Alright, so. There are five basic topics discussed here, all of which you should know when writing your scripts. I tried to explain them as simply as possible but it isn't easy; some parts are still kinda complicated. **Feel free to ping and ask any (or all) @Technician for any questions**.

# The discussed topics are...

1. The basic format for dialogue
2. How to set the scene
   * This is primarily used to change the bg (background)
3. How to use the show statement
   * The most complex portion explained in the document. This is used to…
     + Bring up a character for the first time in a scene
     + Play an animation on a character
     + Focus characters during dialogue (indicate clearly who is speaking)
     + Hide characters (make them leave the scene)
4. How to play audio tracks
   * Playing music tracks in this game is fairly simple. I'll explain how to…
     + Stop existing tracks
     + Start new ones
   * Audio tracks are also easy to do.
5. More on writing dialogue
   * Escape characters
   * Font effects like italic, bold, etc.
   * Regulating text progression

**I would like to thank everyone for their time spent developing this project. Now, time to get cracking…**

One: The basics

Before continuing, you'll see me use the word "statement" a bunch in this document. For our purposes let's just say that a statement is just a line of code.

Use character codes to represent the, well, characters. Sayori is s, Natsuki is n, Monika is m, Yuri is y, MC is mc, and leave blank for a narration (MC’s thoughts). This is case sensitive.

For example: this:

Sayori: “Hello!”

Monika: “Hello!”

Yuri: “Hello!”

Natsuki: “hey”

MC: “Hello!”

Narrator: “this is dumb...”

Becomes this:

s "Hello!"

m "Hello!"

y "Hello!"

n "hey"

mc "Hello!"

"This is dumb..."

Additionally, when you want a character to say the player's name, you need to type in [player]. For example…

s "Hi [player]~"

One thing to notice is that each line is indented, using four spaces and **NOT a tab**.

Two: Setting the scene

The scene statement is used to set the, well, scene. Here’s an example:

scene bg residential\_day

with dissolve\_scene\_full

Let’s break that up, shall we.

* scene bg is used to basically tell Renpy two things: you want to change the scene, and you want to use a specific background.
* residential\_day is the name of the background image. A list of vanilla bg’s can be found in the [scripting guide](https://docs.google.com/document/d/1LCS_5lDERM5VardjJ5IDqdXp9zMVL-irespq11mJr7g/edit). Custom bg’s can be found in the “mod assets” Drive link in #mod-resources.
* with dissolve\_scene\_full is basically adding an effect for switching backgrounds. The only two effects you should worry about are dissolve\_scene\_full and wipeleft\_scene.
  + dissolve\_scene\_full brings the background to a black screen then brings in a new background, takes about a second to complete. You should use this when ending a scene and starting a new one.
  + wipeleft\_scene quickly changes from one background to another. Use this when changing rooms but don't really want to cause a big interruption in the flow of the scene.

Three: The “show” statement

The show statement is used NOT ONLY to bring a character on-screen BUT ALSO to animate on-screen characters (eg make them as hop), focus them, and hide them (move them off the scene).

# Adding characters to a scene:

First, I'll talk about adding a character to a scene. Lemme put up an example and explain:

show sayori 4p at t11 zorder 2

* show tells Renpy that you’re about to bring up an object (for your purposes it’ll mainly be a character).
* sayori specifies which character to bring up. For now, stick to sayori, yuri, monika, and natsuki.
* 4p is the code for the **body** and **facial** expressions. The number indicates the body pose, and the letter indicates the facial expression (excluding special faces, explained below). You can refer to the "expressions 'n poses" cheat sheet in #mod-resources or the [scripting guide](https://docs.google.com/document/d/1LCS_5lDERM5VardjJ5IDqdXp9zMVL-irespq11mJr7g/edit) to see the various poses and faces present.
  + IMPORTANT: There exists two kinds of body and facial expressions, “normal” and "special”. You cannot mix the two (ie you can’t put a “normal” face on a special pose, and vice-versa.)
    - The only exception is Natsuki, her normal faces work on all poses but her special "2 series" faces are *not* compatible with her special pose.
    - The last body pose for each character (5th for Sayori, Natsuki, and Monika, and the 4th for Yuri) are the special poses.
    - To use casual outfits, add b after the number (eg 4bp). For Natsuki's "2 series" faces, put b after the 2 (eg 42ba).
* at Well, not much to say here. It's just syntax, but you have to put it anyways.
* t11 indicates the character's physical location. There's three components to this:
  + The first letter (first digit) indicates the effect. The scripting guide lists the different effects but doesn't actually explain how and where they're used, so I'll take the liberty of explaining the pertinent ones here.
    - t-- is the default effect (nothing special here).
    - l-- is a sliding effect. The character will enter the scene sliding in from the left. Alternatively, use r-- to make the character slide in from right.
  + The first number (second digit) should equal the current number of on-screen characters. So this will be -1- if only sayori is on-screen, -2- if Natsuki and Sayori are on-screen, etc.
  + The second number (third digit) indicates the ordered position, left to right. This basically says where each girl is. So --1 is the leftmost, --2 is second to the left, etc.
    - Example: If Sayori and Natsuki are on screen, and Sayori is to the left, Sayori would be t21 and Natsuki would be t22).
    - If there's only 1 character, put the number 1 here (eg t11).
* zorder 2 indicates the character's zorder. The zorders can be 1, 2, or 3. zorder 2 is used most of the time. zorders 1 and 3 are used in two special cases, each of which will be explained in detail further down below.

# 

# Changing expresssions mid-conversation:

* During dialogue, you'll often want the character who is speaking to change body and/or facial expressions without the need of animations (no hop/sink effects, nor are we changing the focused character). To do that, simply insert the code for the body/facial expression in the code line.
* For example, if sayori started off with both her hands down and her :D face, and you then wanted to switch to her :o face with one hand raised, then back again to having both hands down and her :D face you would do this:

show sayori 1r at t11 zorder 2

s "Hey, [player]!"

s "How are you?"

s 2o "I heard about what you did last night…"

s 1r "Very naughty!"

# Playing animations on characters:

* At times, you'll want to make animate on-screen characters. For example, you may want to make a happy Sayori hop,or you may want to make a disappointed Monika sink. To do that, you'll need to use the show statement again (even though they're already on-scene).
* Here's an example of animating Sayori, with the key variables highlighted in blue.

1: show sayori 4r at t11 zorder 2

2: s "Hey, [player]!"

3: s "Guess what?"

4: show sayori 4r at h11 zorder 2

5: s "I found the plush toy I was looking for all week!"

6: show sayori 1j at s11 zorder 2

7: s "You could have offered some help, y'know!"

8: show sayori 3r at t11 zorder 2

9: s "But still, I'm happy I found it!"

* + On line 1, we used t-- to bring sayori into the scene.
  + On line 4, we used h-- to make sayori hop once.
  + On line 6, we used s-- to make to make a disappointed sayori sink. She stayed lowered until…
  + Line 8, when we used t-- to "un-sink" sayori and make her normal again.
* Here's a key of all the different letters you'll want to use when playing animations:
  + t-- is the default effect. Some animations are persistent through dialogue and need to be reset with t
  + s-- is a mild sink effect. Used mostly when a character is on-screen; when they are disappointed or somethin (eg Sayori's "Awwwwww..." moment). They'll stay sunk/lowered until "reset" with t--.
  + h-- is a hop. Used mostly when Sayori's happy. Also usually used on characters already on-screen. Does not need to be reset with t--
  + d-- makes the character dip. Similar to s-- except they'll almost immediately come back up, unlike s-- during which the character doesn't come back up until reset with t--.

# Focusing characters:

* When there are multiple characters on-screen, the character talking should be focused. To do this, show statements have to be used to focus whomever is speaking at the time. There's two parts to this:
  + The character who is focused should have a physical location f-- and zorder 3.
  + Any other characters should have a physical location t-- and zorder 2.
  + Every time a new character speaks, they should be focused and the character who was originally focused should be de-focused.
  + If only one character is on screen, they should not be focused (physical location t--, zorder 2)

If you forgot what physical location and zorders are, they're highlighted above in yellow.

* Do note that characters will de-focus after using any animated effect (such as a sink with s-- or a dip with d--). This means after the effect is played, you'll have to refocus them. The only exception is hop (h--). There exists a combined hop-focus effect hf-- that lets a character hop while keeping them focused (or focuses them if they aren't already.
* Here's an example: the cookie scene. Sayori on the right; Natsuki on the left:

"Sayori rapidly tears open the wrapper and takes a big bite."

show sayori at f22 zorder 3

s 4q "Sho good..."

show sayori at hf22 zorder 3

s 4o "Mmf--!"

"Sayori suddenly clasps her hands over her mouth."

s 4p "I bit my tongue..."

show sayori at t22 zorder 2

show natsuki at f21 zorder 3

n 3a "Ehehe."

n "You're going through a lot over just one cookie."

"Natsuki takes a bite of her own cookie."

show natsuki at t21 zorder 2

show sayori at f22 zorder 3

s 1c "Ah, yours looks really good too, Natsuki!"

s "Can I try it?"

show sayori at t22 zorder 2

show natsuki at f21 zorder 3

n 4e "Jeez..."

n "Beggars can't be choosers!"

show natsuki at t21 zorder 2

show sayori at f22 zorder 3

s 1h "But yours is chocolate..."

show sayori at t22 zorder 2

show natsuki at f21 zorder 3

n 4c "Yeah, why do you think I gave you that one?"

# Hiding characters:

Hiding a character is more straightforward, compared to the convoluted mess that was everything else in this section. Hiding characters is merely a two-step process:

1. An animation must be played to fade the character out (Thanos anyone?)
2. Then you can use the hide statement to actually hide the character.

Here’s an example:

show sayori at thide zorder 1

hide sayori

* The only variables here you truly need to worry about are sayori and thide. It's fairly obvious that sayori should be replaced with the name of the character you want to remove.
* The other variable is thide. You can use thide to make them fade away, or replace it with lhide to make them slide offscreen.
* zorder should always be set to 1 as shown above.
* If you want to make the character hide instantly, just skip the first line and type only

hide sayori.

Four: Audio

# 

# Music tracks:

So in this game we have music. A real shocker, I know. Coding it in is actually kinda simple too. There's two things to do: start and stop music. I'll start by explaining how to stop music. First, an example.

stop music fadeout 2.0

* stop music Tells Renpy you want to stop the music. Simple enough. This will stop ANY currently playing music track. (It will not interrupt sound effects if there happens to be one playing at the time).
* fadeout 2.0 If you omit this variable, Renpy will stop the music immediately, creating an... unpolished sound. Adding a fadeout makes the transition smoother and more natural. 2.0 is the duration of the fadeout in seconds. Change this to whatever you please.
* NOTE: You should always use the stop music statement BEFORE playing a new track. Just trust me on that one.

Now, as for starting music, it's also really easy.

play music t2

* play music tells Renpy you want to start a music track.
* t2 is the code name of the track you want. Using [this YouTube playlist](https://www.youtube.com/watch?v=BFSWlDpA6C4&index=1&list=PLc5ZKngbAPXMG4yjq9ESGfqblQfL9g4-p) as a reference, I'll make a quick table comparing the track titles to the code names. It's in the [scripting guide](https://docs.google.com/document/d/1LCS_5lDERM5VardjJ5IDqdXp9zMVL-irespq11mJr7g/edit) but I'll narrow it down here. If any quotes appear, include them in your script.

| [Doki Doki Literature Club! (Main Theme)](https://youtu.be/BFSWlDpA6C4) | t1 |
| --- | --- |
| [Ohayou Sayori!](https://youtu.be/BUWuDdfe7v4) | t2 |
| idk but it's like the title theme | t3 |
| [Dreams Of Love and Literature](https://youtu.be/rsBeCQA93-Q) | t4 |
| [Okay, Everyone!](https://youtu.be/Hh2e9_bGSys) | t5 |
| [<Sayori's poem background music>](https://youtu.be/frHGwz7ZhmE) | t5\_sayori |
| [<Natsuki's poem background music>](https://youtu.be/BDsCNVj72ig) | t5\_natsuki |
| [<Yuri's poem background music>](https://youtu.be/VGwfIloNM8w) | t5\_yuri |
| [<Monika's poem background music>](https://youtu.be/Isuc8kwVh64) | t5\_monika |
| [Play With Me](https://youtu.be/BUKN_ySpqU4) | t6 |
| [Poem Panic!](https://youtu.be/7JDlEpvE-cs) | t7 |
| [Daijoubu!](https://youtu.be/FDG1AUN53S8) | t8 |
| [My Feelings](https://youtu.be/IgRUaVHq1Hs) | t9 |
| [My Confession](https://youtu.be/GAhiW1Z3GJY) | t10 |
| [Sayo-nara](https://youtu.be/al1BNB8bKaE) (Consider not using this.) | td |
| [Just Monika.](https://youtu.be/JC0mFHN7r2c) (Consider not using this.) | m1 |
| [I Still Love You](https://youtu.be/QIHUK68L9qQ) | mend |
| [Your Reality (Credits)](https://youtu.be/CAL4WMpBNs0) | "bgm/credits.ogg" |

# Sound effects:

This game also has sound effects. However most are glitch-related and there's only a handful that we can actually use.

To play a sound effect, it's really easy.

play sound "sfx/slap.ogg"

* play sound tells Renpy you want to play a sound effect
* "sfx/slap.ogg" is the code of the sound effect used. Note that the quotes are necessary.

There are only eight vanilla sound effects that are really usable for our purposes. [A list of the ones we'll use can be found here.](https://drive.google.com/drive/folders/159fqCSlLRRdvm_t0K767fizNygISNJDB) I also have a quick table just below listing some sound effects and their in-game codes.

| [Opening a closet door](https://drive.google.com/open?id=1a0lyLFg8FryphqSoLfrNqtQTXf9xIraP) | "sfx/closet-open.ogg" |
| --- | --- |
| [Closing a closet door](https://drive.google.com/open?id=1Zd7hDHKPd5H8F6-IFb4KHmdg1Fg3xhZg) | "sfx/closet-close.ogg" |
| [Natsuki's manga collection falling](https://drive.google.com/open?id=1FWovb1wHH-x0aniGHYhe5Z-iHHYxYPyG) | "sfx/fall.ogg" |
| [Sayori's stuff falling (Heavier sound)](https://drive.google.com/open?id=1eUb_27UTEksDZOb0pHT4kzIxT8I5f2vy) | "sfx/fall2.ogg" |
| [Creepy giggle](https://drive.google.com/open?id=1MI7IrS3WlVsci5kSOCaVKD4qIQXTgOkJ) | "sfx/giggle.ogg" |
| [Flipping a page](https://drive.google.com/open?id=1L_wJRvXrRUyJIBzWrRNuRX_x4FmDpQEr) | "sfx/pageflip.ogg" |
| [generic slap sound](https://drive.google.com/open?id=1Uuxr0i5rtnWX5F-KF9ToUKS4Jp6BD2ZQ) | "sfx/slap.ogg" |
| [More like a slam, but smack!](https://drive.google.com/open?id=1SbPRD9pBrnk7sskW_aJPrhaQ2eLgklrX) | "sfx/smack.ogg" |

Five: More on writing dialogue

In your dialogues, you can spice them up with certain effects. You can put up some font effects (such as bold, italics, yada yada yada). You can also adjust some text effects to introduce pauses, delays, etc. You can even regulate text speed. But first, there is something you need to be aware of:

# Escape characters

Not all text written in your dialogue will appear the way you want it to. Some characters are interpreted by the engine differently and will not appear in your dialogue (and can cause glitches). These characters must be **escaped (**written differently) to appear normally. Here's a quick table showing them:

| **Character** | **Escaped** | **Note** |
| --- | --- | --- |
| " | \" | The double quotes are used to mark the beginning and end of dialogue(these should not be escaped). Any double quotes included WITHIN dialogue must be escaped with \" |
|  | Example: | "Hey, \"Jan\" |
|  | Result | "Hey, "Jan" " |
| ' | \' | Similar to above but with a single quote. |
|  | Example: | "Hey, \'Jan\' " |
|  | Result: | "Hey, 'Jan' " |
| ░ | \░ | A space. I'm using ░ to represent a single press of the spacebar.. Renpy by default collapses **multiple** adjacent spaces into a single space, so to put for example three spaces, you need to write \░\░\░. |
|  | Example: | "hey\░\░\░son" |
|  | Result: | "hey son" |
| \n | \\n | Renpy interprets \n in a dialogue to mean a new line. Use \\n to display \n within your dialogue. |
|  | Example: | "Line 1\n Line 2\\n" |
|  | Result: | "Line 1  Line 2\n" |
| \ | \\ | Backslashes tell Renpy to start looking to escape characters. As such they themselves must be escaped. |
|  | Example: | "yes\\no" |
|  | Result: | "yes\no" |
| [ | [[ | **Left** square brackets by default are used to mark variables (eg. [player] ). Double the left bracket to unmark it. |
|  | Example: | "Hey, [[player]" |
|  | Result: | "Hey, [player]" |
| { | {{ | Similar to above but with **left** curly brackets. Double 'em to display one of em |
|  | Example: | "No, {{baka!}" |
|  | Result: | "No, {baka!}" |

# Font effects:

There's LOTS of these! Most of them are pretty simple too. I'll use a table to list 'em.

| *Effect & Notes* | *Example code* | *Example result* |
| --- | --- | --- |
| Bold | {b}Bold{/b} | **Bold** |
| Italic | {i}Italic{/i} | *Italic* |
| Underline | "{u}Underline{/u}" | Underline |
| Strikethrough | "{s}strikethrough{/s}" | s̶t̶r̶i̶k̶e̶t̶h̶r̶o̶u̶g̶h̶ |
| Alpha -- Controls text opacity (transparency) | "{alpha=0.5}Alpha{/alpha}" | Alpha |
| Color -- [Uses hex value](https://www.webfx.com/web-design/color-picker/) (#rrggbb) | "{color=ff0000}red{/color}, {color=#0000ff}blue{/color}" | red, blue |
| Font -- Notify a technician if you want to use custom fonts. | "{font=impact}YOU DONT SAY{/font}" | YOU DONT SAY |
| Kerning -- Space between letters | "{k=+2}kerning{/k}" | k e r n i n g |
| Plain -- cancels any special effects. | "{b}This is bold. {plain}This is not.{/plain} This is bold.{/b}" | **This is bold.** This is plain. **This is bold.** |
| Text size | "{size=+3}Big{/size}, {size=-3}smol{/size}" | Big, smol |
| Space -- adds pixels of space (not equal to the size of a spacebar press) | "Well {space=10} then" | Well then |
| Vertical space -- adds space between lines | "Line 1 {vspace=20} Line 2" | Line 1  Line 2 |

There are even more text effects but most of them are more complicated and kinda outside the scope of this mod.

# Regulating text progression:

I'm also gonna make and use a table to list the various ways to make text effects. I even made crappy gifs to illustrate them :D

| *Effect & Notes* | *Example code* | *Example result* |
| --- | --- | --- |
| Normal | "Normal" |  |
| Displays text before the tag instantly | "This is instant.{fast} This isn't" |  |
| Does not wait for a click; immediately executes the next dialogue without waiting for a click | "Line 1{nw}"  "Line 2" |  |
| Waits the specified number of seconds. If no seconds are given, waits for a click. | "Text 1 {w} Text 2 {w=1.0} Text 3" | Note: You must click to advance from "Text 1" to "Text 2"  **BUT** "Text 2" advances to "Text 3" automatically. |
| Adds a pause (in seconds). Similar to wait, but creates new lines. | "Line 1 {p} Line 2 {p=1.0} Line 3" | Note: You must click to advance from "Line 1" to "Line 2"  **BUT** "Line 2" advances to "Line 3" automatically. |

