

# Functions - examples

## ■ **isFilled(field)**

- - Returns true if the data element/variable is filled, otherwise false
- - isFilled(DTM/C507[\_2005='137']/\_2379) => true/false

## ■ **string-length(string)**

- - Returns the length of the string
- - string-length(„String“) => result: „6“

## ■ **matches(string content, string pattern)**

- Checks given value with a regular expression pattern
- matches(C506/\_1154,'^[1-9]([0-9]+)?\$') => true/false

## ■ **concat(string, string, ...)**

- - Combines all parts to one string
- - concat(„This “,“is “,“an “,“example“) => „This is an example“

# Functions - examples

- **getInputParameter(string parameterName)**
  - - Reads in compliance component parameter
  - - getInputParameter(„Param\_1“)
- **getInputParameterOrDefault(string parameterName, string defaultValue)**
  - Reads in compliance component parameter, if empty the default value is used
  - getInputParameterOrDefault(„Param\_1“,“Not\_Found“)
- **isInputParameterSet(string parameterName)**
  - Checks if the parameter is set
  - isInputParameterSet(„Param\_1“) => true/false
- **isValidEdifactTime(string date, string format qualifier)**
  - Validates the given date string with the EDIFACT format code
  - isValidEdifactTime(., "101") => true/false

# Functions - examples

- **asEdifactTime(string date, string format qualifier)**
  - Converts a date string to a date object according given EDIFACT format code
  - asEdifactTime(C507/\_2380,C507/\_2379)
- **parseDate(string date, string date pattern)**
  - Converts a date string to a date object according to the format string
  - parseDate(head/messageDate,"yyyyMMddHHmmss")
- **current-dateTime()**
  - Returns a date object of the current date/time
  - current-dateTime()
- **before(date object, date object)**
  - Compare two date objects, the first date has to be previous to the second
  - before(\$Date\_1,\$Date\_2) => true/false

# Functions - examples

- **lower-Case(string)**
  - Returns given string as small letter value
  - lower-Case(„ABC“) => „abc“
- **upper-Case(string)**
  - Returns given string as capital letter value
  - upper-Case(„Abc“) => „ABC“
- **number(numeric string)**
  - Converts a numeric string to a mathematical number
  - Needed for numeric operations e.g. calculations
  - Cuts leading zeros and zeros after the decimal separator
  - number(„0123.456000“) => 123.456

# Functions - examples

- **not(expression)**
    - Inverts the Boolean value
    - not(true()) => false
  - **false()**
    - Returns the Boolean value false
    - false()
  - **true()**
    - Returns the Boolean value true
    - true()
- **More functions and detailed documentation can be found in the Guideline Tool Help Contents:** Guideline Designer > 5 Schema Overlays > 5.5 Assertions > 5.5.4 Function Reference