情報メディア基盤ユニット

座標変換、変数の有効範囲、関数

情報メディア学科佐藤尚

今日やること

- プログラムが動かない-Σヽ(`д´;)ノうおおお！
- 座標変換（前回の続き）
- 変数の有効範囲
- 関数その1

動かない理由

- 無限ループ

```
void setup(){
  size(400,400);
}

void draw(){
  background(255);
  fill(128);
  int x = 0;
  while(x >= 0){
    x = x + int(random(10));
    ellipse(x,height/2,10,10);
  }
}
```

- 拡張機能：PDE Xを使うとデバッグ(debug)が楽になる

座標変換

- 座標軸を移動させる
- translate
- rotate
- scale

変数の有効範囲

- 大域変数（グローバル変数）
- 局所変数（ローカル変数）
- 区別：どこで変数宣言されたか？

大域変数(グローバル変数)

- プログラムの先頭部分
- 変数宣言した後は、プログラム中のどこでもOK
- 大域変数の使いすぎには注意

局所変数（ローカル変数）

- ブロックの中で変数宣言

```
for(int i=0;i < 10;i++){
  // 変数iはこの中で有効
}
```

```
void draw(){
  background(255);
  fill(128);
  int x = 0;
  while(x < height){
    x = x + int(random(10));
    ellipse(x,height/2,10,10);
  }
}
```

関数

- 特定の処理を行う受付窓口
- 内部の細かい処理内容を隠す（隠蔽）

プログラム(パンの作成関数)

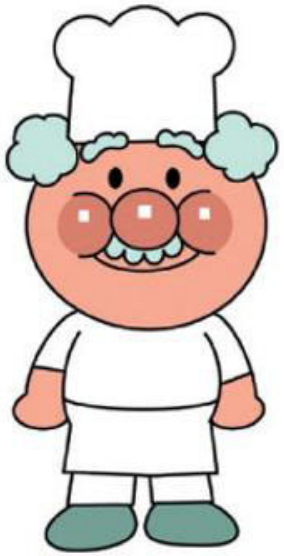
あんぱん

カレーパン

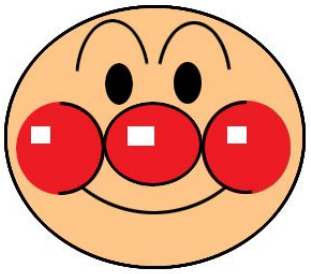
成果物(パン)

関数

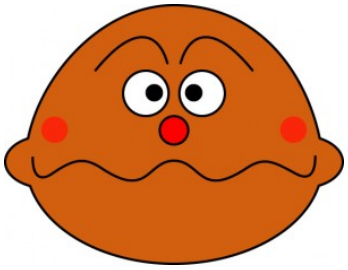
- 特定の処理を行う受付窓口
- 内部の細かい処理内容を隠す（隠蔽）



プログラム（パンの作成関数）

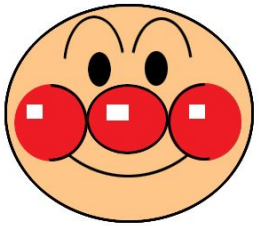


あんパン



カレーパン

あんパンの作り方



こねる

材料を用意
粉などをまぜる
こねる

発酵

生地を整形
放置

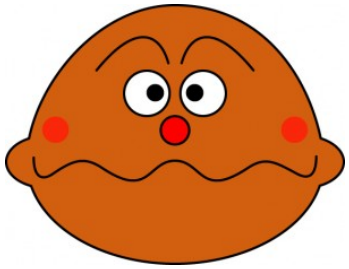
あんパンのたね作り

生地をとる
あんを詰める
丸める

焼く

オーブンを温める
たねを入れる
しばらく放置
取り出す

カレーパンの作り方



こねる

材料を用意
粉などをまぜる
こねる

発酵

生地を整形
放置

あんパンのたね作り

生地をとる
カレーを詰める
丸める

焼く

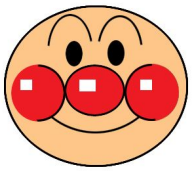
油を入れる
たねをいれれる
しばらく放置
取り出す

関数を使うと

処理の区切りがわかりづらく、重複が多い

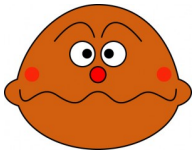
こねる
材料を用意
粉などをまぜる
こねる
発酵
生地を整形
放置
あんパンのたね作り
生地をとる
あんを詰める
丸める
焼く
オーブンを温める
たねを入れる
しばらく放置
取り出す

こねる
材料を用意
粉などをまぜる
こねる
発酵
生地を整形
放置
カレーパンのたね作り
生地をとる
カレーを詰める
丸める
焼く
オーブンを温める
たねを入れる
しばらく放置
取り出す



こねる → 発酵 → あんパンのたね作り → 焼く

発酵 → カレーパンのたね作り → 揚げる



関数による
プログラムの整理

関数：makeあんパン
　こねる
　発酵
　あんパンのたね作り
　焼く

関数：makeカレーパン
　こねる
　発酵
　カレーパンのたね作り
　揚げる

関数：こねる
　材料を用意
　粉を混ぜる
　こねる
関数：発酵
　生地を整形
　放置
関数：あんパンのたね作り
　生地を適量とる
　あんを入れる
　整形
関数：焼く
　オーブンを温める
　たねを入れる
　一定時間待つ
　取り出す
関数：カレーパンのたね作り
　生地を適量とる
　カレーを入れる
　整形
以下略

関数宣言

- 同じ処理はひとかたまりに->関数を利用
- 関数名で使えるもの：変数と同じ

```
void 関数名(){
    処理内容を書く
}

 関数名(); // <-処理内容を実行できる
```

- データを与えて処理を行う

```
void 関数名(データ型名 引数名,...){
    処理内容を書く
}

 関数名(); // <-処理内容を実行できる
```