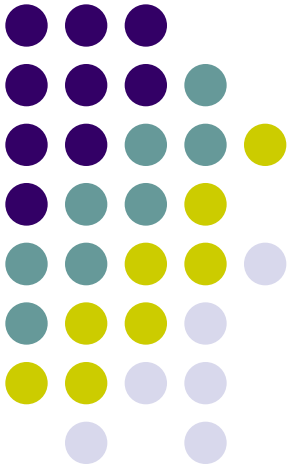
プログラミングにおける見やすいコーディング方法



見やすいソースを作るメリット

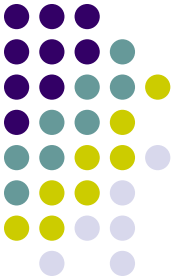
- プログラムは一度作ったらおしまい、ではない
- ある程度大きなプログラムになると、プログラム作成に何日もかかる

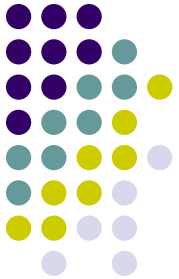
- 後で見直したとき、見やすいソースならスムーズに読むことができる
- 他の人が読みやすいと、アドバイスももらいやすい

プログラムを書くときに最も心がけるのは
安全性でも速度でもなく、**見やすいコード**
（ソースコードがわけが分からなければメンテナンスもできないため）

基本的な使い方

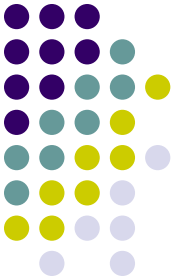
- プロジェクトの作り方
- 実行方法





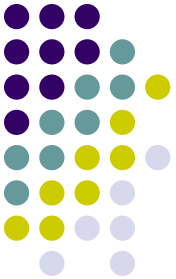
- 字下げを使う
- 改行を使ってコードグループを区別する
- 分かりやすい変数名
- コメント文
- 関数に分割

変数に分かりやすい名前



- １文字の短い名前の場合、変数名をみても何のための変数か、全く分からない
  - 後から利用しようと思っても、どこでどのように使ったか思い出せない、 ということになりやすい
- 短い名前だと、名前の数がかなり少ないので、気づかずに既に使った名前と同じ名前をつけてしまうことにもなりかねない

名前は英語で書こう

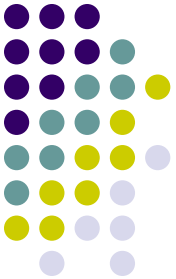


- 1:外国人との共同プロジェクトのため
- 2:日本人の舶来コンプレックス
  - 本来の意味とは違う単語というのもしばしばです
- 3:日本語だと情報量が多いため

命名規則をはっきりさせる

種類	例	規則
定数	MAX_SIZE	すべて大文字で書く
ローカル変数	enemy_pos	全て小文字で書く
関数	EnemyCharacter	単語をはじめに大文字にして単語単位に大文字にする

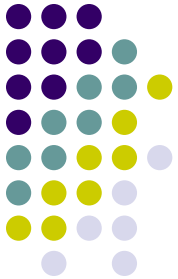
コメントを書く理由



- **プログラム内容の説明**
  - プログラムを書き直す場合や、 新しくプログラムを書くとき、昔書いたプログラムを参考にすることがよくある。しかし、そのころには、どのようなプログラムだったかはたいてい忘れている。コメントが書いてないと、ソースを一行一行確認しないといけない。
  - コメントは書きすぎてもいけない。 全部の行にコメントを書いたりしたら、 コメントが多すぎてかえって読みにくくなる。 関数や処理のまとまりごとに、 何をしているかを書いておくぐらいがいい。

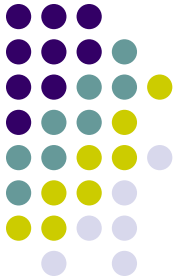
- **ソースの一部をコメントアウト**
  - 作ったプログラムがうまく動かなかったとき、 怪しい個所をとりあえずコメントアウトして、 他の部分でエラーが起きなければ、 今コメントアウトした部分に何か問題があることがわかる

ショートカットキー



Ctrl+F	検索
Ctrl+C	コピー
Ctrl+V	貼り付け
Ctrl+Z	元に戻す
Ctrl+Y	やり直し
Ctrl+}	対応するもう一つの括弧に移動
Ctrl+Shift+}	対応する括弧内を選択
Ctrl+F5	デバッグなしで実行
Ctrl+K+C	選択行のコメント化
Ctrl+K+U	選択行のコメント化解除

デバッグ方法



- ブレークポイントの作成

質問受付