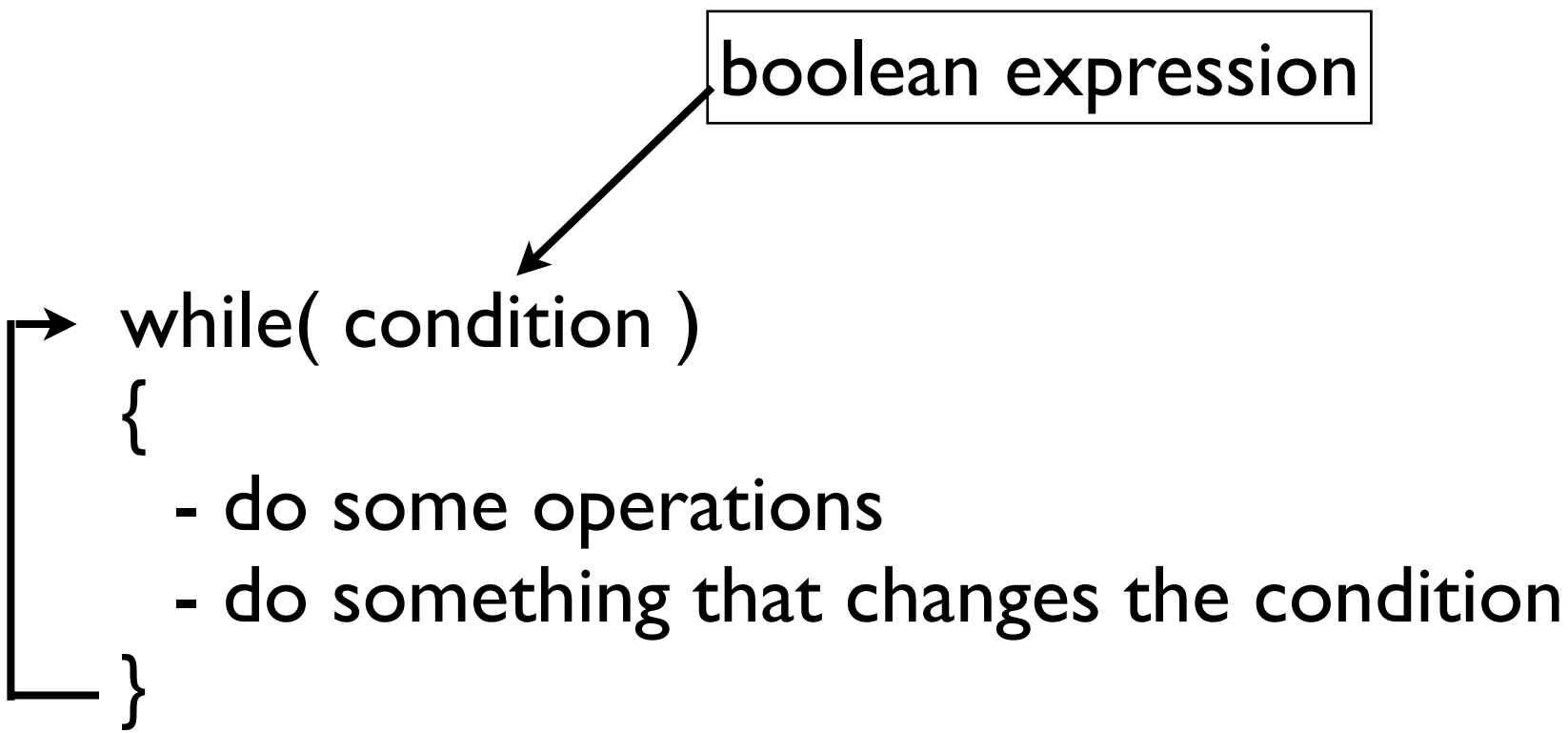
Intro To Loops

Mr. Fahrenbacher Likes Fruit Loops

Purpose of Loops

- To make our code run several times (perhaps forever!)
- An alternate technique to recursion (less complicated, but also less eloquent)

Basic **While** Structure

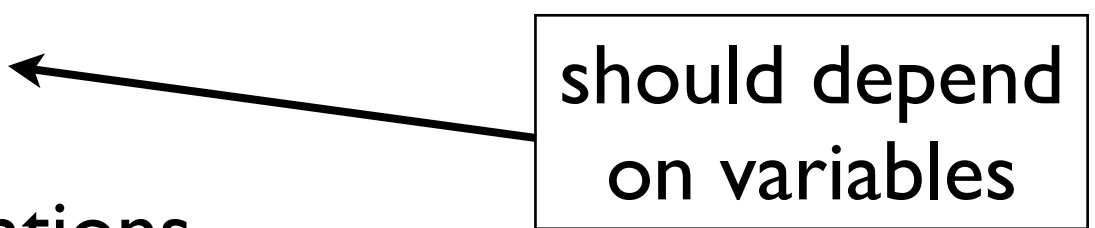


If the condition is never false, the loop will run forever!

Basic **While** Structure

- variable we want to change
- variables we want to effect the condition

```
while( condition )
{
  - do some operations
  - do something that changes the condition
}
```



When the condition evaluates to false, the loop ends

Simple Example

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

What does this code do?

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 0

Display:

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 0

Display:

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 0

Display:
0

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 1

Display:
0

Stepping through code

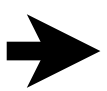
```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 1

Display:
0

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
   System.out.println(loc); //operation
   loc++; //increment
}
```



loc = 1

Display:
0
1

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 2

Display:
0
1

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 2

Display:
0
1

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 2

Display:
0
1
2

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 3

Display:
0
1
2

Stepping through code

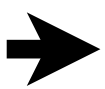
```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```

loc = 3

Display:
0
1
2

Stepping through code

```
int loc = 0; //counter

while(loc < 3) //condition
{
  System.out.println(loc); //operation
  loc++; //increment
}
```



loc = 3

Display:
0
1
2

Simple Example Reprised

```
int loc = 0; //counter

while(loc < 3) //condition
{
  loc++; //increment
  System.out.println(loc); //operation
}
```

What does this code do differently?

Simple Example Repraised

```
int loc = 0; //counter

while(loc < 3) //condition
{
  loc++; //increment
  System.out.println(loc); //operation
}
```

Display:
1
2
3

More Variables Are Good!

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

Can you step through this code?

More Variables Are Good!

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

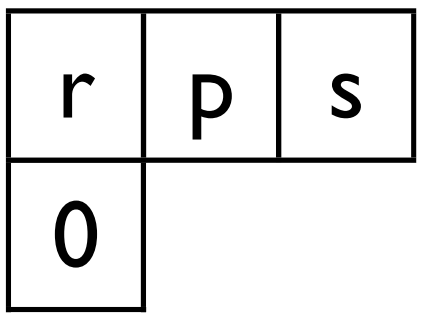
r	p	s

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!




```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0		

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1		

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3		

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5
6		

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5
6	4	

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5
6	4	5

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5
6	4	5
10		

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5
6	4	5
10	5	

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

What value does position have after the while loop is done?

Display:
10

More Variables Are Good!

r	p	s
0	0	5
0	1	5
1	2	5
3	3	5
6	4	5
10	5	5

```
int result = 0;
int position = 0;
int sentinel = 5; //look out for end

while(position < sentinel)
{
   result = result + position;
   position = position + 1;
}

System.out.println(result);
```

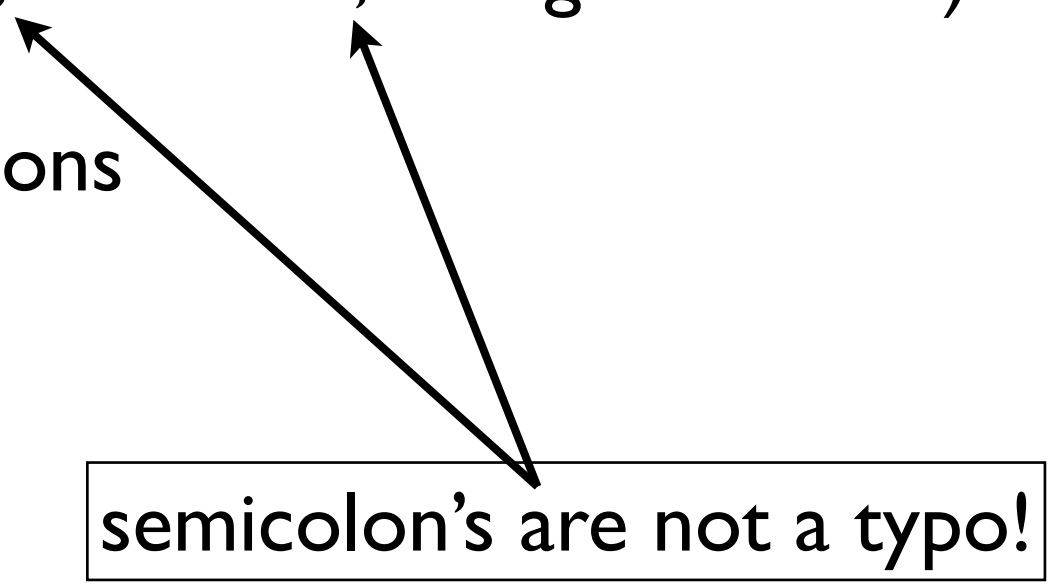
What value does position have after the while loop is done?

Display:
10

For Loop

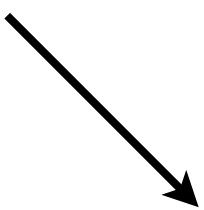
- A way to write a while loop horizontally instead of vertically.

```
for( create counter; condition; change counter )
{
  - do some operations
}
```



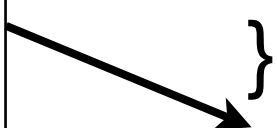
Previous Example For'd

```
int loc = 0;

while(loc < 3)
{
  System.out.println(loc);
  loc++;
}
```



```
for(int loc = 0; loc < 3; loc++)
{
  System.out.println(loc);
}
```

Note: loc will not exist after loop is done!



For Loop Order

```
          (1)     (2)        (4)
for(int loc = 0; loc < 3; loc++)
{
  System.out.println(loc);  (3)
}
```

step 1 is only executed the first time

after that it goes 2-3-4-2-3-4-2-3-4-...-2, stopping when the condition is false

For versus While

- Anything you can do with a for loop, you can also do with a while
- For loops are shorter, but more "complicated"
- Most of my loops are for-loops

Infinite Loops

- If you don't change the condition, the loop will go on forever! (But your program won't crash).

```
int loc = 0;
while(loc > 0)
{
  System.out.println(loc);
}
```

Loops and Scope

- If you declare a variable in a loop
  - it gets recreated every iteration (forgets)
  - you can't use it outside of the loop

```
for(int i = 0; i < 5; i++)
{
  int sum = 0;
  sum += (int)(Math.random()*6);
}
//System.out.println(sum);        Error!
```

Loops and Scope

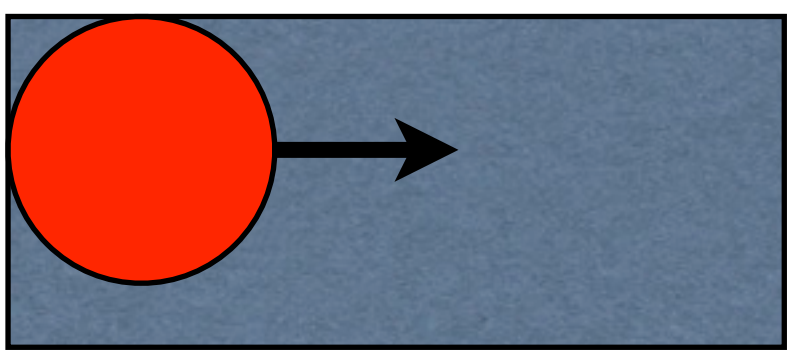
- If you declare a variable in a loop
  - it gets recreated every iteration (forgets)
  - you can't use it outside of the loop

```
int sum = 0;
for(int i = 0; i < 5; i++)
{
  sum += (int)(Math.random()*6);
}
System.out.println(sum);
```

No Error!

Helpful Hints

- If you're going to calculate a value with a loop, create the variable BEFORE the loop
- When you write the loop conditional, think backwards
  - Normal - when should the loop stop?
  - Actual Code - when should it NOT stop?



Stop: x + radius > width

Loop: x + radius <= width