"Boolean Zen"
Better Programming Style

Boolean Method

```java
public static boolean isEven(int num) {

    boolean isEven;

    if (num % 2 == 0) {
        isEven = true;
    } else {
        isEven = false;
    }
    return isEven;
}
```

Correct, but verbose

```java
public static boolean isEven(int num) {

    if (num % 2 == 0) {
        return true;
    } else {
        return false;
    }
}
```

Better

```java
public static boolean isEven(int num) {
    return num % 2 == 0;
}
```

Perfect

"Boolean Zen" Template

Replace...

```
public static boolean methodName(parameter(s)) {

    if ( <expression> ) {
        return true;
    } else {
        return false;
    }
}
```

With...

```
public static boolean methodName(parameter(s)) {
    return <expression>;
}
```

Boolean Method Call

```java
if ( isEven(number) == true ) {
    System.out.println("Even");
}

if ( isEven(number) ) {
    System.out.println("Even");
}

if ( isEven(number) == false ) {
    System.out.println("Odd");
}

if ( !isEven(number) ) {
    System.out.println("Odd");
}
```

Verbose → true == true

Preferred

Verbose → false == false

Preferred

"Boolean Zen"

in CheckVowel.java (Lec13)

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Enter a String: ");
        String str = input.next();

        // String index starts at 0
        char first = str.charAt(0);
        char last = str.charAt(str.length() - 1);

        if( isVowel(first) && isVowel(last) ) {
            System.out.print("The input " + str + " starts and ends with vowels.");
        } else if( isVowel(last) ) {
            System.out.print("The input " + str + " ends with a vowel.");
        } else if( isVowel(first) ) {
            System.out.print("The input " + str + " starts with a vowel.");
        } else {
            System.out.print("The input is " + str + ".");
        }
    }

    public static boolean isVowel(char letter) {
        // Use double equal signs to compare primitive data
        return letter == 'A' || letter == 'a' ||
               letter == 'E' || letter == 'e' ||
               letter == 'I' || letter == 'i' ||
               letter == 'O' || letter == 'o' ||
               letter == 'U' || letter == 'u';
    }
}
```