### Boolean fields

#### Overview

When using a field as a Boolean switch, it's desirable to limit the potential for bad data entry - or your own bad coding! This is similar to using a validation on a phone number field, where strict requirements must be met. Because the validation is strictly enforced, this will increase your level of confidence with regards to the contents of Boolean fields.

#### Guidelines

The following guidelines are suggested for Boolean fields.

- **Use a numeric** field type
- **Prefix the field name with a Boolean word such as** `is` or `isNot` (e.g., `isStudent` or `isCustomer`)
- Specify the field to **Always validate** (image 1) and **turn off** "Allow user to override during data entry" (this assumes your users won't be directly interacting with the field and will likely be using a script or some type of trigger)
- Specify a calculated validation similar to the following (image 2)
  
  ```
  Self = 1 or Self = 0
  ```

- Specify a custom message similar to the following
  
  **Internal developer message: This value should be Boolean!**

  (this means if users actually see this message in production, you've done a **bad job of coding** 😞 (image 3)

#### 1/0 vs. 1/Null

This best practice assumes you're **not** attempting to HIGHLY OPTIMIZE your data storage and you're Ok with storing a zero (0) for a false value. If you're using a 1/Null setup, then you'll need to adjust the validation calculation accordingly.

![Image #1](image1.png)

**Validate data in this field:**
- **Always**
- **Only during data entry**
- **Allow user to override during data entry**

![Image #2](image2.png)

**Validation for field "isBoolean" =**

```
Self = 1 or Self = 0
```
Boolean switching via scripting

There are a number of ways to switch a 1 to a 0 and a 0 to a 1. The most efficient is to use the `xor` operator. Here's an example:

```
$isStudent xor 1
```

or to use the standards specified on this site, the more readable is this

```
$isStudent xor True
```