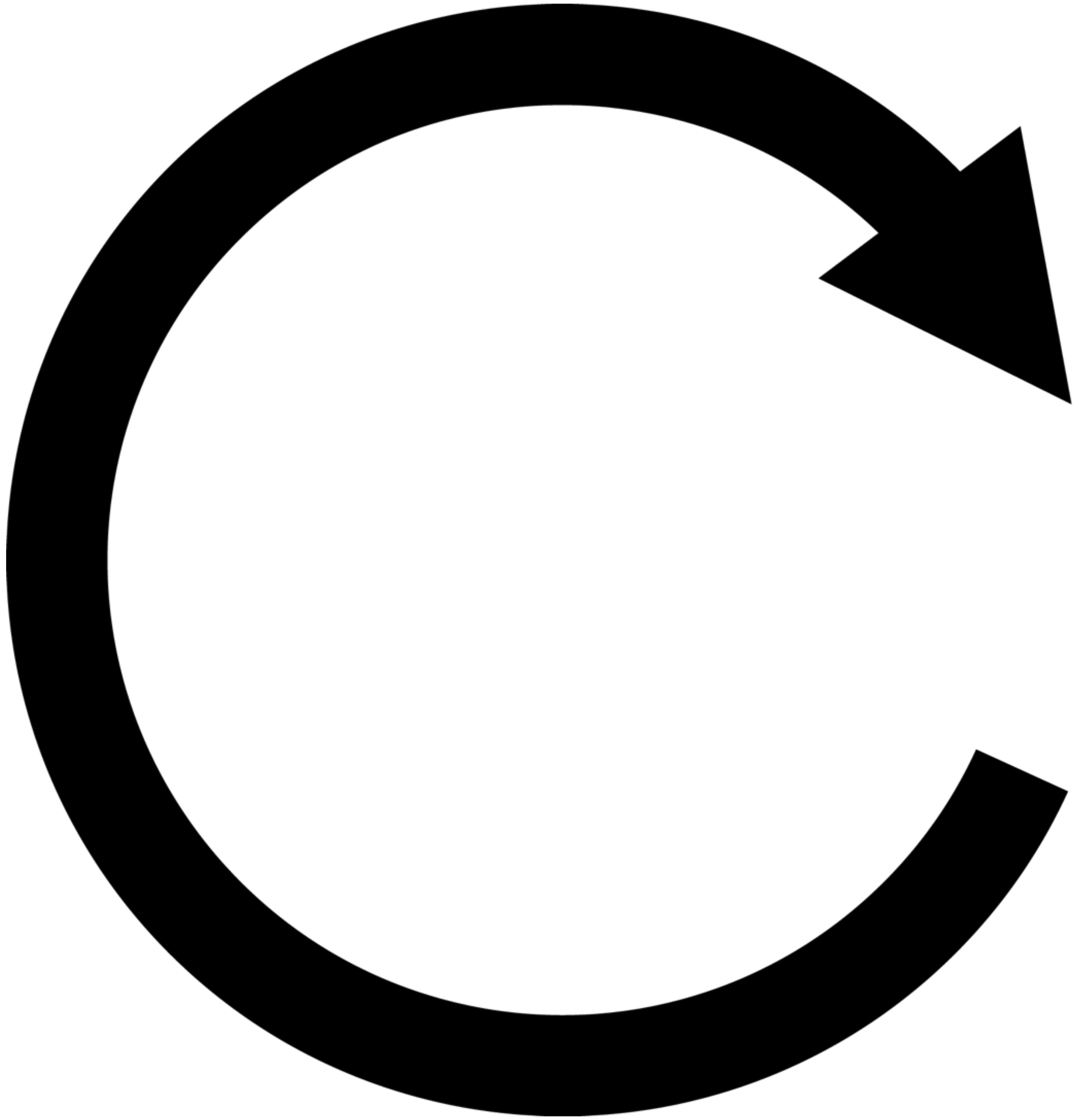
Essential Computing 1

Loops



Three ways of repeating; **while**, **for** and **do-while**.

```
int i = 0;
while( i++ < 100 ){
    // repeat.
}
```

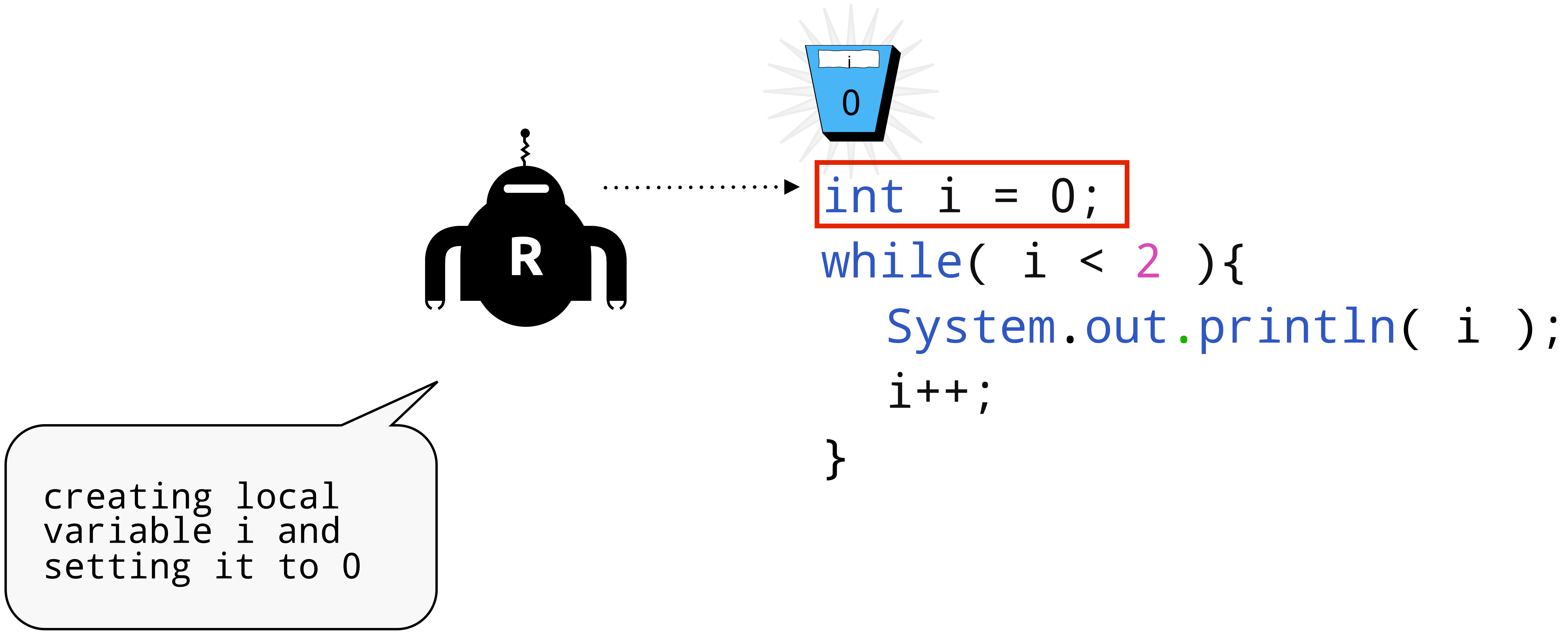
```
for( int j=0; j<100; j++ ){
    // repeat.
}
```

```
int k = 0;
do {
    // Repeat.
} while( k++ < 100 );
```

While loop example

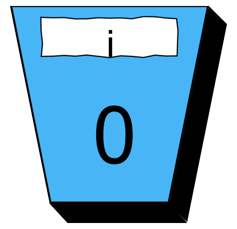
```
int i = 0;
while( i < 2 ){
  // repeat this code
  i++;
}
```

```
int i = 0;
while( i < 2 ){
  System.out.println( i );
  i++;
}
```

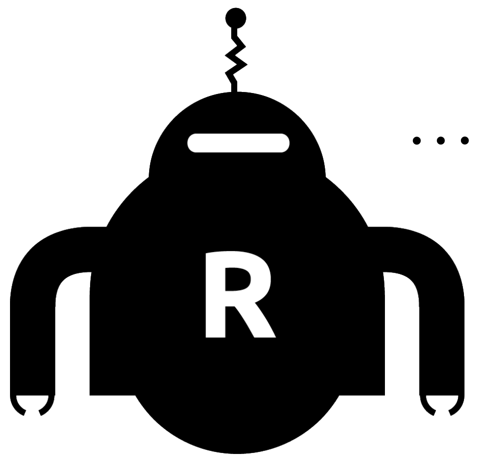


```
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```

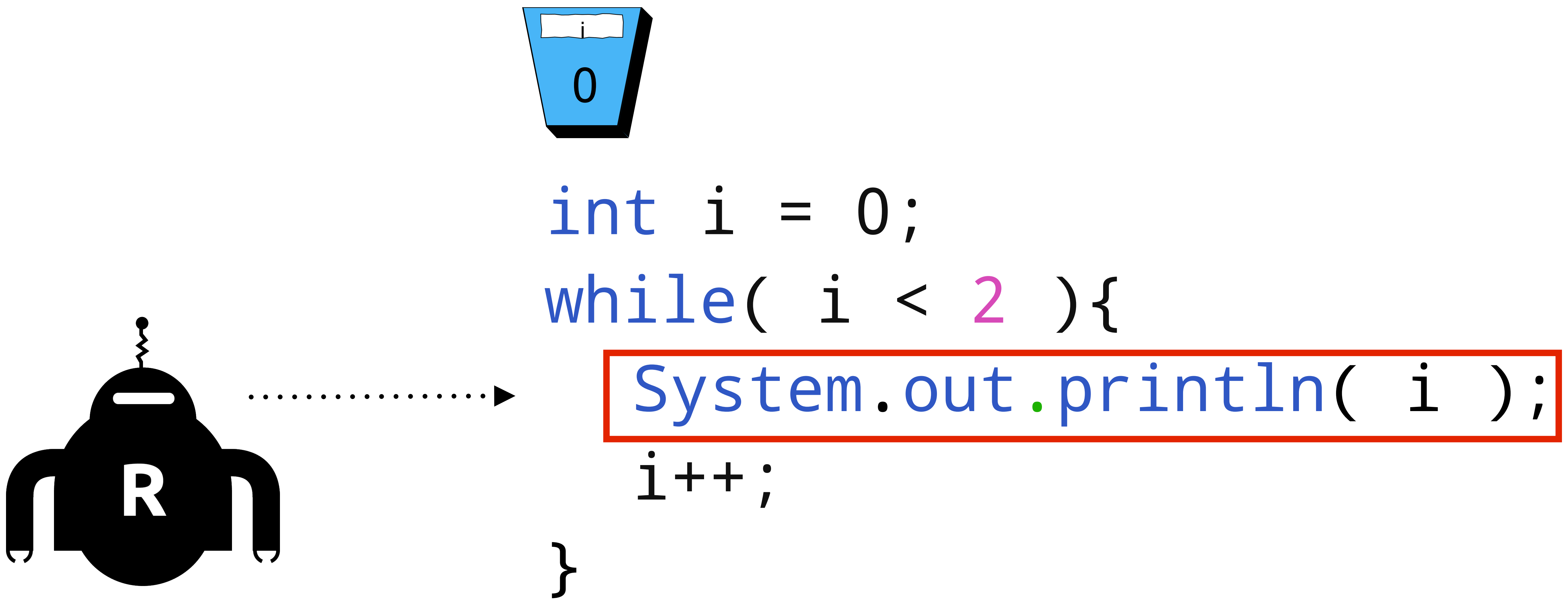
creating local
variable i and
setting it to 0



```
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```



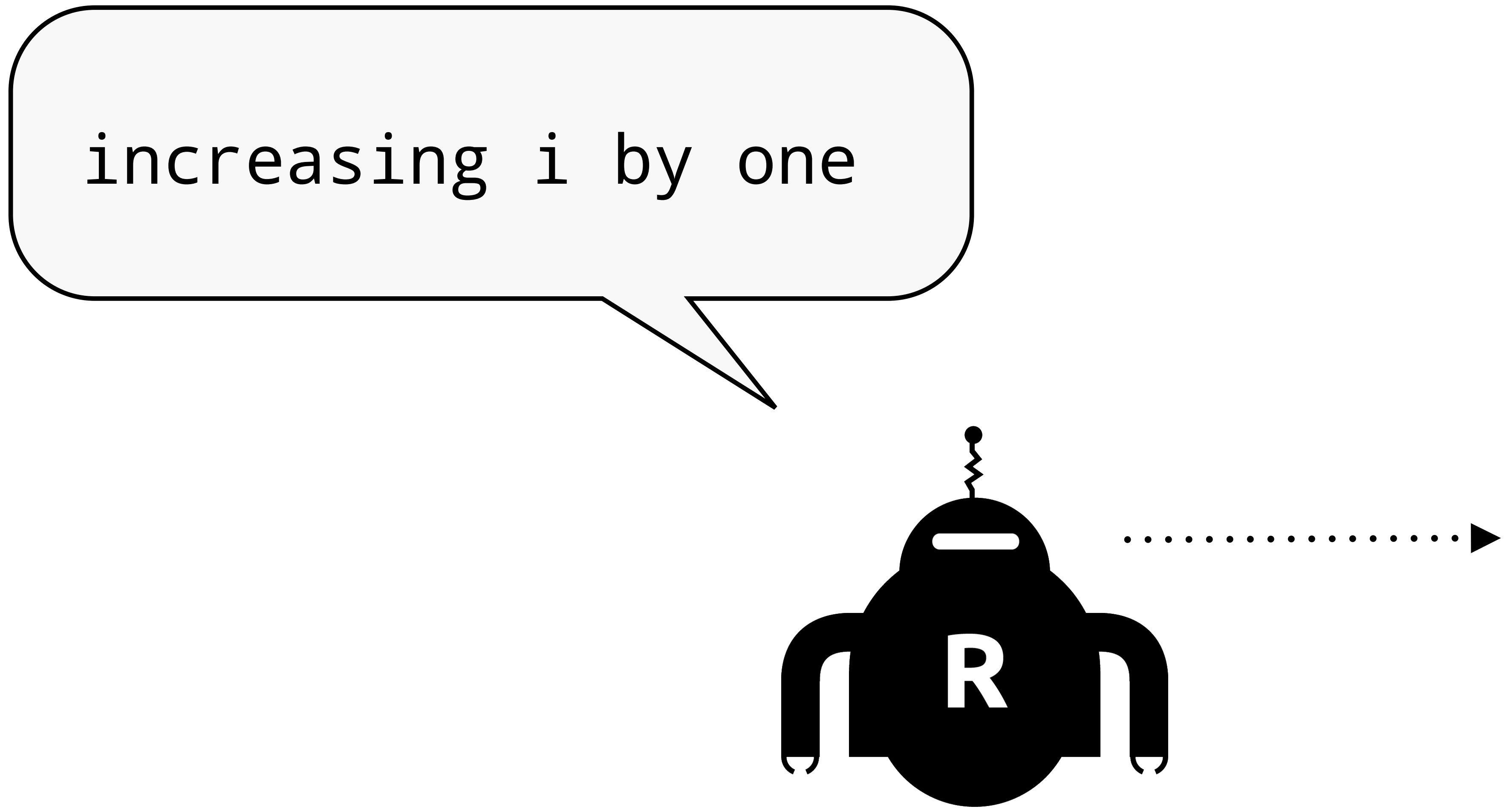
condition is true,
so I can enter the
while loop

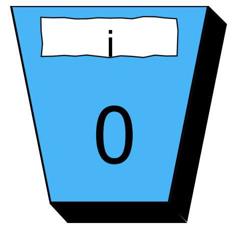


```
int i = 0;
while( i < 2 ){
  System.out.println( i );
  i++;
}
```

Console

```
0
```

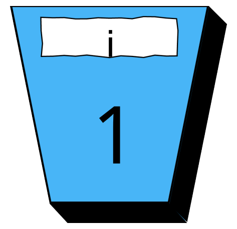




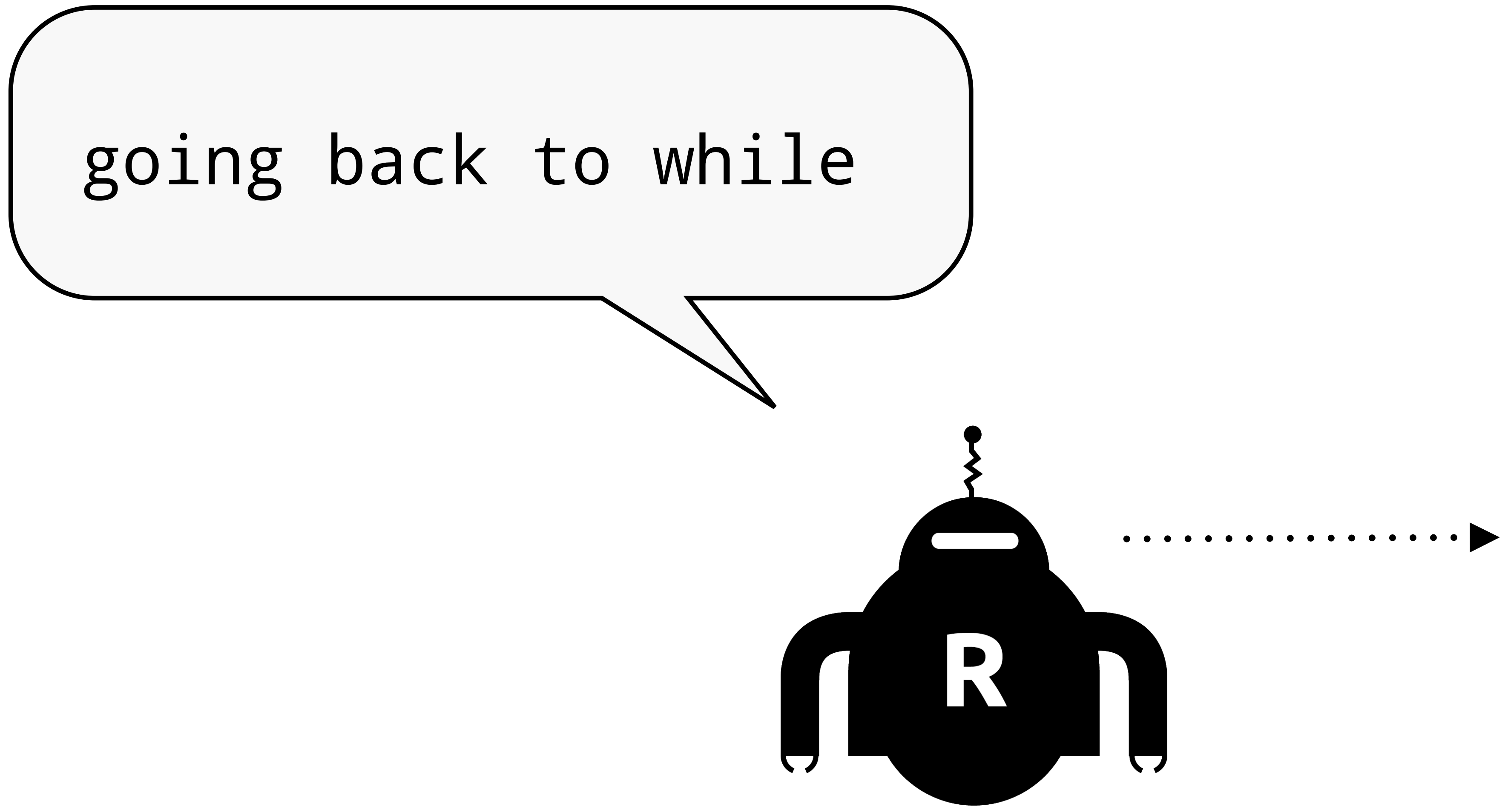
```
int i = 0;
while( i < 2 ){
  System.out.println( i );
  i++;
}
```

Console

```
0
```

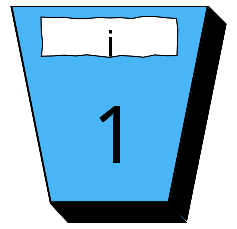


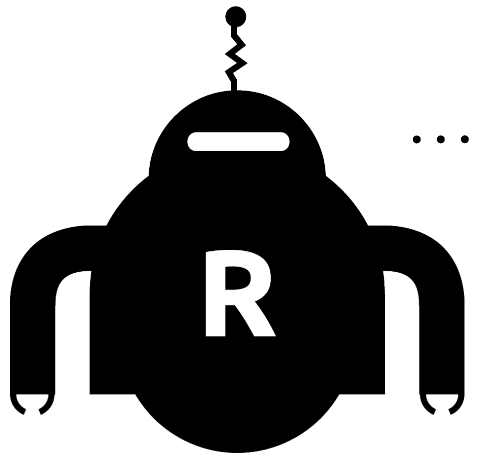
```
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```



Console

```
0
```



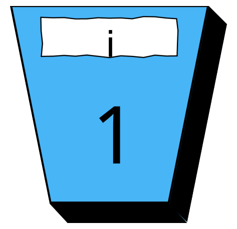


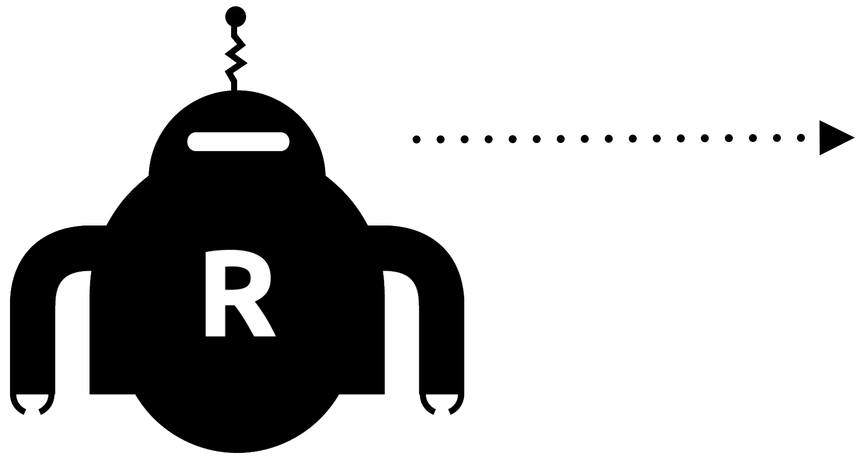
```
int i = 0;
while( i < 2 ){
  System.out.println( i );
  i++;
}
```

condition is true,
so I can enter the
while loop

Console

```
0
```

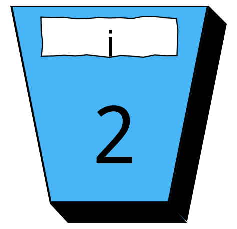




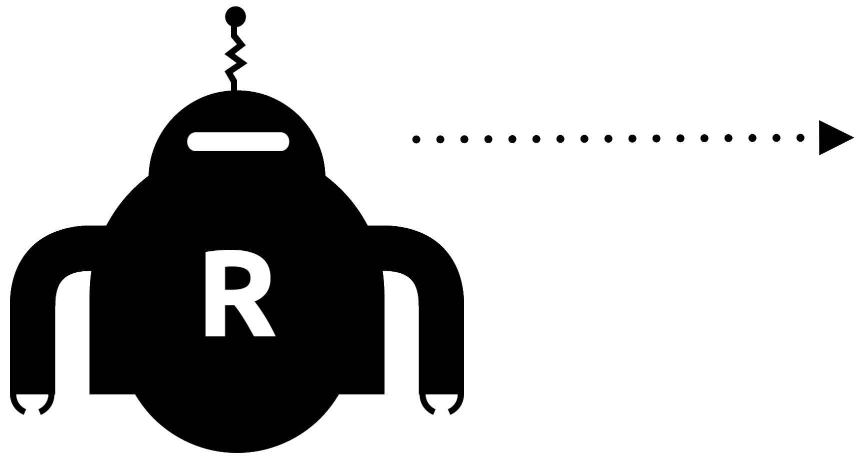
```
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```

Console

```
0
1
```

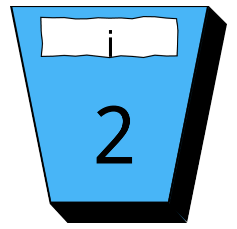


```java
int i = 0;
while( i < 2 ){
  System.out.println( i );
  i++;
}
```

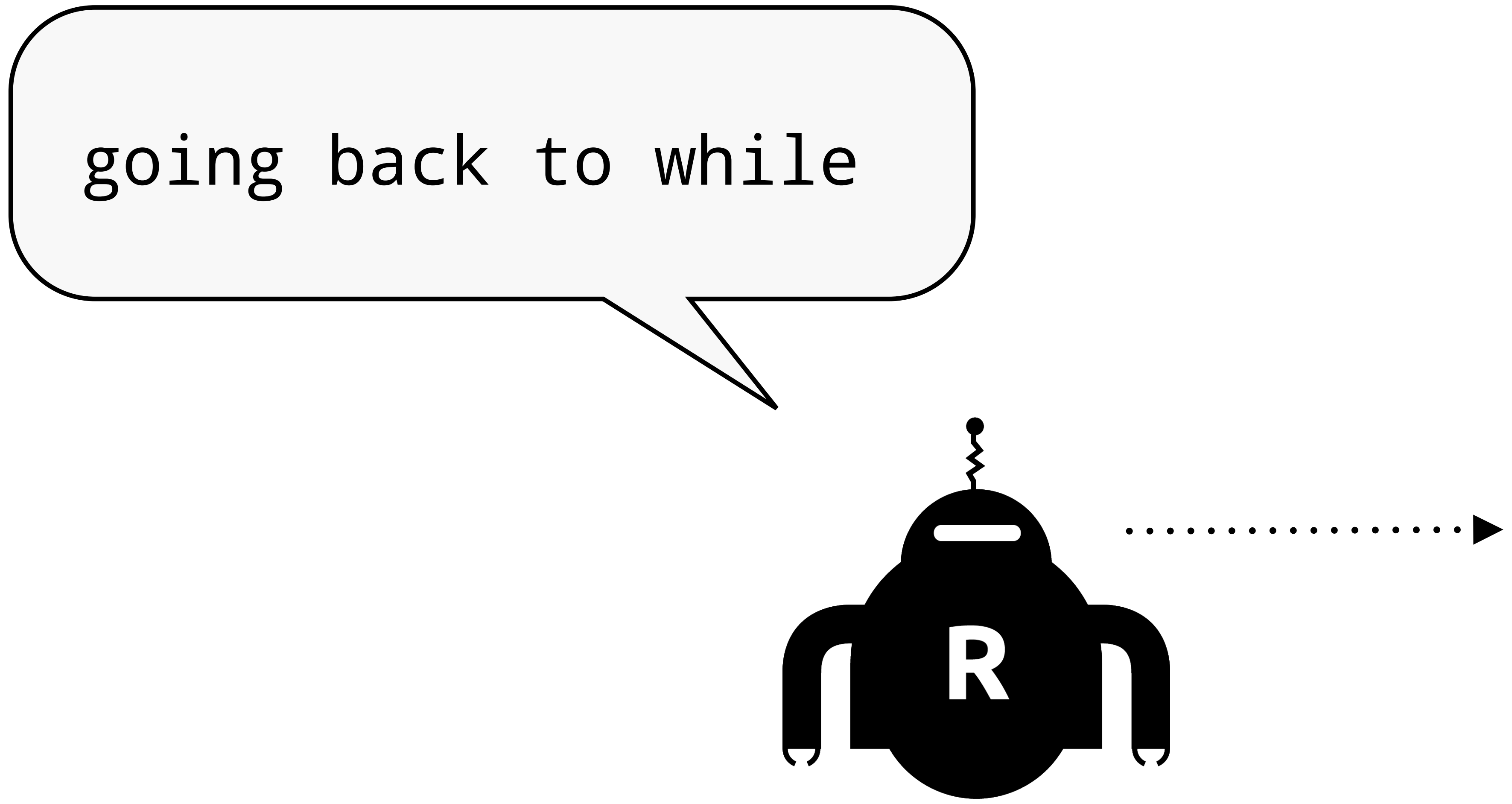


Console

```
0
1
```



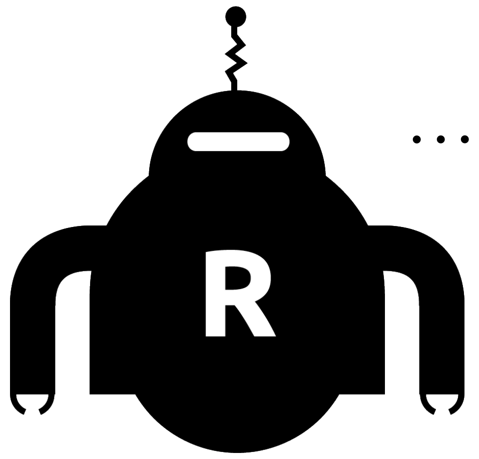
```java
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```

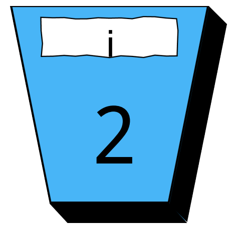


Console

```
0
1
```

condition is no
longer true. exiting
the while loop.

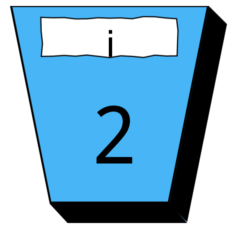




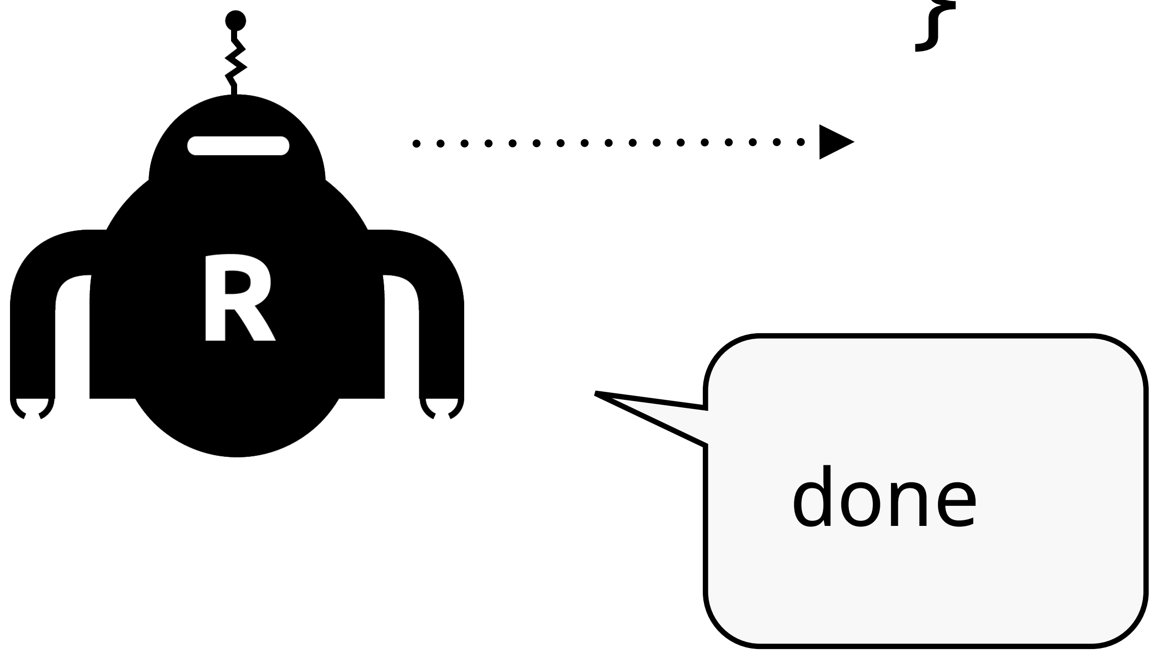
```
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```

Console

```
0
1
```



```
int i = 0;
while( i < 2 ){
   System.out.println( i );
   i++;
}
```



Console

```
0
1
```

```java
int i = 0;
while( i < 2 ){
  System.out.println( i );
  i++;
}
```

Console

```
0
1
```

For loop example

```
for( int i=0; i<2; i++ ){
  // repeat this code
}
```

start with the variable *i* at value 0

repeat as long as condition is true

increase *i* by 1 every iteration

```
for( int i=0; i<2; i++ ){
  // repeat this code
}
```

start with the variable *i* at value 0

repeat as long as condition is true

increase *i* by 1 every iteration

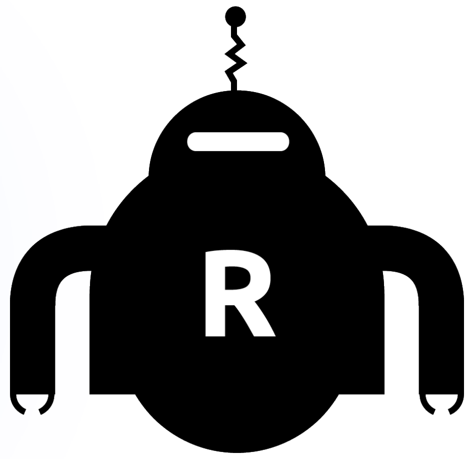
```
for( int i=0; i<2; i++ ){
  // repeat this code
}
```

start with the variable *i* at value 0

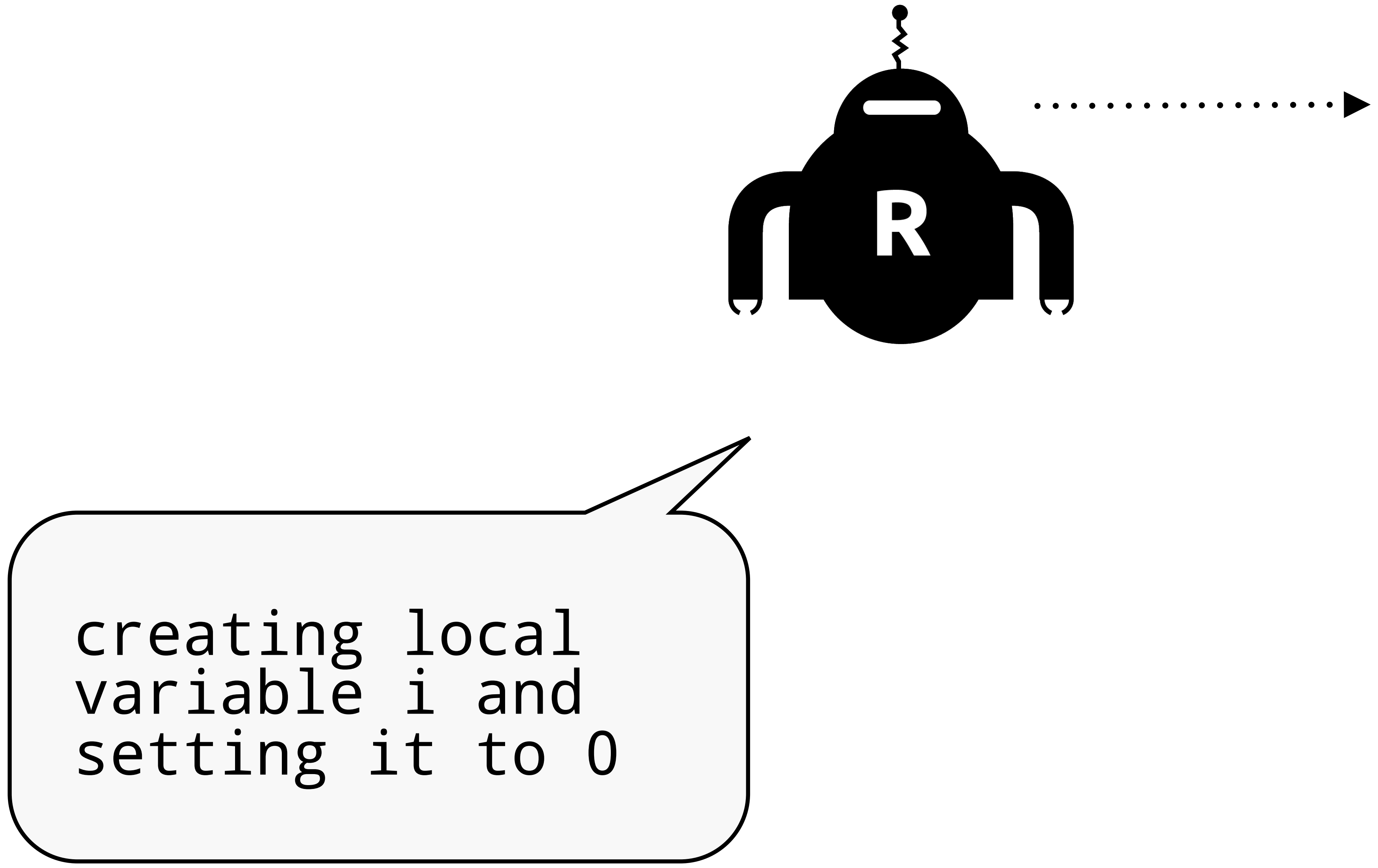
repeat as long as condition is true

increase *i* by 1 every iteration

```
for( int i=0; i<2; i++ ){
   // repeat this code
}
```

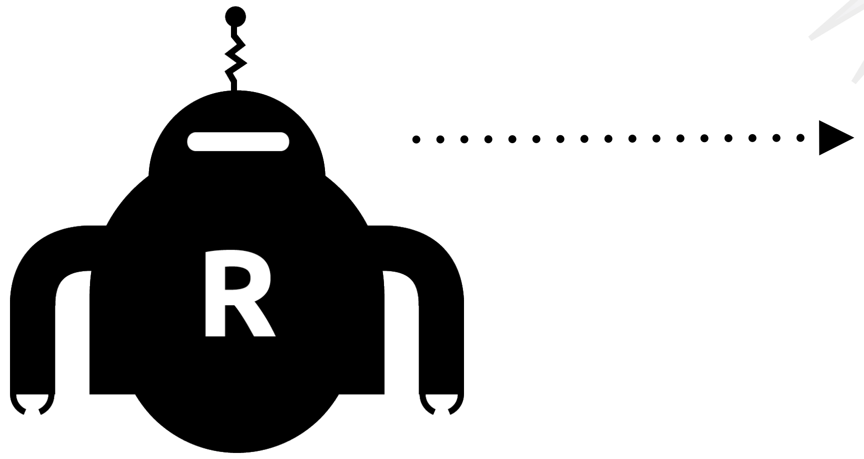


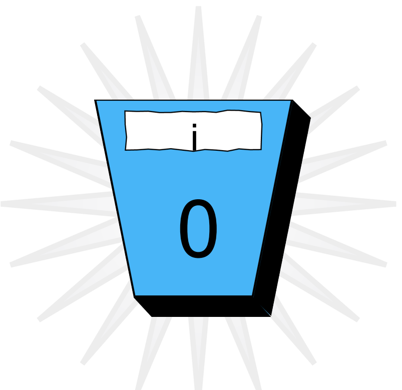
```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```



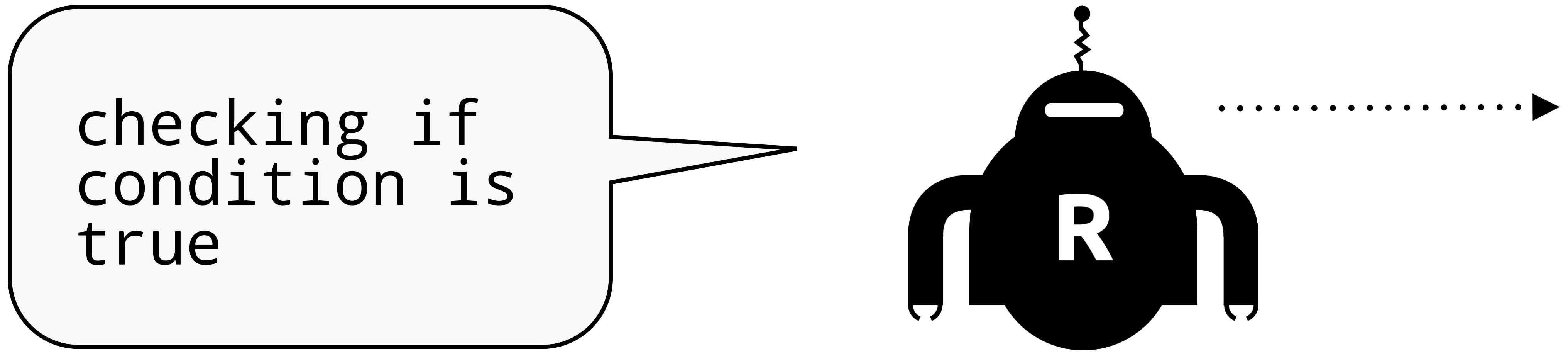
```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

creating local
variable i and
setting it to 0

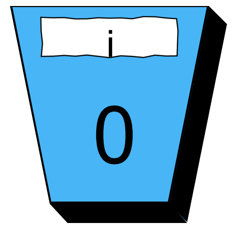




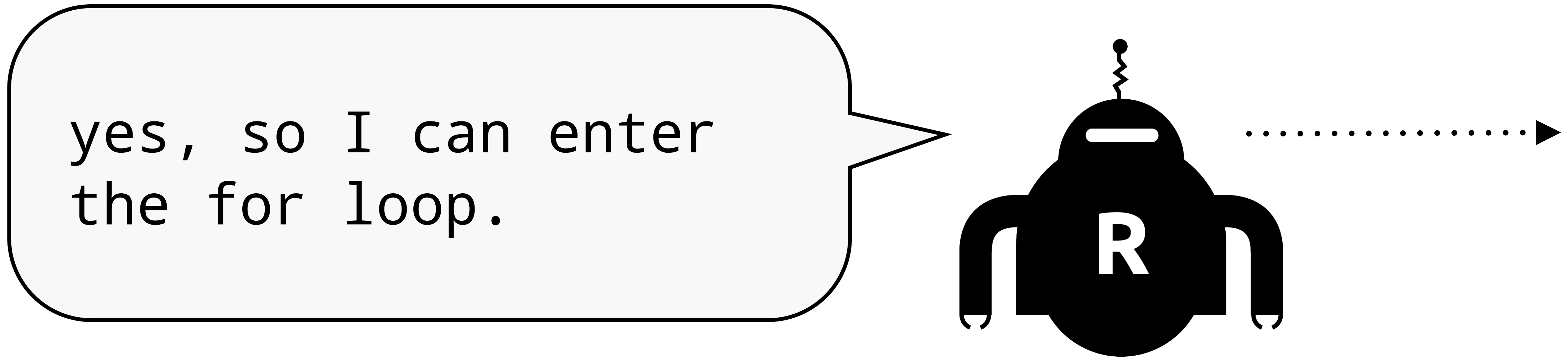
```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

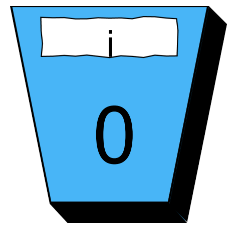


checking if condition is true

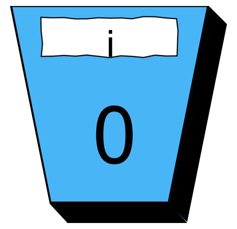


```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

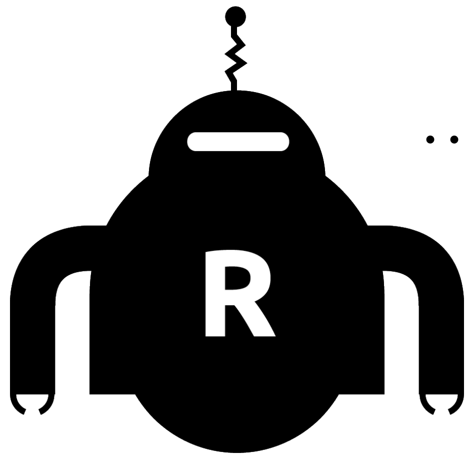




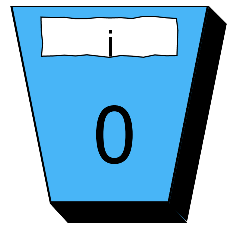
```
for( int i=0; true; i++ ){
   System.out.println( i );
}
```

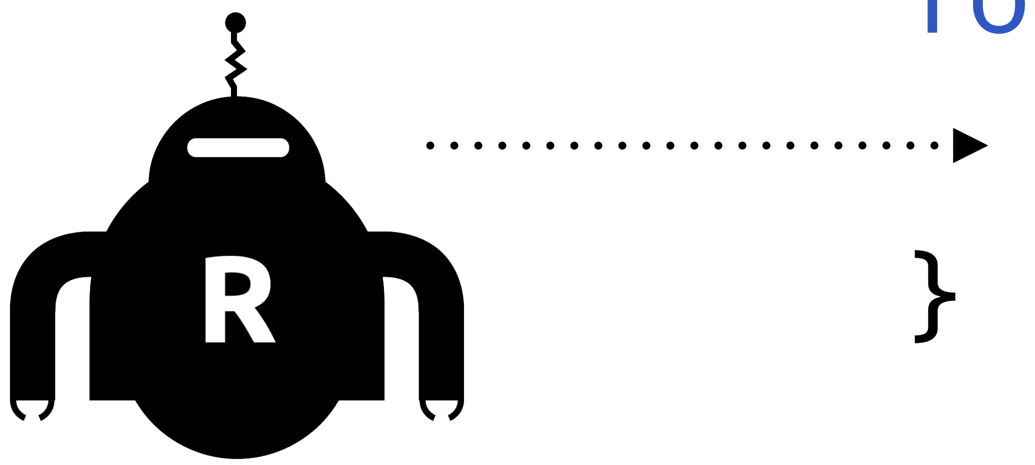


```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```



logging the
value of i

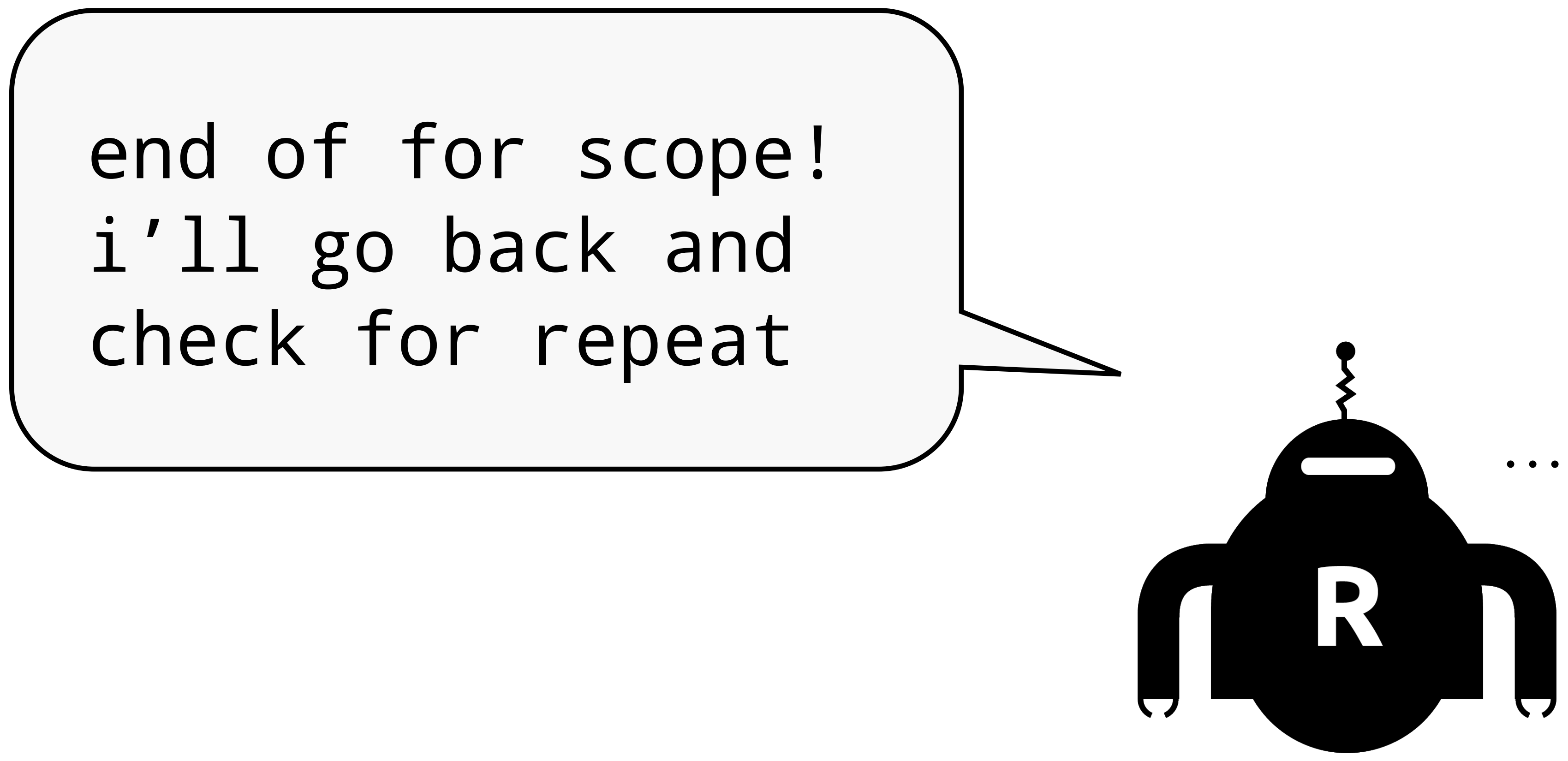


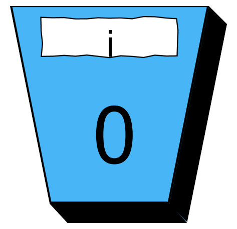


```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

Console

```
0
```

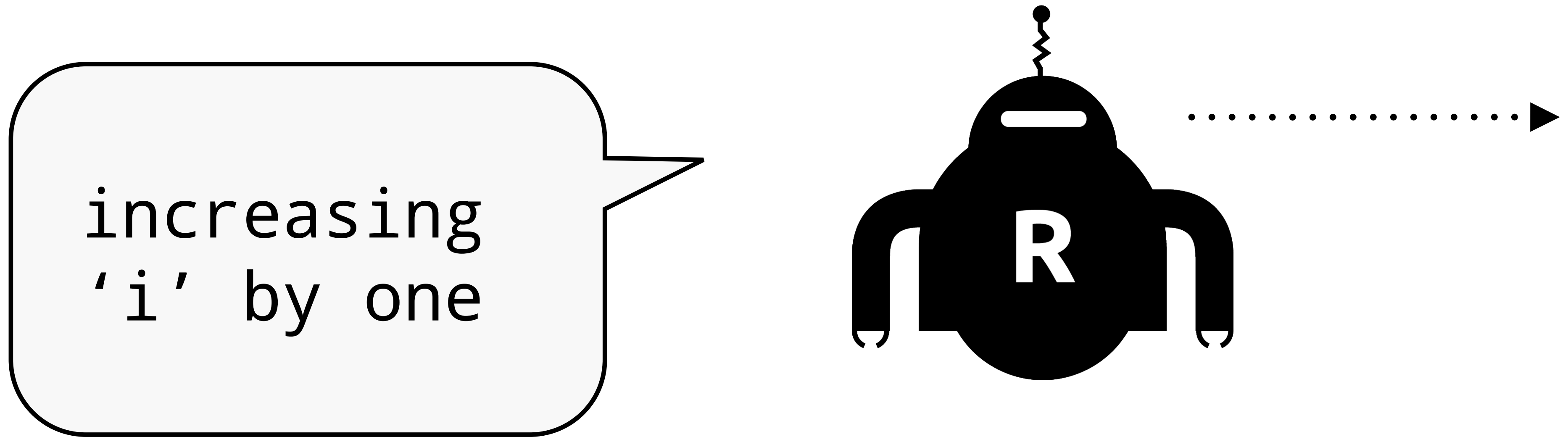


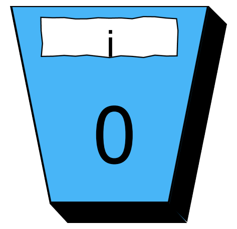


```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

Console

```
0
```

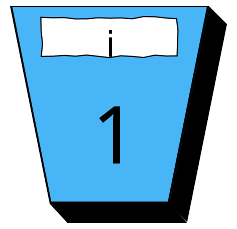


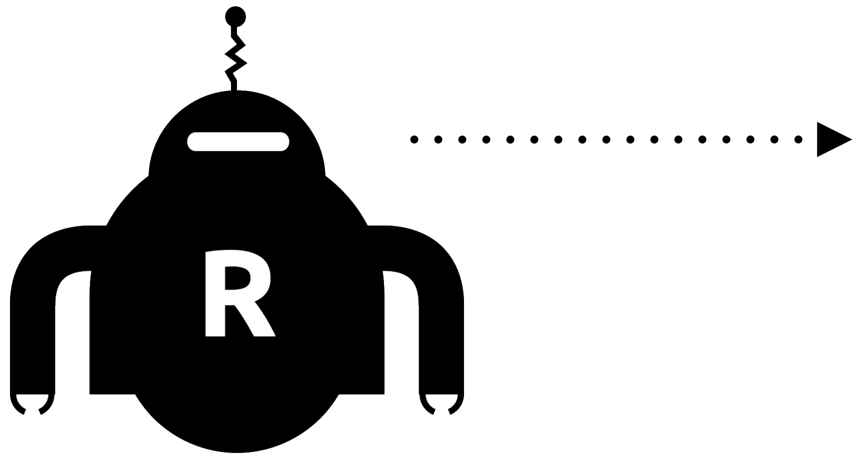


```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

Console

```
0
```



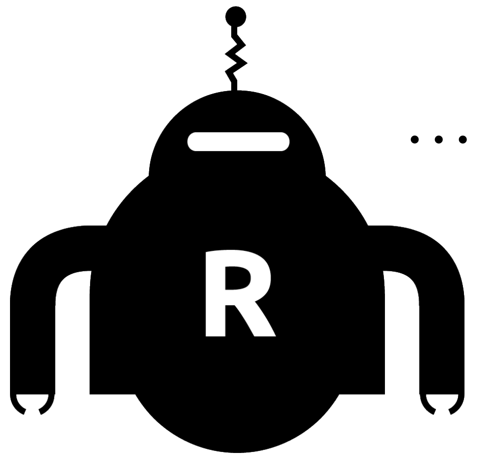


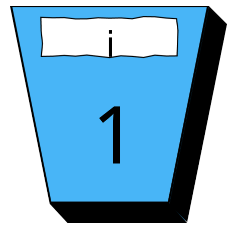
```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

Console

```
0
```

checking if
condition is true

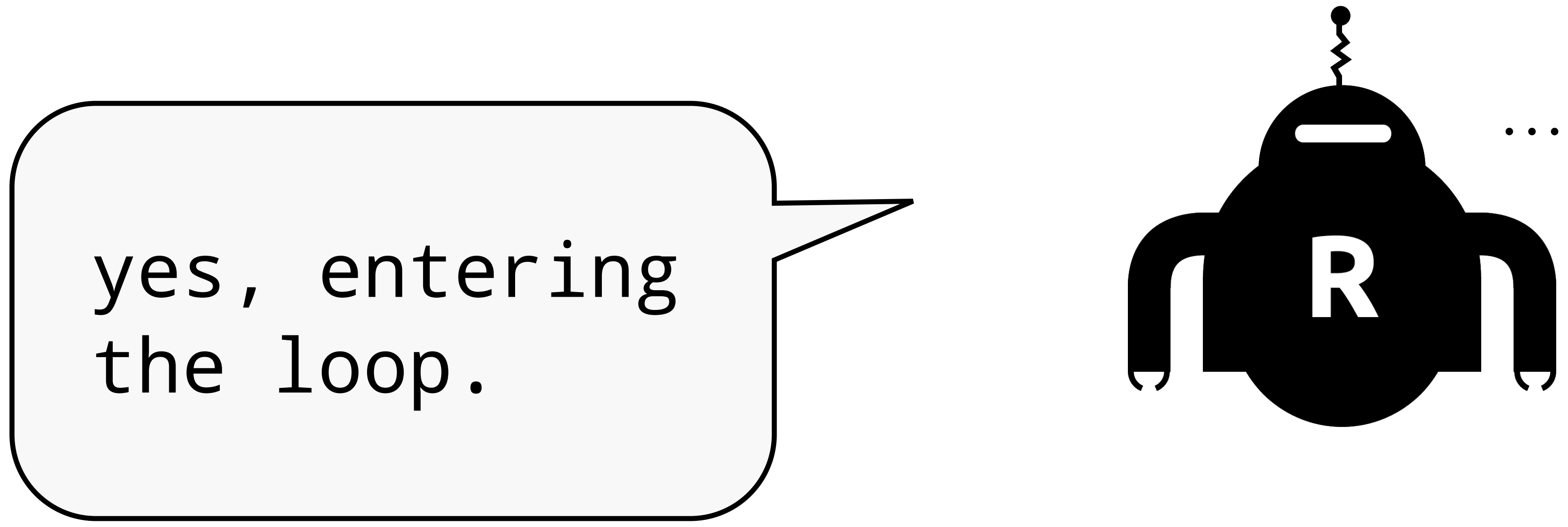




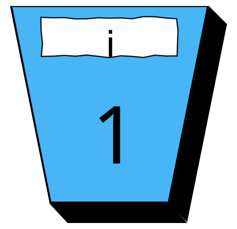
```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

Console

```
0
```



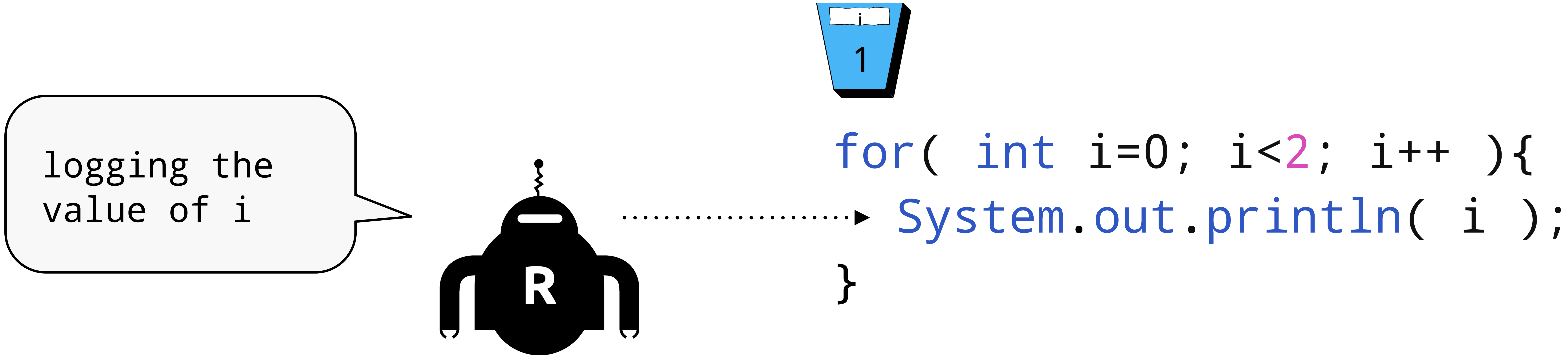
yes, entering the loop.



```
for( int i=0; true; i++ ){
   System.out.println( i );
}
```

Console

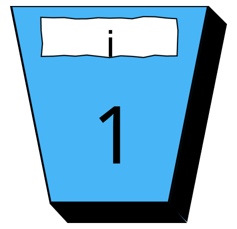
```
0
```

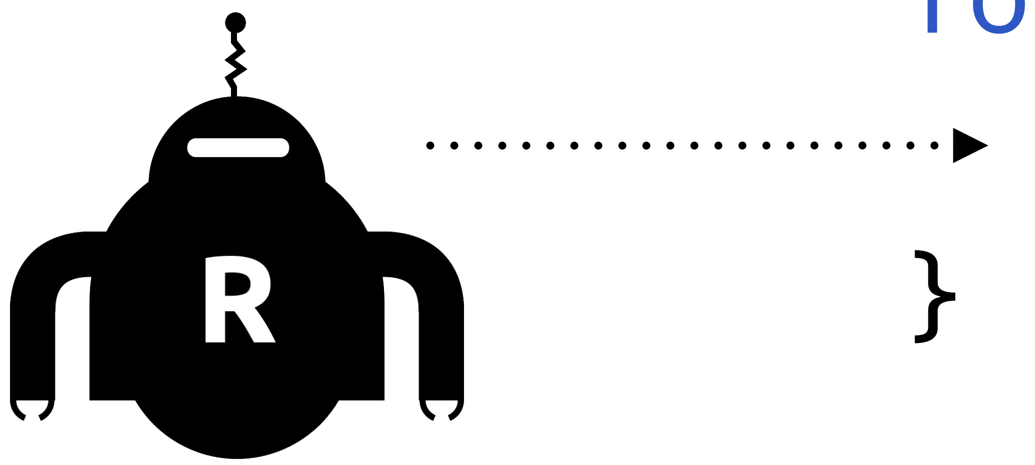


```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

Console

```
0
```



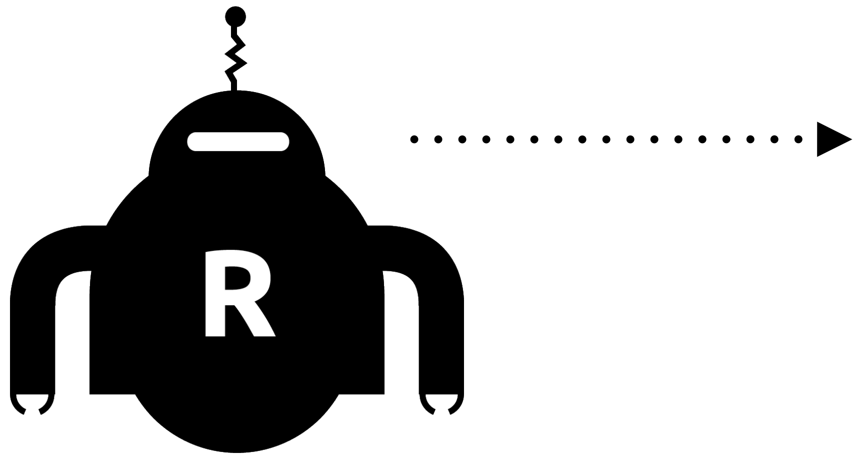


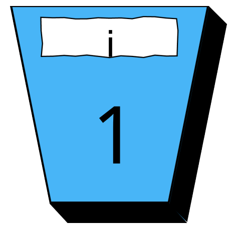
```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

Console

```
0
1
```

back to the top
once again

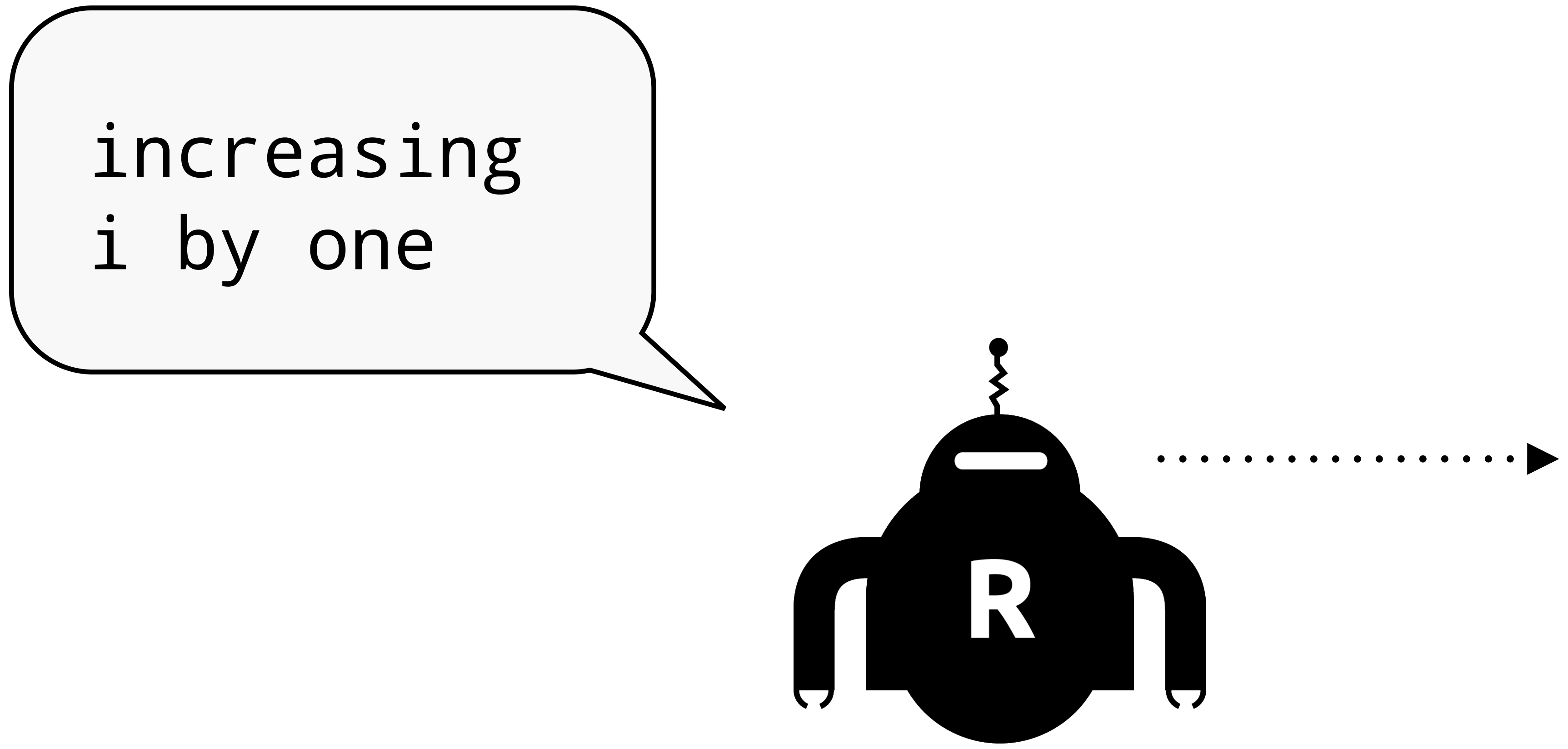


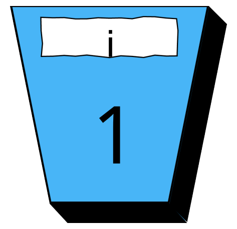


```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

Console

```
0
1
```

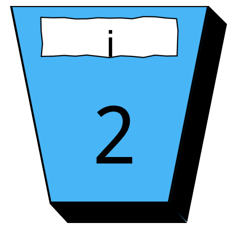


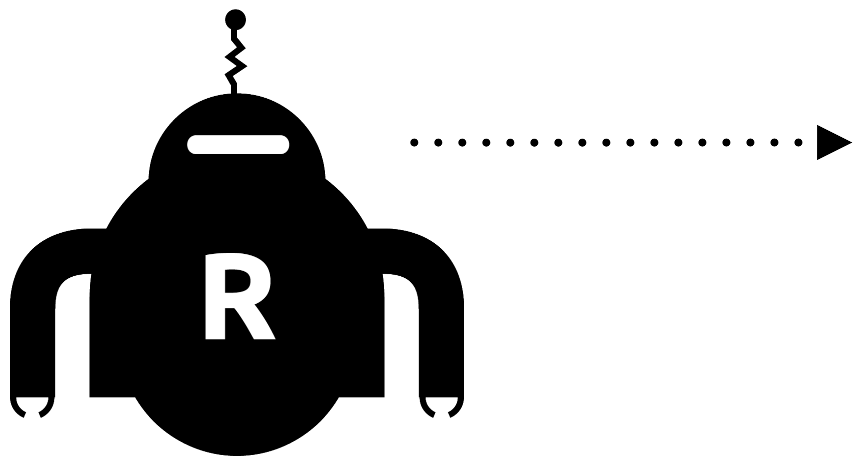


```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

Console

```
0
1
```



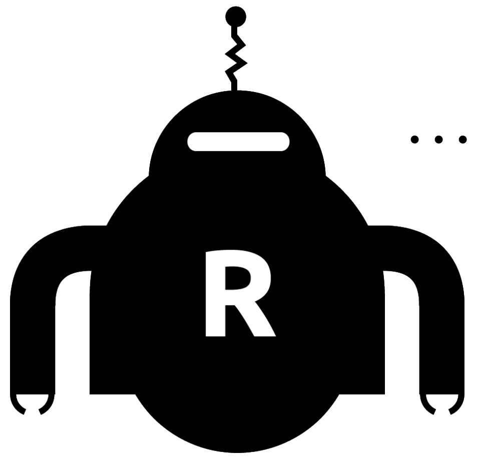


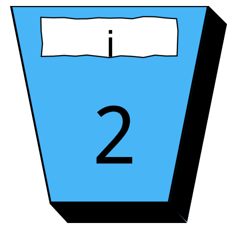
```java
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

Console

```
0
1
```

checking if
condition is true

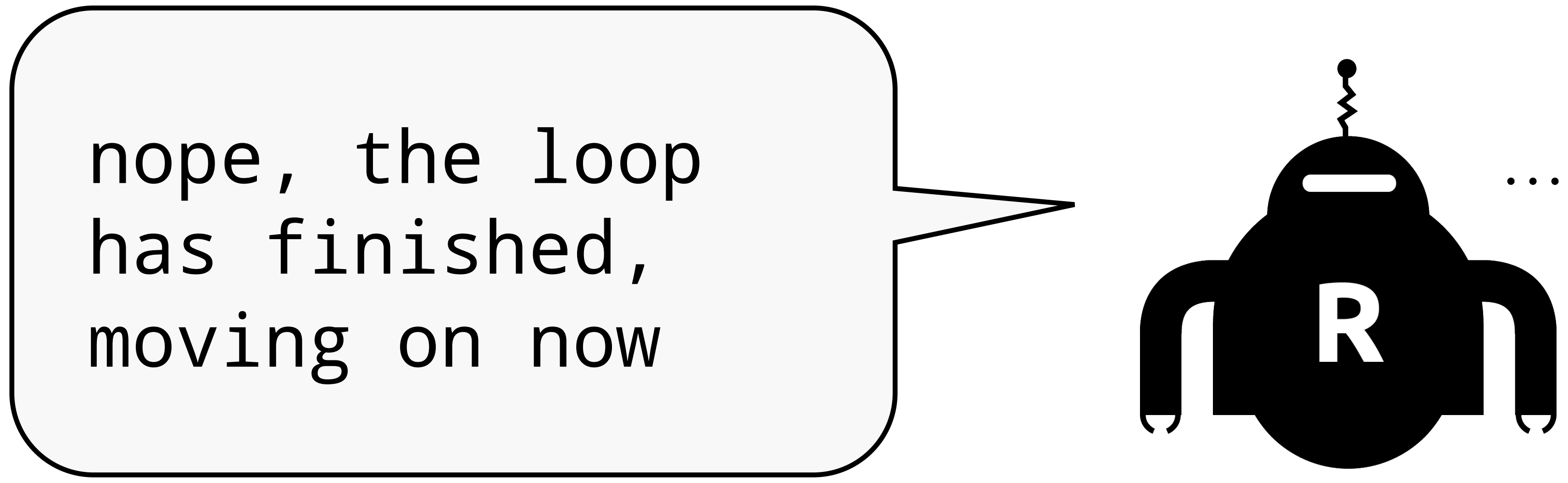


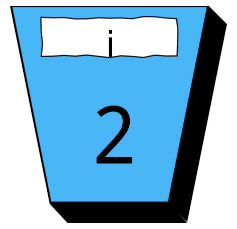


```
for( int i=0; i<2; i++ ){
   System.out.println( i );
}
```

Console

```
0
1
```

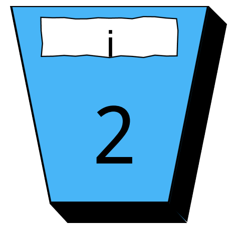




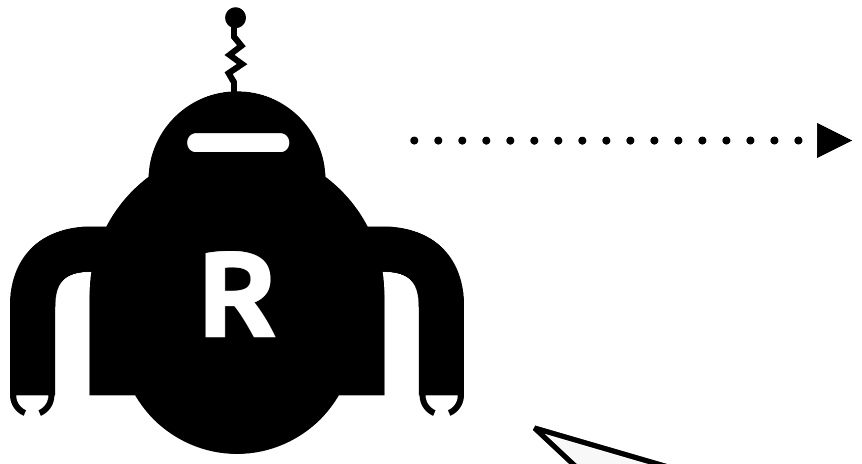
```
for( int i=0; false; i++ ){
  System.out.println( i );
}
```

Console

```
0
1
```



```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```



nothing here.
job done!

Console

```
0
1
```

```
for( int i=0; i<2; i++ ){
  System.out.println( i );
}
```

Console

```
0
1
```

While and **for** loops are pre-test, **do-while** is post-test.

zero or more repetitions

```
int i = 0;
while( i++ < 100 ){
    // repeat.
}
```

```
for( int j=0; j<100; j++ ){
    // repeat.
}
```

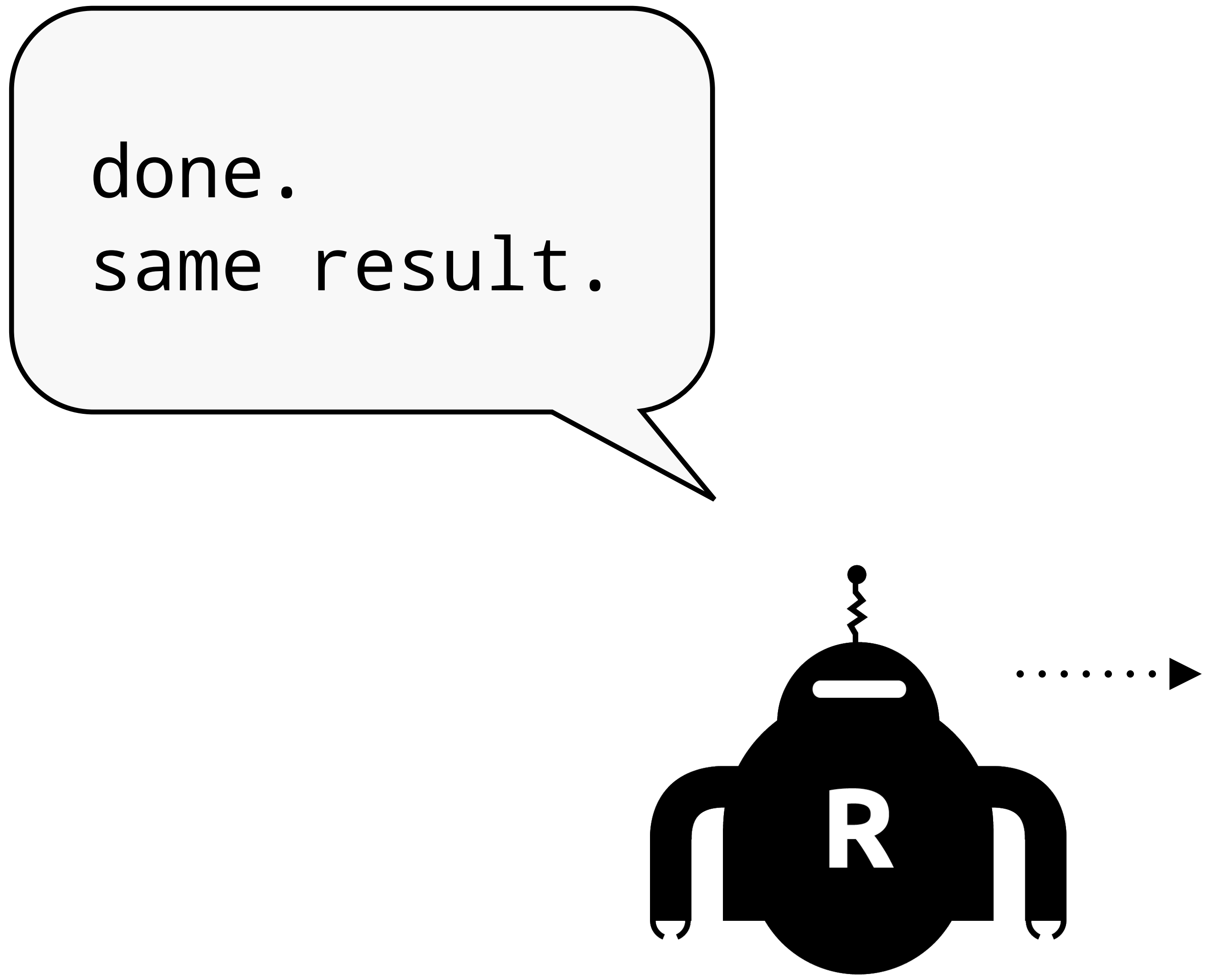
one or more repetitions

```
int k = 0;
do {
    // Repeat.
} while( k++ < 100 );
```

Do-while loop example

```
int i = 0;
do {
  // repeat this code
  i++;
} while( i < 2 );
```

```
int i = 0;
do {
  System.out.println( i );
  i++;
} while( i < 2 );
```



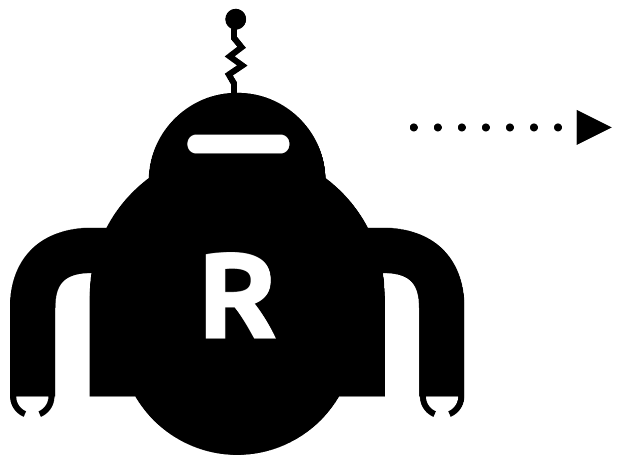
```
int i = 0;
do {
   System.out.println( i );
   i++;
} while( i < 2 );
```

Console

```
0
1
```

```java
int i = 0;
do {
   System.out.println( i );
   i++;
} while( i < 0 );
```

done.
I executed to loop once,
even though the while-
condition was never true.



```
int i = 0;
do {
   System.out.println( i );
   i++;
} while( i < 0 );
```

Console

```
0
```

The **break** statement exits the loop immediately.
Works for all types of loops.

```
int i = 0;
while( true ){
    if( i++ > 100 ) break;
}
```

The **continue** statement jumps to the next iteration immediately. Works for all types of loops.

```
for( int j=0; j<100; j++ ){
    if( j%2 != 0 ) continue;
    // j will always be even here.
}
```