

情報科学概論

プログラム

目次

- プログラム
- ウォーミングアップゲーム：
プログラミングの気分を味わってみる
- 復習？CPUと機械語
- プログラミング言語
- コンパイラ
- オープンソースってどういうこと？

プログラム

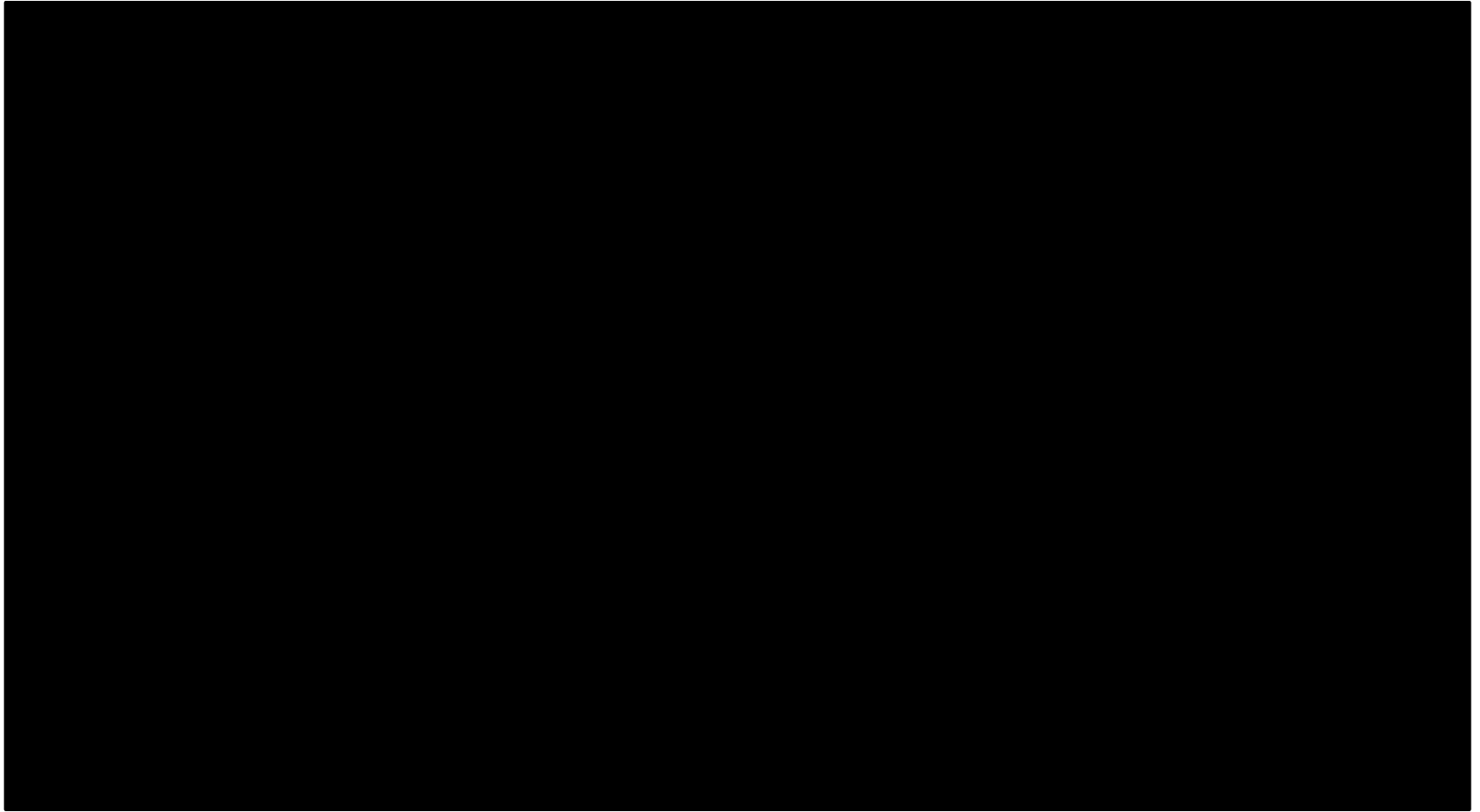
- プログラム：
コンピュータに仕事させるための命令書
- プログラミング：
プログラムを記述すること
- プログラミング言語：
プログラムを記述するための言語

プログラミング
の気分を
味わってみる

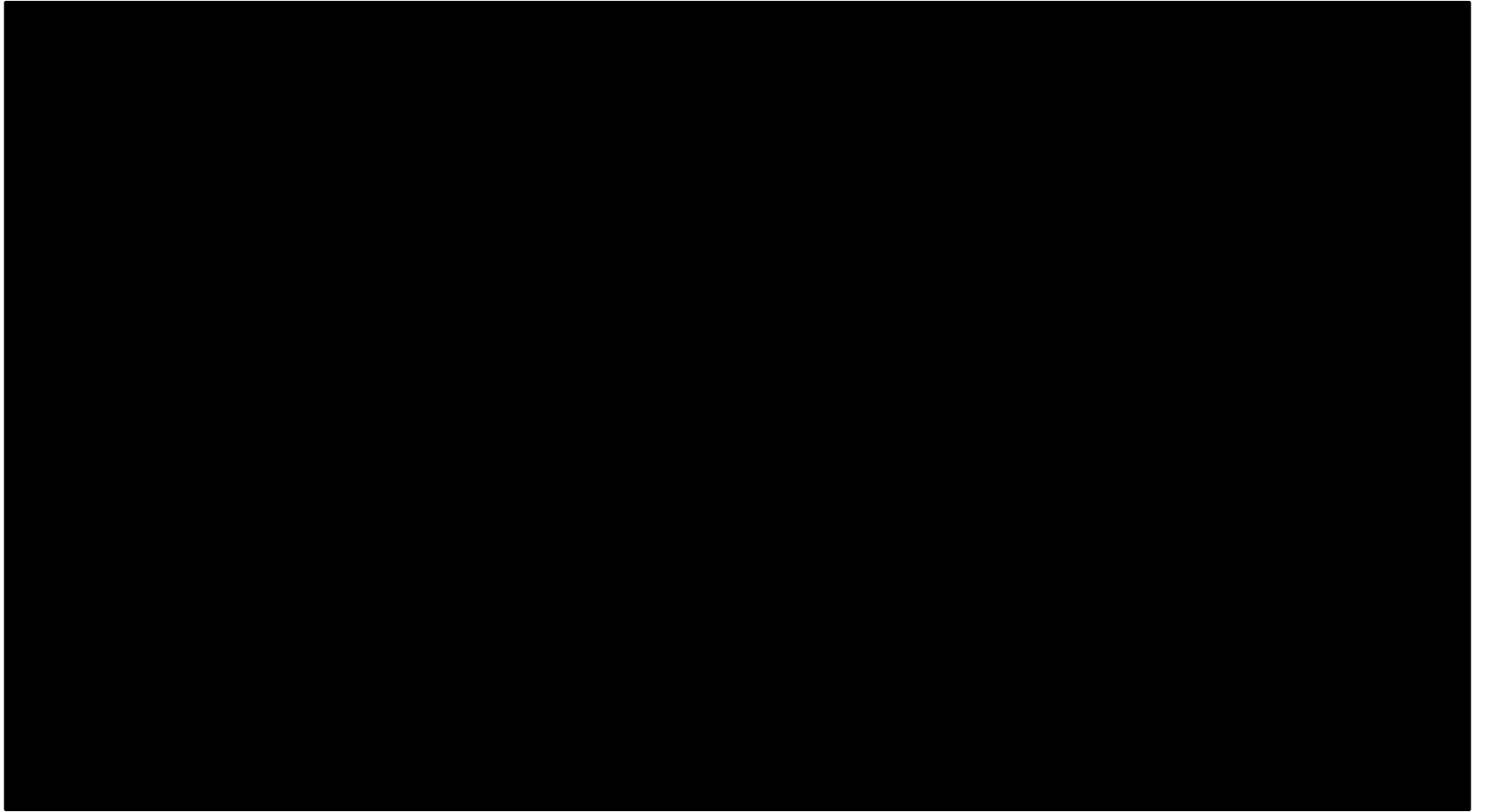
ウォーミングアップゲーム手順

1. 適当に2～3人組になってください
2. 出題役を一人決めてください
3. お題の絵を見せます
 1. 出題役の人はそれをノートかメモに写します
 2. それ以外の人は見ないようにします
4. 出題役がその絵を描く手順を言葉で指示し、ほかの人はそれに従って描いてみましょう
 1. **言葉以外は禁止！ 固有名詞も禁止！**
5. お題の絵がちゃんと描けたら成功！
(出題役を交代して、再挑戦)

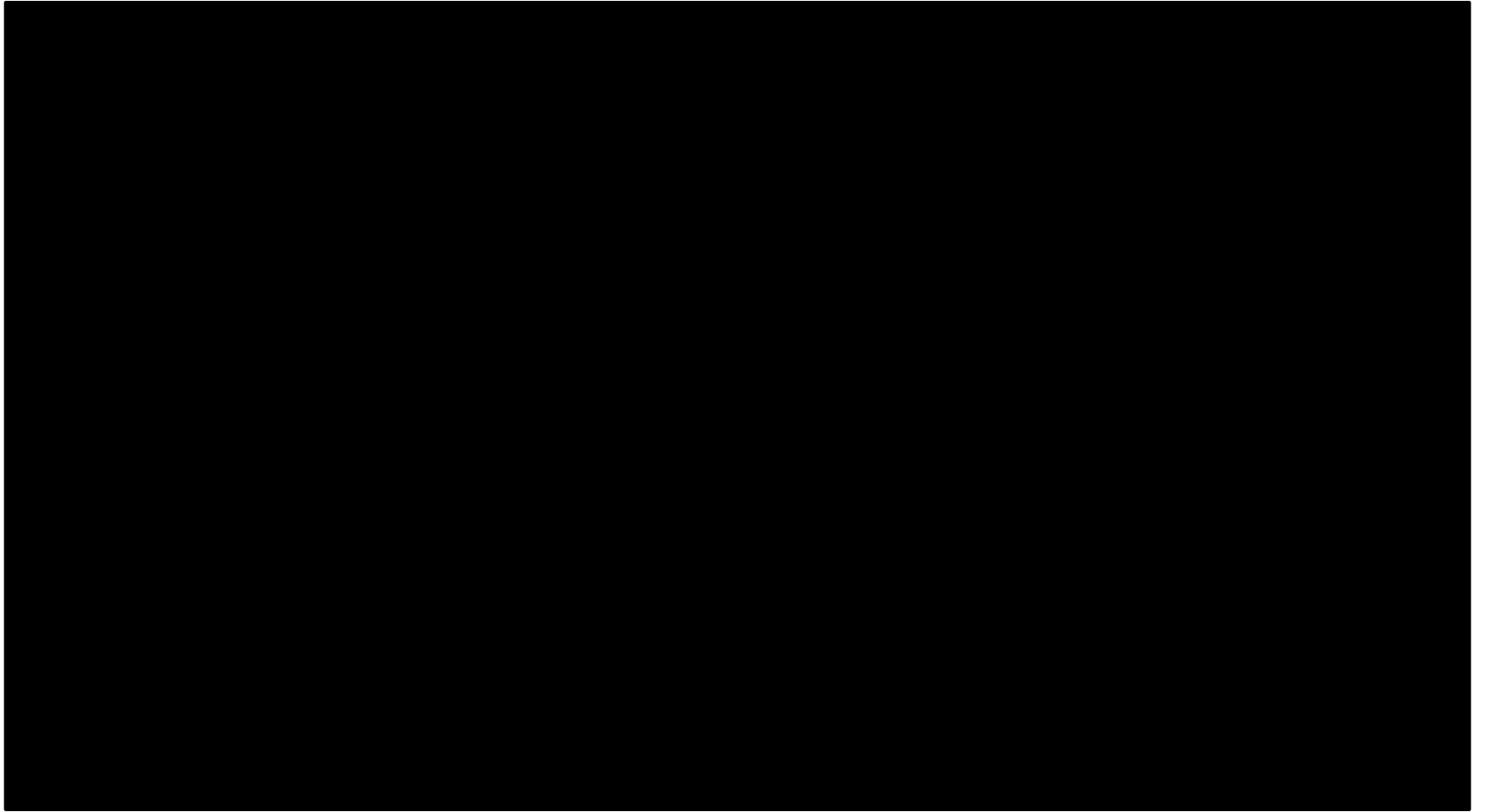
お題1



お題2



お題3



プログラミング

結構、難しい

でも、楽しい

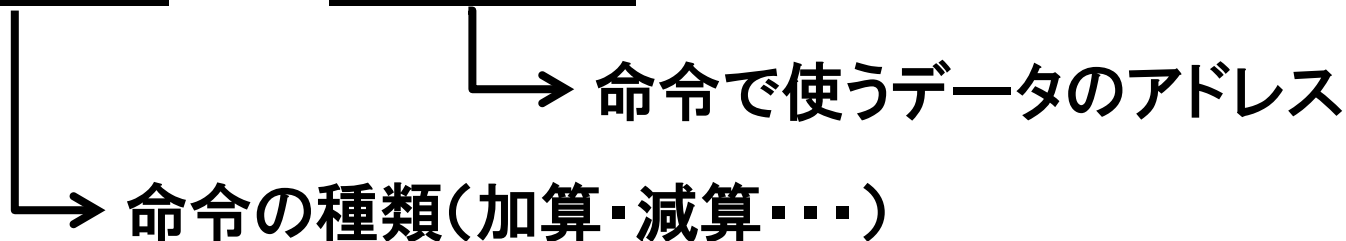
復習？： CPUの基本動作

1. 次に実行する命令をメモリから読み込む
(フェッチ)
2. 読み込んだ命令が何の命令か解読する
(デコード)
3. 実行する

機械語

- CPUが実行する命令を01のパターンで表したものの

- オペコード + オペランド



- CPUの種類が変わると、機械語も変わる

機械語：命令の種類

- 演算命令
 - ▣ 足し算、引き算、掛け算、割り算 etc.
- 入出力命令
 - ▣ メモリ⇔CPUのデータのやりとり
- 分岐命令
 - ▣ プログラムカウンタを書き換える命令
 - ▣ 計算結果によって次に実行する命令を変える

たとえばこんな感じのもの (よくわからない、でOKです)

アドレス	機械語	アセンブリ	意味
0001:	16	ldc.i4.0	0を読み込む
0002:	0A	stloc.0	それを0番地に格納
0003:	16	ldc.i4.0	0を読み込む
0004:	0B	stloc.1	それを1番地に格納
0005:	2B0A	br.s 0011	000e番地に飛ぶ
0006:	06	ldloc.0	0番地の値と
0007:	07	ldloc.1	1番地の値を
0008:	58	add	足す
0009:	0A	stloc.0	それを0番地に格納
000a:	07	ldloc.1	1番地の値と
000b:	17	ldc.i4.1	1を
000c:	58	add	足す
000d:	0B	stloc.1	それを1番地に格納
000e:	07	ldloc.1	1番地の値が
000f:	1F0A	ldc.i4.s 10	10より
0010:	FE04	clt	小さければ
0011:	2DED	brtrue.s 0007	0006番地に飛ぶ
0012:	06	ldloc.0	0番地の値を読み込み、
0013:	28	call	Console.WriteLine を呼び出す

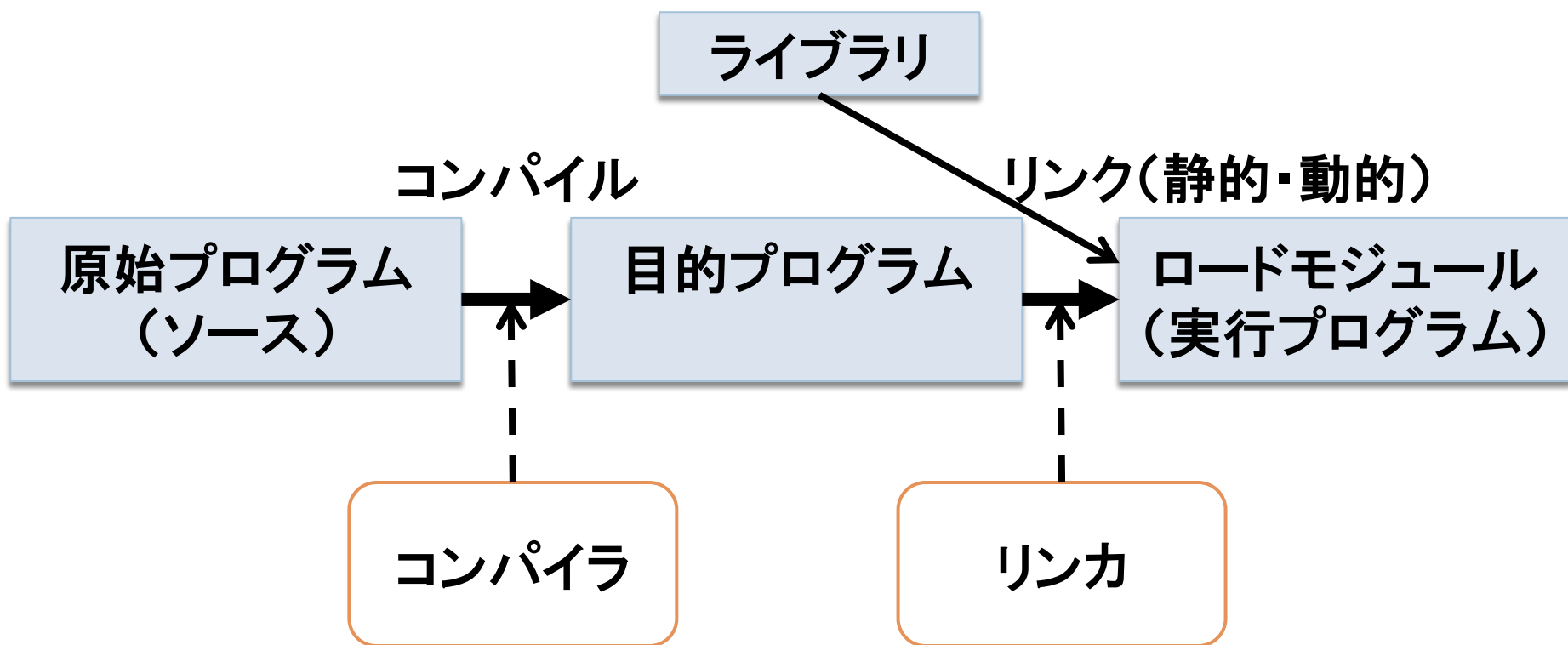
つまり、機械語を書くのはつらい

- ごく簡単な命令しかないのがつらい
 - ▣ 足し算するだけで、読み込んで、読み込んで、足す
- アドレスで書くのがつらい
 - ▣ どこに分岐するか
 - ▣ どこにデータを格納するか

プログラミング言語

- 自然言語(日本語、英語 etc.)よりも厳格な文法
 - 処理 (演算・入出力 etc.)
 - 制御 (条件分岐・繰り返し etc.)
- その他よくある機能
 - 関数 (function)
 - オブジェクト指向 (Object-oriented programming)

コンパイラ



先ほどのプログラムはこんな感じ

```
static void Main(string[] args)
{
    int sum = 0;
    for (int i = 0; i < 10; i++)
    {
        sum += i;
    }
    Console.WriteLine(sum);
}
```

プログラミング言語 C#

sum と i は変数
sum に 0 を代入する
i が10未満の間、以下を繰り返す
 sum に i を足す
 i を 1 増やす
sum を印字する

日本語で書くとこんな感じ

プログラマのすること

□ プログラミング言語を覚える

1. アルゴリズムを考える（予定：次週のテーマ）
└─→ 目的を達成するための手順・問題の解法
2. バグがないように正確に記述する
└─→ プログラムの誤り

オープンソース

プログラムのソースコードが公開されていると
どうなるのか

比較：ソースコードと実行プログラム

ソースコード(原始プログラム)

- プログラムの設計図
- なくてもプログラムは使える
- プログラマなら容易に読み書きできる

実行プログラム

- プログラムそのもの
- 使っている人は持っている
- 読み解き、改造するのは、プログラマでも困難

なんでオープンにするの？

- 誰かがもっとよくしてくれたら利用者としてうれしい
- ドヤ顔したい
- 就職したい
- 会社を宣伝したい
- 公共の福祉のために奉仕したい
- 使っているライブラリの利用規約にそうしろと書いてあったから

オープンソースにも流派がある

強い制約のあるオープンソース

- このライブラリを使ってプログラムを書いたら、それもオープンにしなければだめ！
- 有名なのはGPLというライセンス

制約の緩いオープンソース

- 自由にお使いください
- 最近はこちらが多い（ような気がします）