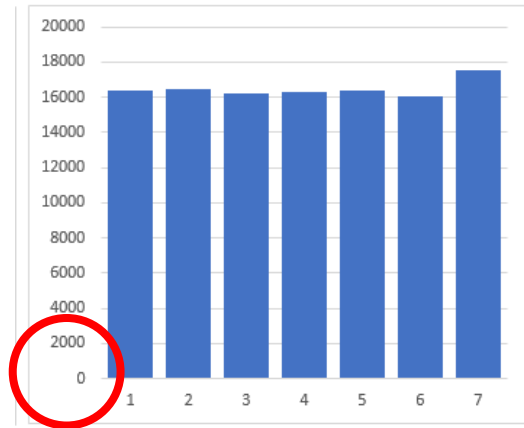
情報の的確な伝え方  
情報デザイン

情報にアクセスする力  
ネットワーク技術

新しい情報の創造  
データ分析

情報を管理する力  
データベース

# 例えば、僕の1時間目と1学期<メディアリテラシー>



参考  
統計的消去で擬似相関を見抜こう!  
<http://hoxo-m.hatenablog.com/entry/20130711/p1>

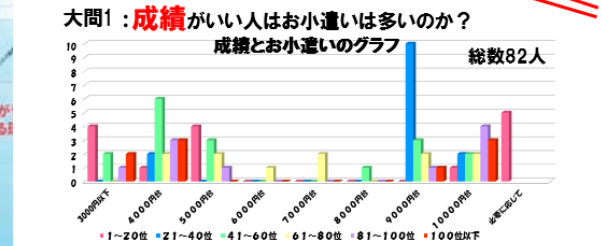
中学生の学習時間のながら行動

中学生が勉強する時

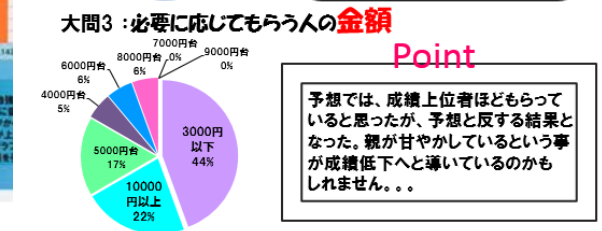
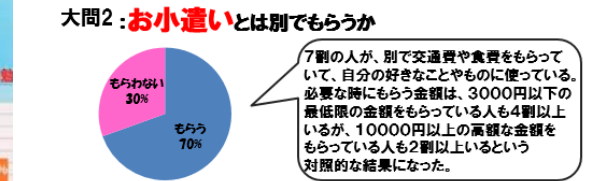
<1979年(235人)>

<成績別に見た資本を覗きながら勉強する中学生>

## 高2のお小遣い事情!! 100



このグラフを見ると3000円以下は1~20位の成績上位者が最も多く、10000円台になると81位~成績下位者が多い傾向になっているのがわかる。



「情報」 = パソコンの時間 → 「情報」の扱い方を学ぶ時間 (パソコンは手段, 人間理解のメタファー)

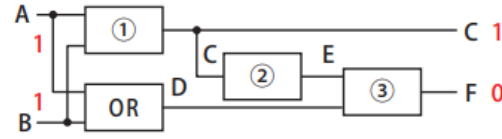
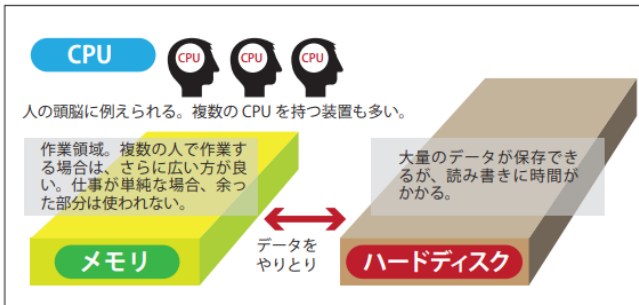
## 「情報 I」のコンセプト

1. コンピュータを活用して、  
問題解決ができるようになること
2. 情報科学(Computer Science)を  
きちんと学ぶこと



# 2. 「情報科学」をきちんと学ぶ

## 1つの学問領域としての「情報科学」



真理値表					
入力		途中経過		出力	
A	B	D	E	C	F
0	0	0	1	0	0
0	1	1	1	0	1
1	0	1	1	0	1
1	1	1	0	1	0



- アプリケーション層
- トランスポート層
- インターネット層
- ネットワークインタフェース層

「情報科学」を学ぶことは、「問題解決」にも役立つ。

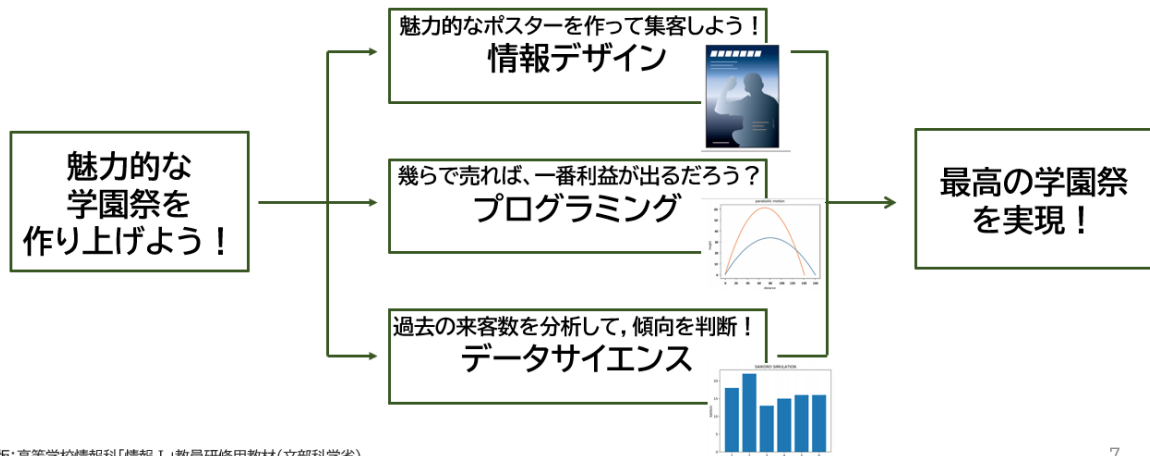


科学的理解に裏打ちされた「情報活用能力」の育成

# この2つを見失わなければ、「情報 I」は乗り切れる

## 1. コンピュータを活用して問題解決ができるようになる

「情報デザイン」「データサイエンス」「プログラミング」は、  
問題解決の三種の神器



7

## 2. 情報科学をきちんと学ぶ

1つの学問領域としての情報科学

CPU  
人の脳速に劣らされる。複雑のCPUを持つ装置も多い。  
作業領域。複雑の人で作業する場合は、さらに広い方がよい。仕事の領域は、決った領域は使われない。  
メモリ  
データをやりとり  
ハードディスク  
大量のデータが保存できるが、読み書きに時間がかかる。

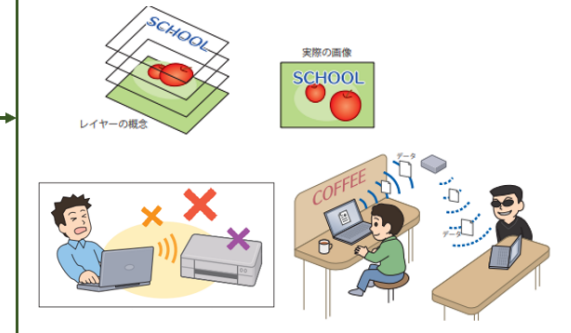
真理値表

入力		途中経過			出力	
A	B	D	E	C	F	
0	0	0	1	0	0	
0	1	1	1	0	1	
1	0	1	1	0	1	
1	1	1	0	1	0	

アプリケーション層  
トランスポート層  
インターネット層  
ネットワークインタフェース層

あ

「情報科学」を学ぶことは、  
「問題解決」にも役立つ。



科学的理解に裏打ちされた「情報活用能力」の育成

図版: 高等学校情報科「情報 I」教員研修用教材(文部科学省)

8

# 「情報Ⅰプログラミング教育」への(外野の)誤解

× 「情報Ⅰ」で「プログラミング」が必修化。  
「プログラミング」の勉強が一番大変！

- ◆全体として「情報Ⅰ」は消費者教育(「情報Ⅱ」はエンジニア養成的)。
- ◆プログラミングも、情報を学ぶ「手段」の1つ。

「プログラミングスクール」は数多あるが、  
「データサイエンススクール」「情報デザインスクール」  
「コンピュータサイエンススクール」はあまり無い。

「他教科」免許状指導者が大活躍する機会かも  
プログラミング > 旧数学B「数値計算とコンピュータ」  
データサイエンス > 数学Ⅰ「データの分析」、理科実験  
情報デザイン > 美術「色彩理論」……。

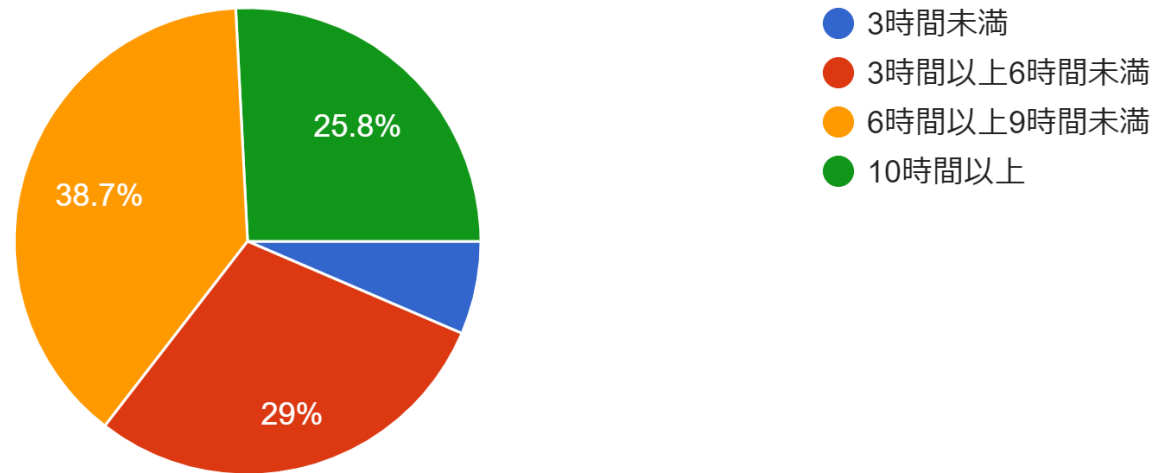
# 「プログラミング教育」の実際のところ

## 【時間的なところ】

様々な研修会に参加しても**6～10時間**くらいとおっしゃる先生が多い。  
(それ以上やると、他の分野を削っていく必要性が出てくる。もちろん複  
合として扱うことも可能だが…)

プログラミングの指導に掛けている（掛ける予定の）年間の授業時間数を教えてください。

31件の回答



# でも……入試にはそれなりに出る予定

## 共通テスト「情報 I」

- おそらく全分野の中で、もっとも授業との乖離が激しい。  
(演習時間的な意味で)

## • ただ、内容自体はそこまで多岐に渡らない。

- 制御構造(繰り返し・分岐)
- 変数・配列
- 演算(剰余計算%)
- 関数
- アルゴリズム(大小比較・合計)

時間的には  
8hで十分

```
(1) kakaku = 46
(2) min_maisu = 100
(3) サ を シ から 99 まで 1 ずつ増やしながら繰り返す:
(4)   shiharai = kakaku + tsuri
(5)   maisu = ス + セ
(6)   もし ソ < min_maisu ならば:
(7)     タ = ソ
(8) 表示する (min_maisu)
```

図2 最小交換硬貨枚数を求めるプログラム

このプログラムを実行してみたところ3が表示された。46円を支払うときの最小交換硬貨枚数は、支払いで50円玉が1枚、1円玉が1枚、釣り銭で5円玉が1枚の計3枚なので、正しく計算できていることが分かる。同様に、kakakuの値をいろいろと変えて実行してみたところ、すべて正しく計算できていることを確認できた。

サ, ソ, タ の解答群 \_\_\_\_\_  
① maisu ② min\_maisu ③ shiharai ④ tsuri

シ の解答群 \_\_\_\_\_  
① 0 ② 1 ③ 99 ④ 100

ス, セ の解答群 \_\_\_\_\_  
① 枚数(shiharai) ② 枚数(kakaku) ③ 枚数(tsuru)  
④ shiharai ⑤ kakaku

では、

何を・どのようにやらせれば良いのか？

# 3.プログラミング・アラカルト

---

# プログラミングのイメージ

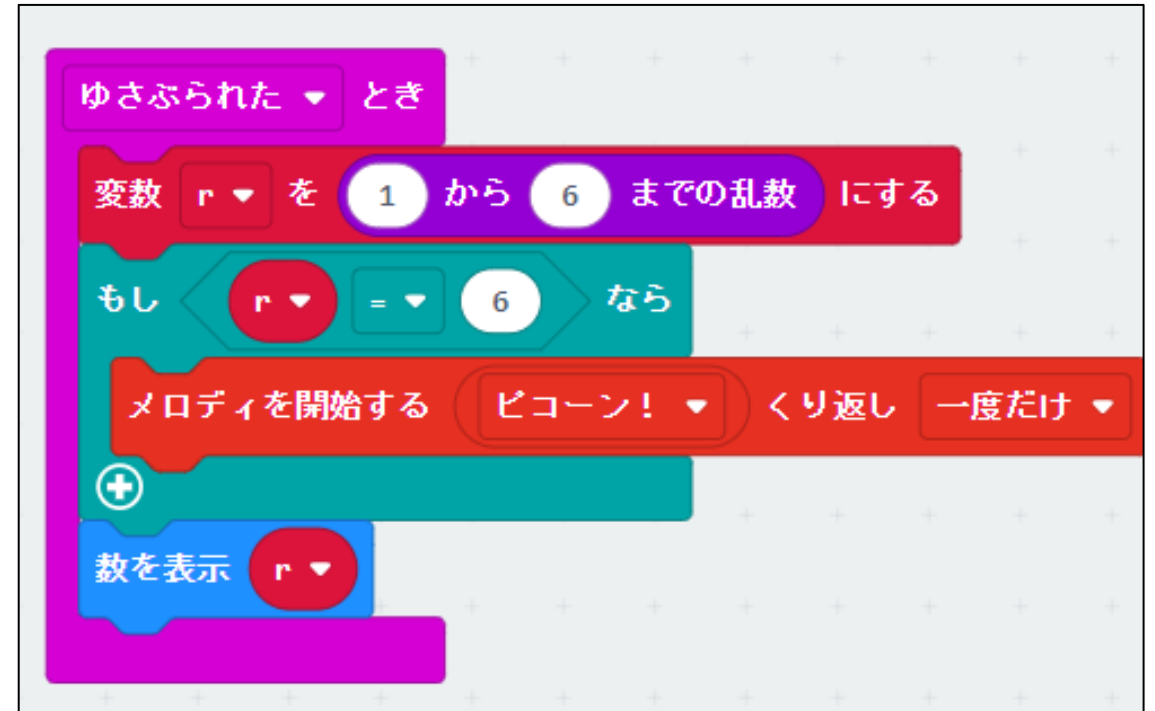
```
[ ] #ここをどんどん書き換えていく
import random
te=['gu','choki','pa']
you=input()
you=int(you)
print(te[you])
r=random.randint(0,2)
print(te[r])
if you==r:
    | | print('draw')
if you==0:
    | | if r==1:
    | | | | print('win')
    | | if r==2:
    | | | | print('lose')
if you==1 and r==0:
    | | print('lose')
```



# 実際は、これらも“プログラミング”

```
かめた=タートル!作る(青)線の色。
ペン=タートル!作る。
ゴール=ペン!(赤)線の色 90 右回り 90 歩く 図形を作る。
ペン!(黒)線の色。
「|n| ペン!(n*25) 歩く 72 左回り」!11 繰り返す。
コース=ペン!図形を作る。
ペン!消える。
コース!140 250 移動する。
ゴール!140 250 移動する。
かめた:衝突=「時計!停止」。
ゴール:衝突=「|相手|
  「相手==かめた」!なら「
    時計!停止。
    ラベル!“ゲームクリア” 作る 30 文字サイズ
  」実行。
```

日本語プログラミング



ブロックプログラミング

# 共通テストも“日本語プログラミング”

- (1) `kakaku = 46`
- (2) `min_maisu = 100`
- (3) `サ` を `シ` から 99 まで 1 ずつ増やしながら繰り返す：
- (4)     `shiharai = kakaku + tsuri`
- (5)     `maisu = ス + セ`
- (6)     もし `ソ` < `min_maisu` ならば：
- (7)     ┌   └ `タ` = `ソ`
- (8) 表示する (`min_maisu`)

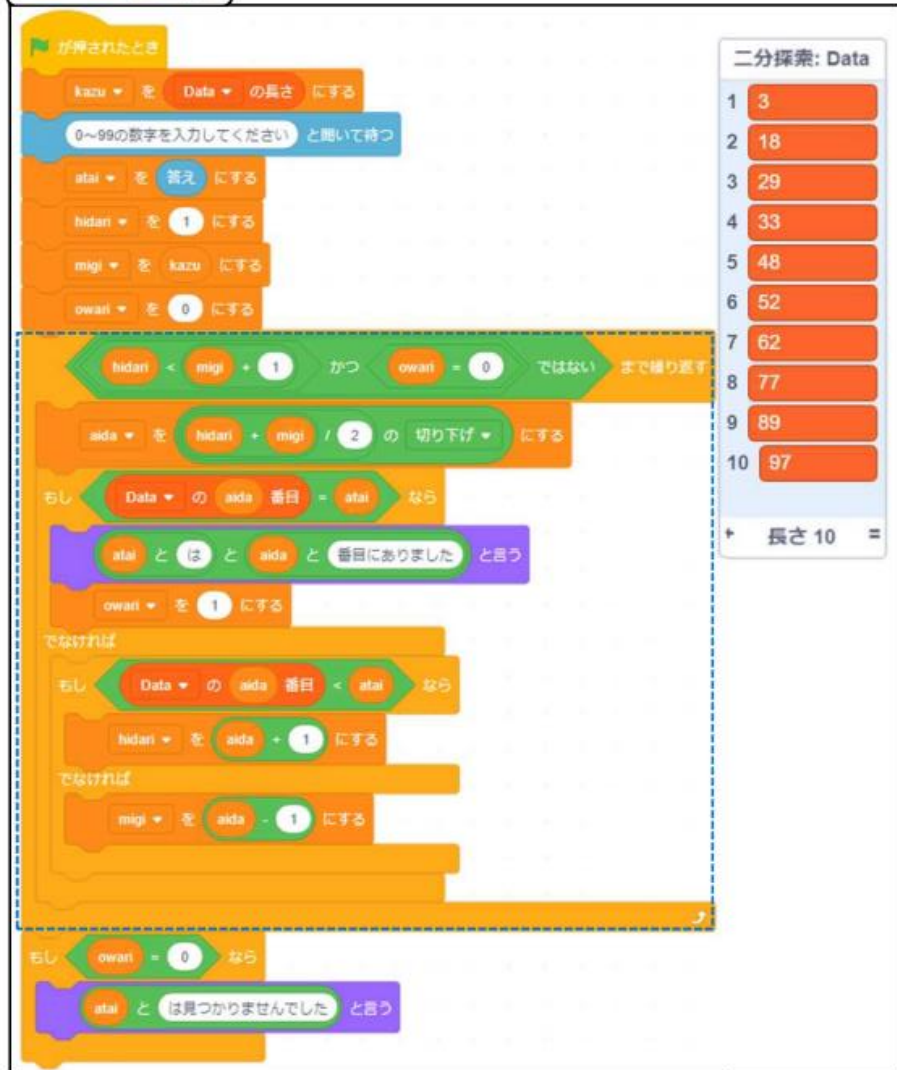
図2 最小交換硬貨枚数を求めるプログラム

# 学習指導要領も共通テストも、テキスト必須ではない

## Python3記述例

```
1 data=[3,18,29,33,48,52,62,77,89,97]
2 kazu=len(data)
3 atai=int(input('0~99の数字を入力してください'))
4 hidari=0;migi=kazu-1
5 owari=0
6 while hidari <= migi and owari==0:
7     aida=(hidari+migi)//2      #演算子//は商の整数部分
8     if data[aida]==atai:
9         print(atai,'は',aida,'番目にありました')
10        owari=1
11    elif data[aida]<atai:
12        hidari=aida+1
13    else:
14        migi=aida-1
15 if owari==0:
16     print(atai,'は見つかりませんでした')
17 print('添字',' ','要素')
18 for i in range(0,kazu,1):
19     print(i,' ',data[i])
```

## Scratch記述例



The Scratch script implements a binary search algorithm. It starts with a 'when clicked' event, sets 'kazu' to the length of the 'Data' list, and prompts the user for a number 'atai'. It initializes 'hidari' to 1 and 'migi' to 'kazu'. A loop continues while 'hidari' is less than or equal to 'migi' and 'owari' is 0. Inside the loop, 'aida' is calculated as the integer part of the average of 'hidari' and 'migi'. A conditional check determines if 'atai' is found at 'aida'. If found, 'owari' is set to 1 and a message is shown. If not found, 'hidari' is incremented if 'atai' is greater than 'data[aida]', or 'migi' is decremented if 'atai' is less. After the loop, a final check for 'owari' determines if the number was found or not. A '二分探索: Data' panel on the right shows the list [3, 18, 29, 33, 48, 52, 62, 77, 89, 97] and its length of 10.

基本的には「構造理解」を求めている。  
細かな言語の文法は問わない。

# ここからの流れ

- 前半

「とっつきやすい」プログラミング環境(無料)を紹介

- 後半

Python + Google Colaboratoryで実習環境を紹介  
(今年は、この内容を膨らませて授業実施予定)

# おすすめのサイト



～紹介するもの～

●「プログラミングっぽい」を体験  
テキスト音楽エディタSakura

●「ブロック」プログラミング  
Scratch,micro:bit,アルゴロジック

●「日本語」プログラミング  
ドリトル

まずはここからプログラミング事例集

[https://ten.tokyo-shoseki.co.jp/ten\\_download/2018/2018018104.htm](https://ten.tokyo-shoseki.co.jp/ten_download/2018/2018018104.htm)

# テキスト音楽エディタ「サクラ」

Q.生徒がプログラミングに対して0ベースの場合、取っ掛かりとしてやるべきこと、やったほうが良いことがあれば知りたい。

・日本語で「作曲」ができる(DTM)

・テキスト音楽「Sakura」

<https://sakuramml.com/>

・制御構造まで学ぶことが可能

## 【目的】

①生徒の実態把握

②テキストによるモノづくり

③デジタル化単元の箸休め

```
プログラム ①  
② トラック 1 0!  
~{ど} = {n36,} //Bass-drum!  
~{た} = {n38,} //Snare!  
~{つ} = {n42,} //Hi-hat hat close!  
~{く} = {n44,} //Hi-hat half open!  
~{ち} = {n46,} //Hi-hat open!  
~{ぱ} = {n49,} //Crash cymbal!  
音符 8 【3 どつたつどつたつ】どつたつたたた!
```



まずはここからプログラミング事例集

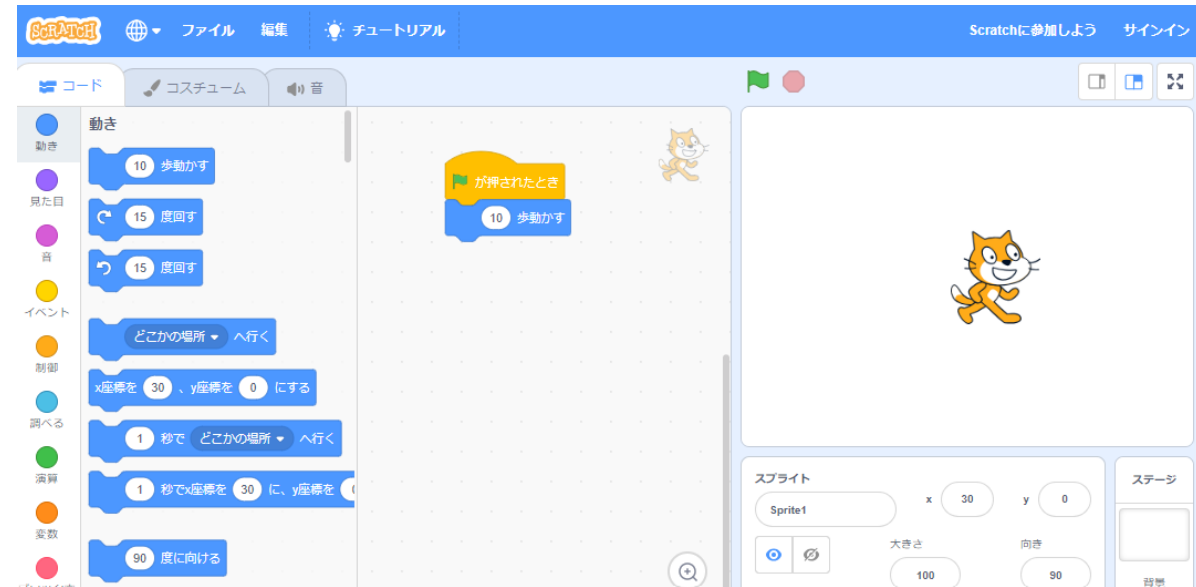
[https://ten.tokyo-shoseki.co.jp/ten\\_download/2018/2018018104.htm](https://ten.tokyo-shoseki.co.jp/ten_download/2018/2018018104.htm)

# Scratch

- 小学校でも多数取り組まれているブロック型プログラミング
- <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

## 【目的】

①モノづくりとしてのプログラミング



# micro:bit

- Scratchのようなブロック
- 実機と接続して使用もできるし、画面上だけでも利用可能

<https://makecode.microbit.org/>

- (去年は)1学期に実施。
- 1台3000円程度  
(昔は2000円くらいだった)

## 【目的】

①ハードウェアを織り交ぜたプログラミング





# アルゴリズム

- アルゴリズム(二重ループ等)を学ばならこのサイト  
<https://algo.jeita.or.jp/prm/2/index.html>
- 夏休みの宿題等でも良いかも

## 【目的】

### ①制御構造の理解

06 無限ループ

設定 リセット 使い方 記録

↑ 1  
↑ 1  
回転 1  
回転 1  
← LOOP ∞  
←

# ドリトル

- 教育用プログラミング言語「ドリトル」
- 日本語でコーディング可能
- ブラウザ環境「Bit Arrow」が使用可能  
(教員用システム有り)

<https://bitarrow.eplang.jp/bitarrow/?Login/form>

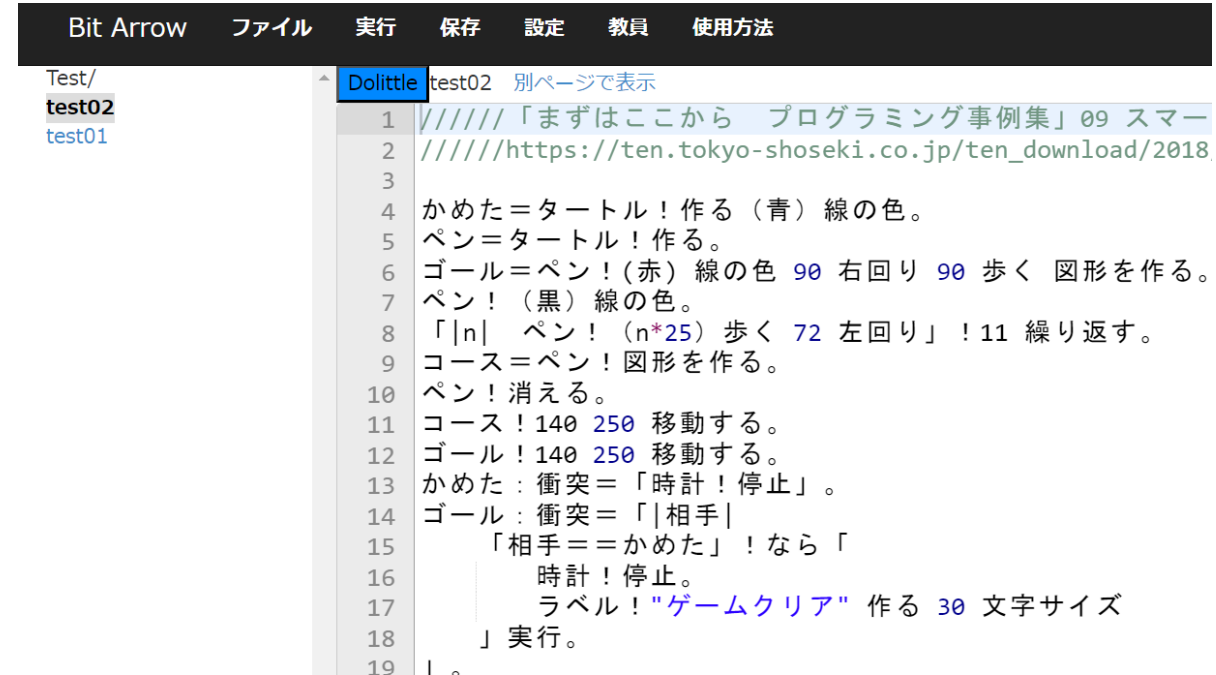
- QRコードを生成し、  
スマホのセンサを利用することも可能  
(プログラミング事例集に記載あり)

- かなり高度なことも指導できる
- 授業では「1時間で学ぶソフトウェアの仕組み」が、導入に丁度よい

<https://dolittle.eplang.jp/d1h>

## 【目的】

- ①「書かなければ」動かないことへの理解
- ②「1行書いて」「実行」の繰り返しによる開発



The screenshot shows the Bit Arrow web-based programming environment. The top menu bar includes 'Bit Arrow', 'ファイル', '実行', '保存', '設定', '教員', and '使用方法'. The file explorer on the left shows a directory structure: 'Test/' containing 'test02' and 'test01'. The main editor area shows a file named 'Dolittle test02 別ページで表示' with the following code:

```
1 ///////////////////////////////////////////////////////////////////「まずはここから プログラミング事例集」09 スマ
2 ///////////////////////////////////////////////////////////////////https://ten.tokyo-shoseki.co.jp/ten_download/2018
3
4 かめた=タートル！作る（青）線の色。
5 ペン=タートル！作る。
6 ゴール=ペン！（赤）線の色 90 右回り 90 歩く 図形を作る。
7 ペン！（黒）線の色。
8 「|n| ペン！（n*25）歩く 72 左回り」！11 繰り返す。
9 コース=ペン！図形を作る。
10 ペン！消える。
11 コース！140 250 移動する。
12 ゴール！140 250 移動する。
13 かめた：衝突＝「時計！停止」。
14 ゴール：衝突＝「|相手|
15     「相手==かめた」！なら「
16         時計！停止。
17         ラベル！"ゲームクリア" 作る 30 文字サイズ
18     」実行。
19 |。
```

まずはここからプログラミング事例集

[https://ten.tokyo-shoseki.co.jp/ten\\_download/2018/2018018104.htm](https://ten.tokyo-shoseki.co.jp/ten_download/2018/2018018104.htm)

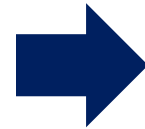
で、結局どれがおすすすめなの？

→全部やりました。

# 昨年度、神奈川県情報部会でこんな発表を・・・

## 1学期・夏休み・2学期で分割実施した プログラミング学習 ～課題と次年度への改善～

- ・年間授業時間は50時間程度
- ・4章構成の「情報1」を満遍なく指導すると、プログラミング指導の時間は8時間程度が限界。



- 1学期 **日本語+ブロック**
  - ・テキスト音楽Sakura (1)
  - ・ドリトル (1)
  - ・micro:bit (2)
- 夏休み **ブロックで長編**
  - ・Scratch (発表会1)
- 2学期 **テキスト**
  - ・Python (3)

# 昨年度、神奈川県情報部会でこんな発表を・・・

## 1学期・夏休み・2学期で分割実施した プログラミング学習 ～課題と次年度への改善～

### (分割理由)

1. 生徒認知における「課題の分離」  
(ex)「Pythonは辛かったけど、  
プログラミングは楽しかった」
2. 夏休みの有効活用
3. 進路決定にプログラミング学習を  
間に合わせたい(文理選択が10月)

### ■ 1学期 日本語+ブロック

- ・テキスト音楽Sakura (1)
- ・ドリトル (1)
- ・micro:bit (2)

### ■ 夏休み ブロックで長編

- ・Scratch (発表会1)

### ■ 2学期 テキスト

- ・Python (3)

# 1学期期末試験

プリントあります↓

<http://high.hinode.ed.jp/share/takeyoshi/class.html>

問1 以下は授業で作成したドリトルによる「宝探しゲーム」の仕様に基づき、作り上げたソースコードである。

<仕様>

プレイヤーが「左ボタン」「右ボタン」によって「かめた」を操作し、「かめた」が🍷の上を通過すると、🍷が消える。

<コード>

- |   |  |
|---|--|
| 1 | かめた=タートル!作る。                               |
| 2 | 左ボタン=ボタン!"左" 作る。                           |
| 3 | 左ボタン:動作="かめた! [ X ] 左回り"。                  |
| 4 | 右ボタン=ボタン!"右" 作る。                           |
| 5 | 右ボタン:動作="かめた! [ X ] 右回り"。                  |
| 6 | 時計=タイマー!作る。                                |
| 7 | 時計!"かめた!10 歩く"実行。                          |
| 8 | タートル!作る "tulip.png" 変身する ペンなし -300 150 位置。 |

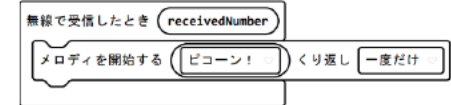
(1) このゲームを実行した後、左ボタンを一定間隔で複数回押すと、「かめた」は正三角形を描きながら、画面上の特定箇所を周回し続けた。ソースコード[ X ]に入る数値は、次のうちどれか。

【解答群】 ① 30    ② 60    ③ 120    ④ 150

(2) このソースコードは不完全であり、仕様を満たせていない。実行すると、どのような現象が生じるか。

- ① プレイヤーが左ボタン、右ボタンでかめたを操作できない。
- ② かめたがチューリップの上に来ると、かめたが止まる。
- ③ チューリップの上を通過しても、チューリップが消えない。
- ④ かめたが動き出さない。

問5. 「送信機 micro:bit は引き出ししに入っており、引き出しを開けると音が鳴る。それと同時に、受信機 micro:bit から音が鳴る」プログラムを受信機・送信機の双方で作成した。受信機のプログラムが右である時、正しい送信機のプログラムはどれか。ただし、無線のグループ設定はともに1番となっている。



①	②
③	④

① 送信機プログラム:  
ずっと  
メロディを開始する (ビコーン!) くり返し 一度だけ  
もし (30 < 明るさ) なら  
無線で数値を送信 (0)

② 送信機プログラム:  
ずっと  
もし (40 < 明るさ) なら  
無線で数値を送信 (0)  
メロディを開始する (ビコーン!) くり返し 一度だけ

③ 送信機プログラム:  
ずっと  
もし (30 > 明るさ) なら  
無線で数値を送信 (1)  
メロディを開始する (ビコーン!) くり返し 一度だけ

④ 送信機プログラム:  
ずっと  
もし (20 < 明るさ) なら  
無線で数値を送信 (1)  
メロディを開始する (ビコーン!) くり返し 一度だけ

## 種類1.本当に基礎的な一問一答問題

問2 次のプログラムの出力結果は **ウエ** である。

```

(01) x=10
(02) h=5
(03) x=x*h*(1/2)
(04) print(x)

```

問3 次のプログラムの出力結果は **オ** である。

```

(01) x=29
(02) if x%6==0:
(03)     print('0')
(04) else:
(05)     print('1')

```

問4 次のプログラムの出力結果は **カキ** である。

```

(01) n=0
(02) for i in range(10):
(03)     n=n+1
(04) print(n)

```

## 種類2「情報」試作問題(暗号)

Tさん: 復号は必ずしも反対にシフトする必要はないよね。例えば9文字右にシフトされていた場合、復号するには9文字左にシフトでも良いけど、右に **アイ** 文字シフトすることもできるね。図2のようにアルファベットに0~25の番号を割り当てて考えてみると、暗号化してx番目の文字になった時、復号はx+**アイ**の値が **ウエ** 以下であればx+**アイ**番の文字に置き換わるけど、**ウエ**より大きい場合は、x+**アイ**-**オカ**番の文字に置き換えれば復号できるよね。

Mさん: 暗号化で文字を何文字シフトしているか分かれば、この復号法で解読できるよね。どうやったら分かるかな。

Tさん: すべての可能性、つまりシフトしない時を除いた **キク** 通りをプログラムで試せばいいんじゃない?

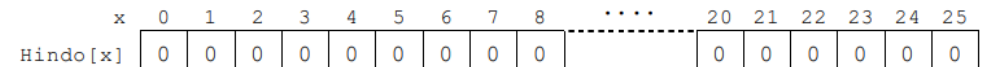


図4 アルファベットの出現頻度を数え上げる配列

```

(01) Angoubun = ["p", "y", "e", "b", ..., (省略) ..., "k", "b", "d", "r", "."]
(02) 配列 Hindo のすべての要素に0を代入する
(03) i を0から要素数(Angoubun)-1まで1ずつ増やしながら:
(04)     bangou = 差分(ケ)           ケ ①Angoubun[i]
(05)     もし bangou != -1 ならば:
(06)         コ = コ + 1           コ ④Hindo[bangou]
(07) 表示する(Hindo)

```

図5 出現頻度を求めるプログラム

# ただ、うまくいかなかった・・・。


適切な字下げができない。

目に見えない構造が取れない？ micro:bitではいけたのに……。

↓このようなスライドを示して説明しても、結構キツイ。

## Pythonは頭の空白で構造を表す

行頭を揃える！  
ブロックを意識する！



```
import random

for k in range(10):
    r=random.randint(1,100)
    if r== 6 :
        print('当たり！')
    else:
        print('はずれ')
```

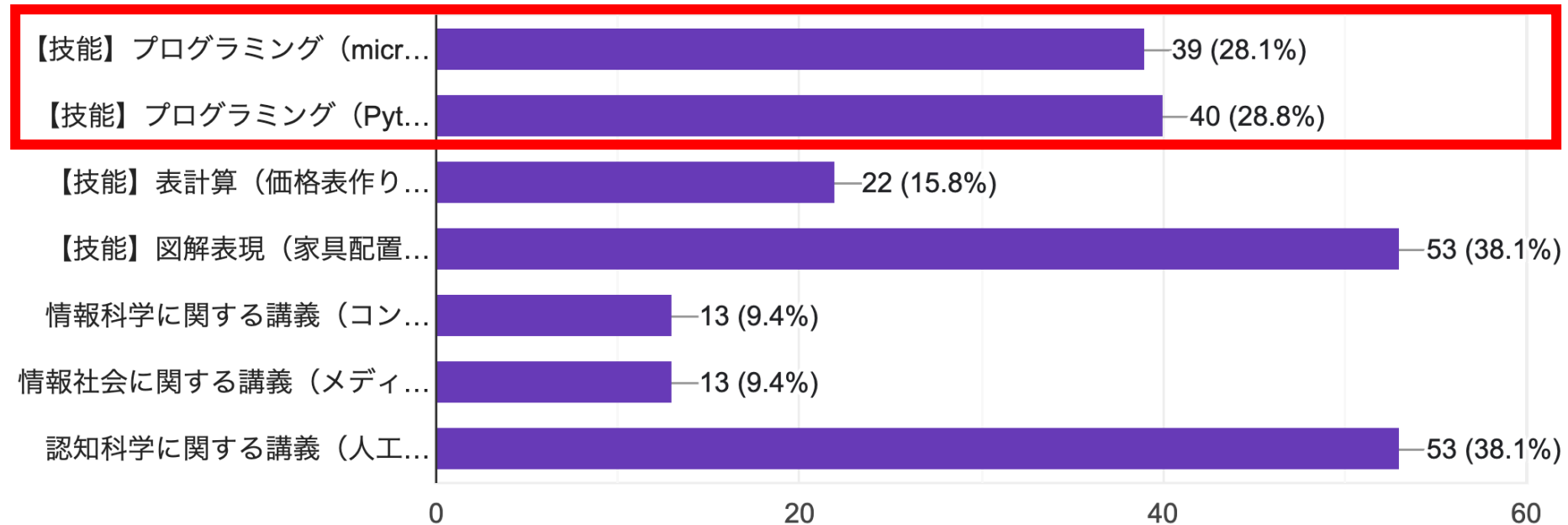
The image shows a comparison between Scratch and Python. On the left, Scratch code blocks are shown: a loop '変数 k を 0 から 9 までくりかえす', a block '変数 r を 1 から 100 までの乱数 にする', an 'もし r = 6 なら' block containing '文字列を表示 "atari!"', and an 'でなければ' block containing '文字列を表示 "hazure!"'. On the right, the equivalent Python code is shown with indentation: 'import random', 'for k in range(10):', 'r=random.randint(1,100)', 'if r== 6 :', ' print('当たり！')', 'else:', ' print('はずれ')'. A vertical dashed line separates the two, and a green box highlights the Python code block.



# 人気度もPythonとmicro:bit等で変わらない

## • 人気度は意外にもPythonとmicro:bitで変わらなかった

ここまでの情報（理論）の授業の中で、あなたが「...感じた内容を選んでください（2つまで選択可）  
139件の回答



# 今年については

- Python(テキストプログラミング)にもう少し寄せていくつもり。
- 「ブロックプログラミング」→「共通テスト」は飛躍が激しすぎるかも。

```
(1) kakaku = 46
(2) min_maisu = 100
(3) サ を シ から 99 まで 1 ずつ増やしながら繰り返す：
(4)   shiharai = kakaku + tsuri
(5)   maisu = ス + セ
(6)   もし ソ < min_maisu ならば：
(7)   |   |   タ = ソ
(8) 表示する(min_maisu)
```

図2 最小交換硬貨枚数を求めるプログラム

# ここからの流れ

- 前半

「とっつきやすい」プログラミング環境(無料)を紹介

- 後半

Python + Google Colaboratory の授業の流れを紹介  
(今年は、この内容を膨らませて授業実施予定)

# 4. Python プログラミング授業の流れ

---

- テキストプログラミング
- 本当は奥が深いが、触りだけならとても簡単。
- 入試にもっとも近い。DNCL2は、ほとんどPython。
- 事前アンケートでも、もっとも選ばれている。

```
[ ] #ここをどんどん書き換えていく
import random
te=['gu','choki','pa']
you=input()
you=int(you)
print(te[you])
r=random.randint(0,2)
print(te[r])
if you==r:
    print('draw')
if you==0:
    if r==1:
        print('win')
    if r==2:
        print('lose')
if you==1 and r==0:
    print('lose')
```

# Pythonの流れ

## ①Python入門(1h)

- 20分一緒に(環境構築)、残りは基本的に**自学**。
- 教科書の例題を一通り写経。Hello World、変数、if else、forまで。

-----しばらく時間をあけて-----

## ②Pythonによるガチャシミュレーション(1h)

## ③表計算シミュ&大数の法則プログラミング(1h)

## ④おみくじ・会話bot・じゃんけん作成(1h)

- 高度な内容:**リスト,論理演算子**
- 長め:**共通テスト意識**

まず初めに…

プログラミング = 問題解決の手段として  
授業内でストーリーを構築

情報 I・社会と情報（理論） 2学期 第1回

# モデル化とシミュレーション

## Modeling and Simulation

教科書 p.78~p.81, p.152~p.153



# 2学期の内容

- 教科書第3章「コンピュータを**活用する**」
- 実践多め (9~13h, 実習4+発表会2+演習1+座学2)
  - 表計算 (Excel or スプレッドシート) やります
  - テキストプログラミング (Python) やります
  - 2進法の計算 やります
  - 統計 やります
- 2学期は「試験以外の比重が大きい」  
(夏休みの宿題③+Pro発表会⑦+シミュ発表会⑦+統計分析レポート③予定)

# 家具配置シミュレーション

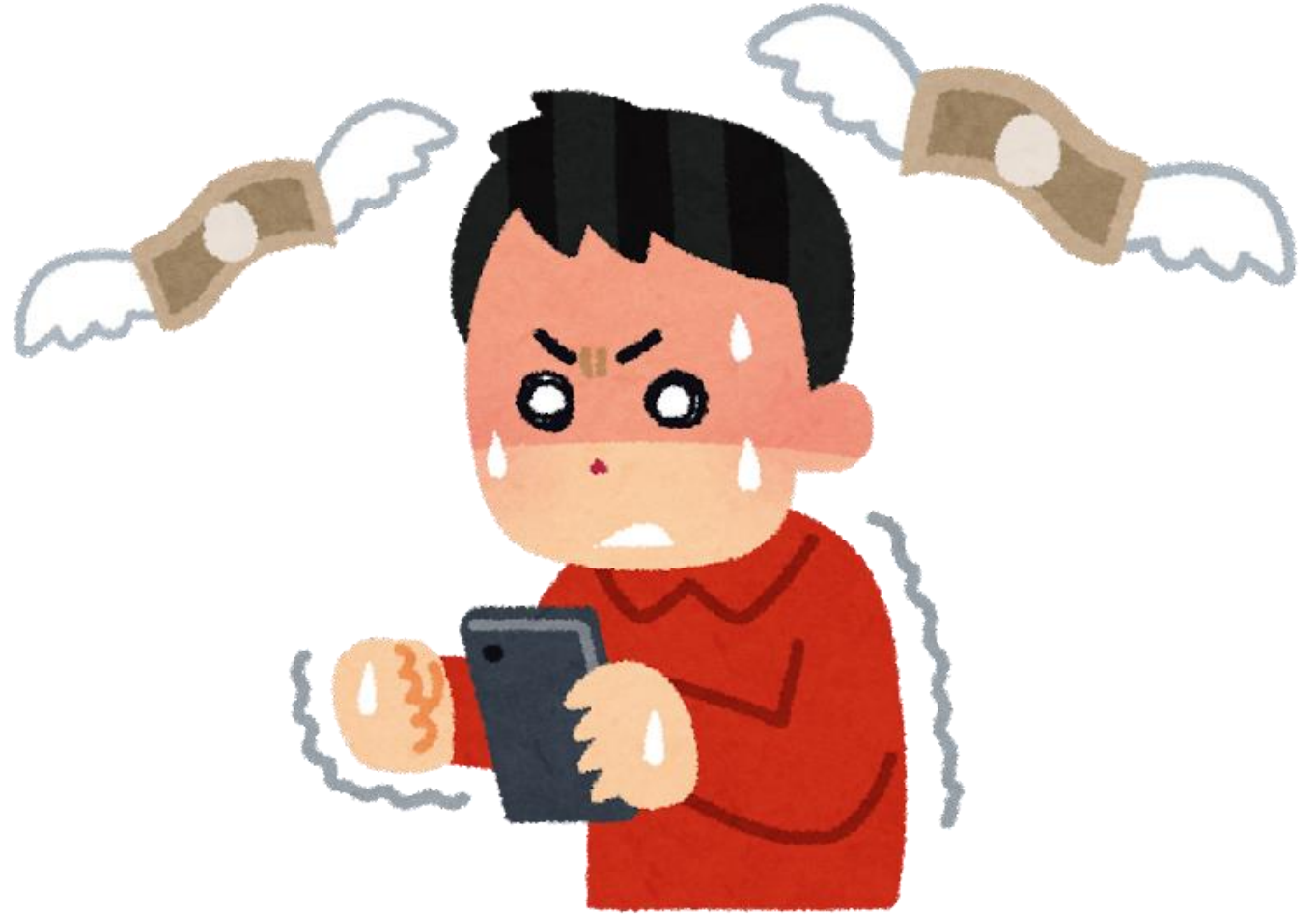
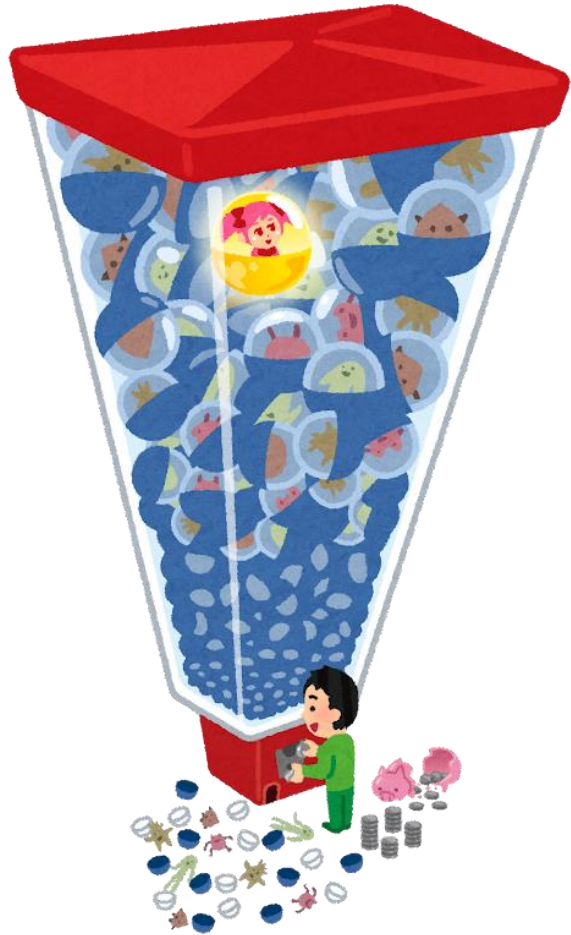
ソファを購入するときのシミュレーション

# この単元の進め方

1. モデル化とシミュレーションの入門  
「家具配置シミュレーション」

2. モデル化とシミュレーション×プログラミング  
「ガチャの確率シミュレーション」

こんな経験ありませんか？



当たり1%のガチャ、  
100回引けば1回は当たる？

排出率X%のキャラ、  
何円課金すれば引くことが出来る？

今度のシミュレーションの道具は…

# Python



```
k=0
```

```
while k<10: #kの値が10より小さい間はずっと繰り返す。
```

```
    print(k)
```

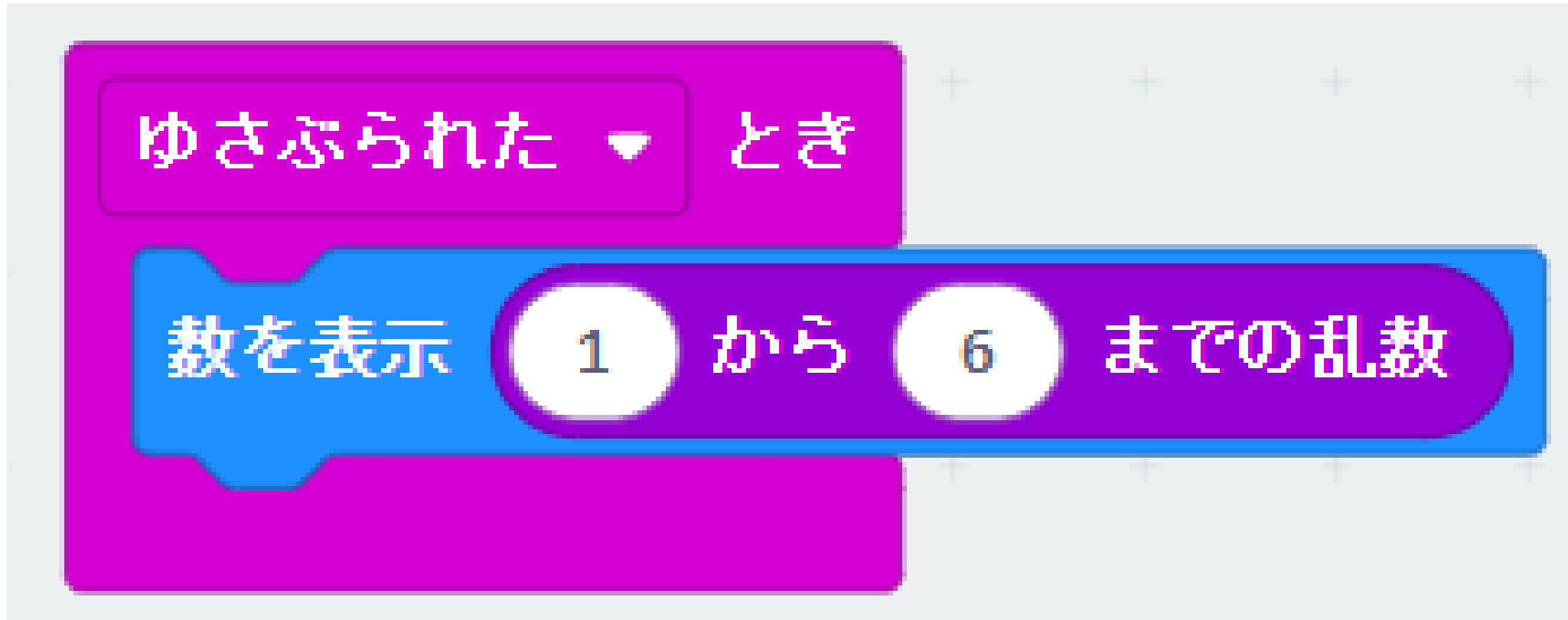
```
    k=k+1 #実行するたびに、kの値が1増えていく。
```

席は自由です。

- 学校のPCでログインしてください。
  - 画面転送用
  - ID: jt01~jt42 or 自分のID
  - PASS: 1111 or 自分で設定したPASS
- ChromebookでGoogle Classroomを開いてください。

※作業PCにマウスを繋いでおきましょう。

# 乱数





# 当たり判定

The image shows a Scratch script for a hit check. It starts with a 'when clicked' event block. The first block is 'set variable r to random number between 1 and 6'. The second block is an 'if r equals 6' conditional block. Inside the 'if' block, there are three sub-blocks: 'start melody with Beep!', a 'plus' block, and 'show number r'. The 'start melody' block is set to repeat once.

```
when clicked ▾ とき  
変数 r ▾ を 1 から 6 までの乱数 にする  
もし < r ▾ = ▾ 6 > なら  
メロディを開始する Beep! ▾ くり返し 一度だけ ▾  
+  
数を表示 r ▾
```

情報 I ・社会と情報 (理論) 2学期

ガチヤシミュレーション  
(表計算)

# クラスごとの結果

	当たり
4-1	73.1%
4-2	55.6%
4-3	51.9%
4-4	67.3%
4-5	56.5%

	当たり
5-1	72.7%
5-2	54.3%
5-3	51.6%
5-4	66.7%

# モンテカルロ法

カジノ都市

モナコ

モンテ・カルロ

理論上はどれくらい当たるのか？



**数理モデル**を作って、  
シミュレーションをしてみよう！

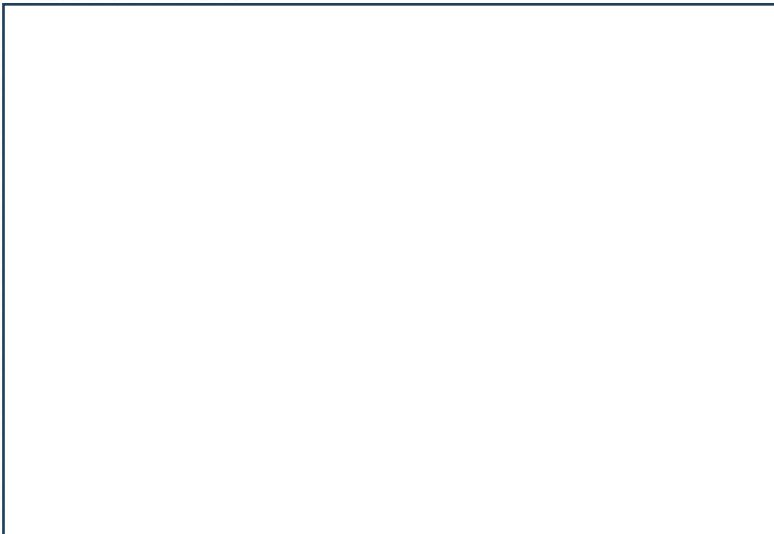
理論と実験はどれくらいで近づくのか？



大規模なシミュレーションをやってみよう！

# 授業で実施したシミュレーション

【配置図】 ←



【集計表】 ←

家具	実際のサイズ		縮小サイズ		価格
	幅	奥行	幅	奥行	
折りたたみベッド(レビン)	91	201	4.55	10.05	10175
折りたたみベッド(レビン)	91	201	4.55	10.05	10175
折りたたみベッド(レビン)	91	201	4.55	10.05	10175
折りたたみベッド(レビン)	91	201	4.55	10.05	10175
折りたたみベッド(レビン)	91	201	4.55	10.05	10175
↑関数で作る→					50875 ←

#ここをどんどん書き換えていく。

```
import random
for cnt in range(10):
    r=random.randint(1, 100)
    if r==6:
        print(' atari ')
    else:
        print(' hazure ')
```

当たり外れ		
試行回数	1回も当たらない確率	1回は当たる確率
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

# シミュレーションは身近にもたくさんある

- ガチャ②シート「シミュレーション」例を開いてみる。



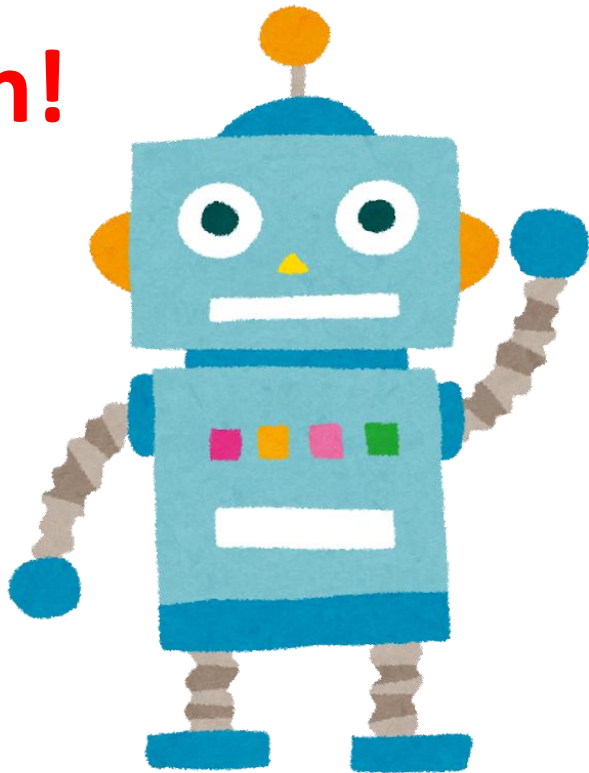
# まとめ

- 今日で「モデル化とシミュレーション」単元は終わりです。
- 日常のあらゆる場面で、「ちょっとシミュレーションしてみよう（スライド・プログラミング・表計算を使ってみよう）」と思えたら嬉しいです。

# 次回予告

- 次回で**プログラミングも最後!**
- 「コンピュータと“じゃんけん”」ゲームを作ります。

Win!



情報 I ・社会と情報（理論） 2学期

実践プログラミング

- プリントが配られたら半分に折って「左側」だけ見えるように。

# 本日は3本立て! (プログラミングの最後)

1. おみくじゲーム【7分】
2. 会話bot(リスペクト・イライザ)【8分】
3. じゃんけんゲーム【30分】

## 今日の目標

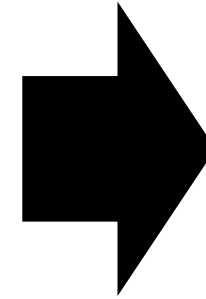
プログラミングで、  
作りたい物を作れるようになるろう!

# 1. おみくじゲーム

---

# 前々回の復習 10連ガチャ

```
import random
for cnt in range(10):
    r=random.randint(1,6)
    if r==6:
        print('atar i')
    else:
        print('hazure')
```



```
hazure
hazure
hazure
hazure
atar i
hazure
hazure
atar i
hazure
hazure
```

# もっと細かく分岐させるには？

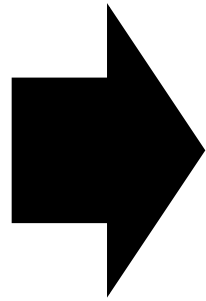
$r=1$  → 凶  
 $r=2,3$  → 小吉  
 $r=4,5$  → 中吉  
 $r=6$  → 大吉





ひたすら分岐で書けば良い。

```
import random
for cnt in range(10):
    r=random.randint(1,6)
    if r==1:
        print('凶')
    elif r==2:
        print('小吉')
    elif r==3:
        print('小吉')
    elif r==4:
        print('中吉')
    elif r==5:
        print('中吉')
    elif r==6:
        print('大吉')
```



凶  
小吉  
小吉  
小吉  
大吉  
小吉  
中吉  
中吉  
小吉

もっと簡単に  
書くには？

# 「変数」と「配列」



## 2. 会話bot(リスペクト・イライザ)

---

# 復習

## 対話型AI

## ②ものしりなAI

ELIZA(1964) 「いらいざ」

- ・頭痛がする→どう痛いの？
- ・母が私を嫌い→誰があなたを嫌いですって？

人々はすぐにプログラムに感情的に没頭

人工無脳（≡Twitterのボット）

発展したものが現在のSiri

### 3.じゃんけんゲーム

---

# 勝敗判定 you-pc

## コンピュータ(pc)

 (gu-,0)

 (choki,1)

 (pa-,2)

プレイヤー(you)

 (gu-,0)

Draw 0

Win -1

Lose -2

 (choki,1)

Lose 1

Draw 0

Win -1

 (pa-,2)

Win 2

Lose 1

Draw 0

# 勝敗判定

コンピュータ(pc)

 (gu-,0)

 (choki,1)

 (pa-,2)

プレイヤー(you)

 (gu-,0)

Draw 0

Win 2

Lose 1

 (choki,1)

Lose 1

Draw 0

Win 2

 (pa-,2)

Win 2

Lose 1

Draw 0

最適解  $\rightarrow (you-pc+3)\%3$



# Pythonの流れ

## ①Python入門(1h)

- 20分一緒に(環境構築)、残りは基本的に**自学**。
- 教科書の例題を一通り写経。Hello World、変数、if else、forまで。

-----しばらく時間をあけて-----

## ②Pythonによるガチャシミュレーション(1h)

## ③表計算シミュ&大数の法則プログラミング(1h)

## ④おみくじ・会話bot・じゃんけん作成(1h)

- 高度な内容:**リスト,論理演算子**
- 長め:**共通テスト意識**

# 4. 他分野との連携

---

# 課題「Pythonの構造を取れない生徒が多すぎる」

適切な字下げができない。

目に見えない構造が取れない？ micro:bitではいけたのに……。

↓このようなスライドを示して説明しても、結構キツイ。

## Pythonは頭の空白で構造を表す

行頭を揃える！  
ブロックを意識する！



```
import random

for k in range(10):

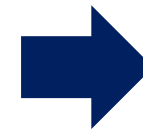
    r=random.randint(1,100)

    if r== 6 :

        print('当たり!')

    else:

        print('はずれ')
```



## 【分析】 「情報デザイン」が 鍵？

- 「可視化」「構造化」に関連する実習を重点的にやるべき？
- 「ベタ書き文書」を「箇条書きにする」ような実習

# 対策:「情報デザイン」で「構造化」を重視

- ・今まで「ピクトグラム」だけだった「情報デザイン」実習を改変
- ・デジタル化単元の最後に、総合実習で「Webページ作成(3h)」

パソコン部です。パソコン部の部員にはパソコン初心者も経験者もあり、年度初めに、自分ができることを考え話し合っ、やる内容を決めます。今年度は、新たに学校へ導入された3Dプリンタを用いた3Dオブジェクトの造形、3Dゲームアプリの制作が特に盛んです。主な活動を紹介すると、ペンタブレットを用いたイラスト製作、DTM (UTAUによる歌声合成, DominoによるMIDI打ち込み, Audacityによる波形編集)、プログラミングによるゲーム制作、動画制作、メディアアートの講習などを行っています。活動日は、火曜日と木曜日と隔週土曜日です。顧問は武善先生・佐久間先生・原田先生です。活動場所は、メディアルームです。

## パソコン部

パソコン部の部員にはパソコン初心者も経験者もあり、年度初めに、自分ができることを考え話し合っ、やる内容を決めます。今年度は、新たに学校へ導入された3Dプリンタを用いた3Dオブジェクトの造形、3Dゲームアプリの制作が特に盛んです。

### 主な活動

ペンタブレットを用いたイラスト製作、DTM (UTAUによる歌声合成, DominoによるMIDI打ち込み, Audacityによる波形編集)、プログラミングによるゲーム制作、動画制作、メディアアートの講習などを行っています。

### 活動日

## パソコン部

パソコン部の部員にはパソコン初心者も経験者もあり、年度初めに、自分ができることを考え話し合っ、やる内容を決めます。今年度は、新たに学校へ導入された3Dプリンタを用いた3Dオブジェクトの造形、3Dゲームアプリの制作が特に盛んです。

### 主な活動

ペンタブレットを用いたイラスト製作、DTM (UTAUによる歌声合成, DominoによるMIDI打ち込み, Audacityによる波形編集)、プログラミングによるゲーム制作、動画制作、メディアアートの講習などを行っています。

### 活動日

Webの構造化!

## パソコン部

パソコン部の部員にはパソコン初心者も経験者もあり、年度初めに、自分ができることを考え話し合っ、やる内容を決めます。今年度は、新たに学校へ導入された3Dプリンタを用いた3Dオブジェクトの造形、3Dゲームアプリの制作が特に盛んです。

### 主な活動紹介

ペンタブレットを用いたイラスト製作、DTM (UTAUによる歌声合成, DominoによるMIDI打ち込み, Audacityによる波形編集)、プログラミングによるゲーム制作、動画制作、メディアアートの講習などを行っています。

### 活動日

火曜日と木曜日と隔週土曜日です。

文章の構造化!

- ・HTMLの見出しタグによる**構造化**
- ・HTMLとCSSの分離で**構造化**
- ・画像配置(フォルダ構成)で**構造化**
- ・箇条書きで**構造化**

etc...

(参考) B-03 毎学期やって定着させる「文章の構造化」(高速スライド作成術)  
東京都立南多摩中等教育学校 御家雄一先生

# プログラミングに関わる潜在能力は何？

## 昨年度

### ■ 1学期 日本語+ブロック

- ・テキスト音楽Sakura (1)
- ・ドリトル (1)
- ・micro:bit (2)

### ■ 夏休み ブロックで長編

- ・Scratch (発表会1)

### ■ 2学期 テキスト

- ・Python (3)

- ・多様な言語で、共通の枠組みを認知
- ・ブロックによる「構造」の認知  
→テキストへの橋渡し

## 今年度

### ■ 1学期 テキスト

- ・テキスト音楽Sakura (1)
- ・Webページ作成 (3)

### ■ 夏休み ブロックで長編

- ・Scratch (発表会1)

### ■ 2学期 テキスト

- ・Python (4)

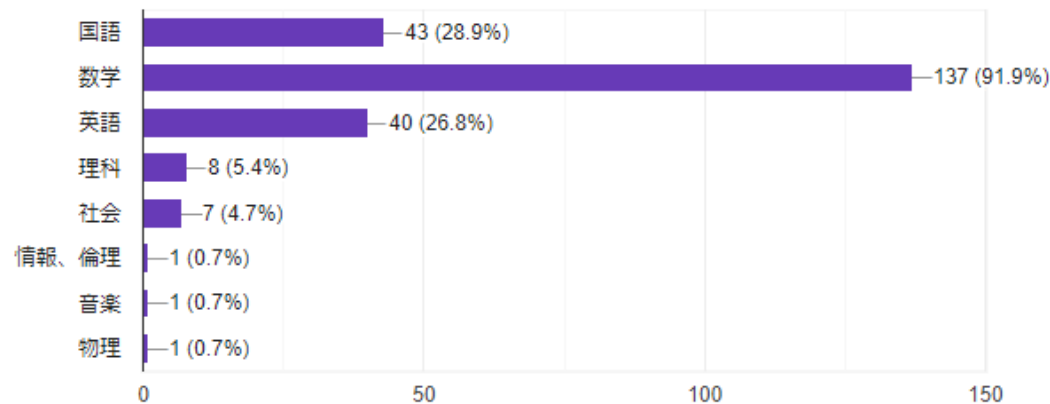
- ・他単元でプログラミング潜在能力  
を育成する視点の重要性
- ・テキストによる「構造」の具現化  
(でも、完成物はビジュアル表現)  
→テキストへの橋渡し

# 細かい感想1

- 共通テストの長文問題、果たしてプログラミング学習で出来るようになるか？
  - 正答率を見ると、プログラミングに入る前に多くの生徒が諦めている。
  - 文章読解能力(情報処理能力)が課題。
  - 1つ前の雑感とも関連するが、どうしても生徒の認知は「数学だから出来ない」に収束。

プログラミングと関係の深い教科は何だと思えますか？ (〇〇が得意だと、プログラミングも出来る、のイメージ)

149 件の回答



- 「理論確率シミュレーション」の回では、特に「数学」を苦手とする生徒が引いていくのがわかった (そして、そんな生徒が大部分)。「理論確率」と「経験確率」の比較は面白いが、少し長めの文章を読ませて「お小遣いシミュレーション」みたいな別題材でアプローチしたほうが良い？
- 定期テストで高得点だった生徒に、「なぜ点が取れたか？」をアンケート調査してみる。

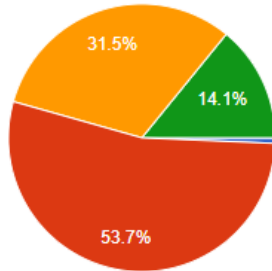
# 細かい感想2

- 写経はやらないと、ダメだった。

- 振り返りアンケートを見ると、そもそも取り組んでいない(アンケート結果)。

「Python入門」はどれくらい取り組みましたか？

149 件の回答

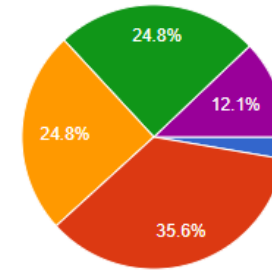


- 全くやらなかった (授業中もやらなかった)
- 授業時間内では指示された時に、取り組んだ
- 授業時間外も取り組んだ (全ては終わらなかった)
- 授業時間外も取り組んだ (全て終わった)

コピー

「Python入門」はどれくらい終わりましたか？

149 件の回答



- 一切手をつけなかった
- 3割ぐらいは終わった (自分の名前を10回表示するなど)
- 5割ぐらいは終わった (変数まで。三角形の面積計算など)
- 7割ぐらいは終わった (分岐まで。年齢でAdult、Child表示)
- 全て解き終わった (文字入力・繰り返しまで)

コピー

- 他の流れは悪くない気がするので、**写経の時間を含めると約10時間は最低限必要？**  
(DTM①、ドリトル①、micro:bit②、Scratch発表会① / 写経①、ガチャ②、じゃんけん①)
- ブロックでは構造が取れるので、アルゴリズムは不要？

- **タイピング速度より、特殊記号の打ち方練習が必要。**

- 最後の授業まで、「かっこ→(」打ちの生徒が、数人。全角文字を打ち続けてエラーする。
- 「絶対に全角文字を使わない」を指示として徹底。「当たり」も「atari」と表記させる。
- 寿司打代わりに、**こういったツール？** <https://typing.twi1.me/game/38473>

# 細かい感想3

## プログラミングを学んでどのような力が身についたと思いますか？

### 「プログラミングの授業を面白かった」と回答した層(25点満点25点)

- ・ 諦めずに最後まで取り組む力／諦めない力
- ・ コンピュータを正しく動かすために必要なことを考える力
- ・ 物事を論理的に考える思考力と、試行錯誤をする力／物事を順序立てる力
- ・ 「こうすればこうなるだろう」ということを考える力
- ・ 間違いを見つける力
- ・ 文を組み立てる力／言葉と言葉をつなぎ合わせる力／どうすれば簡単に伝えられるか
- ・ 色々な工夫をする力
- ・ 問題の推測や改善をする力

### 「プログラミングの授業を面白くなかった」と回答した層(25点満点10点以下)

- ・ コンピュータを動かすための指示の仕方
- ・ 行動を言語化する力
- ・ 間違いを見つける力
- ・ あんまり実感はないが、いい経験をしたと思える
- ・ 分岐
- ・ パソコンが少しいじれるようになった
- ・ プログラミングをやる力
- ・ へこまずに、一生懸命、授業についていく力
- ・ タイピング能力、考察する力

プログラミングを「面白い」と感じた生徒の多くが、「技能」ではない面に着目している。プログラミングを「面白くなかった」と感じる生徒ほど、技能に注目した感想を抱きがち。

→方向性は判断できないが、「プログラミングをどのように捉えるか(思考訓練か？技能か?)」と「興味・関心」には関係がありそう。



# 5. 質疑応答

---

# 6.この場の質疑応答

---

**情報科は、面白いです!**

- 「試行錯誤」こそが、情報科の本質。「試行錯誤」を楽しみましょう。