Referencing variables

**A variable can be referenced with double curly braces. Just like you do in an App.**

Prepared for {{Company}}

They can be bulleted and numbered.

* {{Company}}
1. {{Company}}

For number and date inputs, the format from the App can be carried into the document with :formatted.

Number input without :formatted: {{ExpenseAmount}}

Number input with :formatted: {{ExpenseAmount:formatted}}

Date input without :formatted: {{Date}}

Date input with :formatted: {{Date:formatted}}

[Metavariables](https://support.checkbox.ai/hc/en-us/articles/900000418466-List-of-Meta-Variables) can be used directly in documents. These are variables that aren’t created in the App.

 Prepared by {{[[current\_user\_name]]}}

Adding conditions

Conditions can be applied so that sentences, paragraphs or clauses only appear if that condition is met.

{{#Subscription == "Yes"}}{{Company}} is providing a subscription plan.{{/}}

{{#Subscription != "Yes"}}{{Company}} is not providing a subscription plan.{{/}}

Starting expression example: 

Ending expression example: 

The "!=" expression can be used to denote "not equals to". For a full list of Checkbox Operators, click [here](https://support.checkbox.ai/hc/en-us/articles/360024380253-Operators).

The highlighting on the conditions won't be shown to the end-user but helps us identify any conditions.

***Note:*** *To format variables, the curly braces need to be formatted to. E.g. if we use {{Company}}, note how the variable in the generated document is not green.*

**All Form elements are below**

Radio, Dropdown and Text: {{#Subscription == "Yes"}}Subscription made{{/}}

Checkbox: {{#CBX\_ProductOne == "TRUE"}}[x] {{/}}{{#CBX\_ProductOne == "FALSE"}}[ ] {{/}}

Number and Slider Input: {{#ExpenseAmount <= 1000}}Less than or equal to 1000{{/}}{{#ExpenseAmount > 1000}}Greater than 1000{{/}}

**Conditions can be nested as well. The outer most condition must be met for the inner conditions to be considered.**

{{#Subscription == "Yes"}}{{Company}} is purchasing the {{#SubscriptionPlan == "Starter"}}Starter{{/}}{{#SubscriptionPlan == "Standard"}}Standard{{/}}{{/}} plan.

***Note:*** *Yellow highlighting is used to identify the inner conditions which are contained within the green highlighted condition.*

**AND and OR expressions can also be used to accommodate complex logic.**

{{#Subscription == "Yes" && (SubscriptionPlan == "Starter" || SubscriptionPlan == "Standard")}}{{Company}} is purchasing the Basic plan.{{/}}

**Conditions can be dependent on whether a field is null, (i.e. whether an optional field has been filled out).**

{{#!!OtherTerms}}{{OtherTerms}}{{/}}{{#!!!OtherTerms}}There are no other terms.{{/}}

***Note:*** *Blue highlighting is only used here to emphasise that there is another condition, independent of the green highlighted condition.*

Formatting with conditions

**For conditional paragraphs of mutually exclusive options (e.g. dropdown options)**

{{#SubscriptionPlan == "Starter"}} Starter Plan description can be displayed here for the end-user to see. {{/}}{{#SubscriptionPlan == "Standard"}} Standard Plan description can be displayed here. {{/}}

No spaces should exist between Product A's ending expression and Product B's starting expression. Otherwise, as it's not within either condition, it will always appear and create unwanted space.

**For conditional paragraphs of multi-select options (e.g. checkboxes)**

{{#CBX\_ProductOne == "TRUE"}} Product One's description can be displayed here for the end-user to see.

{{/}}{{#CBX\_ProductTwo == "TRUE"}} Product Two's description can be displayed here for the end-user to see.

{{/}}

Enter lines are used inside the condition (before ProductOne's ending expression) so it creates space for Product Two's description.

**"Keeping lines together". To keep multiple lines of text on the same page (e.g. header and paragraph), the "Keep Lines" function can be used.**

Header

Paragraph paragraph paragraph paragraph paragraph paragraph paragraph paragraph paragraph paragraph paragraph paragraph paragraph.

It's difficult to keep formatting consistent when generating dynamic documents. We may not want our information (paragraphs/headers) to be split across pages. Use the Microsoft Word functionality of "Keep Lines Together" to ensure content is not split.

Reach out to your CSM for more details on how this can be achieved.

Using Conditions in Tables

**Conditions can be easily used in tables, the same way it's used in a document.**

|  |  |
| --- | --- |
| **Product** | **Product Description** |
| Product A/B | {{#Product == "Product A"}}Product A description can be displayed here for the end-user to see. {{/}}{{#Product == "Product B"}}Product B description can be displayed here. {{/}} |

**If a condition is used across multiple cells, the entire row/s will be conditional.**

|  |  |
| --- | --- |
| **Product** | **Product Description** |
| {{#Product == "Product A"}}Product A | Product A description can be displayed here for the end-user to see. {{/}}  |
| {{#Product == "Product B"}}Product B | Product B description can be displayed here. {{/}} |

Using LOOPs

**Repeating/Looping a sentence based on a LIST. A LIST variable can be created with the LIST page, a MERGE, or a Table with "User Generated Rows".**

{{LOOP(TBL4\_A2)}} The services provided include {{TBL4\_A2.item}}
{{ENDLOOP(TBL4\_A2)}}

**Looping a part of a sentence**

The services provided include: {{LOOP(TBL4\_A2)}}

1. {{TBL4\_A2.item}}{{ENDLOOP(TBL4\_A2)}}

Using LOOPs in Tables

**LOOPs can be used Tables so that each item appears in its own row.**

|  |  |
| --- | --- |
| **Service** | **Service Cost** |
| {{LOOP(TBL4\_A2)}}Service {{TBL4\_A2.item}} | {{TBL4\_B2:formatted.item}} {{ENDLOOP(TBL4\_A2)}} |

Formatting can be carried through for items that are formatted in the App.

NOTE: The row will only loop through depending on the number of the items in list "TBL4\_A2".

*E.g. If "TBL4\_A2" had 2 items but the inner list, "TBL4\_B2", had 3 items, only the first 2 items of "ServiceCosts" would appear.*