This is a docx cheat sheet of the fusion ability and operations that can be performed on the docx template

# How to use the fusion system

There is 3 ways to insert operations into a docx template :

* The insertion of a comment
* The inlining
* The docx property

## Insertion of a comment

You can insert operation inside a comment. The advantage is that the document can be written exactly as the target document, the inconvenient is if there is a lot of commented section, it will be harded to read. Then, you add comment to replace parts of the document that need to be replaced.

Example

Dear Mr Wayne,

We would like to inform you….

On fusion, providing the object person with a title and a last name, the new document will be generated.

## The inlining

You can directly insert part of document to be replaced by using the ${…} pattern. This has the advantage to be easy to use but the document will not seems as the target document.

Example

Dear ${title} ${lastname}

We would like to inform you….

## The docx property

And lastly you can use the document custom property. It is more complicated to insert into the document. But then it is more readable. To be processed the property should start with ‘#’.

Example

Dear Mr Wayne,

We would like to inform you….

For all these example, after providing the object person with ‘title’ Ms and ‘lastname’ Moneypenny will be merged as :

Example

Dear Ms Monnypenny,

We would like to inform you….

# Text opérations

The text operations are the following, to use them you can use the prefixe ‘t’ or ‘text’.

* *text.value(‘replacement text’) :* this will replace the text with the value in the parenthesis

|  |  |
| --- | --- |
| *Data:*  {  "val" : "blue"  } | *Template:*  *The color of the sky is red* |
| *Result:*  *The color of the sky is blue* |

* *text.show(boolean) :* this will show the text according to boolean
* *text.hide(boolean) :* this will hide the text according to boolean

|  |  |
| --- | --- |
| *Data:*  {  "isWoman" : true  } | *Template:*  *Welcome Madame Mister President* |
| *Result:*  *Welcome Madame President* |

* *text.hide() :* this will hide the text, usefull when using other operation on a part of text that we want to remove after. In this exemple, we set the variable *title* to ‘Madame’ if *isWoman* is true and ‘Mister’ if *isWoman* is false. On peut ensuite cacher l’instruction setTitle avec ‘and t.hide()’

|  |  |
| --- | --- |
| *Data:*  {  "isWoman" : false  } | *Template:*  *setTitleWelcome Madame President* |
| *Result:*  *Welcome Mister President* |

* *text.join(list of string) :* this will join the given string
* *text.join(list of string, separator) :* this will join the given string with the given separator
* *text.join(list of string, separator, lastSeparator) :* this will join the given string with the given separator and for the last word use the last separator

|  |  |
| --- | --- |
| *Data:*  {  "ducklets" : [  "Riri", "Fifi", "Loulou"  ]  } | *Template:*  *Donalds nephews are the ducklets*  *Donalds nephews are the ducklets* |
| *Result:*  *Donalds nephews are Riri, Fifi, Loulou*  *Donalds nephews are Riri, Fifi and Loulou* |

* *text.join(strings) :* this will concat the given strings

|  |  |
| --- | --- |
| *Data:*  {  "title" : "Mr"  "name" : "Alligator"  } | *Template:*  *Welcome Mister Wayne* |
| *Result:*  *Welcome Mr Alligator* |

# Paragraph Opérations

A paragraph is a piece of text between two carriage return. If you display the space characters, it is text between two ¶. To use them you can use the prefixe ‘p’ or ‘paragraph’.

* *paragraph.show(boolean) :* this will show the text according to boolean
* *paragraph.hide(boolean) :* this will hide the text according to boolean

|  |  |
| --- | --- |
| *Data:*  {  "isAgent" : false  } | *Template:*  *This paragraph is for everybody.*  *This should show only for agent.* |
| *Result:*  *This paragraph is for everybody.* |

* *paragraph.hide() :* same as text.hide(), provide a way to do operation without interfering with the final word document.

|  |  |
| --- | --- |
| *Data:*  {  "isWoman" : false  } | *Template:*  *setTitle This will be fully removed.*  *Here this one is unaffected.* |
| *Result:*  *Here this one is unaffected.* |

* *paragraph.repeat(list of objects, variable name) :* This will repeat the paragraph for every object in the list and provide a variable to access the object every time

|  |  |
| --- | --- |
| *Data:*  {  "people" : [  {"name" : "Sam", "firstname" :"Sung"},  {"name" : "Burger", "firstname" :"Joe"}  ]  } | *Template:*  *Here are the people invited :*  *- John Wayne* |
| *Result:*  *Here are the people invited :*  *- Sung Sam*  *- Joe Burger* |