

# 「難しいけど楽しい」を目指した プログラミング授業の実践

—Google Colaboratoryを用いたPython学習教材の作成—

沖縄県立陽明高等学校 情報科  
教諭 安次嶺 主紋(あしみね しゅもん)  
ashimnes@open.ed.jp

あしみね しゅもん  
安次嶺 主紋



- 沖縄県立高等学校情報科教員

- 教員年数：採用10年目  
指導科目：情報C(3年)  
：社会と情報(5年)  
専門校勤務(4年)

(昨年度) 沖縄県立総合教育センター  
1年長期研修

# 現所属高校の紹介

## 県立陽明高等学校

・総合学科(男女共学)

1学年6クラス約240名

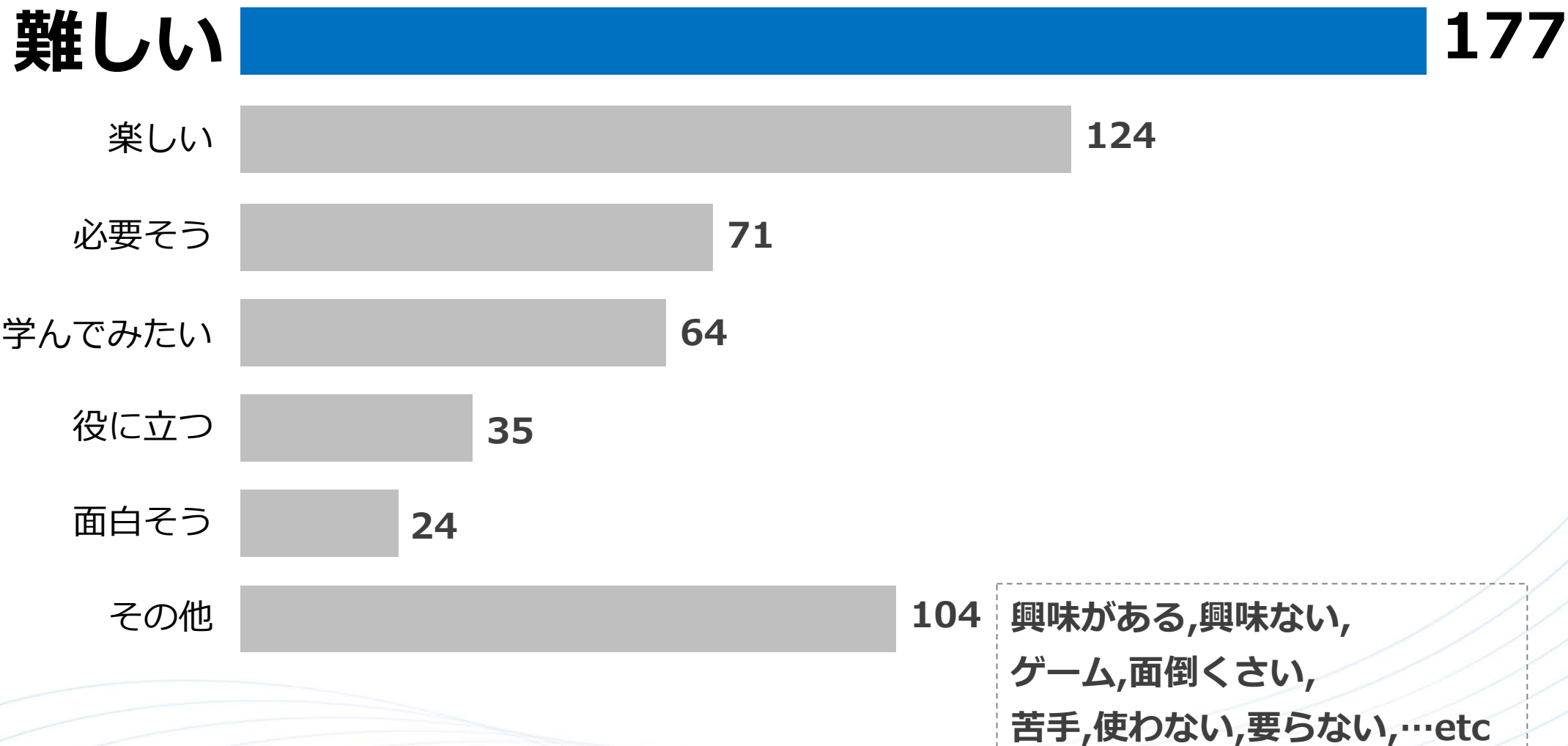
### <進路状況>



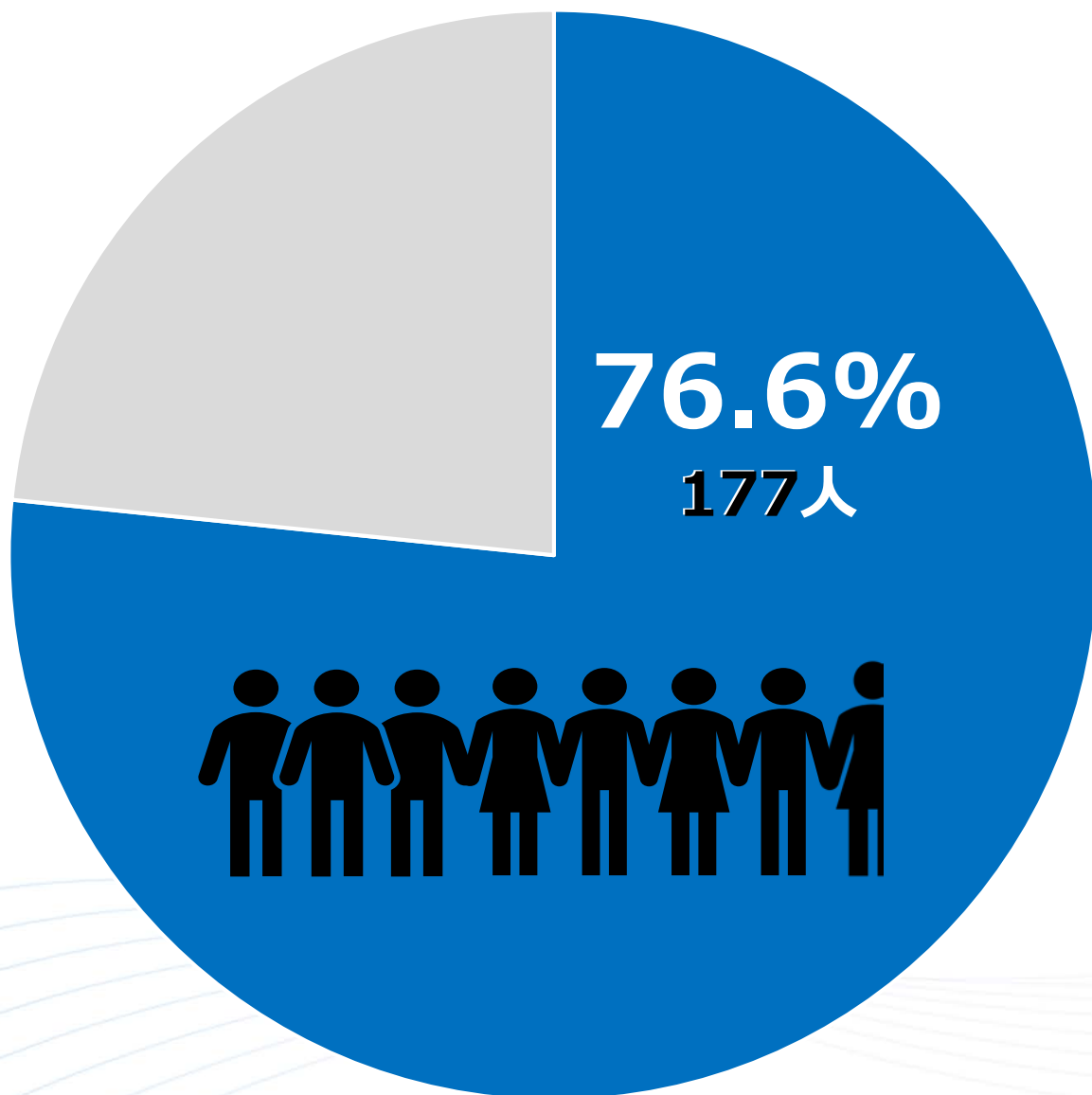
	R1	R2
大学	58名	70名
専門学校	104名	96名
就職	39名	29名
進路決定率	92.7%	87.1%

国公立大 3名  
県内私大 63名他

## プログラミング学習と聞いて連想する言葉(回答人数：231)



## 「難しい」と回答した生徒の割合 (回答人数：231)

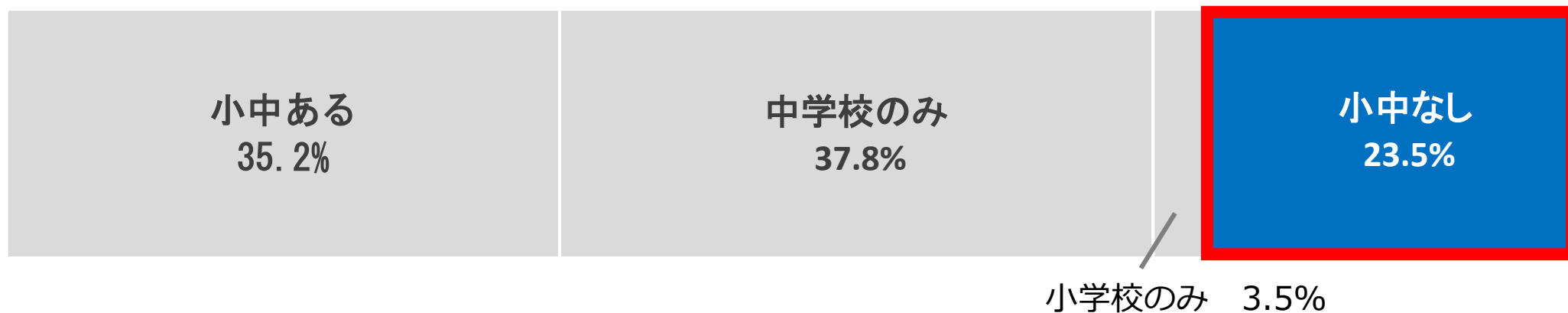


回答した生徒の

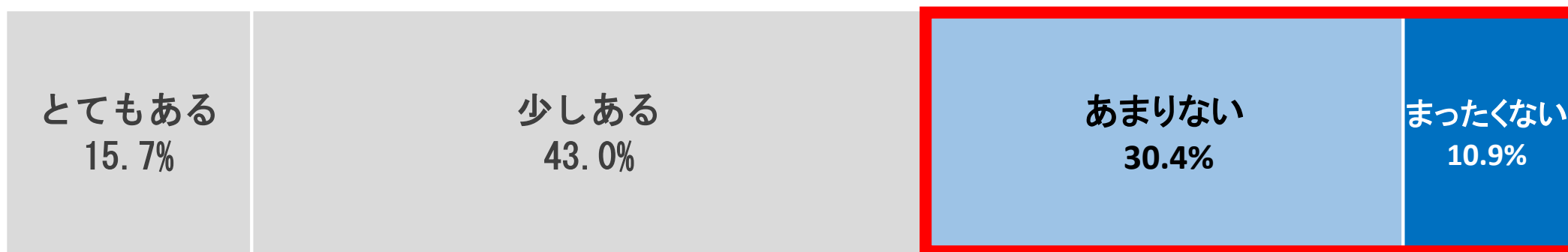
**約4分の3**

「難しい」印象を持っている

## ■ 高校入学以前のプログラミング学習経験



## ■ プログラミングに興味がありますか？



プログラミング学習の経験がない生徒が **23.5%**

「興味・関心が低い」生徒の割合が **約 40%**

「難しいけど楽しい」を目指したのは・・・



回避

学習への抵抗感

学習意欲の低下

初心者歓迎



プログラミングに**不慣れな生徒に**  
**配慮**した学習計画（**指導の工夫**）



とても悩みました・・・

テキストコーディングとか難しそう

プログラミングの授業どうしよう

アプリ制作の指導とか必要なの？





# そんな時、文科省教材の公表 . . .

高等学校情報科「情報Ⅰ」  
教員研修用教材

文部科学省  
MINISTRY OF EDUCATION,  
CULTURE, SPORTS,  
SCIENCE AND TECHNOLOGY

情報処理学会 IPSJ MOOC (登録不要、無料)

ホーム ご案内 学び方 教材 関係者

## IPSJ MOOC

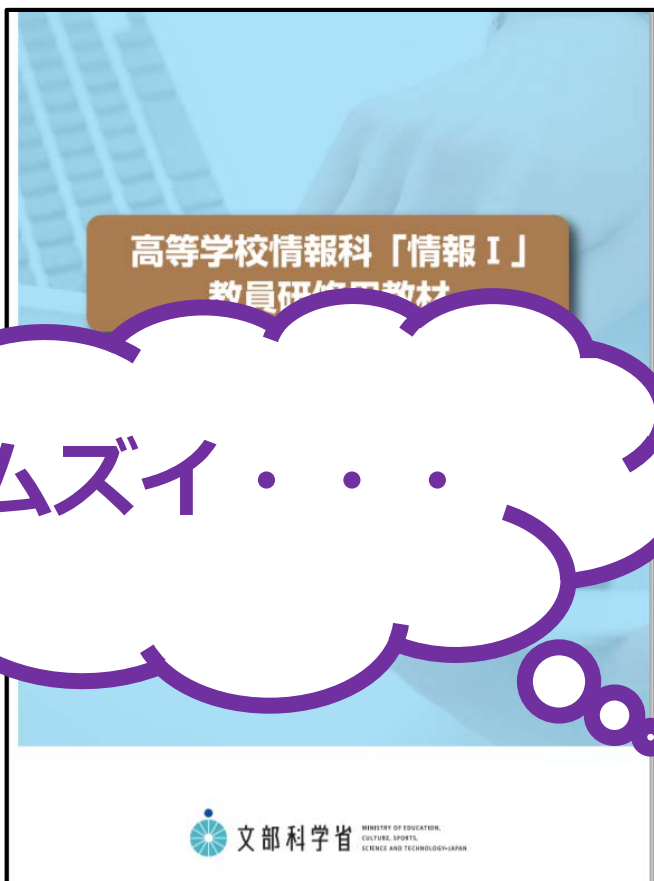
情報処理学会 公開教材

教員・生徒・学生の皆様、  
ぜひ、ご活用ください。

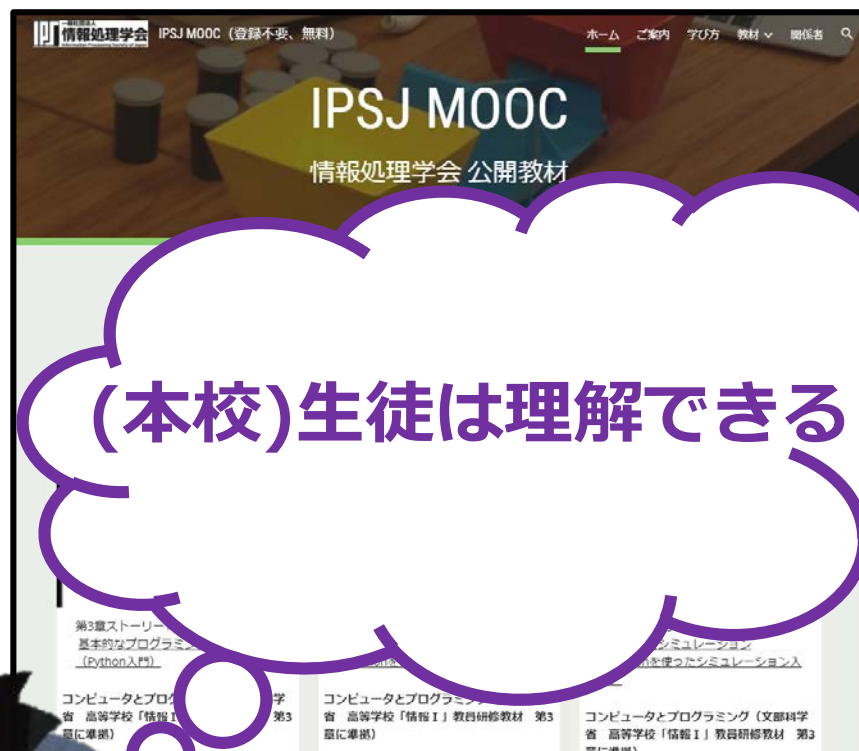
第3章ストーリー1	第3章ストーリー2	第3章ストーリー3
基本的なプログラミング (Python入門)	アルゴリズム (Pythonを使ったアルゴリズム入門)	モデル化とシミュレーション (Pythonを使ったシミュレーション入門)
コンピュータとプログラミング (文部科学省 高等学校「情報Ⅰ」教員研修教材 第3章に準拠)	コンピュータとプログラミング (文部科学省 高等学校「情報Ⅰ」教員研修教材 第3章に準拠)	コンピュータとプログラミング (文部科学省 高等学校「情報Ⅰ」教員研修教材 第3章に準拠)



# そんな時、文科省教材の公表 . . .



ムズイ . . .



(本校)生徒は理解できる？



## ゴールをどこに置くか・・・

- 「探求」に活用できるようになる
- ソースコードが読めるようになる
- 大学や専門学校へ繋がるプログラミング
- 「ソースコードを書けるようになったかも！」体験
- プログラミング的思考を育む
- プログラミングに興味をもってもらう

対象も大事



- 大学や専門学校へ繋がるプログラミング

基本制御構造や順次、繰り返し、分岐などができるといレベル

大学入試については、別枠で

- 「ソースコードを書けるようになったかも！」体験

「まずはやってみよう」の気持ちで・・・

作ってみた



# プログラミング言語の選択について



- Python  
文法が単純  
記述しやすい  
豊富なライブラリ
- JavaScript
- VBA
- ドリトル
- Swift



# Google Colabratoryを使用した理由

## ■プログラミング学習環境について

生徒が**学校外でも学習ができる**

①  
インストールが  
不要

②  
共有が  
容易

③  
マークダウンが  
利用可能



# 作った教材

## Google Colab による学習ファイル

授業時数: 10コマ  
環境: Chromeブラウザ

No.	学習項目
1	print関数・変数の使い方
2	if関数
3	リストについて
4	for関数①
5	for関数②(if関数との組み合わせ)
6	for関数③(リストとの組み合わせ)
7	while関数①
8	while関数②(if関数との組み合わせ)
9	random関数
10	数当てゲームを作ろう

1.print関数/変数



6.for関数③



2.if関数



7.while関数①



3.リスト



8.while関数②



4.for関数①



9.random関数



5.for関数②



10.数当てゲーム





# 作った教材

## Google Colab による学習ファイル

No.	学習項目
1	print関数・変数の使い方
2	if関数
3	リストについて
4	for関数①
5	for関数②(if関数との組み合わせ)
6	for関数③(リストとの組み合わせ)
7	while関数①
8	while関数②(if関数との組み合わせ)
9	random関数
10	数当てゲームを作ろう

- ・偶数or奇数
- ・3ケタの数当て
- ・ヒット&ブロー

```
+ コード + テキスト 接続
```

### 数当てゲーム

①プログラムの手順

【ステップ1】 random関数を用いて、1~100までのランダムな数をコンピュータに選ばせる。  
※random関数のimportも忘れないようにする。

```
# 1~100までの整数をランダムに選んで、指定した変数(今回はpcという名前の変数)に代入する命令  
pc = random.randint(1, 100)
```

【ステップ2】 print文を用いて、『1~100までの数を選んだよ』と表示する。  
【ステップ3】 input関数を用いて、『数を入力して下さい:』と表示し、入力を求める。入力された変数(yosouという名前の変数)に代入する。  
【ステップ4】 if文を用いて、変数pcの値と変数yosouの値を比較する。

## ■ スモールステップ学習

**ステップ 1 : 誤りを見つけ、正しく実行する**

**ステップ 2 : 指定された条件に従い変更を加え、  
正しく実行する**

**ステップ 3 : 指定された条件に従い、プログラム  
コードを記述し、正しく実行する**

## ステップ1： 誤りを見つけ、正しく実行する

練習① 下のプログラムはこのままではエラー表示になります。  
誤っている個所を見つけて、正しく実行できるように修正してください。

```
▶ 1 number = 1
   2
   3 if number == 1
   4 | print("好き")
   5 else:
   6 | print("嫌い")
```

「:」の入力漏れによるエラーに気づけるか？

## ステップ2：指定された条件に従い変更を加え、正しく実行する


練習① 下のプログラムの条件の式を、「scoreが65点以上なら」に修正してください。

<期待する出力表示>

合格です

**if関数の条件部分についての修正を問う課題。  
修正後、<期待する出力表示>どおりに表示されるか確認する。**

```
[ ] 1 score = 75
     2
     3 if score >= 80:
     4 |     print("合格です")
     5 else:
     6 |     print("不合格です")
```



## ステップ3：指定された条件に従い、プログラムコードを記述し、正しく実行する

問題6 点数によって、成績を表示するプログラムを変数名testを用いて作成してください。  
if文を追加して、次のように表示するようなプログラムを完成させてください。

testの値が80以上のとき、「成績は5です」と表示する

testの値が65以上のとき、「成績は4です」と表示する

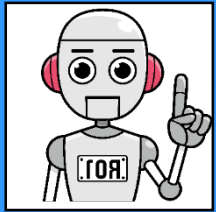
testの値が50以上のとき、「成績は3です」と表示する

testの値が35以上のとき、「成績は2です」と表示する

その他のとき、「単位保留です」と表示する

**問題文に沿ったプログラム  
コードを記述できるかを確認  
する**

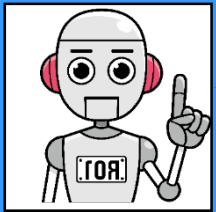
# 学習の見通し（文字の表示編）



ロボット

名前を入力してください

沖縄太郎

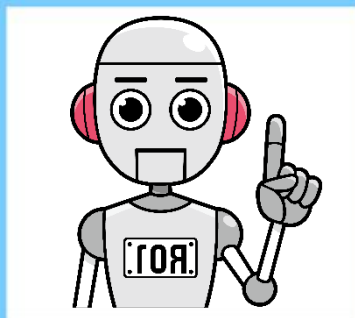


ロボット

沖縄太郎さんですね

```
1 name = input( “名前を入力してください” )  
2  
3 print ( name + “さんですね” )
```

## 会話ロボットをつくろう♪



ロボット

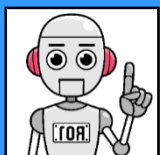
こんにちは

Hello

- input関数
- if関数

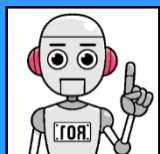
```
1 word = input():  
2  
3 if word == ("こんにちは"):  
4     print(" Hello ")  
5 else:  
6     print(" 別のあいさつを入力してください")
```

# 学習の見通し（くり返し、配列）



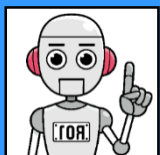
ロボット

Hello



ロボット

Good Morning



ロボット

こんにちは

こんにちは

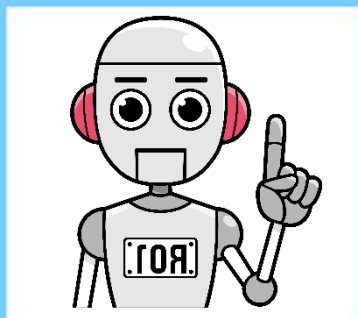
おはよう

- input関数
- if関数/for関数

```
1 import random
2 dic = ["Hello", "Good Morning", "Hi!"]
3
4 for i in range(5):
5     msg = input(random.choice(dic)+">")
6     if msg not in dic:
7         dic.append(msg)
```



## WebAPIを使って、天気予報を確認♪



ロボット

明日の天気は？

くもり

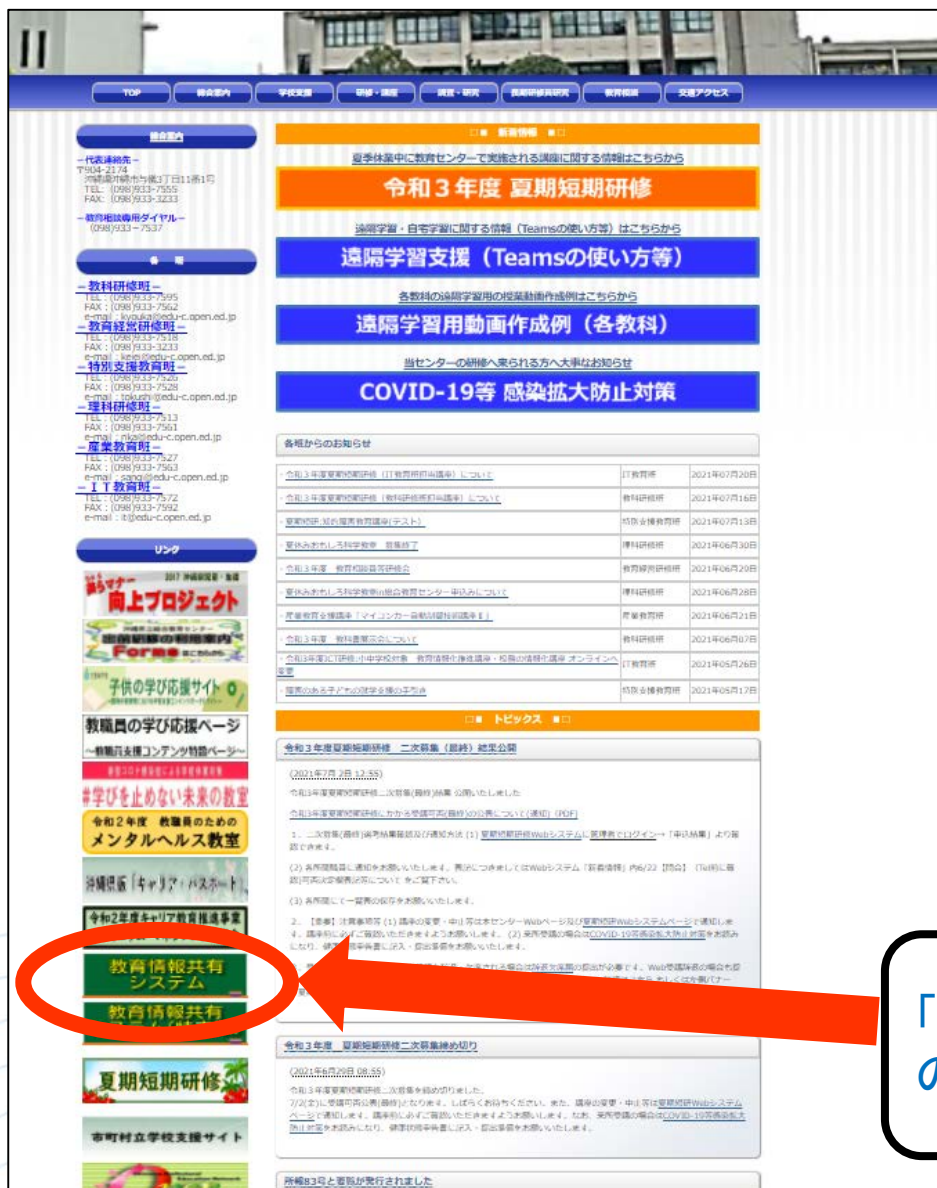
- input関数

- requestsライブラリ

```
1 import requests
2 url = "/api/weather"
3
4 test = input();
5
6 params = {
7     'code' : 4410,
8     'date' : "tomorrow"
9 }
10
11 response = requests.get(url, params)
12 print(response["description"])
```

# 作成した教材のURL

## 沖縄県立総合教育センター 情報共有システム



Pythonによるプログラミング授業  
安次嶺主紋

「沖縄県立総合教育センター」と検索頂き、Topページの左列の真ん中より下にあります。

# マークダウン記法が良かった点

## Google Colaboratryのテキストの入力 (マークダウンの利用)

The screenshot shows the Google Colaboratory interface. At the top, there's a title bar with 'Untitled0.ipynb' and a star icon. Below it, there are menu items: 'ファイル', '編集', '表示', '挿入', 'ランタイム', 'ツール', 'ヘルプ', and 'すべての変...'. On the right, there are icons for 'コメント', '共有', '設定', and a user profile. Below the menu, there are '+ コード' and '+ テキスト' buttons. On the right side, there are indicators for 'RAM' and 'ディスク' usage, and an '編集' button. The main content area shows a text input field with a table of comparison operators and their meanings. The table is rendered in a code block with markdown syntax. The table has two columns: '比較演算子' and '意味'. The rows are: 'a == b | aはbと等しいとき', 'a > b | aはbよりも大きいとき', 'a < b | aはbよりも小さいとき', 'a >= b | aはb以上のとき', 'a <= b | aはb以下のとき', and 'a != b | aはbと等しくないとき'. A yellow box highlights the first three rows, and a green box highlights the last three rows. A yellow arrow points from the first row of the yellow box to the first row of the green box, and a green arrow points from the last row of the green box to the last row of the yellow box.

比較演算子	意味
a == b	aはbと等しいとき
a > b	aはbよりも大きいとき
a < b	aはbよりも小さいとき
a >= b	aはb以上のとき
a <= b	aはb以下のとき
a != b	aはbと等しくないとき

# マークダウン記法が良かった点

## 教師によるカスタマイズ

+ コード + テキスト

▼ for関数の例題③ **タイトル**

・指定された範囲のくり返し

`for x in range(値①, 値②):`  
値①から値②-1まで1刻み (値①, 値①+1, ..., 値②-1)  
(例: `range(2,5): 2,3,4`)

**説明**

**同じ処理をくり返す(ループ) for編**  
for関数を使った「1」～「5」までを表示するプログラムをみてみましょう!

**図や表の挿入**

```
1 for x in range(1, 6):
2   print(x)
```

① 変数 x に1が代入される  
↓  
print(x)で、xの値である1が表示される  
② 変数 x に2が代入される  
↓  
print(x)で、xの値である2が表示される  
⋮  
⑤ 変数 x に5が代入される  
↓  
print(x)で、xの値である5が表示される

出力結果

```
1
2
3
4
5
```

for x in range(a, b):  
とある場合、  
xには、aから始まり、  
(b - a)回の要素が代入される。

比較演算子	意味
<code>a == b</code>	aはbと等しいとき
<code>a &gt; b</code>	aはbよりも大きいとき
<code>a &lt; b</code>	aはbよりも小さいとき
<code>a &gt;= b</code>	aはb以上のとき
<code>a &lt;= b</code>	aはb以下のとき
<code>a != b</code>	aはbと等しくないとき

練習(9) 下のプログラムコードに修正を加えて、「1から10までを表示する」プログラムコードを完成させてください。

```
[ ] 1 #数値を1から10まで表示するプログラム
2
3 for x in range(): #ここの部分を修正しましょう。
4     print(x)
```

**演習問題文 + コード入力欄**

# 学習指導の工夫(授業展開)

導入

## 前時の復習 (Quizizz)

```
1 year = 3
2
3 if year == 1:
4     print("1年生です")
5 else if year == 2:
6     print("2年生です")
7 else if year == 3:
8     print("3年生です")
9 else:
10    print("卒業生です")
```

このプログラムを実行したときに出力表示されるのは、次のうちどれですか？

エラーの表示      卒業生です      3年生です      1年生です

展開

## 本時の文法の説明 (Google Colabratory)

同じ処理をくり返すループ for 文

練習(9) 下のプログラムコードに修正を加えて、「1から10までを表示する」プログラムコードを完成させてください。

```
1 #数値を1から10まで表示するプログラム
2
3 for x in range(): #この部分を修正しましょう。
4     print(x)
```

ふりがえり

## 本時のおさらい (Quizizz/Google Colab)

```
1 test = 0
2
3 if test == 10:
4     print("真赤はすごい!")
5 else:
6     print("次は満点を目標そう!")
```

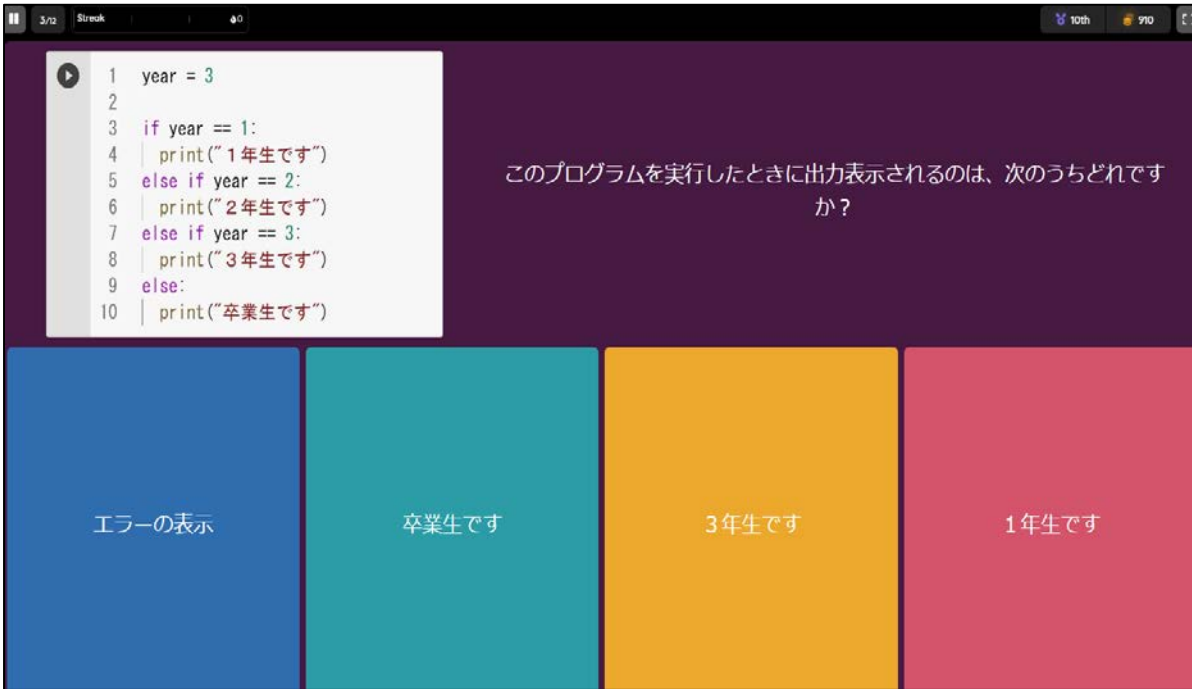
問題1 下のプログラムは、このままではエラー表示になります。 誤っている箇所を見つけて、正しく実行できるように修正してください。

<正しい出力表示>  
次は満点を目標そう!

問題2 下のプログラムは、このままではエラー表示になります。 誤っている箇所を見つけて、正しく実行できるように修正してください。

# クイズアプリについて

## 学習の振り返り



The screenshot shows a quiz application interface. On the left, there is a code editor with the following Python code:

```
1 year = 3
2
3 if year == 1:
4     print("1年生です")
5 else if year == 2:
6     print("2年生です")
7 else if year == 3:
8     print("3年生です")
9 else:
10    print("卒業生です")
```

On the right, there is a question in Japanese: "このプログラムを実行したときに出力表示されるのは、次のうちどれですか?" (Which of the following will be output when this program is executed?). Below the question are four colored buttons representing possible answers: "エラーの表示" (Error display), "卒業生です" (Graduate), "3年生です" (3rd year student), and "1年生です" (1st year student).

- ・一人でも学習できる
- ・いろいろな問題が作れる
- ・無料
- ・教師側が正解率を即時に把握できる  
→振り返る内容を焦点化できる

ゲーム性が高いので楽しい



# プログラミング学習らしいテスト



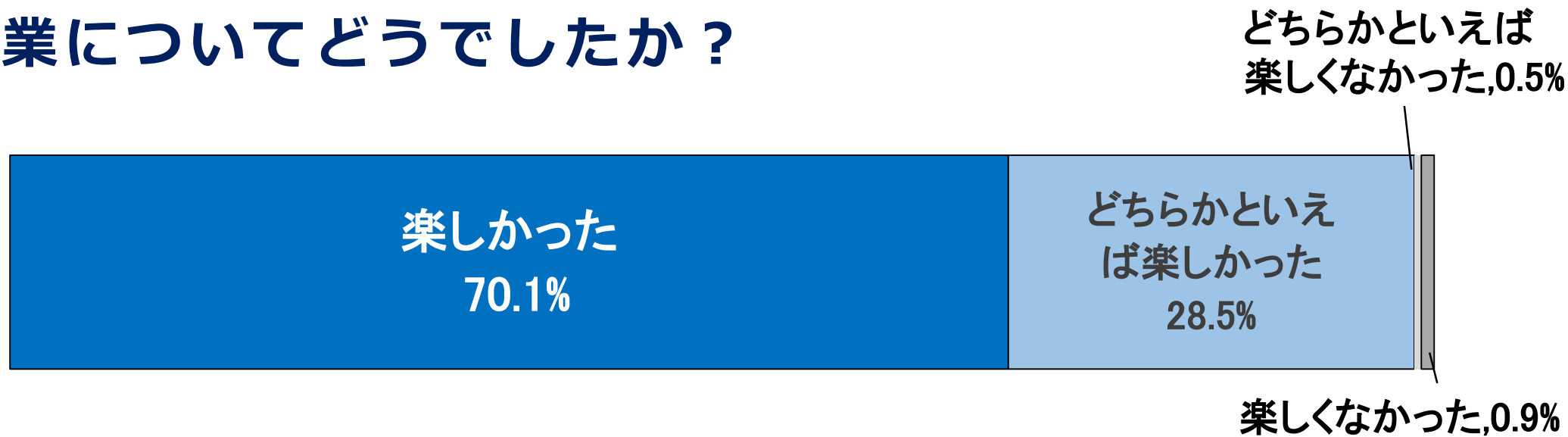
Google Colabratry



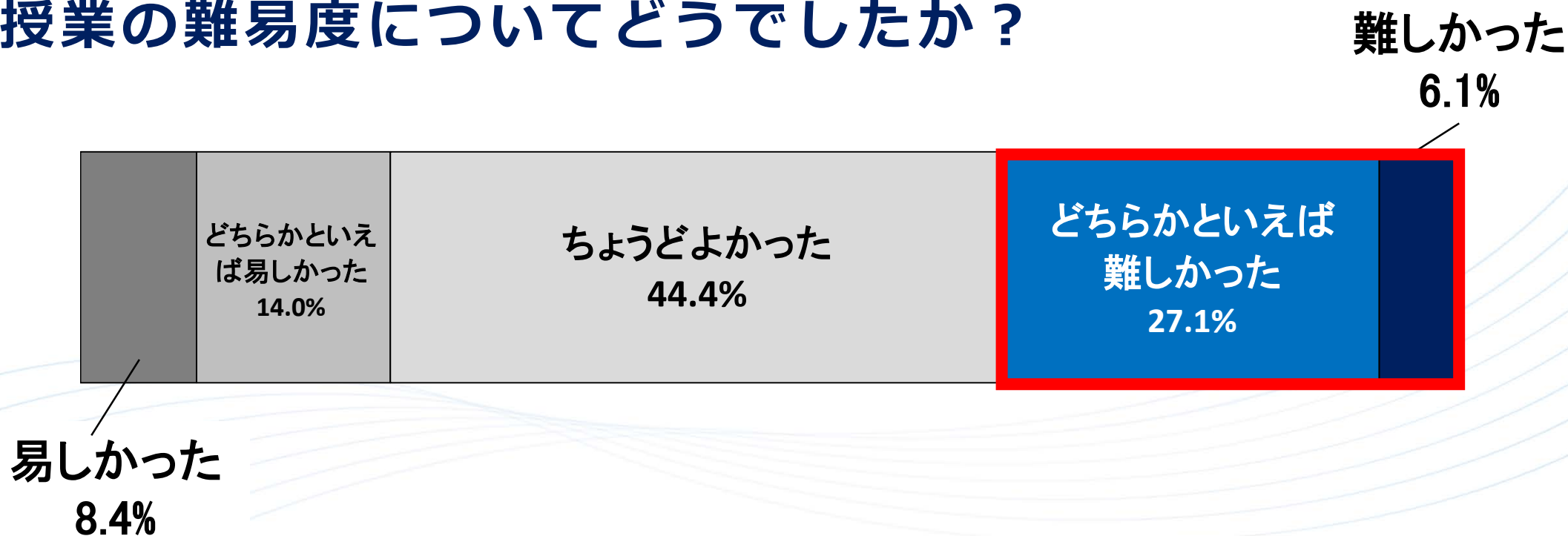
Google Classroom

# 検証授業後アンケート

## 【授業についてどうでしたか？】



## 【授業の難易度についてどうでしたか？】





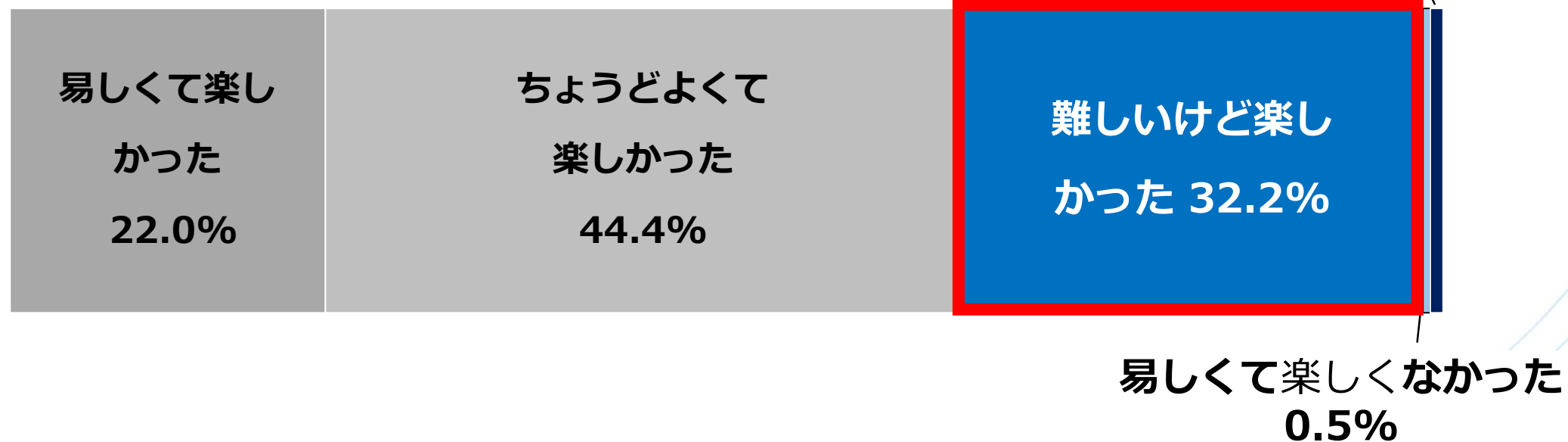
# 検証授業後アンケート

## 授業についてのクロス集計

「授業についてどうでしたか？」

×

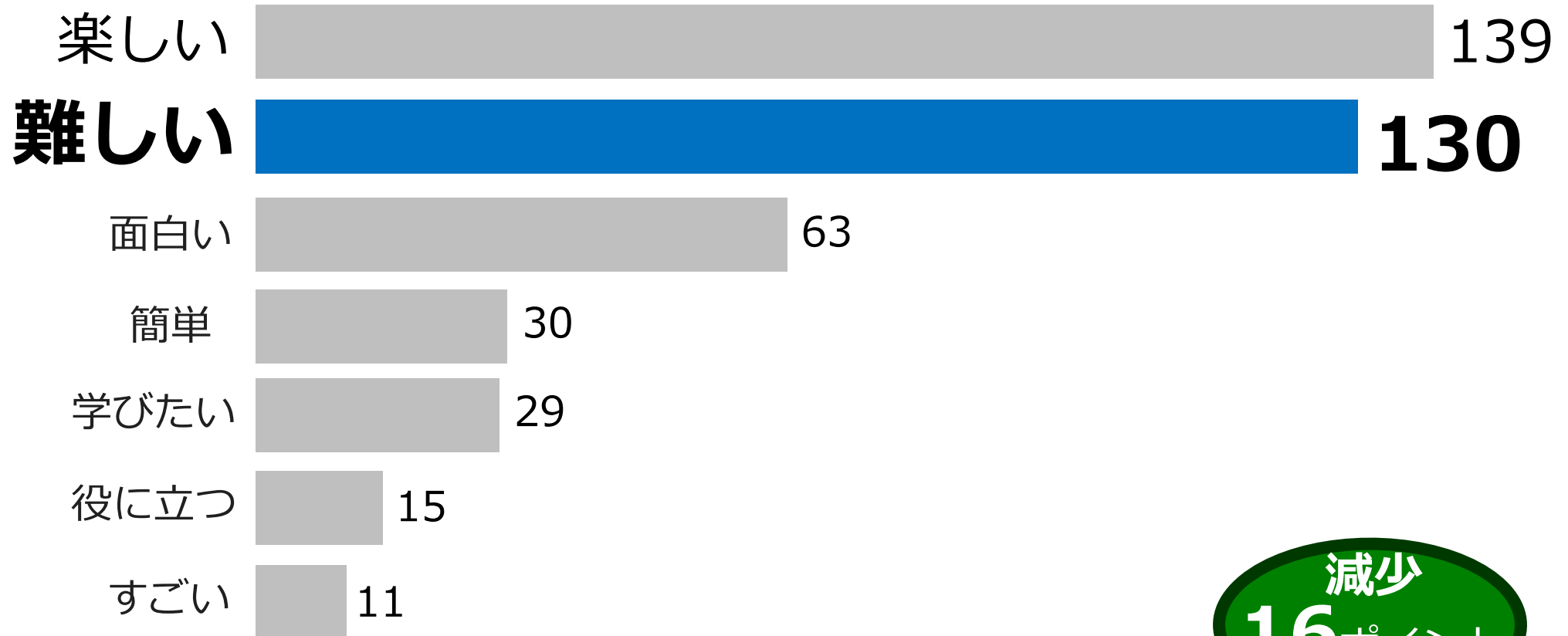
「授業の難易度についてどうでしたか？」



生徒は、**難しい課題にも前向きに取り組んだ**

# 検証授業後アンケート

## プログラミング学習と聞いて連想する言葉（検証授業後）



	授業前	授業後
「難しい」と連想する生徒	76.6%	60.7%

減少  
16ポイント

# 検証授業後アンケート

## ■プログラミング学習と聞いて連想する言葉（検証授業後）

楽しい



139

難しい



130

プログラミング学習については、  
**依然として「難しい」イメージ**はあるが、  
**学習には前向き**に捉えている

- **ビジュアルプログラミングによる事前学習の必要性**
- **Google Colabのテキストセルに保護機能がない**
- **オンライン授業では、難しい**
- **大学入試に向けては、要検討**

質問やご助言、ご指導よろしくお願いたします



あしみね しゅもん  
安次嶺 主紋

- 沖縄県立陽明高等学校
- [ashimnes@open.ed.jp](mailto:ashimnes@open.ed.jp)