#### モバイルシステム演習

Processing

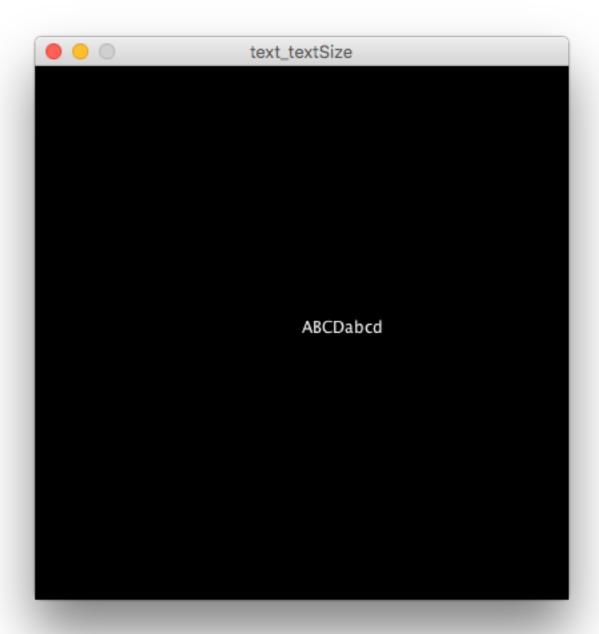
- Typography -

株式会社GOCCO.

watanabe@gocco.co.jp

# 文字を使う

```
size (400, 400);
background(0);
// y座標は文字のベースライン
text("ABCDabcd", width / 2, width / 2);
```



#### 文字サイズの変更

```
size (400, 400);
background(0);
// y座標は文字のベースライン
text("ABCDabcd", width / 2, width / 2);

textSize(20);
text("ABCDabcd", width / 2, width / 2 + 20);

textSize(40);
text("ABCDabcd", width / 2, width / 2 + 60);
```

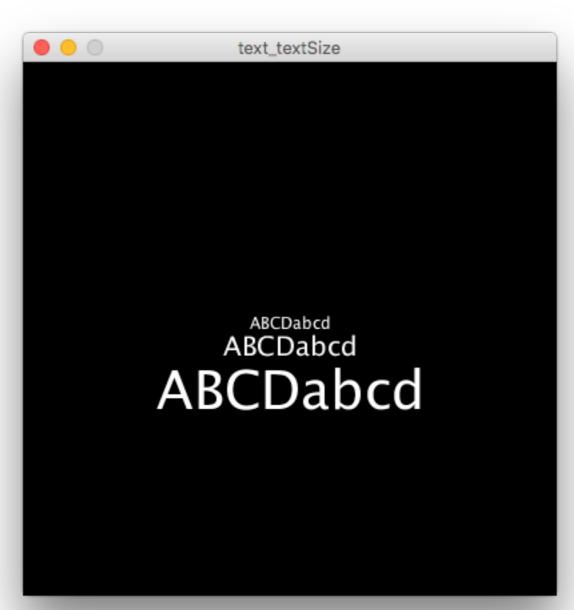


# 文字のAlignmentの変更

```
size (400, 400);
background(0);
// y座標は文字のベースライン
textAlign(CENTER); // LEFT / CENTER / RIGHT
text("ABCDabcd", width / 2, width / 2);

textSize(20);
textAlign(CENTER);
text("ABCDabcd", width / 2, width / 2 + 20);

textSize(40);
textAlign(CENTER);
text("ABCDabcd", width / 2, width / 2 + 60);
```

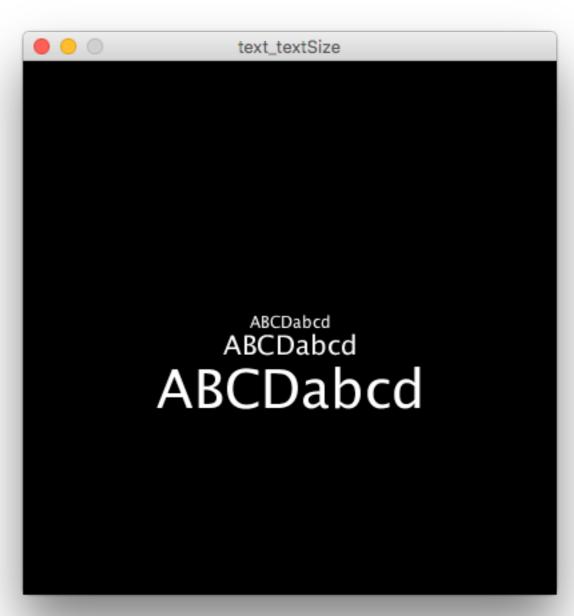


# 文字のAlignmentの変更

```
size (400, 400);
background(0);
// y座標は文字のベースライン
textAlign(CENTER); // LEFT / CENTER / RIGHT
text("ABCDabcd", width / 2, width / 2);

textSize(20);
textAlign(CENTER);
text("ABCDabcd", width / 2, width / 2 + 20);

textSize(40);
textAlign(CENTER);
text("ABCDabcd", width / 2, width / 2 + 60);
```



#### 文字の背景色

```
size (400, 400):
background(0);
String s0 = "Processing";
String s1 = "Processing is flexible.";
String s2 = "Processing is a flexible software.":
textSize(16);
fill(255):
rect(20,24,textWidth(s0),20); //s0の文字列と同じ幅をの四角形を描く
fill(0):
text(s0,20,40);
fill(255,0,0):
rect(20,54,textWidth(s1),20); //slの文字列と同じ幅をの四角形を描く
fill(0):
text(s1,20,70);
fill(200,200,255):
rect(20,84,textWidth(s2),20); //s2の文字列と同じ幅をの四角形を描く
fill(0):
text(s2,20,100);
```



### 文字のフォントデータを作成する

```
size(400,400);
background(255);

String[] fontList = PFont.list();
printArray(fontList);

PFont times_roman;

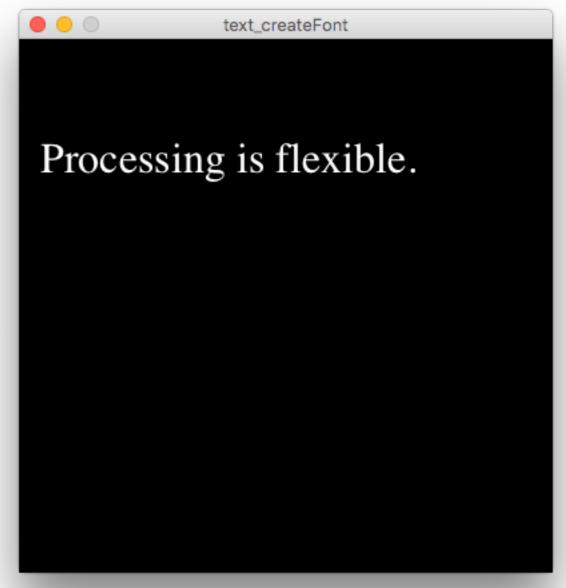
//createFont関数にsketch_006_localfontlistで表示された
//フォント名(Times-Roman)とベースのサイズ (32)を指定して、
//プログラム中で使用するフォントを作成し、times_roman変数に代入する。

times_roman = createFont("Times-Roman",32);

//textFont関数で、描画に適用されるフォントを指定

textFont(times_roman);

fill(0);
text("Processing is flexible.", 16, 100);
```



#### 課題

# マウスの挙動に応じてテキストが変化する デザインを実装してみよう



