

プログラミング教室のテクノロ



# プログラミングの 世界の歩き方

## 「ビルド②」



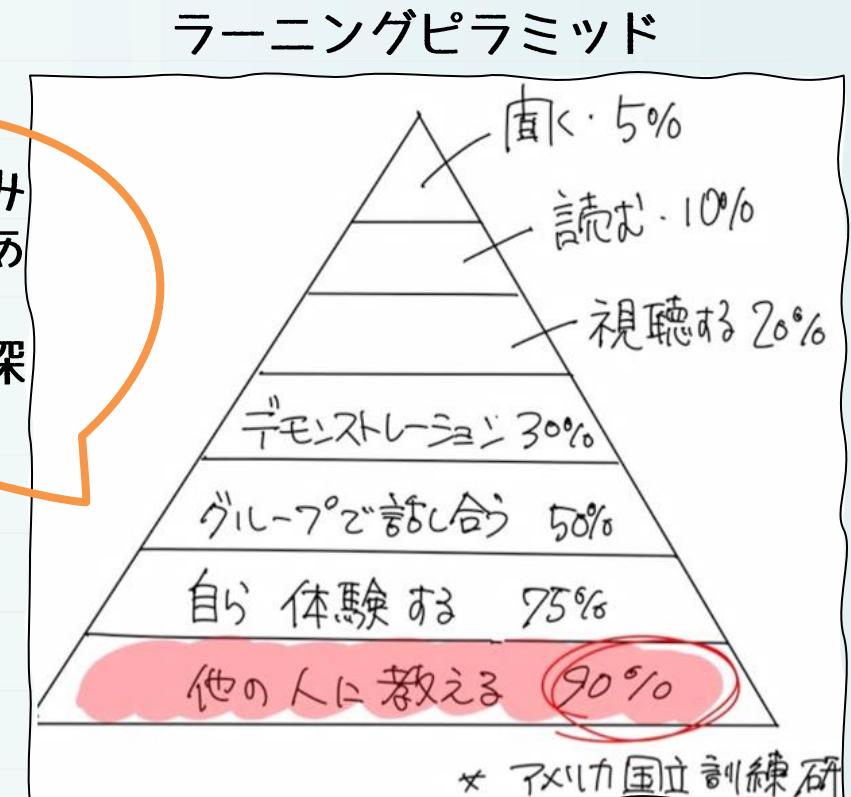
# プログラミングの世界を歩こう！

プログラミングの世界を知るにはその世界で使われていることばを知ることが大切だ。

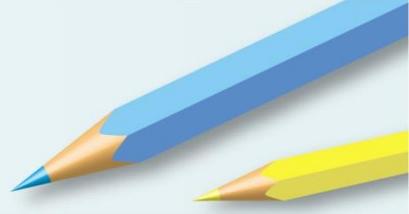
これはプログラミングに限らず、スポーツでも音楽でも何かを習得するには、その世界のことばを知ることから始まるよ。



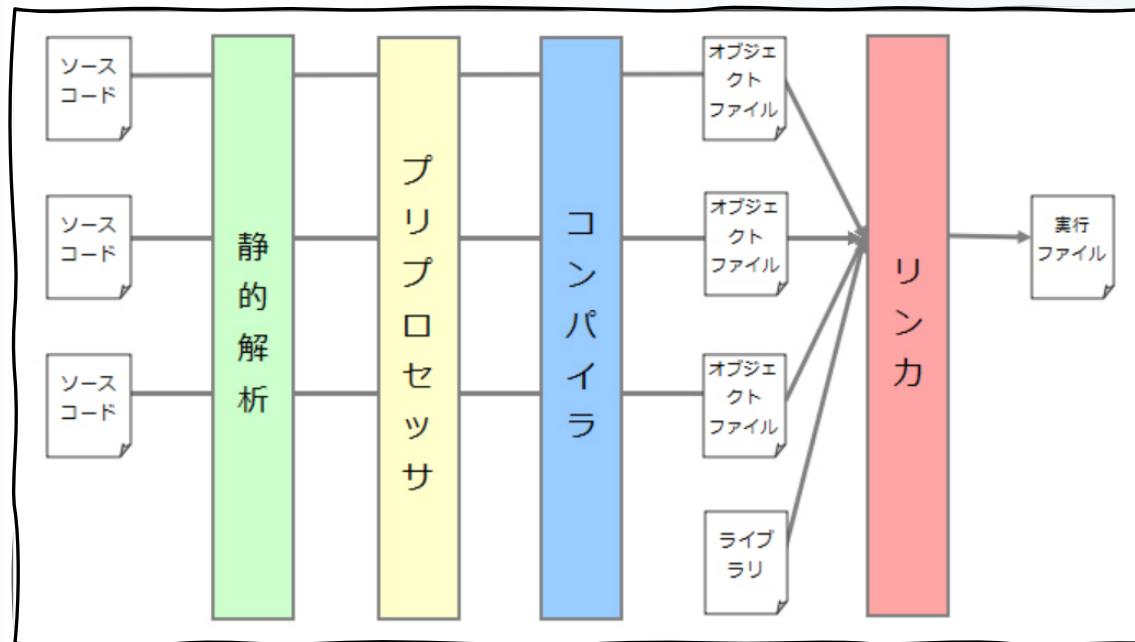
学んだことはみんなに教えてあげよう！  
もっと理解が深まるよ！



# ビルドとは？（前回のおさらい）



ビルドとは、ソースコードに問題（バグ）がないか解析を行い、問題がなければ実行できる形のファイルに変換し、組み立てるこ<sup>ト</sup>。1つの作業のことを指すのではなく、解析から実行可能ファイルを組み立てるところまでの一連の流れをいう。



単一なソースコードのプログラムであればコンパイルするだけでいいが、複雑な仕組みのソフトウェアは、多くのソースコードを使って構成されていて、それらをリンクして組み合わせるという作業が必要になる。

# ビルドとは？（前回のおさらい）

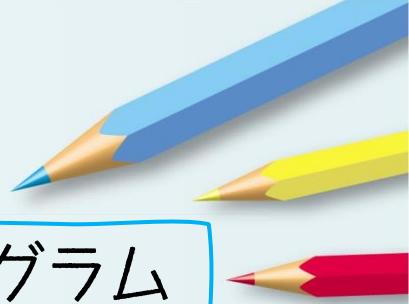
実行ファイルを作る製作工程で、欠かせない担当がいる。それが次の3つの担当である。

## 実行ファイルの製作工程で活躍する担当

- プリプロセッサ … 編集担当
- コンパイラ … 校正・翻訳担当
- リンカ … 製本担当



# ソースコードとは？



ソースコードとはプログラムの基になるもので、人がプログラム言語を使って記述したコードのこと。

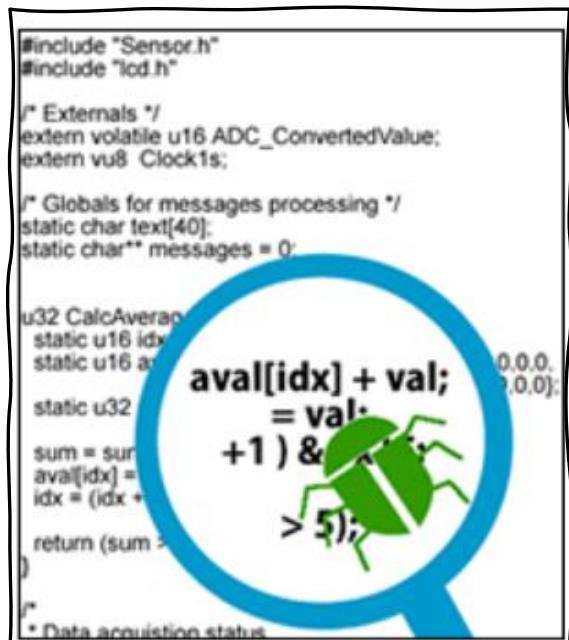
基本的に命令を並べた文章になるが、必要に応じてソースコードに対しての説明（コメント）を書き加えることもある。

```
one_hour_dungeon.py - C:\Users\sator\Documents\プログラミングスクール\01_教材\02_Pythonの道\参考書籍\pythonでつくるゲーム開発入門講座\Chapter12\one_hour_dungeon.py (3.8.1)
File Edit Format Run Options Window Help
1 import pygame
2 import sys
3 import random
4 from pygame.locals import *
5
6 # 色の定義
7 WHITE = (255, 255, 255)
8 BLACK = (0, 0, 0)
9 RED = (255, 0, 0)
10 CYAN = (0, 255, 255)
11 BLINK = [(224, 255, 255), (192, 240, 255), (128, 224, 255), (64, 192, 255), (128, 224, 255), (192, 240, 255)]
12
13 # 画像の読み込み
14 imgTitle = pygame.image.load("image/title.png")
15 imgWall = pygame.image.load("image/wall.png")
16 imgWall2 = pygame.image.load("image/wall2.png")
17 imgDark = pygame.image.load("image/dark.png")
18 imgPara = pygame.image.load("image/parameter.png")
19 imgBt1BG = pygame.image.load("image/bt1bg.png")
20 imgEnemy = pygame.image.load("image/enemy0.png")
21 imgItem = [
```

ソースコードをつくる作業のことをコーディングと呼び、これを機械語に翻訳することで、プログラムを作ることができる。

# 静的解析とは？

ビルドを開始すると、まずソースコードの静的解析を行う。この静的解析では文法や構文の誤りを検出しておき、バグを見つけるための処理が行われている。バグがあった場合はここで処理がストップし、コンパイルには進まずバグの検出箇所をお知らせして終わりになる。



静的解析では、命名規約違反、インデントなど書式の問題がバグとして検出できる。すべてのバグがここで検出できるわけではない。

# プリプロセッサ（編集）の役割

プリプロセッサはソースファイルを翻訳できる形に編集・成形する役割がある。



原稿として渡された「ソースコード」に必要な前処理を行う。「前処理済みソースコード」とは、次工程のコンパイラが翻訳できる状態になるように加工されたファイルのこと。

# コンパイラ（校正・翻訳）の役割

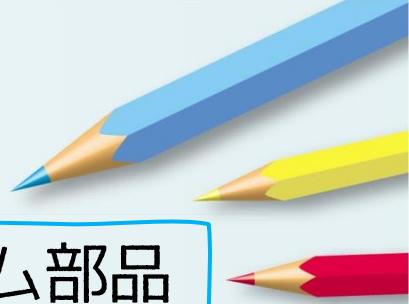
コンパイラはソースコードの文法が正しいかをチェックし、コンピュータが理解できる機械語に翻訳する役割がある。



プリプロセッサから渡される「前処理済みソースコード」を加工し、次工程のリンクに渡す「オブジェクトファイル」を作成する。

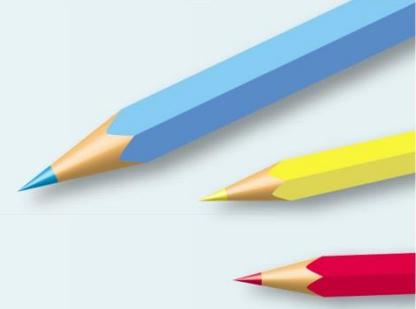
# リンク（製本・出版）の役割

リンクはバラバラに存在する関数や変数といったプログラム部品を、1つに結合しファイルに統合する役割がある。



コンパイラが渡される複数の「オブジェクトファイル」と、外部から提供される「ライブラリファイル」を結合し、最終成果物の「実行ファイル」を生成する。

×モ



# プログラミング教室の テクノロ



なまえ：