The FOR Loop

A byte size lesson in Java programming.

Why use a FOR Loop?

- In Java, loops are used to repeat a block of code.
  - Ex: if you want to show a message 100 times.
  - Ex: counting from 1 to 10
- There are 3 types of loops; FOR loop, WHILE loop, and DO..WHILE loop
- The FOR loop has this basic structure

```
for (initialExpression; condition; updateExpression) {
    // body of the loop
}
```

Breaking it down

```
for (initialExpression; condition; updateExpression) {
    // body of the loop
}
```

- **initialExpression**: initializes and/or declares variables. Executes only once.
- **condition** is evaluated. If true, the body is executed
- **updateExpression**: after each loop, updates the value of initialExpression
- condition is evaluated again, until condition is false

The simplest example: Repeat text

```
int n = 3;
// for loop
for (int i = 1; i <= n; ++i) {
  System.out.println("Java is fun");
}
```

Output

```
Java is fun
Java is fun
Java is fun
```

- Initializes i to 1
- The condition will be true as long as i is less than or equal to n
- Every loop, i will be incremented by 1

Let's test your understanding!

- What output would the following give?

```
class Main {
  public static void main(String[] args) {

    int n = 4;
    // for loop
    for (int i = 1; i <= n; ++i) {
      System.out.println(i);
    }
  }
}
```

Infinite Loops

- If the Test Condition and Update Expression are not written well, you might end up with an infinite loop!

```
class MyLoop {
    public static void main(String[] args) {

        int sum = 0;

        for (int i = 1; i <= 10; --i) {
            System.out.println("Hello");
        }
    }
}
```

- What would this code output?

```
class Main {
  public static void main(String[] args) {

    int sum = 0;
    int n = 3;

    // for loop
    for (int i = n; i >= 1; --i) {
      // body inside for loop
      sum += i;
    }

    System.out.println("Sum = " + sum);
  }
}
```