情報Ⅰにおける授業改善
～コンピュータとプログラミング

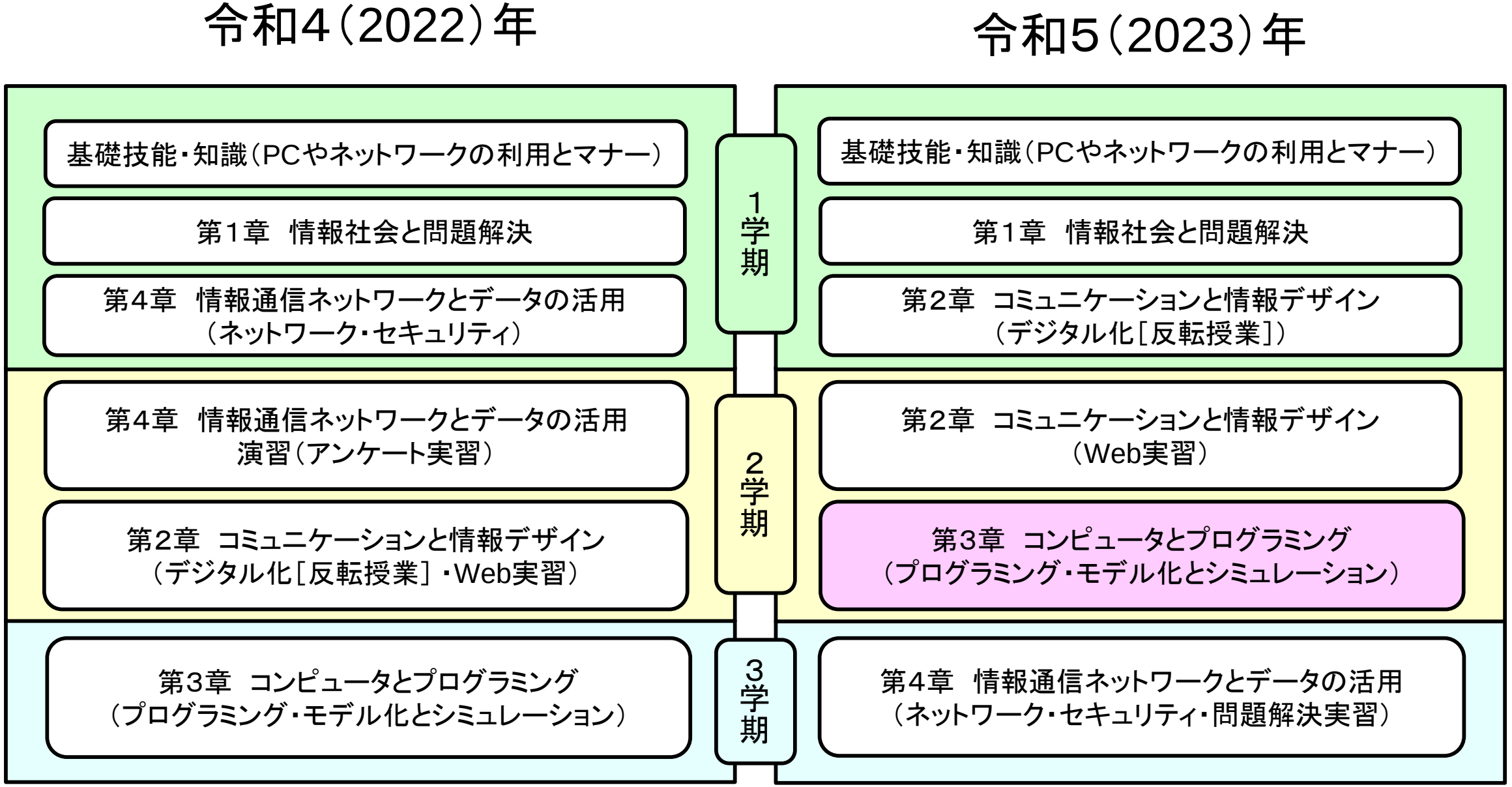
東京都立町田高等学校
指導教諭　小原 格

本日の内容

- 今年度の課題と改善
  - 年間を通した視点
  - 単元の視点
  - 授業方法や教具の視点
- まとめ

【再掲】年間を通した視点

- 年間指導計画を再考する
  - 単元の順番
  - 「大きな実習」の順番・整理
  - 単元の時間数調整
  - 関連内容の整理


令和4（2022）年
令和5（2023）年
基礎技能・知識（PCやネットワークの利用とマナー）
第1章　情報社会と問題解決
第4章　情報通信ネットワークとデータの活用
（ネットワーク・セキュリティ）
1学期
基礎技能・知識（PCやネットワークの利用とマナー）
第1章　情報社会と問題解決
第2章　コミュニケーションと情報デザイン
（デジタル化［反転授業］）
第4章　情報通信ネットワークとデータの活用
演習（アンケート実習）
第2章　コミュニケーションと情報デザイン
（デジタル化［反転授業］・Web実習）
2学期
第2章　コミュニケーションと情報デザイン
（Web実習）
第3章　コンピュータとプログラミング
（プログラミング・モデル化とシミュレーション）
第3章　コンピュータとプログラミング
（プログラミング・モデル化とシミュレーション）
3学期
第4章　情報通信ネットワークとデータの活用
（ネットワーク・セキュリティ・問題解決実習）

【再掲】実習内容と時間の再検討

令和4（2022）年

コンピュータのしくみ（論理回路）
コンピュータのしくみ（演算と計算の限界）
コンピュータと情報処理（アルゴリズムの基礎）
アルゴリズムとプログラム（アルゴリズムの表現方法）
アルゴリズムとプログラム（言語による表現：変数と演算）
アルゴリズムとプログラム（言語による表現：配列と関数）
アルゴリズムとプログラム（言語による表現：制御構造）
アルゴリズムとプログラム（課題制作1：問題の発見）
アルゴリズムとプログラム（課題制作2：試作と実装）
アルゴリズムとプログラム（課題制作3：評価改善）
モデル化とシミュレーション（モデルの種類・状態遷移図）
モデル化とシミュレーション（確定的な数式モデル）
モデル化とシミュレーション（確率的な数式モデル）
モデル化とシミュレーション（課題制作1：問題の発見）
モデル化とシミュレーション（課題制作2：試作と実装）
モデル化とシミュレーション（課題制作3：評価改善）

令和5（2023）年

コンピュータとプログラミング（論理回路）
コンピュータとプログラミング（四則演算）
コンピュータとプログラミング（計算の限界）
コンピュータとプログラミング（アルゴリズムの表現方法）
コンピュータとプログラミング（言語による表現：変数と演算）
コンピュータとプログラミング（言語による表現：配列と関数）
コンピュータとプログラミング（言語による表現：制御構造1）
コンピュータとプログラミング（言語による表現：制御構造2）
モデル化とシミュレーション（状態遷移図）
モデル化とシミュレーション（確定的な数式モデル）
モデル化とシミュレーション（確率的な数式モデル）
プログラミングとシミュレーション（課題制作1：問題の発見）
プログラミングとシミュレーション（課題制作2：試作と実装）
プログラミングとシミュレーション（課題制作3：試作と実装）
プログラミングとシミュレーション（課題制作4：評価改善）

【再掲】進め方の再検討

コンピュータとプログラミング

・プログラミング言語（Python）の基本的な技術

- 令和4年度まで
  - ・教員がデモンストレーションをしながら一斉に
- 令和5年度から
  - ・外部のサービス・チュートリアル等を用いて、個別にドリル形式で

【再掲】実習内容と時間の再検討

令和4（2022）年

コンピュータのしくみ（論理回路）
コンピュータのしくみ（演算と計算の限界）
コンピュータと情報処理（アルゴリズムの基礎）
アルゴリズムとプログラム（アルゴリズムの表現方法）
アルゴリズムとプログラム（言語による表現：変数と演算）
アルゴリズムとプログラム（言語による表現：配列と関数）
アルゴリズムとプログラム（言語による表現：制御構造）
アルゴリズムとプログラム（課題制作1：問題の発見）
アルゴリズムとプログラム（課題制作2：試作と実装）
アルゴリズムとプログラム（課題制作3：評価改善）
モデル化とシミュレーション（モデルの種類・状態遷移図）
モデル化とシミュレーション（確定的な数式モデル）
モデル化とシミュレーション（確率的な数式モデル）
モデル化とシミュレーション（課題制作1：問題の発見）
モデル化とシミュレーション（課題制作2：試作と実装）
モデル化とシミュレーション（課題制作3：評価改善）

オンライン
プログラミングツール
を活用
（「状態遷移図」は除く）

令和5（2023）年

コンピュータとプログラミング（論理回路）
コンピュータとプログラミング（四則演算）
コンピュータとプログラミング（計算の限界）
コンピュータとプログラミング（アルゴリズムの表現方法）
コンピュータとプログラミング（言語による表現：変数と演算）
コンピュータとプログラミング（言語による表現：配列と関数）
コンピュータとプログラミング（言語による表現：制御構造1）
コンピュータとプログラミング（言語による表現：制御構造2）
モデル化とシミュレーション（状態遷移図）
モデル化とシミュレーション（確定的な数式モデル）
モデル化とシミュレーション（確率的な数式モデル）
プログラミングとシミュレーション（課題制作1：問題の発見）
プログラミングとシミュレーション（課題制作2：試作と実装）
プログラミングとシミュレーション（課題制作3：試作と実装）
プログラミングとシミュレーション（課題制作4：評価改善）

昨年度まで：自作教材を活用

アルゴリズムとプログラム：Python（基本編その4）

実習4　テキストエディタを使って簡単なプログラムを作ってみよう（判断分岐）

例8）分岐1：if ～ else ～

```
x = int(input('x='))
if x >= 0:
    result = x
else:
    result = -x
print(result)
```

解説：
「x=」と表示して手操作入力したものを数値に変換して x に代入
「以上」は「>=」（不等号を前にすること）
さらにインデント（4文字）を忘れずに
成り立たない場合の処理の指定

result を表示

例9）分岐2：変数を少なくシンプルに

```
x = int(input('x='))
if x < 0:
    x = -x
print(x)
```

解説：
「x=」と表示して手操作入力したものを数値に変換して x に代入
x が負となる時のみ、
x の符号を入れ替えたものを x に再代入
x を表示

練習4）　次のプログラムをつくりなさい。なお、変数は指定されたものを利用すること。
① 2つの数 a, b を手操作入力すると、大きい方の数を表示するプログラム max2.py
② 4つの数 a, b, c, d を手操作入力すると、最大値を表示するプログラム max4.py

例10）分岐3：elif

```
x = int(input('x='))
if x >= 10:
    print('Excellent!')
elif x >=5:
    print('Good!')
else:
    print('Try Again!')
```

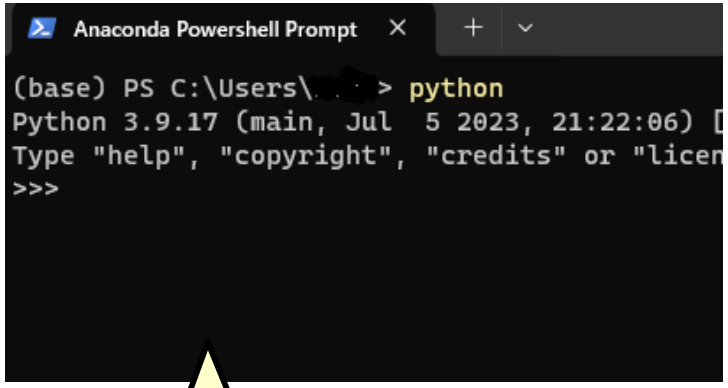
解説：
「x=」と表示して手操作入力したものを数値に変換して x に代入
「以上」は「>=」（不等号を前にすること）
さらにインデント（4文字）を忘れずに
「elif」は、「条件に当てはまらない場合」に「さらに条件を加える場合」を決める

この場合の「else」は、「全ての条件に当てはまらない場合」

練習5）　次のプログラムをつくりなさい。なお、変数は指定されたものを利用すること。
① 年齢（age）を手操作入力し、その数値によって画面へ出力する価格を以下のように変えるプログラム kakaku.py
【注意！】価格の「1,200yen」などは、「文字列」として表示させること！
60歳以上：　1,200yen
18歳以上60歳未満：　2,000yen
13歳以上18歳未満：　1,500yen
13歳未満：　1,000yen

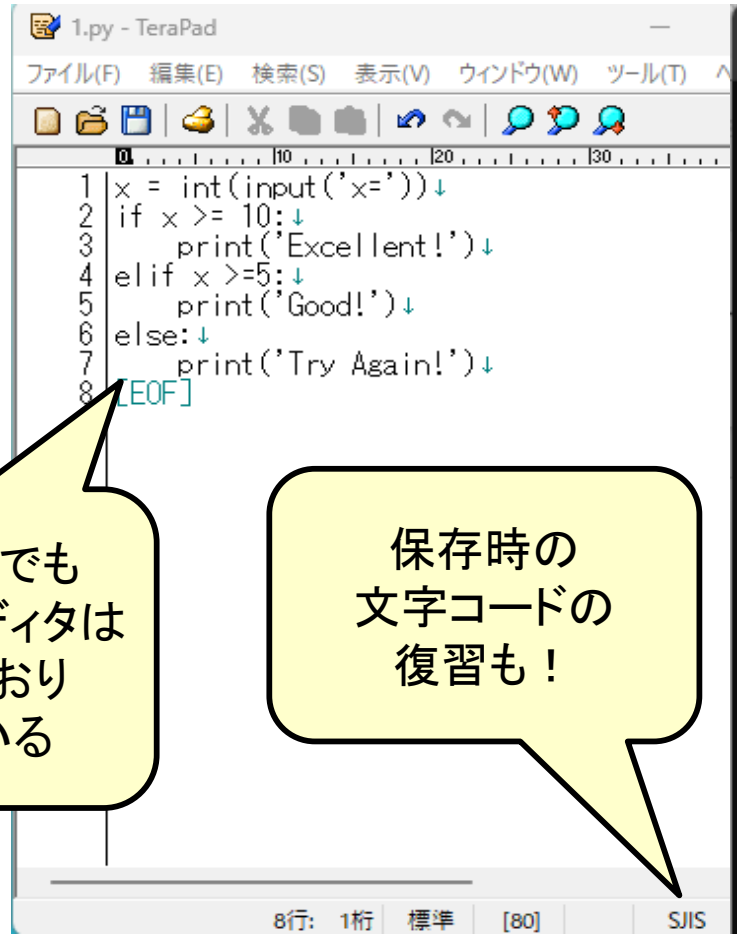
昨年度まで

まずはインタープリターで

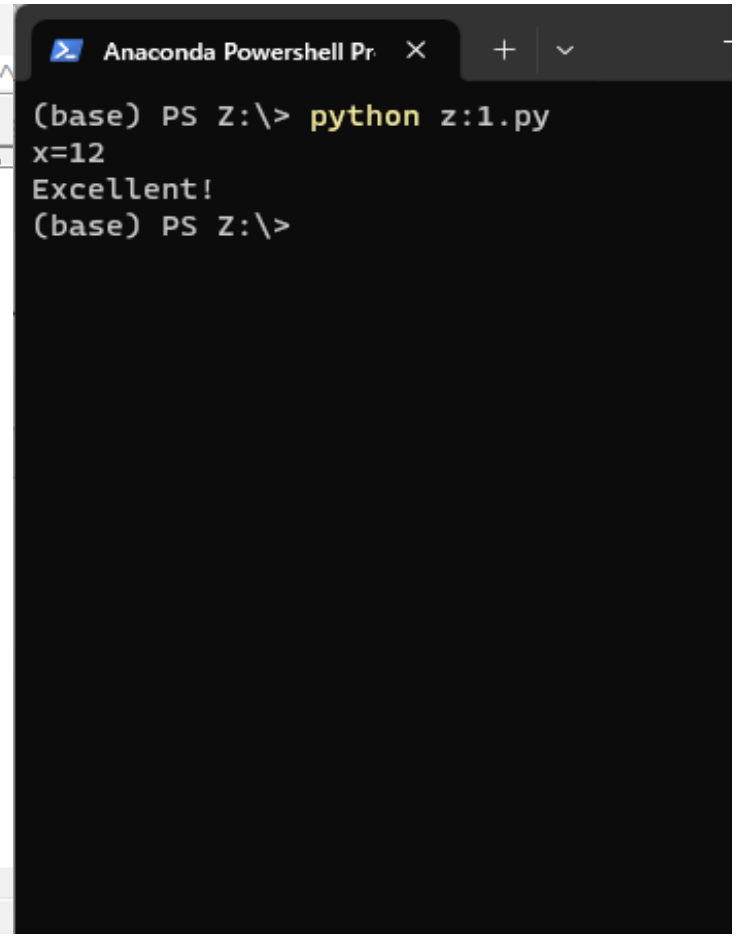


情報デザインでの
CUI,GUIの指導にも
関連付けて

ある程度進むと、テキストエディタで保存し別ファイルを実行



Web制作でも
テキストエディタは
利用しており
慣れている



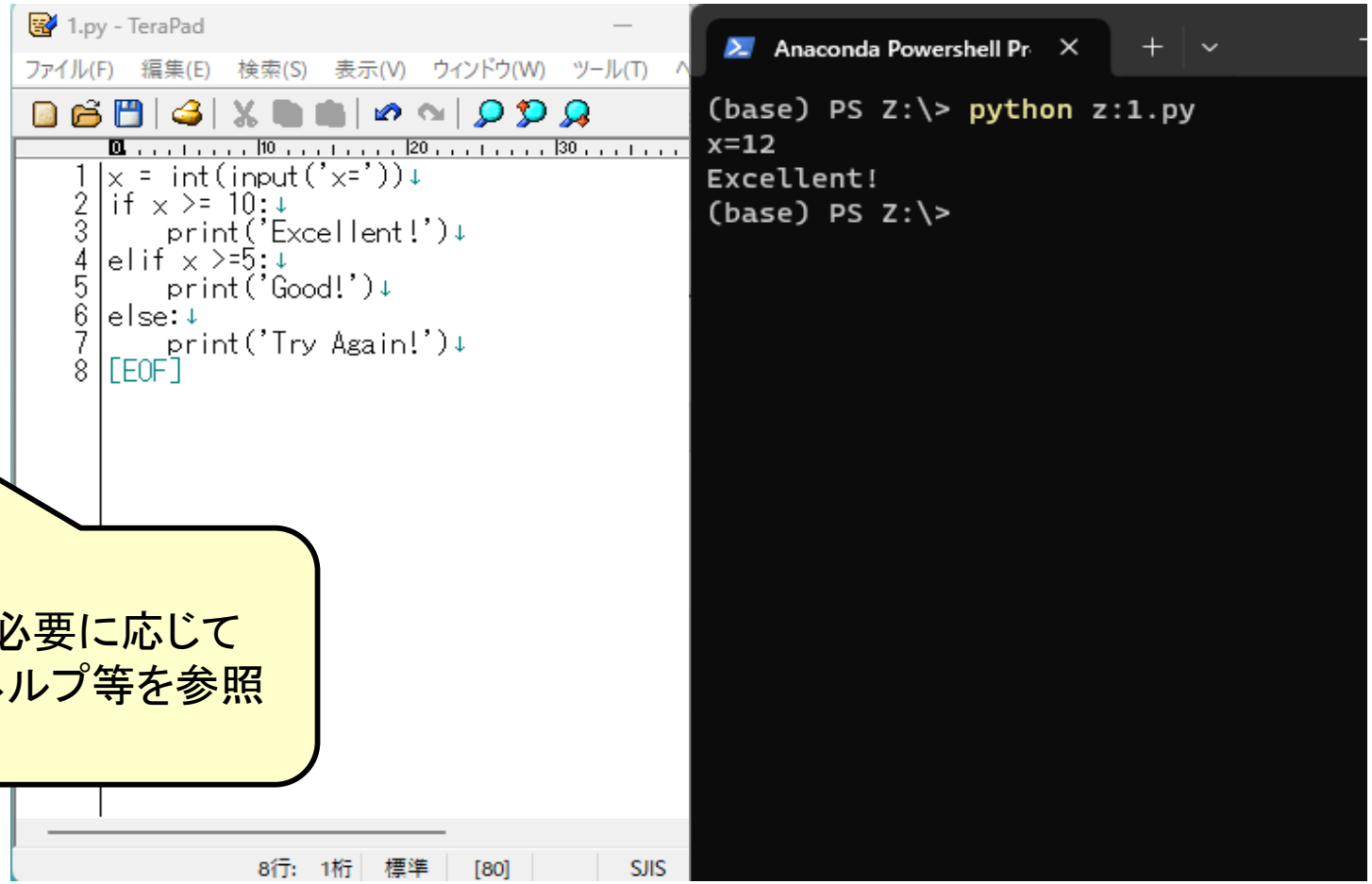
保存時の
文字コードの
復習も！

本年度は・・・

まずはオンラインサービスで演習

ブラウザベースなので
一人1台端末を活用し
好きなタイミングで
予習・復習を

課題制作は、テキストエディタで保存し別ファイルを実行



必要に応じて
ヘルプ等を参照

オンラインサービスに求めたこと

- PythonがWebベースで生徒自身のタイミングで練習可能
- わからない生徒向けにヒントやフォローアップが充実
- 無料で活用できる
- ある程度の自由度があり、自作の教材と併用できる
- 個人情報に配慮され、取り組み状況を教員が把握できる
- 演習問題が充実
- スモールステップと適度な難易度である

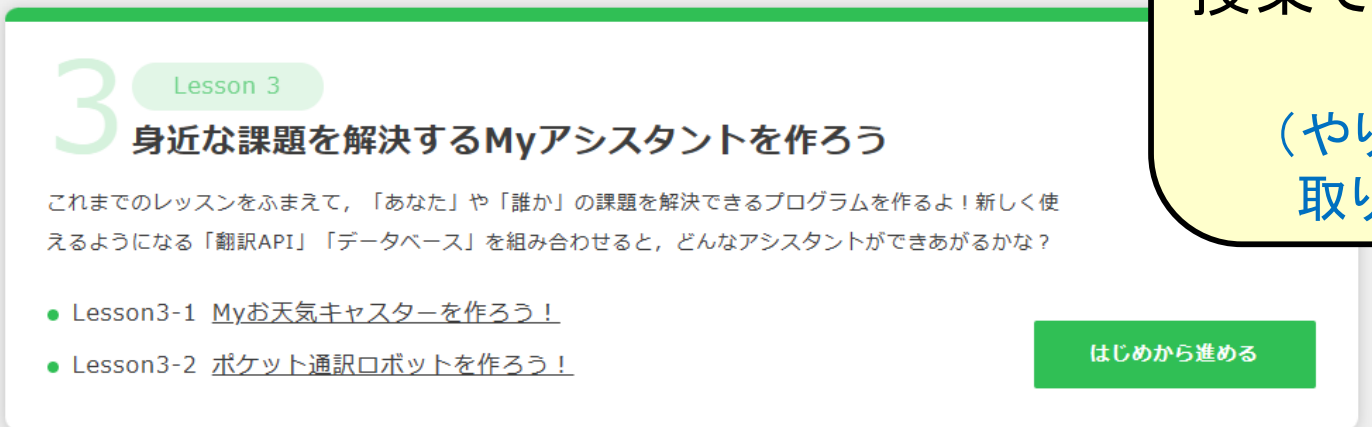
「プログル情報」

- 特定非営利法人みんなのコード
- https://high.proguru.jp/

順次・反復・判断の
基本部分

順番に解説しながら
演習形式で実施

分かる生徒は
どんどん自分で進め、
他の生徒に
教えてあげることを
推奨



応用的な部分

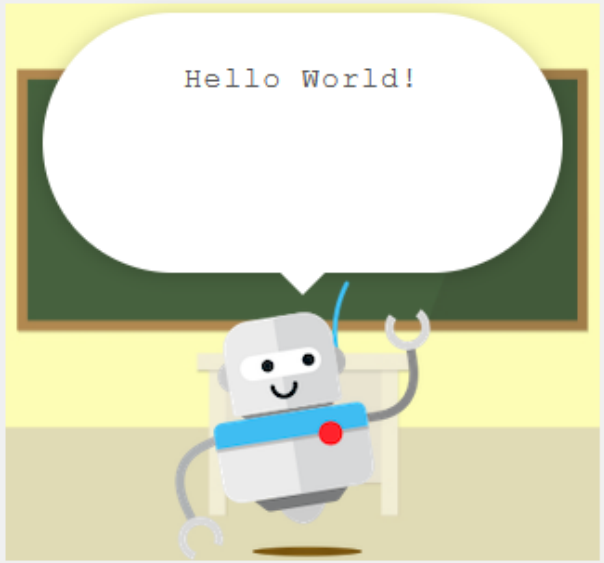
本年度は
授業では実施しない

（やりたい生徒は
取り組む程度）

ステージ1

'Hello World!' という文字列を画面に表示しよう。

ヒントを見る　確認する



実行時プログラム

print()

'Hello World!'

```
1  print('Hello World!')
```

リセット　ステージを最初からやり直す

校　情報科

基本を一通り行いつつ課題を

11月末までの課題として配信、教えあい自由。(答え丸写しはダメ!)
→ 自由課題対策・テスト対策も兼ねて

自由に Python のプログラミングをしよう

リファレンスを見る　保存する

実行結果

1行目でエラーが発生しています

SyntaxError: invalid syntax

プログラム1

```
1  inport ramdom
```

「ヒント機能」の活用

→　後述

AI にヒントを聞く

リセット　最初からやり直す


自由に Python のプログラミングをしよう
リファレンスを見る
保存する
1行目でエラーが発生しています
SyntaxError: invalid syntax
このエラーは「構文エラー」と呼ばれ、プログラムの文法が正しくない場合に発生します。
このプログラムでは、 inport というスペルミスがあるため、構文エラーが発生しています。正しくは import とスペルする必要があります。
このエラーを修正するには以下を確認してみてください。
import のスペルが間違っていないか確認しましょう。
Pythonでは、標準ライブラリや外部ライブラリを使うときは import を使います。 import について調べてみましょう。
※ この文章は OpenAI によって生成されたものです
この情報は役に立ちましたか？
役に立った
役に立たなかった
最初からやり直す

情報科

モデル化とシミュレーション（教科書P150～）

人口予測のシミュレーション

（1）2000年から2001年までの増加割合がずっと続いたと仮定すると？

（2）2008年から2009年までの割合がずっと続いたと仮定すると？

翌年の人口 ＝ 今年の人口 × 増加率(ratio)

※数値を超えない最も大きい整数にする関数： int(数値) → 正の数の場合は「切り捨て」
例:C15の正の数値を整数にする → =int(C15)

		増加率	
年	人数(千人)		
2000	126926		
2001	127316		
2002			
2003			
2004			
2005			
2006			
2007			
2008			
2009			
2010			
2011			
2012			
2013			
2014			
2015			
2016			
2017			
2018			
2019			
2020			
2021			
2022			

1．増加率を求める。
2．2002年に、(前年の数)×(増加率)の式を入力
※絶対セル番値を用いるのを忘れずに！
3．2003年から下にコピー

応用:
2002年の式を、int関数を使って整数にした上で、下にコピーしてみよう。

```
#Python Program
popl = 127316
ratio = 127316 / 126926
for i in range(2002,2036,1):
    popl = int(popl * ratio)
    print(i,popl)
```

		増加率	
年	人数(千人)		
2000			
2001			
2002			
2003			
2004			
2005			
2006			
2007			
2008	127692		
2009	127510		
2010			
2011			
2012			
2013			
2014			
2015			
2016			
2017			
2018			
2019			
2020			
2021			
2022			

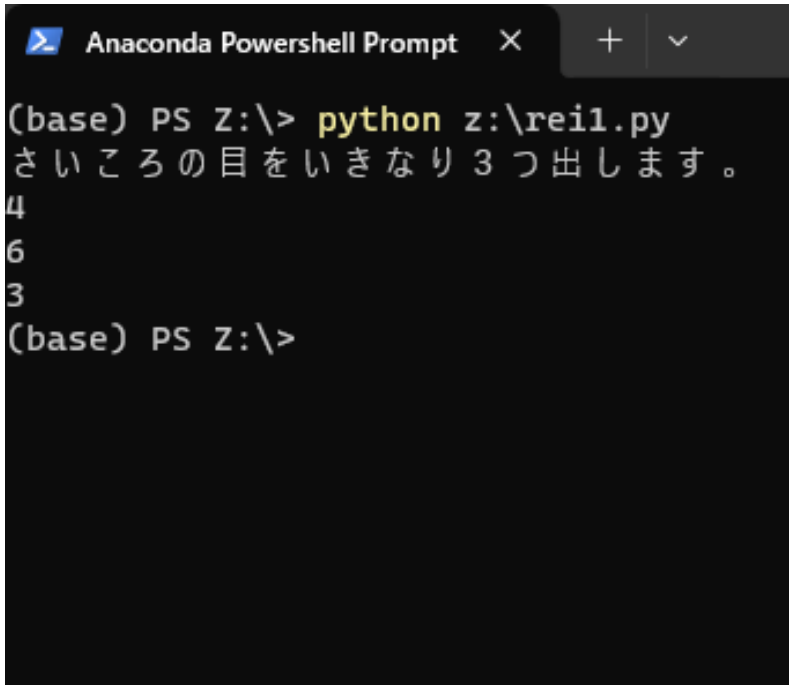
```
#Python Program
（自分で考えて表示させてみよう！）
```

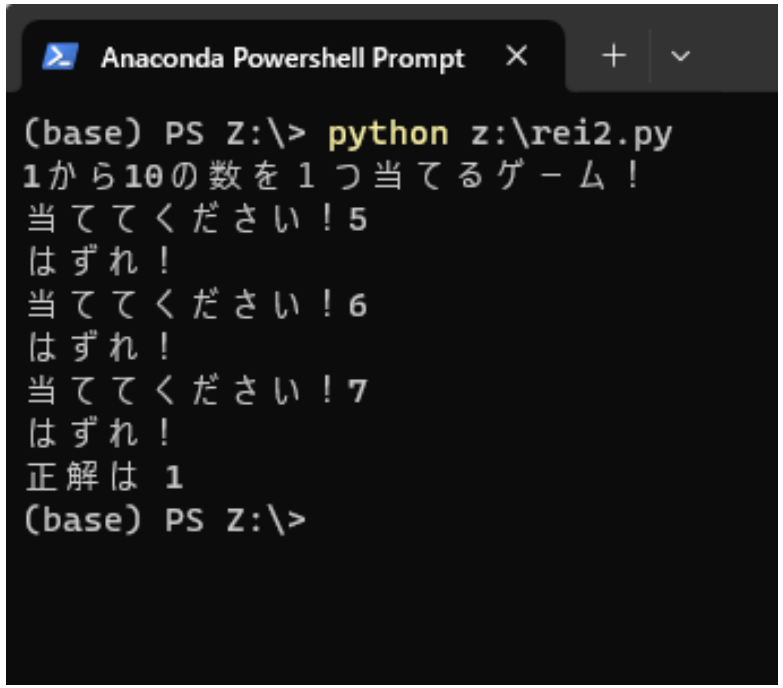
実習　乱数で「何か」を作ろう

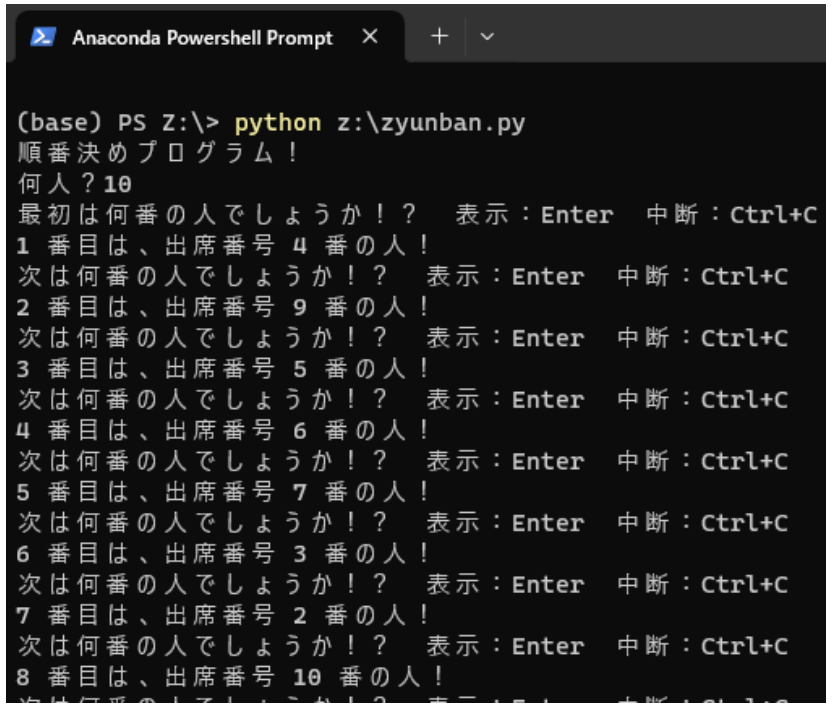
- 乱数と、判断分岐または繰り返しの機能を使って、「何か」のプログラムを「グループで最低1つ」作ろう。
- 楽しいもの、役に立ちそうなもの、など、何でもOK。
- 意味がわからないものはダメ。誹謗中傷も厳禁!!
- グループ作業です。グループ内はたくさん助け合ってください。

  ただし、この時間はグループ以外の人と相談するのは禁止!
- グループメンバー全員が、同じプログラムを再生できるように打ち込み作成しておいてください。
- 本日残り時間を入れて3回が作業時間です。

教員がサンプルを提示


Anaconda Powershell Prompt
(base) PS Z:\> python z:\rei1.py
さいころの目をいきなり３つ出します。
4
6
3
(base) PS Z:\>


Anaconda Powershell Prompt
(base) PS Z:\> python z:\rei2.py
1から10の数を１つ当てるゲーム！
当ててください！5
はずれ！
当ててください！6
はずれ！
当ててください！7
はずれ！
正解は 1
(base) PS Z:\>

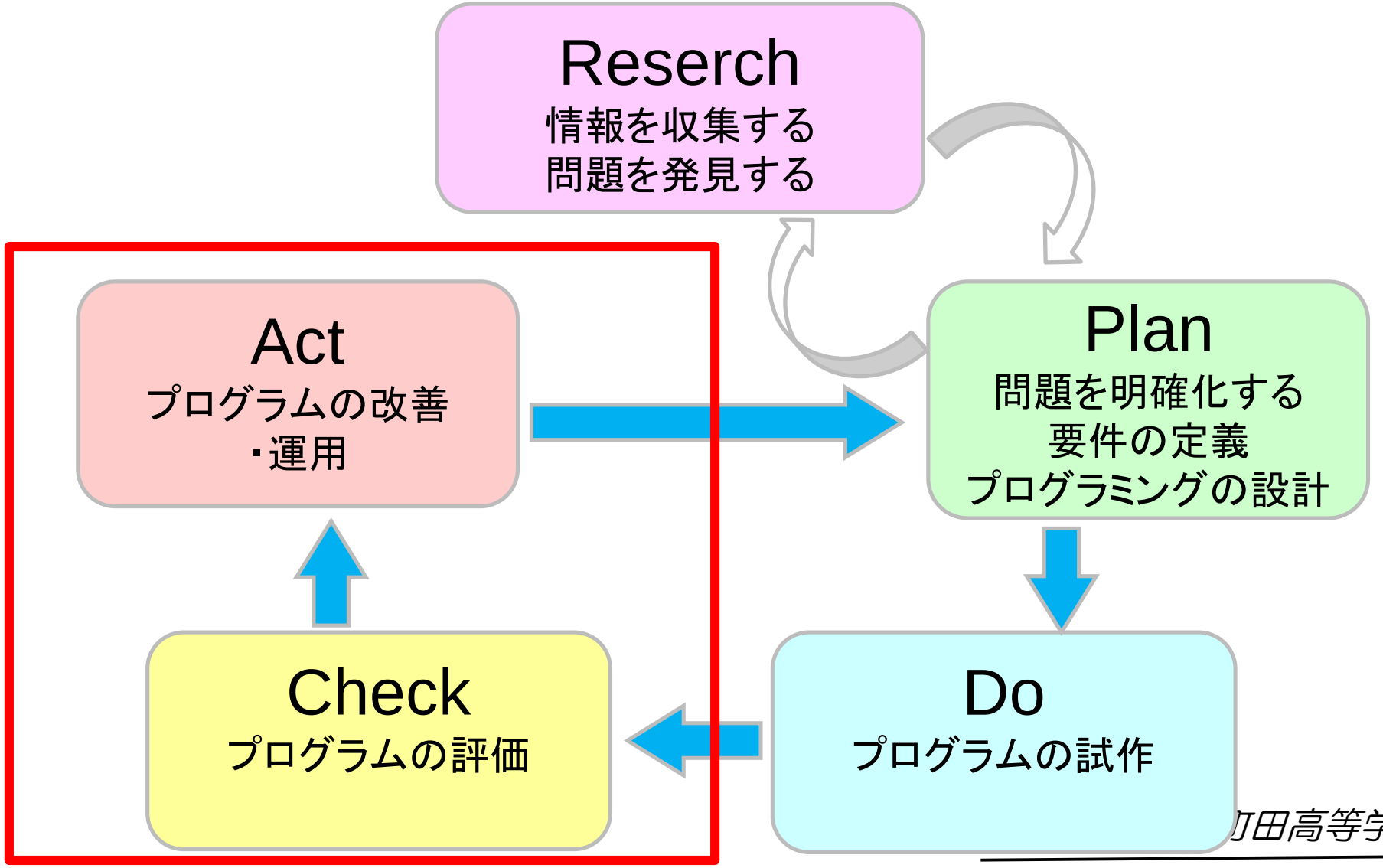

Anaconda Powershell Prompt
(base) PS Z:\> python z:\zyunban.py
順番決めプログラム！
何人？10
最初は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
1 番目は、出席番号 4 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
2 番目は、出席番号 9 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
3 番目は、出席番号 5 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
4 番目は、出席番号 6 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
5 番目は、出席番号 7 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
6 番目は、出席番号 3 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
7 番目は、出席番号 2 番の人！
次は何番の人でしょうか！？　表示：Enter　中断：Ctrl+C
8 番目は、出席番号 10 番の人！

プログラミングと問題解決（発表）

情報Ⅰ　第39回授業

05コンピュータとプログラミング

本日の作業


Reserch
情報を収集する
問題を発見する
Plan
問題を明確化する
要件の定義
プログラミングの設計
Do
プログラムの試作
Check
プログラムの評価
Act
プログラムの改善
・運用

発表の進め方

発表の打合せ（５分）

発表グループで
デモンストレーション

１概要説明

・実演
・プログラムの解説
・工夫した点
・課題　など

（合計3分）

見学者のコメント

（一人ずつ）

・良かった点
・改善提案

をコメントする

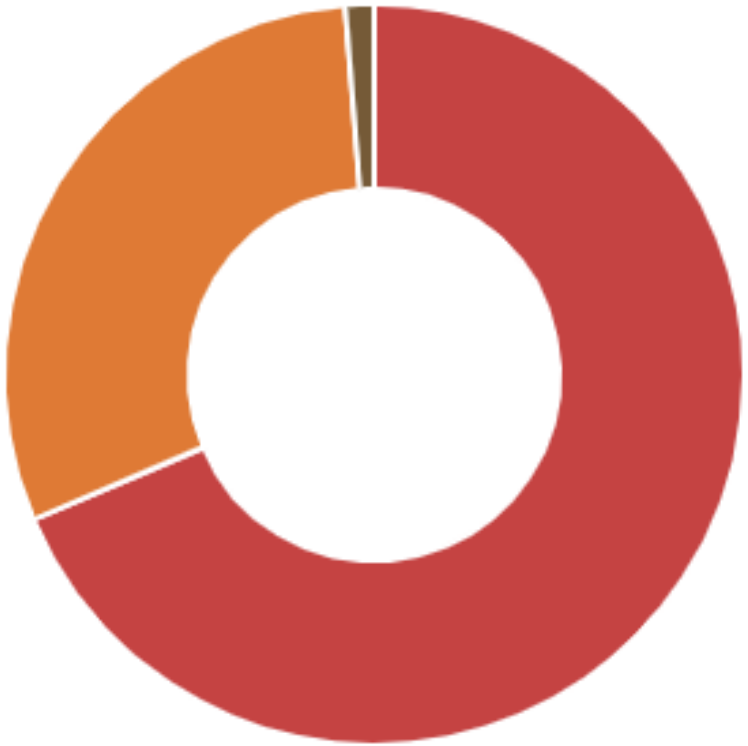
（合計2分）

※この時間帯に、不明な点や質問のやりとりをすると良い

指摘された箇所の吟味
修正作業（終了時刻まで）

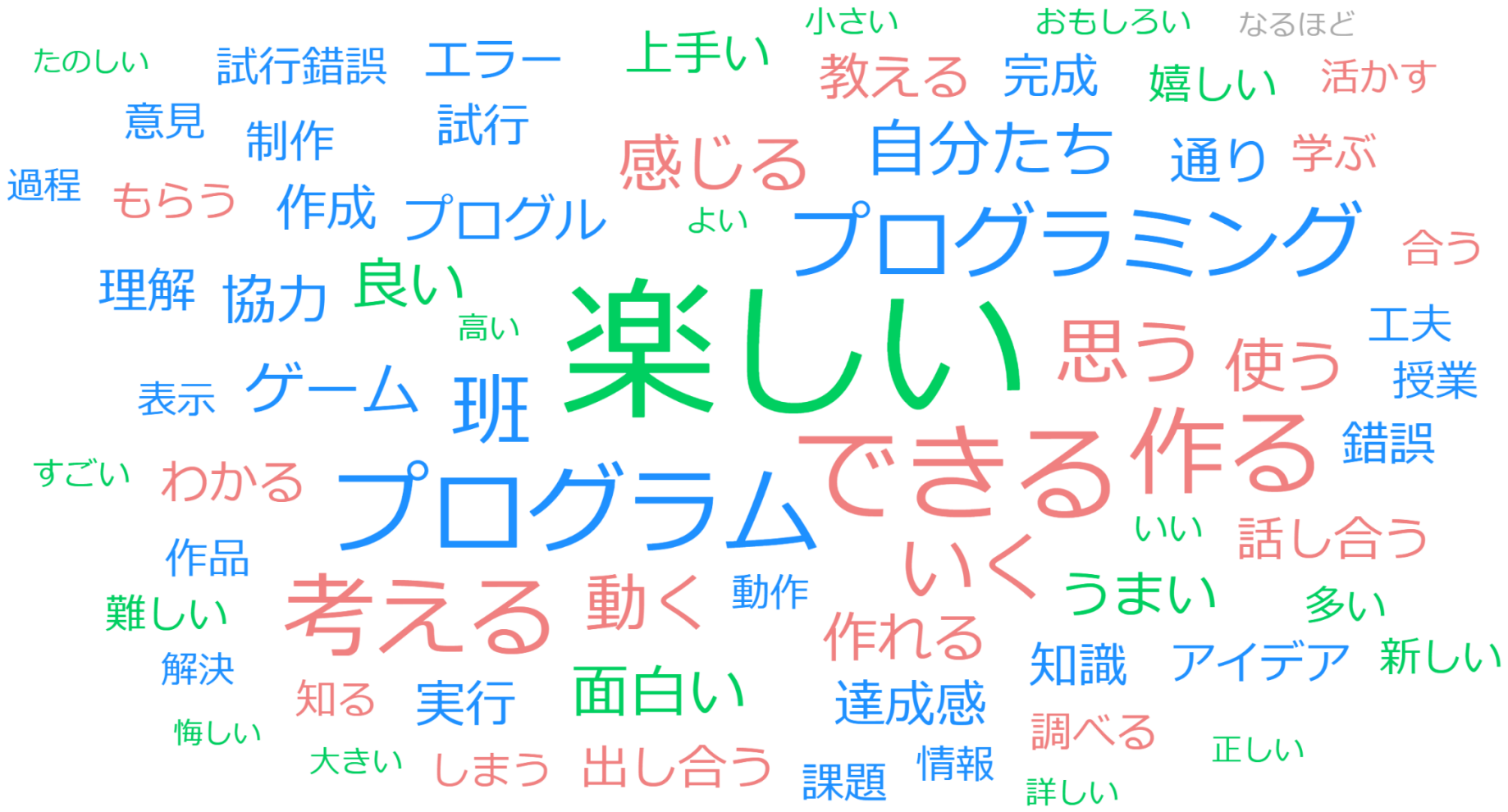
設問 2 課題制作は楽しかったですか

未回答を含める　回答数 239



- 選択肢 1　164人(68.62%)　とても楽しかった
- 選択肢 2　72人(30.13%)　まあ楽しかった
- 選択肢 3　3人(1.26%)　あまり楽しくなかった
- 選択肢 4　0人(0%)　全く楽しくなかった

どのような点が楽しかったり楽しくなかったりしましたか。



「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/

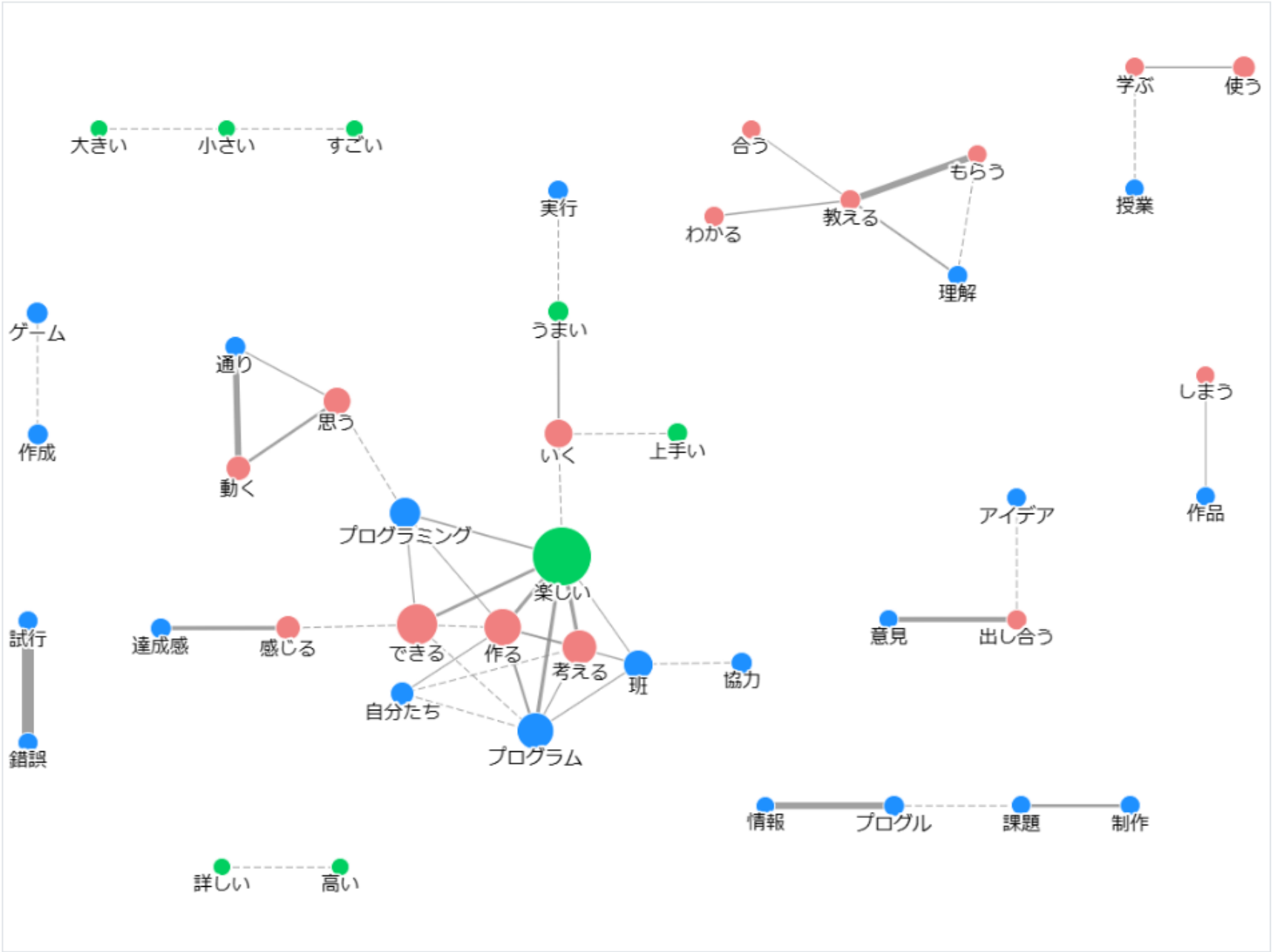
単語出現頻度

名詞	スコア	出現頻度
プログラム	632.36	144
プログラミング	514.65	105
班	339.67	92
自分たち	125.21	47
ゲーム	6.06	35
協力	28.43	31
プログル	348.64	28
達成感	85.63	28
作成	40.97	28
通り	11.27	28
実行	57.19	27
知識	20.74	25
完成	16.54	25
理解	8.30	24
錯誤	80.52	22

動詞	スコア	出現頻度
できる	23.96	141
作る	37.92	119
考える	28.37	104
いく	8.66	68
思う	2.24	63
動く	19.98	44
感じる	9.91	44
使う	2.65	34
作れる	13.26	25
教える	2.71	21
話し合う	43.71	20
わかる	0.77	20
出し合う	95.13	19
学ぶ	9.85	15
調べる	2.41	14

形容詞	スコア	出現頻度
楽しい	118.76	245
面白い	3.16	29
良い	0.98	27
うまい	5.41	24
上手い	4.47	20
嬉しい	0.58	15
難しい	1.34	13
新しい	1.17	12
多い	0.42	12
いい	0.05	8
すごい	0.06	5
正しい	0.72	4
大きい	0.17	4
高い	0.10	4
よい	0.03	4

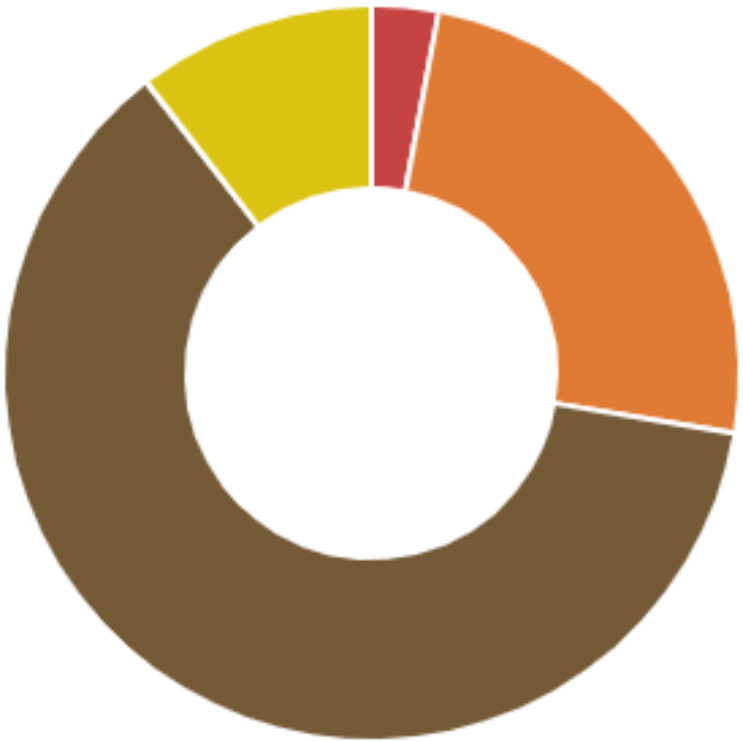
「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/



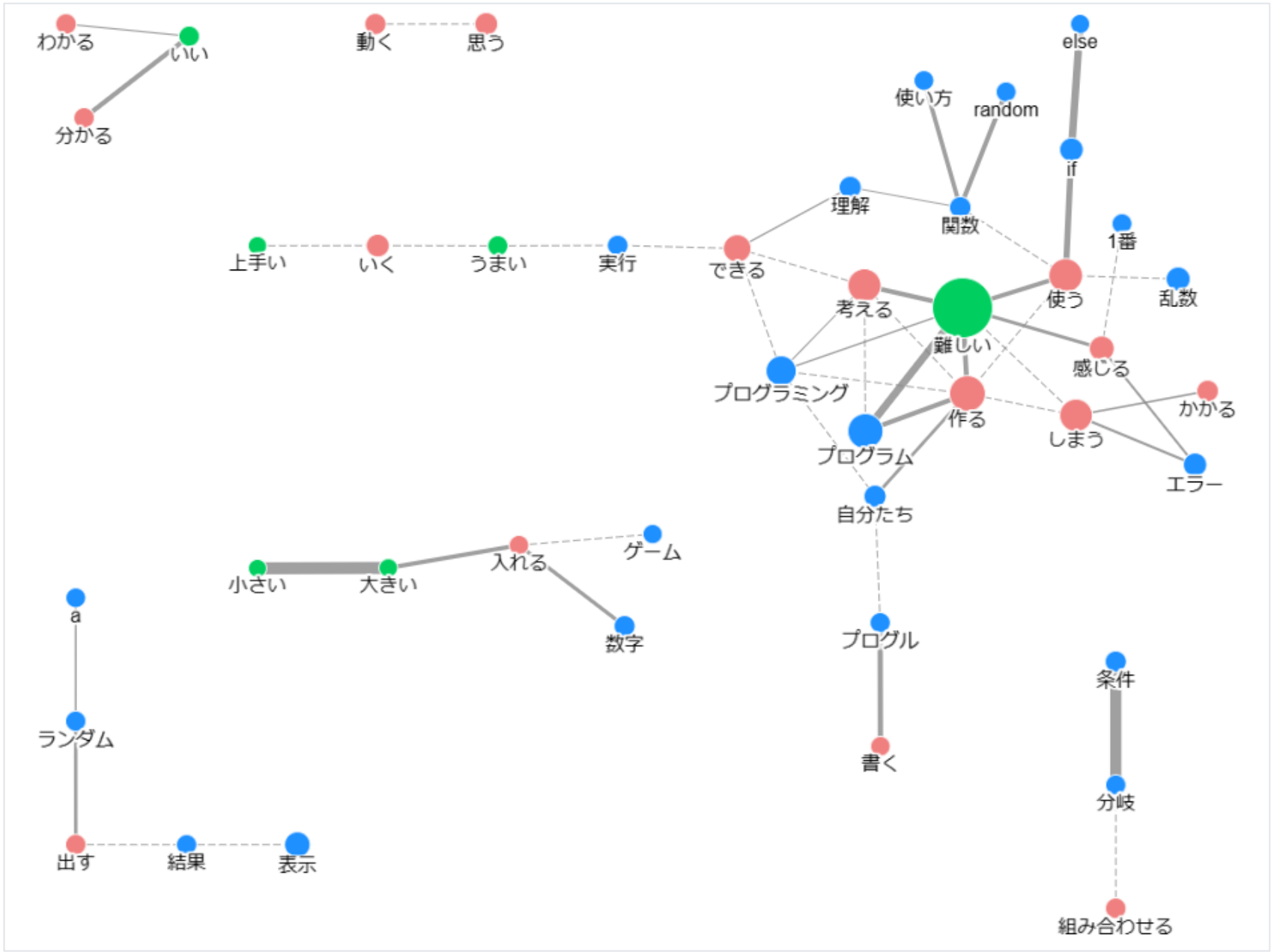
「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/

設問 4 内容は簡単でしたか

未回答を含める　回答数 239



- 選択肢 1　7人(2.93%)　とても簡単だった
- 選択肢 2　59人(24.69%)　まあ簡単だった
- 選択肢 3　148人(61.92%)　やや難しかった
- 選択肢 4　25人(10.46%)　とても難しかった



「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/

■ 名詞 - ■ 動詞

名詞 - 動詞	スコア	出現頻度
プログラム - 作る	4.03	17
乱数 - 使う（否: 11.11%）	1.34	9（否: 1）
プログラム - しまう	1.14	8
エラー - しまう	0.89	7
プログラム - 組む	4.29	5
エラー - 出る	1.50	5
関数 - 使う	0.45	5
自分たち - 作る	0.39	5
班 - 作る（否: 20.00%）	0.39	5（否: 1）
プログラミング - 作る	0.39	5
エラー - 起きる（否: 25.00%）	2.86	4（否: 1）
頭 - こんがらがる	2.50	4
ランダム - 出す	1.43	4
自分たち - 考える	0.31	4
スペース - 開ける	2.00	3

「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/

設問 7 「プログル情報」の「AIにヒントを聞く」は活用しましたか。

未回答を含める　回答数 236

- 選択肢 1　114人(48.31%)　とても活用し、非常に役に立った
- 選択肢 2　86人(36.44%)　それなりに活用し、まあ役に立った
- 選択肢 3　18人(7.63%)　時々は活用したが、あまり役には立たなかった
- 選択肢 4　18人(7.63%)　ほとんど活用しなかった

名詞	スコア	出現頻度
エラー	206.04	77
プログラム	268.25	76
ミス	71.83	55
ヒント	151.32	52
ai	230.50	50
プログラミング	83.03	27
活用	33.02	21
スペル	68.04	20
指摘	25.37	20
間違い	6.95	20
関数	69.14	18
場面	17.78	18
使い方	14.30	18
if	40.33	17
実行	22.71	15

動詞	スコア	出現頻度
くれる	8.42	85
できる	6.43	72
わかる	8.15	66
間違う	77.18	62
役に立つ	137.76	54
教える	10.88	43
使う	4.01	42
聞く	2.96	35
分かる	4.03	33
役立つ	68.56	28
しまう	1.06	26
違う	2.22	24
気づく	4.29	23
忘れる	1.94	22
思う	0.28	22

「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/

■ 名詞 - ■ 動詞

名詞 - 動詞	スコア	出現頻度
ヒント - 聞く	18.06	25
エラー - 出る	14.62	19
エラー - 起きる	11.38	13
ミス - くれる	1.81	12
エラー - しまう	4.89	11
ヒント - 見る	3.50	6
関数 - 使う	0.98	6
エラー - 起こる	5.00	5
ヒント - もらう	2.00	5
つけ - 忘れる	1.30	5
場面 - 役に立つ	0.55	5
原因 - わかる（否: 80.00%）	0.45	5（否: 4）
プログラム - いく（否: 75.00%）	1.54	4（否: 3）
プログラム - 作る	1.25	4
全角 - 打つ	1.11	4

「UserLocal AIテキストマイニング」 https://textmining.userlocal.jp/

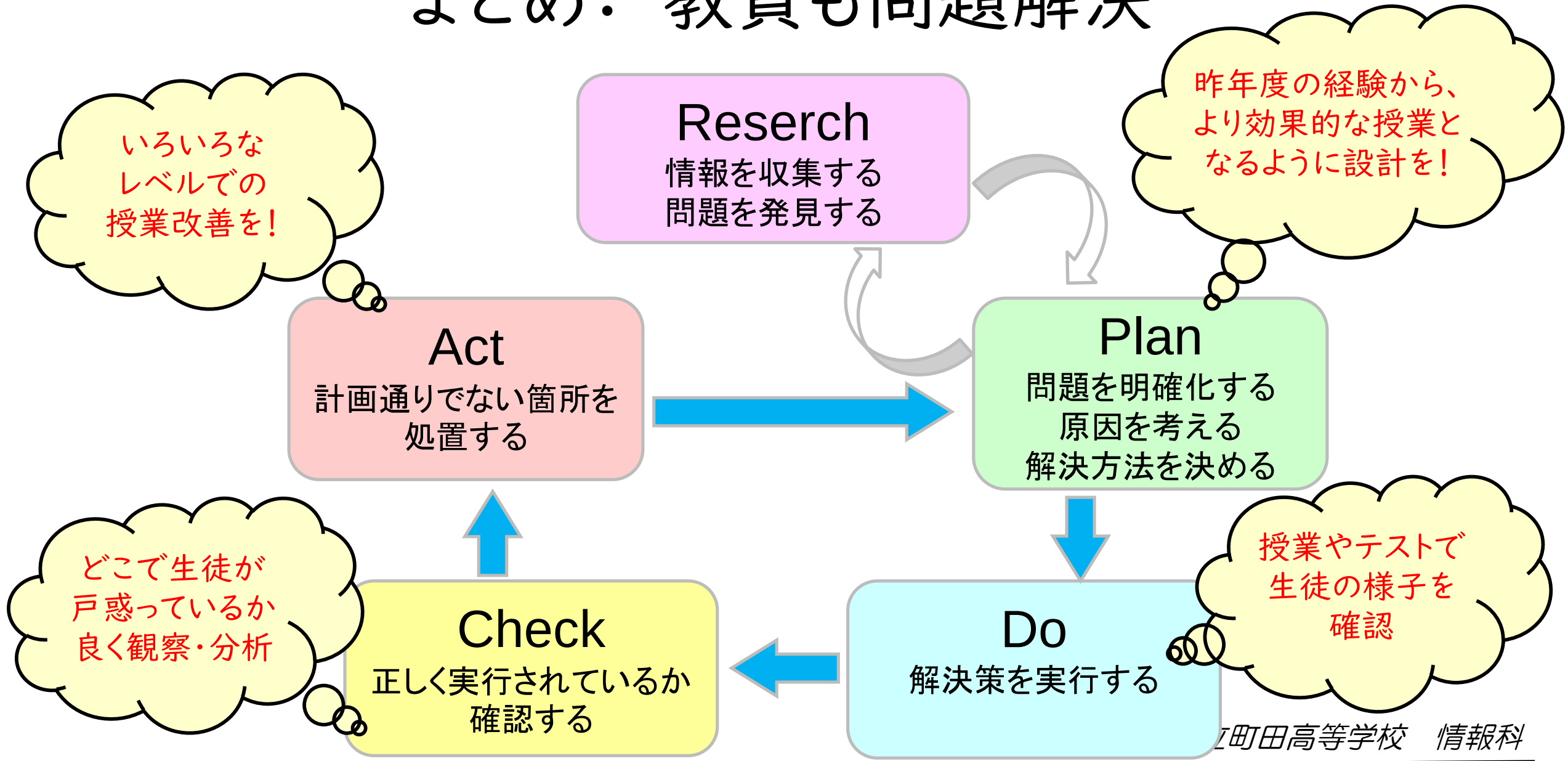
具体的な記述より

- どこが間違っているか分からず時間がかかっていた時にAIヒントを使ったことでインデントがあっていないことが分かりすぐに作業を進められた。
- 書いたコードの何行目がどう違っていて別のどのやり方で書けばいいのかを教えてもらった。例えば空白が足りなかったり余分なコードの指摘があったりとヒントをもらった。
- プログラムにカッコをつけ忘れたり、改行ミスなど、指示文の内容ではなく文法的なところの間違いを指摘してくれる場面で役に立った。
- 「if」「elif」「else」の使い方が合っているか確認する時に一番役に立った。
- 自分では合っていると思っているのに実はカンマをつけていなかったり、セミコロンをつけていなかったりと、非常に気づきにくい些細なミスをしている時に役に立った。

まとめ

- オンラインサービスの活用は非常に効果的
  - 「自分のペースで」「どこででも」「わからない所をすぐ聞ける」環境
  - 生徒の個人差を埋め、生徒どうしの教えあいも活発に
  - 「どの部分を」「どのように活用するか」という設計は重要
- プログラミングを「楽しかった」と思っている生徒は多い
  - 実は、「ケアレスミス」で動かず嫌になっている？
  - 「いつでも機械にフォローしてもらえる」環境は非常に有益
- サービスでカバーしきれない内容のフォローも必要

まとめ： 教員も問題解決


Reserch
情報を収集する
問題を発見する
昨年度の経験から、
より効果的な授業と
なるように設計を！
Plan
問題を明確化する
原因を考える
解決方法を決める
授業やテストで
生徒の様子を
確認
Do
解決策を実行する
Check
正しく実行されているか
確認する
どこで生徒が
戸惑っているか
良く観察・分析
Act
計画通りでない箇所を
処置する
いろいろな
レベルでの
授業改善を！

参考文献等

- 文部科学省 「高等学校学習指導要領（平成30年告示）解説 情報編 平成30年7月」 開隆堂（2019）
- 黒上晴夫 堀田龍也 村井純 編 「情報I」　日本文教出版（2022）
- 特定非営利活動法人みんなのコード「プログル情報」 https://high.proguru.jp/
- UserLocal「AIテキストマイニング」https://textmining.userlocal.jp/
- 小原格 「情報科準備室～小原研究室」 http://www.johoka.info/　（2024.1.8参照）