SSH コンソーシアム TOKAI の 1・2年生を対象とした「高大接続探究ゼミ」

Pythonでフラクタル を描画しよう ベーシックコース

名古屋大学 山里敬也 yamazato@nagoya-u.jp



山里敬也 (YAMAZATO, Takaya) <u>yamazato@nagoya-u.jp</u> 名古屋大学 教養教育院 教授 ハイブリットラーニングセンター・センター長 兼務:工学研究科情報・通信工学専攻, 情報基盤センター教育情報メディア研究部門, 附属図書館研究開発室 名大の授業:http://ocw.nagoya-u.jp 研究 可視光通信、ITS、確率共鳴、Open Educational Resources (OER) 趣味 自転車、ジョギング、バンド

ROUTE1+ HALL ROUTE23





lyoda 伊与田 友貴

- ・研究テーマ
- ▶ 可視光通信/機械学習 ・好きなもの

Yuki



≫ 水族館/猫/ゲーム/バイク/…

"なんでも聞いてください"



> 名前:石田匠

> 学年:博士後期課程1年

▶ 研究:水中光無線通信

> 趣味:麻雀,スポーツ観戦

Reinforcement Learning-Based Link Switching in Underwater Optical Wireless Communication Systems Using Intelligent Reflecting Surfaces

Takumi Ishida, Chedlia Ben Naila, and Hiraku Okada Nagoya University, Japan

Carpe di

IEEE International Conference on Communications (ICC2025) 2025/6/9 Montréal, Canada



名大で実際し	こ行っている
プログラミング及	び演習(3.0 単位
科目区分	専門基礎科目
授業形態	講義及び演習
対象学科	電気電子情報工
開講時期1	2年春学期
必修/選択	必修
担当教員	山里 敬也 教授



学科

2021年度は山里が担当しました. 2022年度から小川先生が担当されています.

拓郎 准教授





本講義の目的(シラバス) C言語による演習を通じて、計算機を用いたより高度なプログラミング 技法・問題解決技法を学ぶ。具体的には比較的大きなプログラム (500~1000行程度)を書く実力をつける。 これにより、情報リテラシーのような基礎力に加え、論理的思考力や問 題解決力といった応用力も涵養する。さらには、プログラムの構成を設 計(デザイン)することにより、創造力を鍛えることも狙いとする。

達成目標:

-目的・仕様に従いプログラムの構成要素を論理的に設計できる -設計に従い、効率の良いプログラム(**C言語**)の実装ができる



今回は Python を使います









	日付
ガイダンス・	4/16
	4/23
	4/30
	5/07
ポ	5/14
	5/21
	5/28
	6/04
ビジ	6/11
ビジュアリゼー	6/18
	6/25
	7/02
	7/09
	7/16
	7/30
	8/06









スケシ	ジュール	
日時	ベーシック	アドバンス
7月31日(木) 10:30~12:00 講義	Google Colabortory (Python)入門 Turtle Graphics入門	Google Colabortory (Python)入 Turtle Graphics入門
7月31日(木) 13:00~14:30 演習	Turtle Graphics で 自分のイニシャルを描こう	Turtle Graphics で多角形を描こ
8月1日(金) 10:30~12:00 講義	Turtle Graphics で多角形を描こう	再帰関数とフラクタル (コッホ曲線、シェルピンスキーのガスケ ト、2分木、Levy曲線、Drangon曲線)
8月1日(金) 13:00~14:30 演習	Turtle Graphics で絵を描こう	Turtle Graphics で フラクタルを描こう



資料について

日時	ベーシック	アドバンス
7月31日(木) 10:30~12:00 <mark>講義</mark>	Google Colabortory (Python)入門 Turtle_Graphics_Basic.ipynb	Google Colabortory (Python)入 Turtle Graphics入門 Turtle Graphics Basic.jpynb
7月31日(木) 13:00~14:30 演習	を使います 自分のイニシャルを描こう	を使います Turtle Graphics で多角形を描こ
8月1日(金) 10:30~12:00 <mark>講義</mark>	Turtle Graphics で多角形を描こう Turtle_Graphics_Basic.ipynb を使います	再帰関数とフラクタル (コッホ曲線、シェルピンスキーのガスケ Turtle_Graphics_Advanced.ipynb を使います
8月1日(金) 13:00~14:30 <mark>演習</mark>	Turtle Graphics で絵を描こう	Turtle Graphics で フラクタルを描こう



最終目標: Turtle Graphics で絵を描こう













2日目にチャレンジします!





Google Colabortory **Python入門**



11

準備 (事前課題) **Google Colaboratory**

- この講義では <u>Google Colaboratory</u> (コラボと呼びます)を使います.
- さい
- コラボはPCのブラウザから利用できます。自宅からでもOK!
 - <u>Google Chromeをダウンロードしてご利用ください.</u>
- 事前学習をやりましたか?
 - <u>https://yamazato.nuee.nagoya-u.ac.jp/research/OER/wwl/</u>

コラボの利用には <u>Google アカウント</u>が必要です。事前に取得しておいてくだ

ブラウザは何でも良いと思います。もし、うまくいかないようであれば



コラボノートブック Colab Notebook

- コラボを立ち上げると右のような画面になります
- Zフォルダからファイルをダ ウンロードし、ご自身のコラ ボにアップロードしてください。
- <u>コラボの使い方</u>
 - <u>https://sites.google.com/</u>
 <u>view/ipsjmooc/How2</u>

 → C ● colat
 Colaboratory ファイル 編集
 目次
 はじめに データ サイエンス 機械学習
 その他のリソース
 機械学習の例
 セクション

olab.research.google.com/n	otebooks/intro.ipynb#recent=true	\$	6	•
oryへようこそ 潗 表示 挿入 ランタイム	ツール ヘルプ		d	ю
×	+ コード - + テキスト - ▲ ドライブにコピー	撼	镜	~

🔘 Colaboratory とは

Colaboratory(略称: Colab)は、ブラウザから Python を記述、実行できるサービスです。次の特長を備えています。

- 環境構築が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、**学生**からデータ サイエンティスト、AI リサーチャーまで、皆さんの作業を効率化します。詳しくは、<u>Colab の紹介動画</u>をご覧くだ さい。下のリンクからすぐに使ってみることもできます。

例	最近	Google ドライブ	GitHub	アッ	70-	- 14	
レートブックを	絞り込む	Ŧ					量できるインタラクティ
タイトル	L.		最終閲覧 👻	最初に聞いた日 時 マ		T.	T.
CO Colabor	alory へようこそ		14:10	6月15日		Ø	
🔼 ස්රීන්ඝ	Lipynb		14:09	6月30日	۵	Ø	キーボード ショートカ
📥 turtleGr	aphi cs.ipyn b		14:09	6月25日	۵	Ľ	
Q 1-1.ipyn	b		5月30日	6月15日	۵	Ø	
🔘 1-0.ipyn	b		5月30日	6月15日	۵	Ø	
		L GAUIZ GAL <u>22</u> 27 -	/-	トブックを新規作成	1	キンセル	ュメントで記述できま 同僚や友人と簡単に共 ab ノートブックを作成
Colab ノートプッ	ックは、Colab がホス	トする Jupyter ノートブックで	す。Jupyter プロ:	ジェクトの詳細に	วเก	cit. jupy	ter.org をご覧くださ





コラボの基本 ^{コラボの code セルを使ってみよう}

[] 1+1

押下すると pythonプログラム が実行できます.

セル

ノートブックはセルから成り立っています。

主に次の二種類のセルを使います。

- Codeセル(コードセル): Pythonのコー ドを実行するには、[]のところにマウン
 ▷が表示されます。これはプレイボタンジョ
 す。(Shiftを押しながらEnterを押しても)
- Markdownセル(テキストセル): 説明た







警告がでるけど心配いりません.



このノートブックは <u>GitHub</u> から読み込まれています。Google に保存されているデータへの アクセスが求められたり、他のセッションからデータや認証情報が読み取られたりする場 合があります。このノートブックを実行する前にソースコードをご確認ください。

キャンセル このまま実行





Codeセルにプログラムを入力してみよう Pythonが実行できます

- cut, copy, paste, undo ができます
 - Ctrl+c, Ctrl+x, Ctrl+v, Ctrl+z
- 四則演算もできます

演算子	説明	例	結果
**	累乗	2**3	8
*	かけ算	3*5	15
/	割り算	10/3	3.33
//	整数の割り算 (小数点以下切り捨て)	10//3	3
%	剰余(余り)	10%3	1
+	足し算	1+2	3
_	引き算	10-3	7











Pythonの命名規則 ほぼスネークケース(単語と単語をアンダースコアでつなぐ)

変数	スネークケース	snake_case
定数	スネークケース(全て大文字)	SNAKE_CASE
グローバル変数	スネークケース	snake_case
関数	スネークケース	snake_case
関数の引数	スネークケース	snake_case
クラス	パスカルケース	PascalCase
インスタンス変数	スネークケース	snake_case
メソッド	スネークケース	snake_case
パッケージ	スネークケース	snake_case
モジュール	スネークケース	snake_case





for i in range(4):
 print(i)



range() 整数列のリストを返す関数

range(start, stop[, step])

range(0,4,1)> 0, 1, 2, 3 range(4)> 0, 1, 2, 3 range(0,4,2)> 0, 2 range(4,0,-1)> 3, 2, 1, 0



タートルグラフィックス









- 前進:forward(距離)
- 回転:right(角度), left(角度)

ペンの上げ下げ : up(), down()

pen down



https://en.wikipedia.org/wiki/Turtle_graphics



タートルグラフィックスの主な関数 https://github.com/tolgaatam/ColabTurtle

関数	短縮形(別表記)	
forward(units)	fd(units)	
backward(units)	bk(units)	
right(degrees)	rt(degrees)	
left(degrees)	lt(degrees)	
speed(s)		亀を
penup()	up()	
pendown()	pd()	
setpos(x,y)	goto(x,y)	
bgcolor()		-
pencolor()	color()	
pensize(w)	width(w) pensize(w)	
position()	pos()	
heading()	getheading()	

説明

亀を units (ピクセル)前進

亀を units (ピクセル)後進

亀を degrees 度 右に向ける(回転)

亀を degrees 度 左に向ける(回転)

をスピード s で動かす.1が最も遅く、13が最も早い.初期値 4

ペンを上げる(以後、描画しない)

ペンを下げる(以後、描画をはじめる)

座標 (x, y) へ 亀を移動

背景の色を指定 (<u>https://www.w3schools.com/colors/colors_names.asp</u>)

ペンの色を指定 (<u>https://www.w3schools.com/colors/colors_names.asp</u>)

ペンの大きさ w を指定

亀の現在の座標(x,y)を取得

亀が向いている方向(角度)を取得



Colabでのタートルグラフィックス

https://github.com/tolgaatam/ColabTurtle

- まず、以下のコマンドでタートルグラフィックスをインストール !pip3 install ColabTurtle from ColabTurtle.Turtle import *
- 図を書くには初期化が必要になります. initializeTurtle()
- 描画領域は以下のように設定されています.
- 画面サイズ: 800 x 500
- 亀の位置:画面の中央(400,250)
- 亀の向き:上
- 亀の動く速さ:speed = 4
- ペンのサイズ:width(4)
- ペンの色:color("white")
- 背景の色:bgcolor("black")







タートルグラフィックス まずは設計から

- 800x500の画面に一辺が 100の正方形を亀の軌跡で 描く
- 画面は左上が(0,0), 右下が(800,500)
- 亀の最初の位置は方眼紙の 中心 (400,250)



(0,0)



中心から右へ亀さんを動かす forward

- 800x500の画面に一辺が 100の正方形を亀の軌跡で 描く
- 画面は左上が(0,0), 右下が(800,500)
- 亀の最初の位置は方眼紙の 中心 (400,250)

(0,0)







試してみよう!

Google Colaファイルを開いて自分で試してみてください.



 亀さんを上下左右に動 かすことで数字の0~
 9, そしてアルファベ ットのA, C, E, F, G, H, J, L, O, P, S, Uを描く ことができます.











演習:自分のイニシャルを描こう







タートルグラフィックス まずは設計から (0,0)

- 800x500の画面に一辺が 100の正方形を亀の軌跡で 描く
- 画面は左上が(0,0), 右下が(800,500)
- 亀の最初の位置は方眼紙の 中心 (400,250)





$(800, 500)_{30}$

(500, 150)





タートルグラフィックス 亀を動かそう

- 100前進 1)
- 90度回転 2)
- 100前進 3)
- 90度回転 4)
- 100前進 5)
- 90度回転 6)
- 100前進
- → 正方形が描画できる









タートルグラフィックス コマンドで書くと

forward(100) right(90) forward(100) right(90) forward(100) right(90) forward(100) right(90)







同じ処理のくり返しはループでまわす

for i in range(4): forward(100) right(90)

range(start, stop[, step])

range(0, 4, 1)> 0, 1, 2, 3 range(4) > 0, 1, 2, 3 range(0,4,2) > 0, 2 range(4,0,-1) > 3, 2, 1, 0







やってみよう:正三角形 ??は自分で考えてね

for i in range(?): forward(100) right(??)





やってみよう: 正六角平 ??は自分で考えてね

for i in range(?):
 forward(100)
 right(??)





一般化:正n角形 ??は自分で考えてね length=100 angle = ?? for i in range(?): forward(length) right(angle)





関数 polygon ??は自分で考えてね

def polygon(n, length):
 length = ??
 angle = ??
 for i in range(??):
 ??
 ??





円はどう描く?





少し複雑な図形に挑戦

Star
angle = ??
for i in range(?):
 forward(200)
 right(angle)



だんだん小さくなる

Star2 angle = ?? for i in range(?): forward(200-i*10) right(angle)





39

回りながら三角形をくり返し描く

回転

Triangle
angle = ??
for i in range(?):
 forward(200)
 right(angle + ??)









ペンアップ:描くのをやめる



ペンダウン:描きはじめる

```
## Star
angle = ??
for i in range(?):
    forward(100)
    right(angle)
right(70)
for i in range(3):
    penup()
    goto(480+i*10,150+i*10)
    pendown()
    left(8)
    for i in range(10):
        forward(20)
        right(2)
    penup()
```





演習:絵を描こう







