Processing のよくあるエラーや間違い集

理工学部　情報科学科

よくあるエラー

- **セミコロンを書き忘れている**

```
size(400, 400)
x = 10
```

セミコロン ";" がありません

- **セミコロンとコロンを書き間違えている**

```
size(400, 400):
```

":" でエラー

- **宣言していない変数を使おうとしている**

```
x = 10;
```

変数 "x" は存在しません

  - 変数を使うためには、必ずその変数を宣言しておかなければいけません。
  - setup(){ の上に、 型　変数名;　で変数を宣言しましょう

- **大文字、小文字を間違えている（変数宣言はしたはずなのに、と言う場合）**

```
int vX;
void setup() {
  size(400, 400)
  vx = 10;
```

変数 "vx" は存在しません

- **メソッド宣言の際に　セミコロンを入れている**

```
void setup();{
```

This method requires a body instead of a semicolon

  - void draw(); { や、void fade();{ なども同様

- **閉じ括弧がない、開き括弧がない**

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

void draw() {

}
```

```
void draw()

}
```

```
size(400, 400;
```

Missing right curly bracket "}"

Missing left curly bracket "{"

  - 括弧　は開いたら必ず閉じましょう。 { を書いたら、すぐに } を書く癖をつけておくと良い。

- **メソッド呼び出しの際の引数の数や値の型が違う**

```
line(0, 0, 100, 100, 20);
```

```
text(100, 20, "hellow");
```

The function "line()" expects parameters like: "line(float, float, float, float)"

The function "text()" expects parameters like: "text(int, float, float)"

  - メソッドの引数の数（入れられる値の数）、引数の順番はメソッドごとに決められています
  - 決められた順番に、決められた型の値を入れましょう

- **メソッド呼び出しの際の引数に何も入れていない**

```
line(0, 0, 100, );
```

  - カンマ（ , ）を一つ多く打ち込んでしまった場合などもこのエラーが出る

- **カンマとピリオドを打ち間違えている**

```
line(0, 0, 100. 0);
```

  - 100 の後の文字が ピリオド （ . ）になっている（引数の数が足りないエラーとなる）。

- **整数型の変数に小数を代入しようとした**

```
int x = 0;
x = 0.5;
```

Type mismatch, "float" does not match with "int"

  - 整数型の変数に小数を代入することは出来ません
  - float x; として、 x の型を変えるか、「値の変換（キャスト）」を行いましょう。

```
x = (int)0.5;
```

（型）値; とすると、(型)の後ろの値をその型に変換できる（エラーにならない）。

- **同じ名前の変数を宣言しようとした**

```
int x = 0;  x = (int)0.5;
int x;
```

Duplicate local variable x

  - 同じ名前の変数を作ることは出来ません（変数のスコープによる例外はある）

- **プログラムは打ち間違えていないのにエラーで実行できない**

sketch_200626a | sketch_2000aa

```
void setup(){
  size(100, 100);
  rect(X, 0, 100, 100);
}
```

  - 一見、分かりにくいが、タブが 2 つ作られている

sketch_200626a | sketch_2000aa

```
int x = 100;
void setup(){
  size(100, 100);
  rect(X, 0, 100, 100);
}
void draw(){
  if( x > 100)
```

  - 別のタブの中にも setup 等があった場合には、setup が同じプログラムの中に 2 つ存在することになってエラーとなる。
  - 「クラス」について良く理解できるまでは、複数のタブを作らないようにすること

- **同じ名前のメソッドが複数宣言されている**

```
void setup() {
  size(100, 100);
}
void setup() {
  size(100, 100);
}
```

  - 一つのプログラムの中に、同じ名前のメソッドを複数宣言できない
- **メソッドの中にメソッドがある**

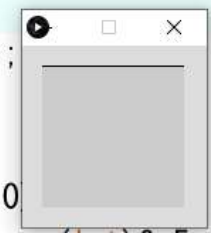
```
void setup() {
  size(100, 100);
void draw() {

}

}
```

  - メソッド宣言の中にメソッドを宣言してはいけない
  - 閉じ括弧の位置を間違えているパターン
  - 深く考えずに別のプログラムからコピー＆ペーストした場合に起こりがち

よくある間違い（エラーは出ないが、期待通りの結果にならないなど）

- **実行画面が小さい**

```
void setuo(){
  size(400,400);
}
void draw(){
  line(0,0,100,0
```



  - void setup(){　を打ち間違えている
  - 打ち間違えたメソッドは実行されないので、size(400,400);の命令も実行されない

- **実行したのに何も表示されない**

```
void setup(){
  size(100,100);
}
void drow(){
  rect(0,0,100,100);
}
```

  - void draw(){ を打ち間違えている

```
void draw(int x){
  rect(0,0,100,100);
}
```

  - void draw(){ を打ち間違えている
  - void draw(){　に引数を設定してしまっている(メソッドのオーバーロードという書き方だが、draw に関してはオーバーロードしてはいけない)。

- **変数宣言した、x,y,z を大文字で打ち間違えた**

```
int x = 100;
void setup(){
  size(100,100);
  rect(X,0,100,100);
```

  - 大文字の ＸＹＺ は特殊な変数になっていて、宣言しなくても参照出来てしまう
    - Ｘ は 0, Ｙ は 1, Ｚ は 2 が最初から代入されている（別の値を代入しようとするとエラーになる）
  - 大文字と小文字は全く別の変数なので、エラーが出なくても打ち間違えには十分注意すること

- **if 文の中の処理が実行されない、繰り返しの中の処理が実行されない**

```
if( x > 100);{
  println("xは100より大きい");
}
```

  - if 文の() の後に（余計な）セミコロンが入力されている
  - このプログラムを書き換えると↓になる。if 文の条件を満たした場合は何もしない、との命令になってしまうため、期待通りの処理が実行されなくなる

```
if( x > 100)
;
{
  println("xは100より大きい");
}
```

  - for 文(繰り返し)でも同様

```
for(int i = 0; i < 10; i++);{
  println("i:" + i);
}
```

  - この場合は、for 文の外でループ変数の i を使おうとしてエラーになっている