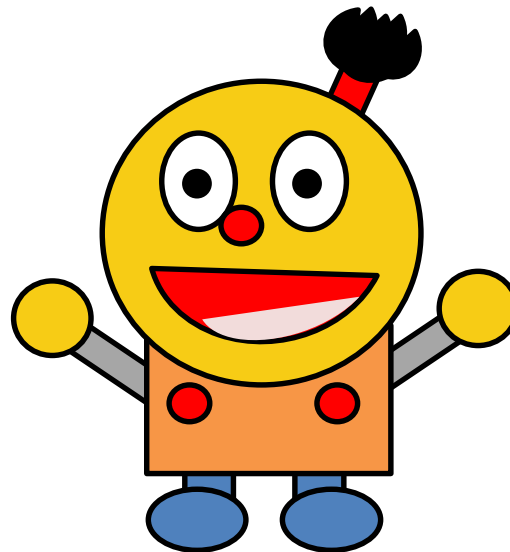
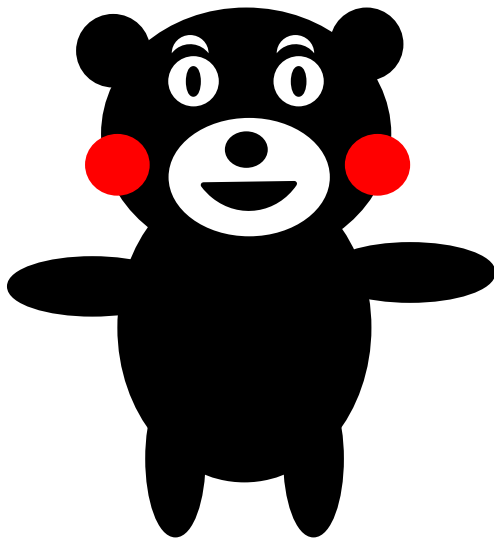


プログラミング演習(2) 課題

• 基本課題1 スケッチ名: chara1

- 何らかのキャラクターをプログラムで描いてください。
- 手、足、頭、黒目・白目があること。
- 標準出力(エディタの下の部分)に組番号と名前を出力せよ

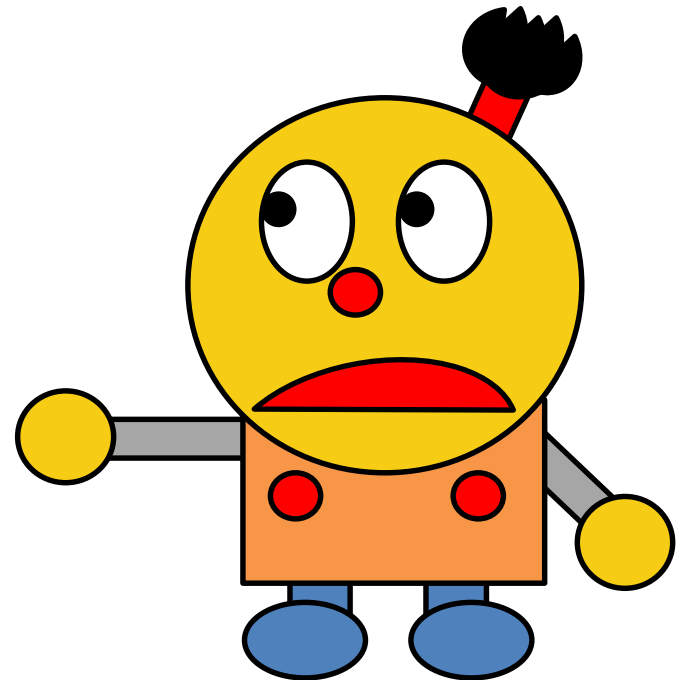
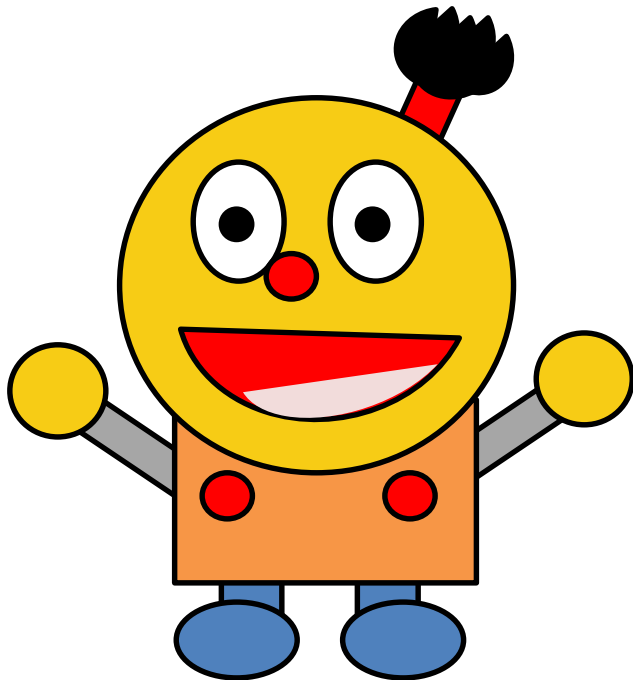
(今後の課題で動かすので必ず保存しておくこと!)



プログラミング演習(2) 課題

• 基本課題2 スケッチ名: chara2

- 課題1のキャラクターの表情とポーズを変えたものを作ってください。
- 基本課題で作成したキャラクターは次週紹介します。



プログラミング演習(2) 課題

- 基本課題3 スケッチ名: logo

- FMSのロゴを考えてプログラムで描いてください。
- ロゴは次週紹介します。また、いいロゴがあれば学科サイトで使うかも？

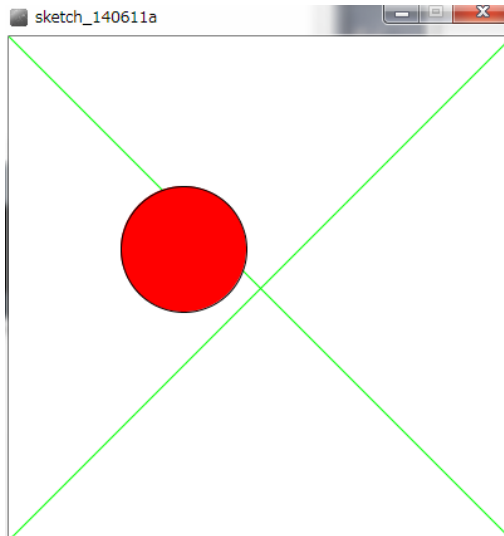
例)

FMS

プログラミング演習(2) 課題

• 基本課題4 スケッチ名: error

- 本プログラムは400x400のウィンドウ上に背景を白色で、ウィンドウの左上から右下、右上から左下にそれぞれ緑色の線を引き、マウスカーソルを中心としてまで赤色で塗りつぶした円(黒縁)を描くプログラムである。しかしエラーばかり表示されて動かない。プログラムの問題を修正し、動作するようにせよ。プログラムはDesktopに配布します。



```
void setapp {  
  size( 400 400 ):background( 255 );}  
  
void drau {  
  LINE( 0 0 400 400 );LINE( 400 400 0 0 );  
  fill( 255 0 0 );  
  sutroku( 0 );  
  ellipse ( mouseX mouseY 50 50 );}
```

プログラミング演習(2) 課題

- 発展課題1 スケッチ名: sign
 - 男女のトイレのサインを並べて描け
 - 多くの人(できれば海外の人)も男女間違わないように工夫せよ

今日の上級テクニック

- arc命令で角度を指定するとき、単位は【ラジアン】。でも単位を【度】で指定したい！ ということは *radians()* という変換命令を使おう。

```
arc( 100, 100, 30, 50, radians(0), radians(180) );
```

0度をラジアンに 変換して指定	180度をラジアン に変換して指定
--------------------	----------------------

- arc命令には7番目の隠しパラメータがある！
以下の3つの結果を比べてみよう。

```
arc(50, 50, 80, 80, 0, 2.5 );
```

```
arc(50, 50, 80, 80, 0, 2.5, CHORD );
```

```
arc(50, 50, 80, 80, 0, 2.5, PIE );
```

今日の上級テクニック

- 図形を移動させる時には translate を使う。translate によって座標系がどんどん移動していく。
- 同じ図形を描画していても座標系が移動していくので、平行移動した図形が描画できる！
- 座標系を戻すのを忘れないように注意！

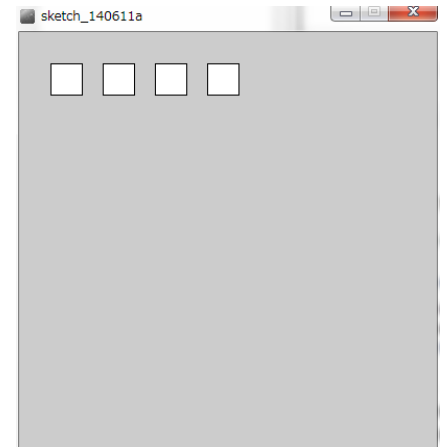
```
translate( 50, 30 );
```

座標を +50, +30 する

```
translate( -50, -30 );
```

座標を -50, -30 する

```
size( 400, 400 );  
rect( 30, 30, 30, 30 );  
translate( 50, 0 );  
rect( 30, 30, 30, 30 );  
translate( 50, 0 );  
rect( 30, 30, 30, 30 );  
translate( 50, 0 );  
rect( 30, 30, 30, 30 );
```



今日の上級テクニック

- 図形を回転させる時には rotate を使う。回転は原点 (0, 0) を中心にして時計回りに回転。
- 角度はラジアンで指定するが、度で指定したい場合は radians() で角度変換をする。
- 角度変換は維持されるので戻す場合は rotate で！

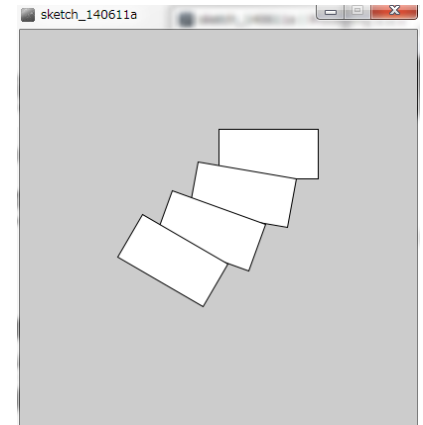
```
rotate( PI/6 );
```

π/6 だけ回転

```
rotate( radians( 30 ) );
```

30度をラジアン
に変換して回転

```
size(400,400);  
rect( 200, 100, 100, 50 );  
rotate( radians(10) );  
rect( 200, 100, 100, 50 );  
rotate( radians(10) );  
rect( 200, 100, 100, 50 );  
rotate( radians(10) );  
rect( 200, 100, 100, 50 );
```

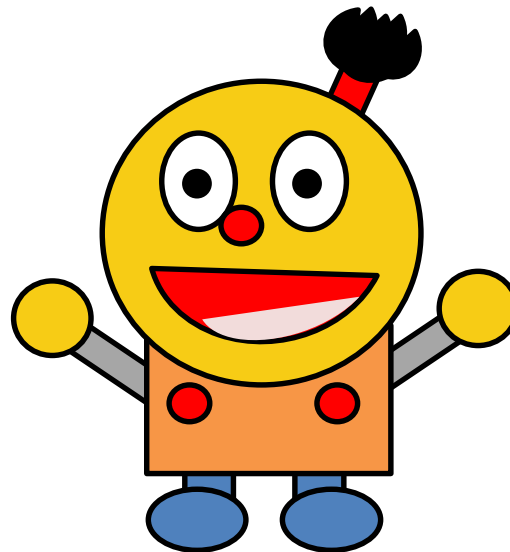
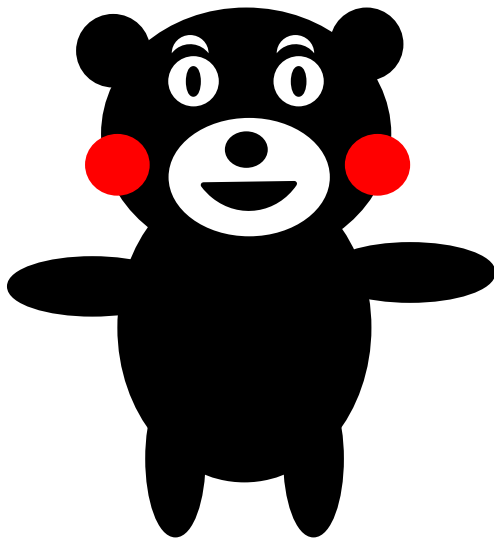


プログラミング演習(2) 課題

• 基本課題1 スケッチ名: chara1

- 何らかのキャラクターをプログラムで描いてください。
- 手、足、頭、黒目・白目があること。
- 標準出力(エディタの下の部分)に組番号と名前を出力せよ

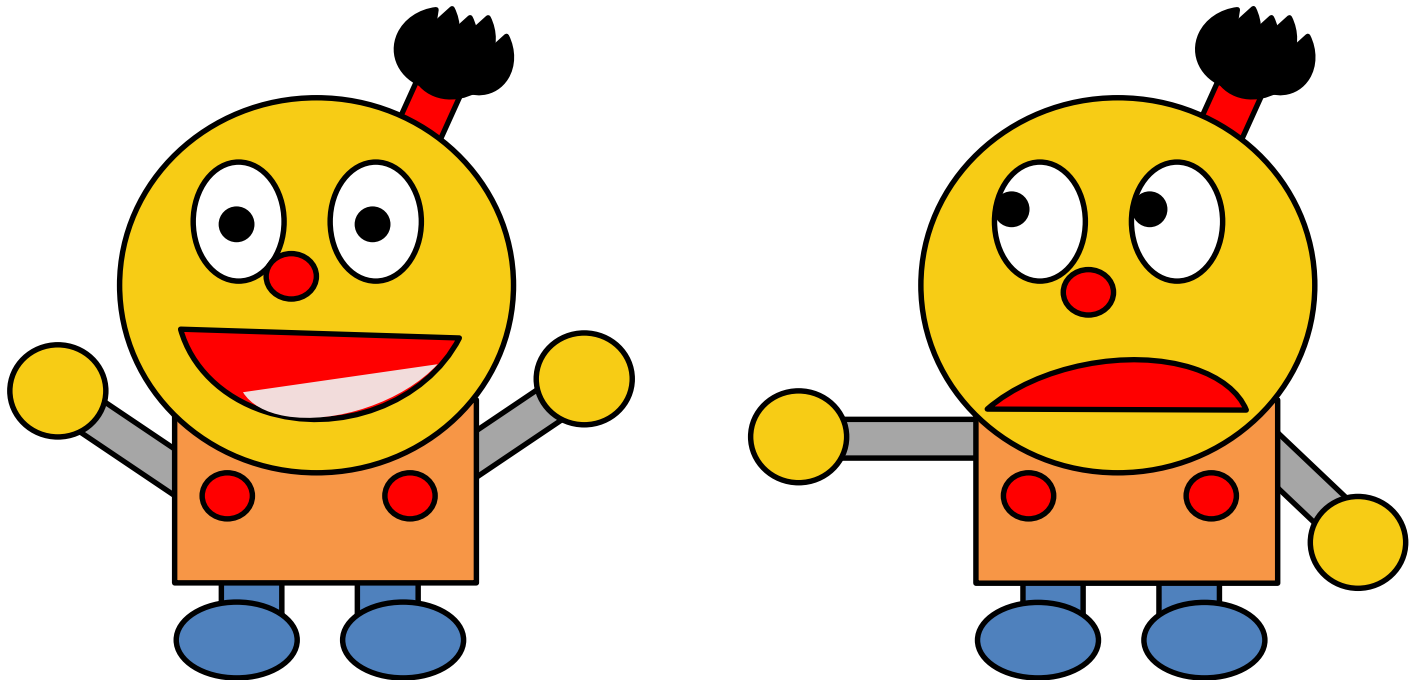
(今後の課題で動かすので必ず保存しておくこと!)



プログラミング演習(2) 課題

• 基本課題2 スケッチ名: chara2

- 課題1のキャラクターの表情とポーズを変えたものを作ってください。
- 基本課題で作成したキャラクターは次週紹介します。



プログラミング演習(2) 課題

- 基本課題3 スケッチ名: logo

- FMSのロゴを考えてプログラムで描いてください。
- ロゴは次週紹介します。また、いいロゴがあれば学科サイトで使うかも？

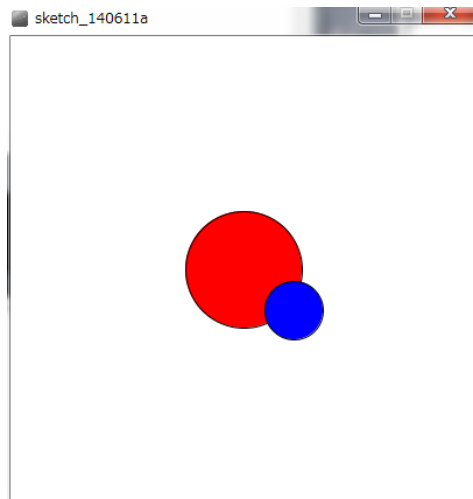
例)

The image shows the letters 'FMS' in a very bold, blocky, and stylized font. The 'F' has a thick vertical stem and a horizontal top bar. The 'M' is composed of thick vertical stems and a wide, flat top bar. The 'S' is also very thick, with a vertical stem on the left and a horizontal top bar that curves into a vertical stem on the right, which then curves back into a horizontal bottom bar. The overall appearance is that of a custom, heavy-duty typeface.

プログラミング演習(2) 課題

• 基本課題4 スケッチ名: error

- 本プログラムは400x400の背景白色のウィンドウを用意し、中央に赤色で塗りつぶした半径50の円を、マウスカーソルの下に半径25の青色で塗りつぶした円を描くプログラムである。また赤色の円に比べ青色の円を手前に表示したい。しかしエラーばかり表示されて動かない。プログラムの問題を修正し、正しく動作するようにせよ。プログラムはDesktopに配布します。



```
void setup {  
  size( 400, 400 )  
  background( 255 ) }  
void draw {  
  fill( 0, 0, 255 )  
  fill( 255, 0, 0 )  
  ellipse( mouseX, mouseY, 50, 50 )  
  ellipse( 200, 200, 100, 100 ) } }
```

プログラミング演習(2) 課題

• 発展課題1 スケッチ名: sign

- エレベータの開くボタンと閉じるボタンを並べて描け
- なるべく開閉が一目で識別できるようにし, 多くの人
が間違わないように工夫せよ

今日の上級テクニック

- arc命令で角度を指定するとき、単位は【ラジアン】。でも単位を【度】で指定したい！ ということは *radians()* という変換命令を使おう。

arc(100, 100, 30, 50, *radians(0)*, *radians(180)*);

0度をラジアンに
変換して指定

180度をラジアン
に変換して指定

- arc命令には7番目の隠しパラメータがある！
以下の3つの結果を比べてみよう。

arc(50, 50, 80, 80, 0, 2.5);

arc(50, 50, 80, 80, 0, 2.5, *CHORD*);

arc(50, 50, 80, 80, 0, 2.5, *PIE*);

今日の上級テクニック

- 図形を移動させる時には `translate` を使う。`translate` によって座標系がどんどん移動していく。
- 同じ図形を描画していても座標系が移動していくので、平行移動した図形が描画できる！
- 座標系を戻すのを忘れないように注意！

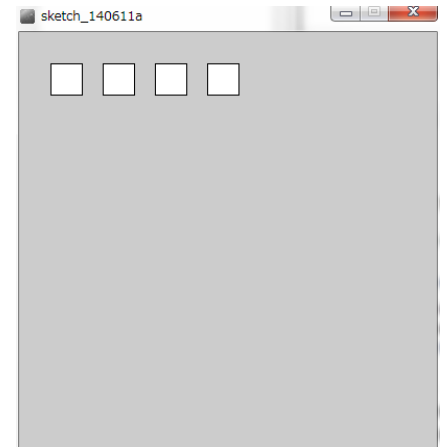
```
translate( 50, 30 );
```

座標を +50, +30 する

```
translate( -50, -30 );
```

座標を -50, -30 する

```
size( 400, 400 );  
rect( 30, 30, 30, 30 );  
translate( 50, 0 );  
rect( 30, 30, 30, 30 );  
translate( 50, 0 );  
rect( 30, 30, 30, 30 );  
translate( 50, 0 );  
rect( 30, 30, 30, 30 );
```



今日の上級テクニック

- 図形を回転させる時には rotate を使う。回転は原点 (0, 0) を中心にして時計回りに回転。
- 角度はラジアンで指定するが、度で指定したい場合は radians() で角度変換をする。
- 角度変換は維持されるので戻す場合は rotate で！

```
rotate( PI/6 );
```

π/6 だけ回転

```
rotate( radians( 30 ) );
```

30度をラジアン
に変換して回転

```
size(400,400);  
rect( 200, 100, 100, 50 );  
rotate( radians(10) );  
rect( 200, 100, 100, 50 );  
rotate( radians(10) );  
rect( 200, 100, 100, 50 );  
rotate( radians(10) );  
rect( 200, 100, 100, 50 );
```

