# Informational and Conversion Functions

This exercise will test your understanding of informational functions. You will need to create expression rules to create each of the functions below. For exercises that don’t require the creation of a new rule, simply create a new expression rule called **testRule** and use it to perform the tasks listed.

**Exercise 1: cast()**

Use the cast function to convert text to a user data type.

1. Create a rule input called ‘username’ of data type text
2. Use the **cast** function to convert values from the username rule input into a user data type
   1. Hint: You can find the appropriate number for the User data type by using the typeof() function and passing in a rule input of data type user
3. Provide a value for the username rule input and test the rule
4. Modify the expression to user the touser() function to find another way to convert a text to user data type.

*Review the Skills practice solutions sheet for the correct result.*

**Exercise 2: isnull()**

Use the isnull() function to prevent a null value from throwing an error

1. Add the following rule input and expression:
   1. Rule Input: **user** (data type User)
   2. Expression: **user**(ri!user,”firstName”) & “ “ & user(ri!user, “lastName”)
2. With the if() and isnull() functions, create a rule that allows the expression to not error-out if there is no user.

*Review the Skills practice solutions sheet for the correct result.*

**Exercise 3: displayvalue()**

Use the display value in an interface to return an image that corresponds to an rating selected from a radio button field.

The interface should have the following components:

* A radio button field that contains options for Good, Average, and Bad
* A icon displayed below the field that corresponds to the selected option. The icons should be faces available from the icon indicator function

Step-by-step

1. Create a constant called **D202\_RATING\_CHOICES** that will allow users to select from three values: Good, Average, or Bad
2. Create a constant called **D202\_RATING\_ICONS** that will return an array of three text values. These text values will refer to options within the Indicator Icon function that allow you to return different faces (happy, indifferent, or sad) depending on which one is chosen. Make sure the values for the constant are as follows: FACE\_HAPPY, FACE\_INDIFFERENT, FACE\_SAD.
3. Create an interface called **D202\_Ex1\_3** that will display two components: a radio button to allow users to select a rating, and a display field that shows the corresponding icon.
   1. Create a rule input of type text called “rating” to store the rating selected by the user
   2. Use a form within the interface and add the radio button and image display fields. *Note: You can use either the design or expression view to create this interface, but the solution will use the expression view.*
   3. Modify the choice values and choice labels of the radio button field to reference your constant, and use the rule input for the value and saveInto
   4. In the image field, add a document image using the a!documentImage() function. Then add a!iconIndicator() to reference an icon to display. Inside of this function, use display value to match the values of your choices constant to the list of images constant. *Hint: Make sure to add a default value as another image for when no value of the radio button has been selected. For example, you can use another icon indicator, such as STATUS\_WARN.*
   5. *Stretch Goal: Update your interface so that the image field is hidden if no rating has been chosen.*