

## Text Component — A Brief Summary

**Range of values:** any string of characters

**Initial value:** "" (the *empty string* of characters)

**Assignment operator:** =

### Relational operators:

==	(equal)
!=	(not equal)
<	(less than and not equal)
<=	(less than or equal)
>	(greater than and not equal)
>=	(greater than or equal)

### Input and output:

Assume that `input` is an object of type `Character_IStream`, that `output` is an object of type `Character_OStream`, and that `t` is an object of type `Text`.

- To input a value for `t` use `input >> t`.
- To output the value of `t` use `output << t`.

Note: For `input >> t`, the value of `t` becomes the string of characters from the stream *input up to and not including* the next newline character, or, if there is no newline character in the remaining stream, up to the end of the stream. The newline character is removed from `input` and discarded.

### Conversion operators:

Assume that `t` is an object of type `Text`.

- To convert the value of `t` to a `Boolean` use `To_Boolean (t)`.  
Note: The value of `t` must be optional white space followed by 0 or 1 followed by more optional white space.
- To convert the value of `t` to a character use `To_Character (t)`.  
Note: The value of `t` must contain *one or more characters* and the value of `To_Character (t)` is the left-most character in `t`.
- To convert the value of `t` to an integer use `To_Integer (t)`.  
Note: The value of `t` must be optional white space followed by a string of characters that represents an integer value (including an optional leading + or -) followed by more optional white space.
- To convert the value of `t` to a real use `To_Real (t)`.  
Note: The value of `t` must be optional white space followed by a string of characters that represents a real value (including using scientific notation) followed by more optional white space.

### Important Note:

Later, there will be additional reading material for the Text component.