

```
int x = 6;  
int y = x;  
x = 10;
```

| | |
|--------------|-----|
| x | y |
| 6 | 6 |
| 10 | |

```
int grade = 80;  
if (grade >= 90) {  
    println("A");  
}
```

```
else if (grade >= 80) {  
    println("B");  
}
```

```
else if (grade >= 70) {  
    println("C");  
}
```

```
else {  
    println("Not passing");  
}
```

B

```
int grade = 80;  
if (grade >= 90) {  
    println("A");  
}
```

```
if (grade >= 80) {  
    println("B");  
}
```

```
if (grade >= 70) {  
    println("C");  
}
```

B
C

boolean a = true;
boolean b = false;

a || b true
a && b false
a && !b true

!(!a || b) true

!(a && !b) || !(a || !b) false
false false

< > <= >= == !=

22 / 5 → 4

22 % 4

22 % 6

3
6 | 22
 18

 4

5
4 | 22
 20

 2

While loop

Loops are used when we want to repeat code without rewriting it.

```
while (condition) {
```

```
    // body
```

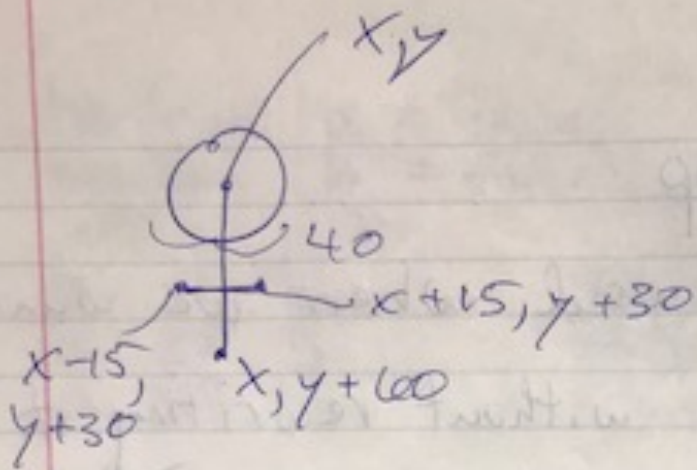
1.) Check condition

2.) If condition is true, execute body, and go back to step 1.

If false, exit while loop block, continue with program.

```
while (true) {
```

* Don't use while loop for performing animations. Use it when you need shapes repeated, or some other repetition.



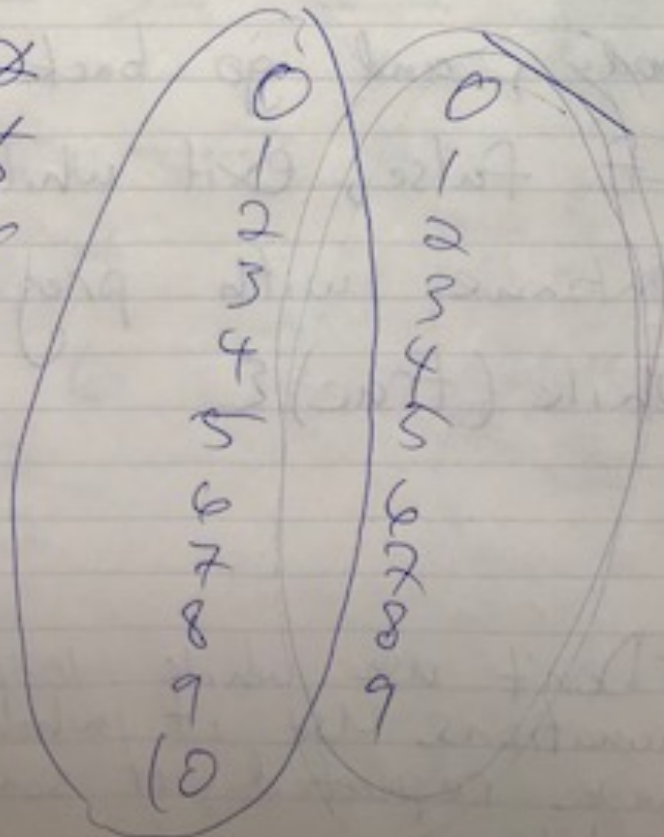
```
int x = 0;
```

```
while (x <= 10) {
```

```
    line(x, x, x+20, x+20);
```

```
    x = x+1;
```

~~x = 0~~ x = 2
 2 4 6
 8 10
 12



```
int x = 0;
while (x < 10) {
```

```
    line(x, x, x+10, x+20);
}
```

Infinite Loop

* When dealing with loops, always save before running program.

$$\sum_{i=0}^{20} i \rightarrow \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

```
int sum = 0;
int i = 0;
while (i <= 20) {
    sum = sum + i;
    i = i + 1;
}
```

I

```
int i = 1;
while (i != 20)
    point(i, i);
    i = i + 2;
}
```

10
infinite

| | | |
|---------------|---------------|---------------|
| 1 | 9 | 17 |
| 2 | 4 | 19 |
| 3 | 10 | 21 |
| 4 | 16 | 23 |
| 5 | 25 | 25 |
| 6 | | 27 |
| 7 | | 29 |
| 8 | | 31 |
| 9 | | 33 |
| 10 | | 35 |
| 11 | | 37 |
| 12 | | 39 |
| 13 | | 41 |
| 14 | | 43 |
| 15 | | 45 |
| 16 | | 47 |
| 17 | | 49 |
| 18 | | 51 |
| 19 | | 53 |
| 20 | | 55 |
| 21 | | 57 |
| 22 | | 59 |
| 23 | | 61 |
| 24 | | 63 |
| 25 | | 65 |

```
int index = 0;
```

initialization
condition

```
while (index < 10) {
```

```
    point(index, index);
```

```
    index = index + 1; — update
```

```
}
```

```
int x = 0;
```

```
void draw() {
```

```
    background(200);
```

```
    while (x <= width) {
```

```
        rect(x, x, 20, 20);
```

```
        x = x + 20;
```

```
    }
```

```
}
```

x = 0;

x
0
~~20~~
~~40~~
~~60~~
~~80~~
~~100~~
~~120~~
0

What is shown by this program in the second frame and beyond?

```

int x = 0;
while (x > 0) {
    rect(x, x, 20, 20);
    x = x + 20;
}

```

⊙ rect.s

```

int x = 0;
if (x < 100) {
    rect(x, x, 20, 20);
    x = x + 20;
}

```

Not a while loop

```

int x = 0;
while (x < 10) {
    rect(x, x, 20, 20);
    x = x + 20;
}

```

initialization

condition

update

```

for (int x = 0; x < 10; x = x + 10) {
    rect(x, x, 20, 20);
}

```

initialization condition update

for(initialization; condition; update) E

{ // body }

1.) Initialization - only once

2.) Check condition

if true, go to step 3
if false, leave for loop

3.) execute for-loop body

4.) Update, go back to step 2

For loop - used when we know how many times we want to iterate.

While loop - We may not know how many times we want to iterate.

99.9% of the time, a while loop and for loop will be interchangeable.

Same as $i = i + 1$

```
for (int i = 0; i < 10; i++) {
```

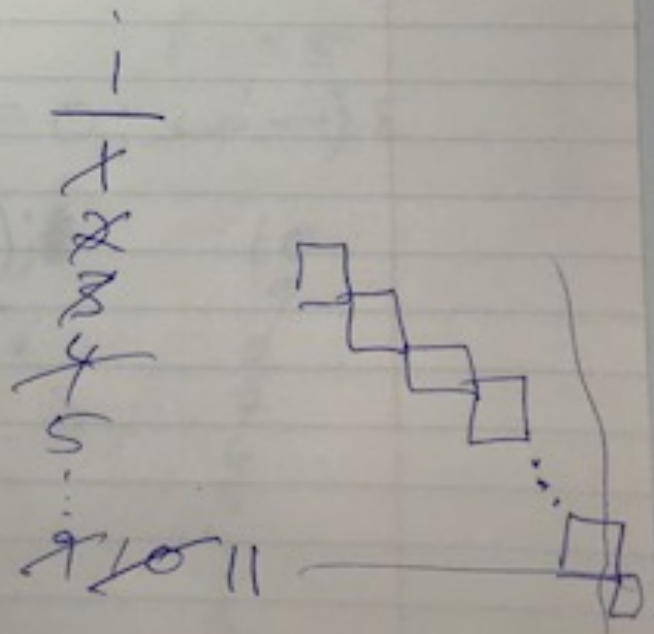
```
    // body
```

```
int i = 0;  
while (i < 10) {
```

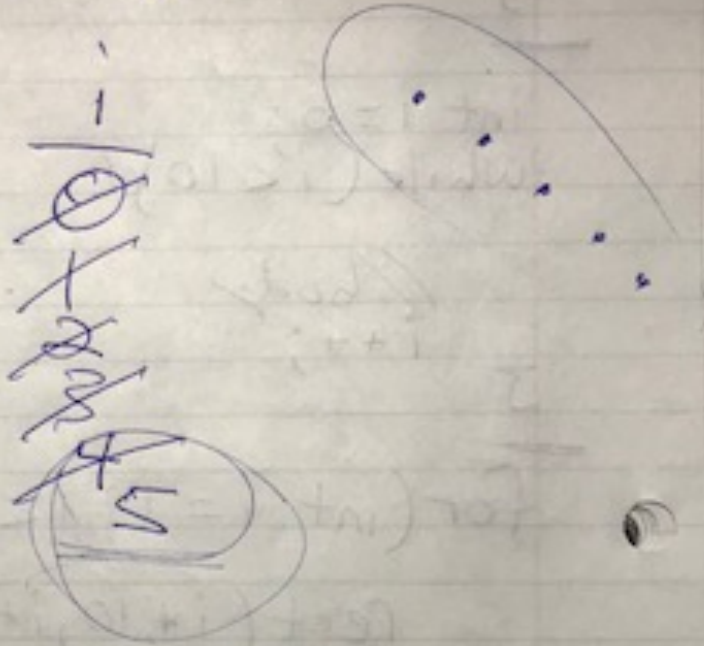
```
    // body  
    i++;
```

```
for (int i = 1; i <= 10; i++) {
```

```
    rect(i * 10, i * 10, 10, 10);
```



```
for(int i=0; i<5; i++){
    point(i*10, i*10);
}
```



int index = 0;

index++; // same as index = index + 1;

index--; // same as index = index - 1;

index += 1; // same as index = index + 1;

index += 5; // same as index = index + 5;

index *= 10; // same as index = index * 10;

index /= 2; // same as index = index / 2;

index -= 3; // same as index = index - 3;

index %= 5; // same as index = index % 5

for(int i = 10; i >= 0; ~~i--~~^{i -= 5}) {

 point(i + 10, i + 10);

 ...
 10
 9
 8
 7
 6
 5
 4
 3
 2
 1
 0