

This is your Cheatsheet!

It's as simple as using:

"\$" for Event Commands

"@" for Dialog Options

"%" for Sections

"&" to Skip to the Next Line

---

So, what does it look like?

Well, like this:

For Normal text: "<NAME>: <SENTENCE>"

For Events: "\$ <EVENT\_NAME> <PARAM\_VALUE>"

For Options: "@ <OPTION\_TEXT>"

For Sections: "% <IDENTIFIER> <SECTION\_NAME>"

For Skipping: "<NORMAL\_SENTENCE>  
&"

---

Let me explain them in more detail.

For example:

Events, we have an EventManager that has been imported. EventTypes is an enum associated with it, though this name will differ from project to project. You can find them in a separate file with that exact name. You need to type exactly one of these enum values, otherwise it won't trigger the event. You can add any value after that.

For example, a float "\$ SET\_TYPE\_TIME 0,01".

Floats should be written with a comma and a leading zero if it's less than 1.

For example, a bool "\$ SET\_INTERACT\_STATE true".

Booleans should be lowercase, and they are pretty straightforward.

For example, a string "\$ SET\_SPRITE angry\_willem".

Strings should be exactly as you can find them in a list or similar and they are also quite simple.

---

Sections are pieces of dialogue where you can easily navigate based on the chosen option. But they can also be executed as commands. I'll explain options next, but for now, the rest!

For Sections, we have "jump," "start," "end," "wait" and "stop." These are the <IDENTIFIERS>.

use "jump" if you want to go to a section; "% jump SectionOne"

use "start" to mark the start of that section; "% start SectionOne"

use "end" when you have reached the end of that section; "% end SectionOne"

use "wait" when you want it to do nothing if the player clicks, this can be used for options, or if the dialog system is waiting for something else.

use "stop" when you want the dialog to stop entirely

Something that can be helpful when creating good dialog is `<CONDITIONS>`.

These Conditions simple to implement, but should be done with the same caution as doing Commands.

Using the proper names, as they should match the Condition in list found in the inspector of the dialog system.

Make sure to include the correct name and pay attention to capitalization because it is very important.

So, if you have `"% jump SectionOne"` in your line, the dialogue will go to the first `"% start SectionOne"` it encounters.

That's why it's important not to have multiple sections with the same name in a file.

---

Options are also straightforward, but have an extra thing.

An example of an option is:

```
"@ Haven't you killed him?"
"% jump SectionName1"
"@ I think you killed him!"
"% jump SectionName2"
```

These lines are stacked vertically, which is necessary in this case.

If you click the button that says `"Haven't you killed him?"` you will go to the section `"SectionName1."`

Conditions can also be used here.

---

Skipping is also quite simple, but you need to know that the `"&"` should be on the line below.

---

You can do a lot more cool things, such as:

Style!:

`<b>Text</b>` for **Bold**

`<i>Text</i>` for *Italic*

`<u>Text</u>` for Underline

`<color = "colorName">Text</color>` for color

But also Inline Commands!

These are very similar to the commands before, but they work in a single line.

For example: `"Unknown: I'm talking to you now $ SET_TYPE_TIME 0.1 $ BUT $ SET_TYPE_TIME 0.05 $ I can also speak slower!"`

In this case, it adjusts the typing speed to be slower only for the word "BUT," and then it resets.

You can do many things here. If you want to play a sound, you can do it in the same way, but with different parameters and a different EventType, of course.

---

You can also be quite flexible with tabs and line breaks, but try to keep it to 1 tab and 2 line breaks, or it might still get upset.