社会と情報
プログラミング超入門③

前回の復習

- 変数を使うことでアニメーションを表現できる

変数定義エリア

```
let x=250;
let vx=5;
let y=250;
let vy=10;
let r=20;
```

描画エリア

```
x=x+vx;
y=x+vy;
setColor("red");
drawCircle(x,y,r);
```

前回の復習

変数定義エリア

```
let x=250;
let vx=5;
let y=250;
let r=20;
```

キャンバスのリセット

描画エリア

```
x=x+vx;
setColor("red");
drawCircle(x,y,r);
```

xの値をx+vxにし（xをvxだけ増やし）、
(x,y)を中心に半径rの円を描く

（復習）アルゴリズムの３構造

順次（順接）

・Ａをしたら次にＢをする

分岐

・もし〇〇ならＡをする（そうでなければＢをする）

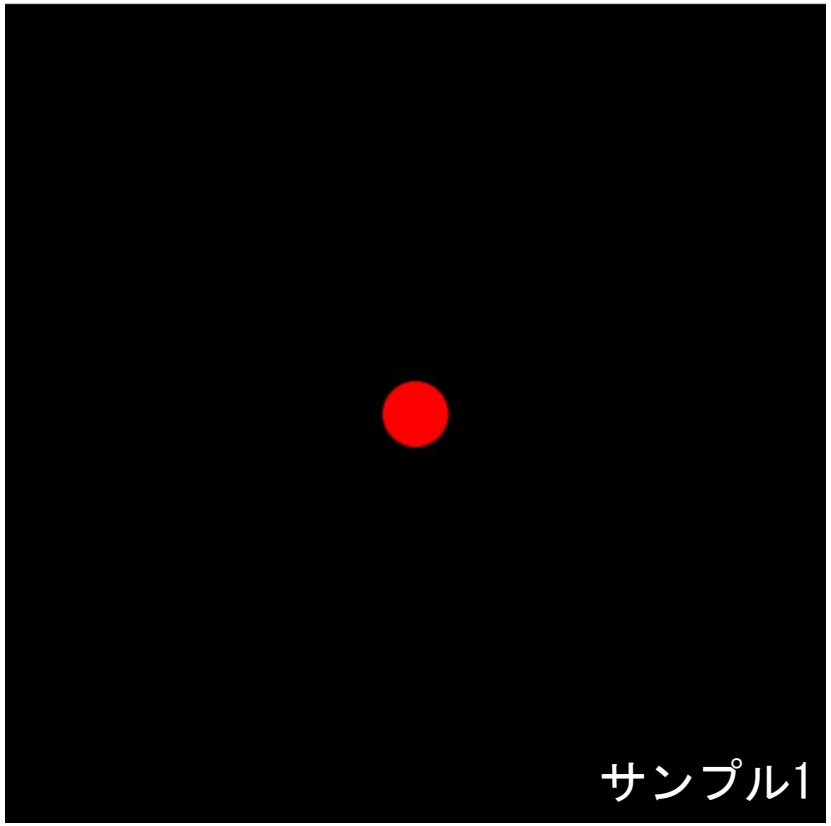
反復（繰り返し、ループ）

・〇〇である限りずっとAを繰り返す

すべてこの３つで
表現できる

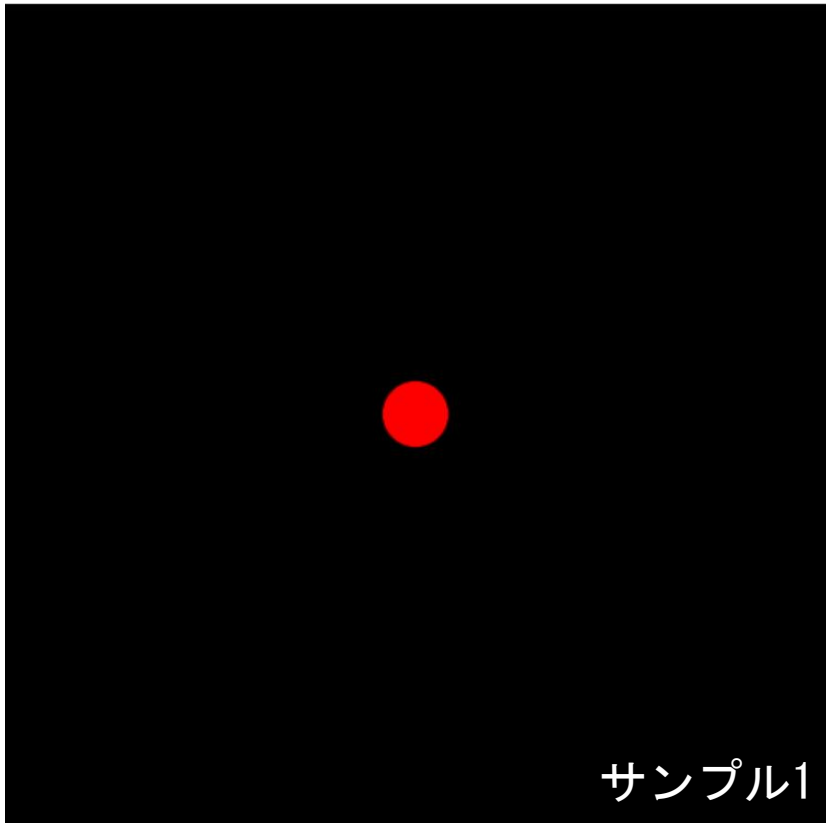
今回の目標

- 分岐を使い、図形が壁で跳ね返るアニメーションを作れるようになる



分岐

- 「もし〇〇ならＡをする（そうでなければＢをする）」というもの
- 今回は「もし壁に当たっていたら跳ね返らせる（そうでなければ何もしない）」という処理を実現させる



分岐の表現

```
if(条件式1){
   条件式1が成り立っているときの処理;
}
else if(条件式2){
   条件式1は成り立たず条件式2は成り立っているときの処理;
}
else{
   条件式も条件式2も成り立っていないときの処理;
}
```

条件式（比較演算子）

比較	表記
aとbは等しいか	a == b
aとbは等しくないか	a != b
aはb以上か	a >= b
aはbより大きいか	a > b
aはb以下か	a <= b
aはbより小さいか	a < b

壁での跳ね返り（準備）

変数定義エリア

```
let x=250;
let y=250;
let vx=5;
let vy=10;
let r=20;
```

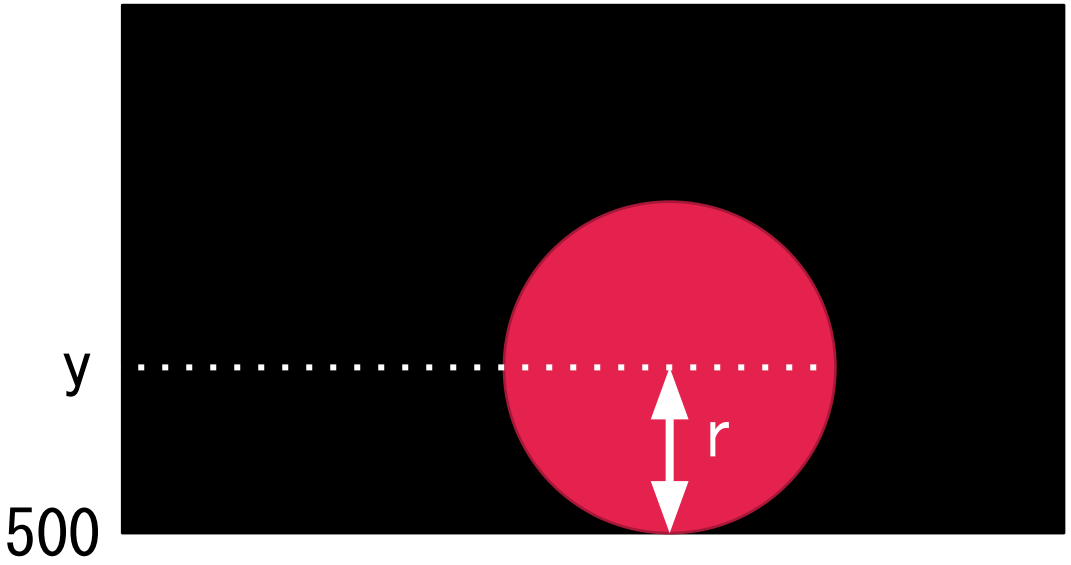
描画エリア

```
x=x+vx;
y=y+vy;
setColor("red");
drawCircle(x,y,r);
```

下の壁に当たっているかの判定

- yが500-r以上ならば当たっている

```
if(y>=500-r){
   当たっているときの処理;
}
```

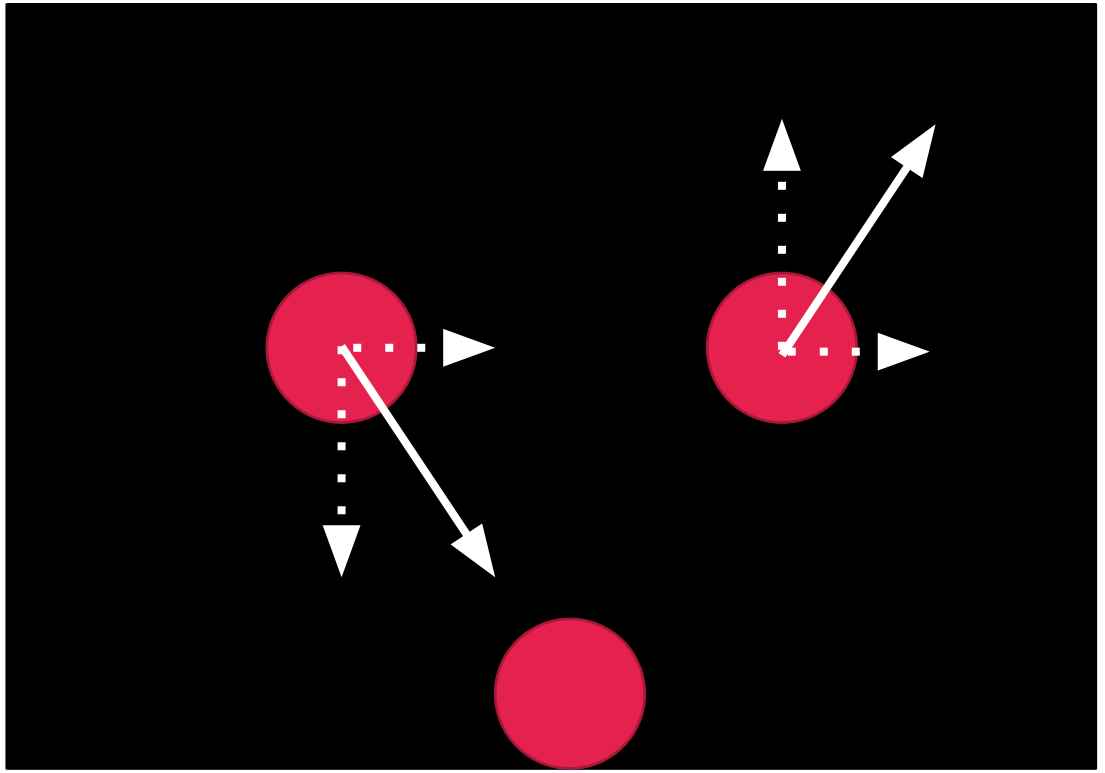


上図の位置かそれより下なら(y座標がこれ以上なら)当たっている

跳ね返ったときの動き

- y方向の速度が逆転する

```
if(y>=500-r){
    vy=-1*vy;
}
```



x方向とy方向に分解して考える

跳ね返りの実装（下の壁のみ）

変数定義エリア

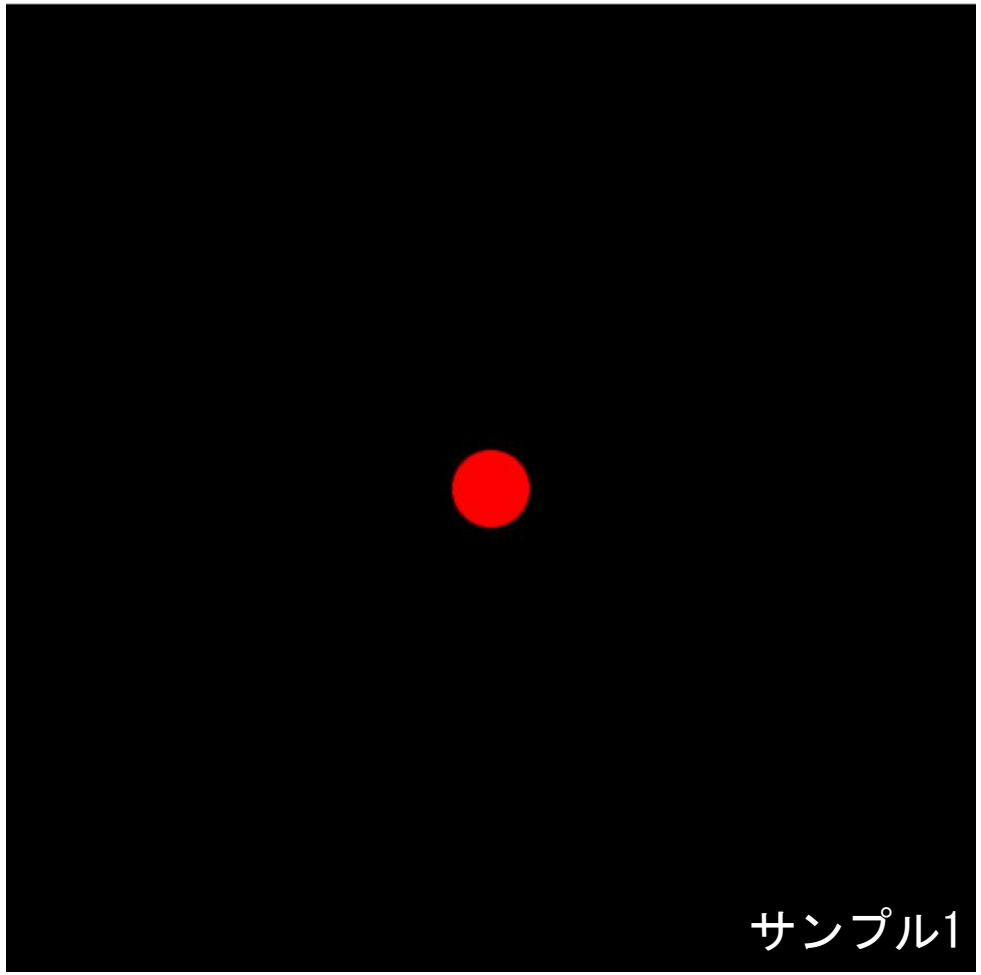
```
let x=250;
let y=250;
let vx=5;
let vy=10;
let r=20;
```

描画エリア

```
x=x+vx;
y=y+vy;
if(y>=500-r){
   vy=-1*vy;
}
setColor("red");
drawCircle(x,y,r);
```

動作の結果

- 下の壁のみ実装したので
  こういった結果になる



変数定義エリア

```
let x=250;
let y=250;
let vx=5;
let vy=10;
let r=20;
```

描画エリア

```
x=x+vx;
y=y+vy;
if(y>=500-r){
   vy=-1*vy;
}
setColor("red");
drawCircle(x,y,r);
```

演習

1. 左のコードを写し、分岐の仕組みと動作について理解する
2. 他の方向で跳ね返る際の処理を作る
3. 【応用】跳ね返った回数を変数countに入れ、display(count);を描画エリアの末尾に追記して画面に表示させる