Download

Day 5: For Loops

Download

Agenda

- Quiz 2
- Nested Conditionals
- Unary Operators
- For Loops
- Continue
- Nested Loops
- In Class

Download

Quiz 2

Download

Nested Conditional Statements

We can nest conditional statements as we can in Python. Pay attention to the curly brackets!

Review of Nested Conditional Statements

```
if(condition 1) {
    if (inner condition 1) {
        // code if cond 1 and inner cond 1 are true
    } else if(inner condition 2) {
        // code if cond 1 and inner cond 2 are true
    }
} else {
    // code if condition 1 is false
}
```

Unary Operators

In Java, we have a concept called "unary" operators.

These are an even simpler way of updating a variable by adding or subtracting 1 to it.

```
int x = 0;

x++; // same as x += 1;

x--; // same as x -= 1;
```

Why?

This one can actually be placed and used as an expression in the println code!

```
int x = 0;

System.out.println(x++); // prints
0

System.out.println(x); // prints 1
```

Why?

Works with subtraction too!

```
int x = 10;

System.out.println(x--); // prints
10

System.out.println(x); // prints 9
```

Location matters!

`x++;` will increment x **after** its used.

`++x;` will increment x **before** its used.

```
int x = 0;

System.out.println(x++); // prints
0 but x becomes 1

System.out.println(x); // prints 1

System.out.println(++x); // prints
2 as x becomes 2 before we use it.
```

Location matters!

`x--;` will decrement x **after** its used.

`--x;` will decrement x **before** its used.

```
int x = 10;

System.out.println(x--); // prints
10 but x becomes 9

System.out.println(x); // prints 9

System.out.println(--x); // prints
8 as x becomes 8 before we use it.
```

Download

For Loop

Outside of while loops, we also have for loops!

Download

For loops work a little differently than they do in Python.

For Template

Here's the template:

```
for(<variable>; <condition>;
<change>) {

  // code

}
```

Example

Let's use an actual example and break down the example. The code to the right will print out the numbers from 1 to 100.

```
for(int i = 1; i < 101; i++) {
  System.out.println(i);
}
```

Example breakdown

We start this loop by initializing a variable to 1.

```
for(int i = 1; i < 101; i++) {
  System.out.println(i);
}
```

Example breakdown

We then tell our code to keep going as long as i is less than 101.

```
for(int i = 1; i < 101; i++) {
  System.out.println(i);
}
```

Example breakdown

And at the end, we tell it how to get there (by incrementing by 1.)

```
for(int i = 1; i < 101; i++) {
  System.out.println(i);
}
```

Decrementing example

What about counting downward?

```
for(int i = 100; i > 0; i--) {
System.out.println(i);
}
```

For v While

```
for(int i = 0; i < 100; i++) {
  System.out.println(i);
}
```

```
int i = 0;
while(i < 100) {
  System.out.println(i);
  i++;
}
```

For v While

For Loop

- Finite number of iterations
- Counter is necessary

While Loop

- Number of iterations may be unknown
- Run until some condition is met, may not be a numeric condition

continue keyword

Continue is a keyword that will skip an iteration and go to the following iteration.

The code to the right will print out all the numbers between 0 and 99, but it'll skip each multiple of 5.

```
for(int i = 0; i < 100; i++) {
   if (i % 5 == 0) {
      continue;
   }
   System.out.println(i);
}
```

Nested Loops

As a review, we can nest loops as we can with conditional statements

```
for(int i = 0; i < 10; i++) {
  for(int j = i; j < 10; i++) {
    System.out.print(j + " ");
  }
  System.out.println();
}
```

Nested Loops

Output of the right hand code gives:

```
0123456789
123456789
23456789
3456789
456789
56789
6789
789
89
9
```

```
for(int i = 0; i < 10; i++) {
  for(int j = i; j < 10; i++) {
    System.out.print(j + " ");
  }
  System.out.println();
}
```

Nested Loops

Once the inner `j` loop finishes, the outer `i` loop goes to the next iteration. This keeps resetting the inner loop until the outer loop finishes.

```
for(int i = 0; i < 10; i++) {
  for(int j = i; j < 10; i++) {
    System.out.print(j + " ");
  }
  System.out.println();
}
```

Download

In Class: Iterative Practice