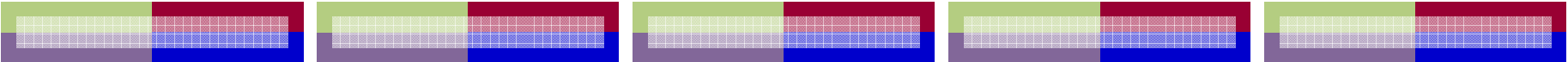


CS101


Problem Solving and Object-Oriented Programming

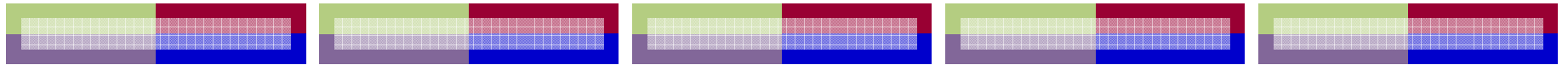
L18: Loops



Anatomy of a While Loop

```
while (penguin.isWiderThan (whichHole)) {           condition
    penguin.turn (LEFT, 1 revolution);
    whichHole.resize (1.1);                          Statements to repeat
}
```

- Evaluate condition
 - If true, execute statements, evaluate condition, execute statements, evaluate condition, etc.
 - If false, skip to statement after while loop
 - Must be some way to update conditional so that loop eventually ends!
- 



While loop example

```
int count = 1, total = 0;
```

```
While (count <= 5){
```

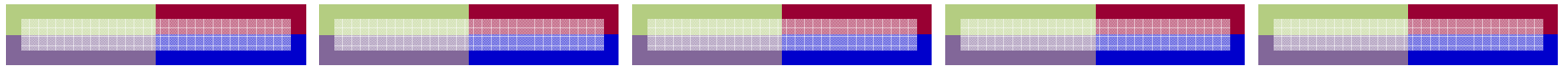
```
    total = total + count;
```

```
    count = count + 1;
```

```
}
```

<u>Iteration</u>	<u>count<=5?</u>	<u>count</u>	<u>total</u>
<i>Before loop</i>	true	1	0
1	true	2	1
2	true	3	3
3	true	4	6
4	true	5	10
5	true	6	15
6	false		





While loop example

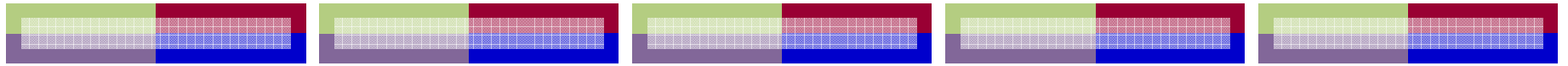
- What if we want to sum the even integers from 1 to 10?
e.g. sum $2+4+6+8+10$

```
int count = 2, total = 0;
```

```
while (count <= 10){  
    total = total + count;  
    count = count + 2;  
}
```

<u>Iteration</u>	<u>count<=5?</u>	<u>count</u>	<u>total</u>
<i>Before loop</i>	true	2	0
1	true	4	2
2	true	6	6
3	true	8	12
4	true	10	20
5	true	12	30
6	false		





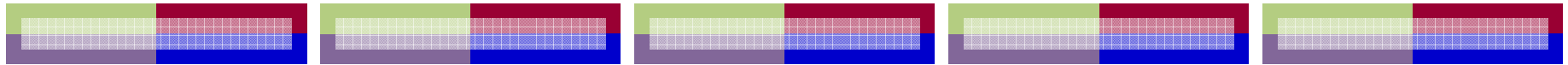
While loop example

- Write a while loop that outputs the numbers from 0-100 by 10's

e.g. 10 20 30 40 50 60 70 80 100

<code>num = 10;</code>	<u>Iteration</u>	<u>num<=100?</u>	<u>num</u>
<code>while (num <= 100){</code>	<i>Before loop</i>	true	10
<code> system.out.println(num</code>	1	true	20
<code>);</code>	2	true	30
<code> num = num+10;</code>	3	true	40
<code>}</code>	4	true	50
	5	true	60
	6	true	70
	7 ...		



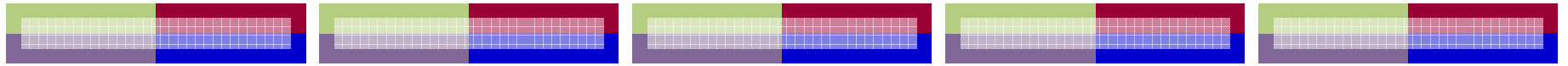


/** Drawing railroad tracks on the canvas:

- On each iteration, draw 1 tie
- Stop when we have reached the right edge of the canvas
- On each iteration, increase tiePosition so that the next tie is drawn further to the right.

```
public void begin () {  
    new FilledRect(0, 0, SCREEN_WIDTH, SCREEN_HEIGHT,  
                  canvas).setColor(GROUND_COLOR);  
    double tiePosition = 5;  
    while (tiePosition < SCREEN_WIDTH) {  
        new FilledRect(tiePosition, TIE_TOP, TIE_WIDTH,  
                      TIE_LENGTH, canvas).setColor(TIE_COLOR);  
        tiePosition = tiePosition + TIE_WIDTH + TIE_SPACING;  
    }  
    new FilledRect(0, TRACK_TOP, SCREEN_WIDTH, RAIL_WIDTH,  
                  canvas).setColor(RAIL_COLOR);  
    new FilledRect(0, TRACK_TOP + GAUGE, SCREEN_WIDTH,  
                  RAIL_WIDTH, canvas).setColor(RAIL_COLOR);  
}
```





Summary

- Keys to implementing a loop
 - What should the loop do?
 - What should happen each time through the loop?
 - Under what condition should the loop continue?
 - What update happens inside the loop to ensure the loop eventually ends?
 - What initialization must happen before reaching the loop?

