

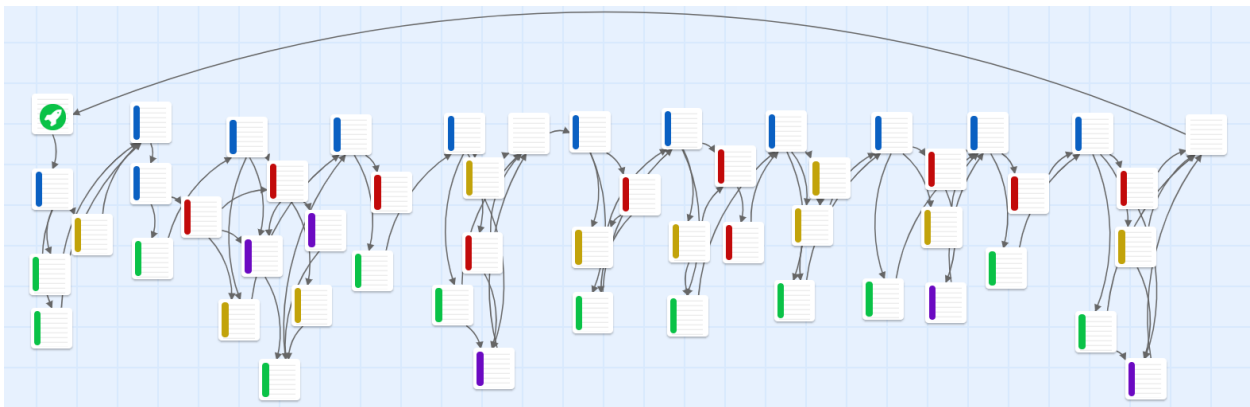
CREATING CAPTIVATING BRANCHING SCENARIOS IN 90 MINUTES

Learning Rebels: Learn Something New

June 13, 2023

Christy Tucker

christy@christytuckerlearning.com



How to Use Twine	3
Download and install Twine	3
Caution for the browser-based version	3
Create a new story in Twine	4
Scenario introduction.....	6
Add and link passages to create the branching structure.....	6
First decision point.....	8
Scenario information.....	9
Ideal Path.....	10
Tags in Twine.....	11
Story and proofing formats	13
Build and test a functional prototype	14
Play mode	14
Test mode	15
Visual Design in Twine	16
Harlowe Toolbar	16
Formatting text	16
Look and Feel in Twine	17
Page settings.....	18
Enchant macro.....	19
Passage Settings.....	20
Link settings.....	21
Background Image.....	22
Hide the sidebar	22
Collapse whitespace	23
Publish a Twine Story	24
Share scenarios on itch.io	24
Publish and prepare your story.....	24
Get started on itch.io	25
Upload your project on itch.io.....	26
Additional Resources	28

How to Use Twine

Download and install Twine

You can download and install Twine from twinery.org. Make sure you get the right version of the desktop app (Windows or Mac). If your workplace doesn't allow downloads, try the online version instead.

Windows users: Select the file named Twine-2.x.x-Windows.exe. You may see a warning about Twine being an "unrecognized app." If you see this message, select **More info**, then select **Run anyway** to run the installer.

Note for Mac users: Select the file named Twine-2.x.x-macOS.dmg. You may see a warning that the Twine app wasn't opened because it's from an unknown source. Select **Open anyway** to launch it. If you're still having trouble, try changing your OS settings to allow unsigned apps.

Note that Twine doesn't automatically update itself; you have to download and install the newest version yourself. At the time this guide was created, the current version of Twine was 2.6.0, and the current version of Harlowe was 3.3.4.

Caution for the browser-based version

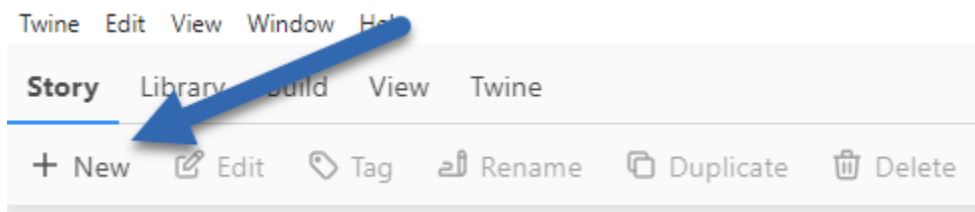
If you can install Twine locally, I strongly recommend it. While the browser-based version of Twine will work for all of the activities in this workshop, your work will **only be saved in your browser**. If you clear the saved data from your browser, your story will be gone, likely with no way of recovery.

Therefore, if you use the online version of Twine, please use the **Archive** button regularly. This will save a local copy of your work.

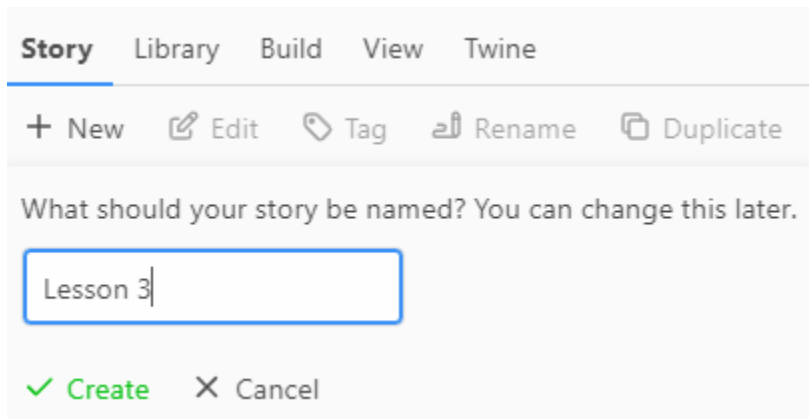


Create a new story in Twine

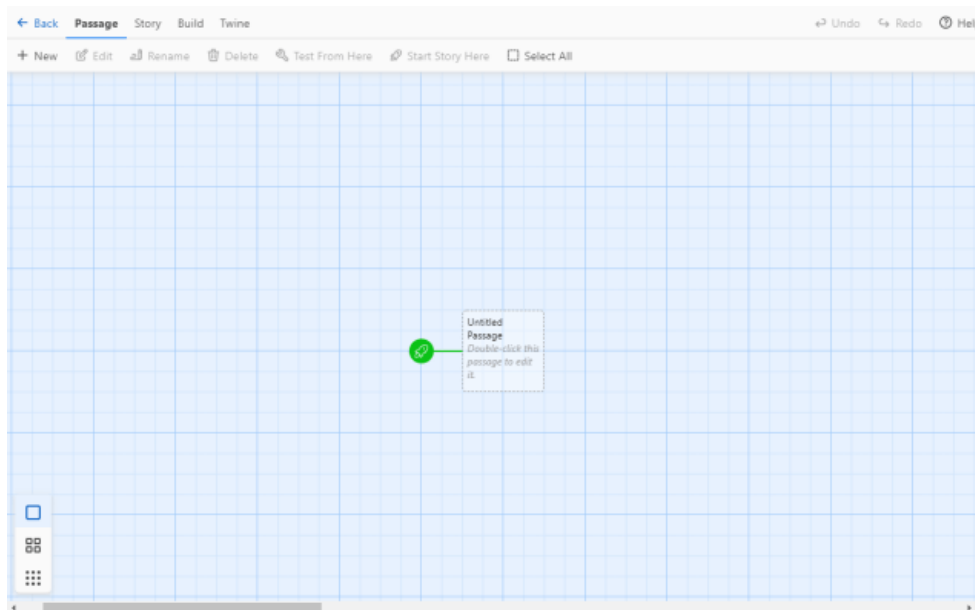
In the **Story** menu, click the **+New** button to create a new story.



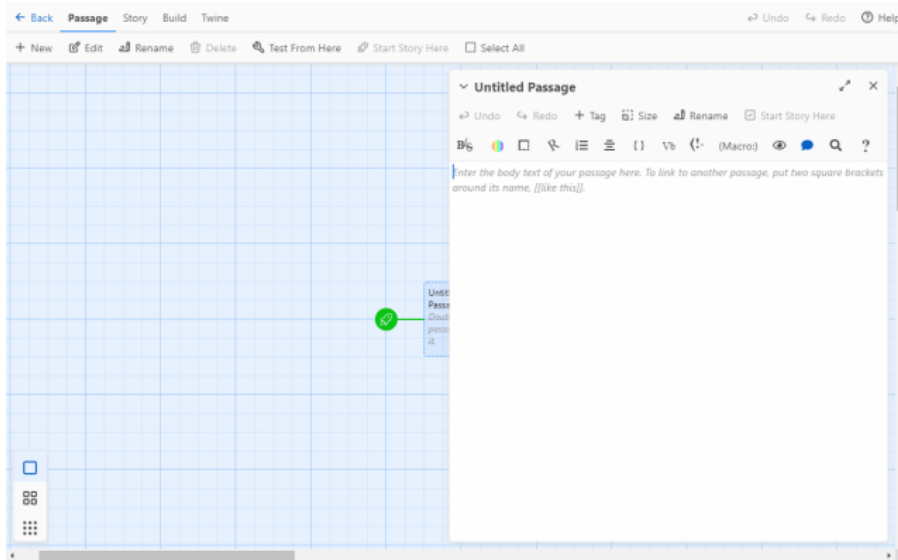
Give your story a name, then click **Create**.



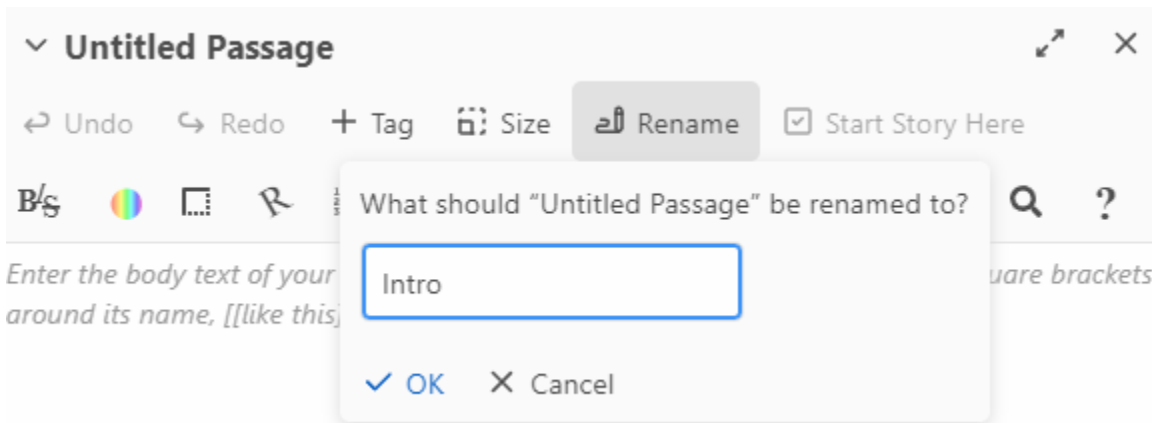
Your story will automatically open. You'll see a single Untitled Passage in the center to start.



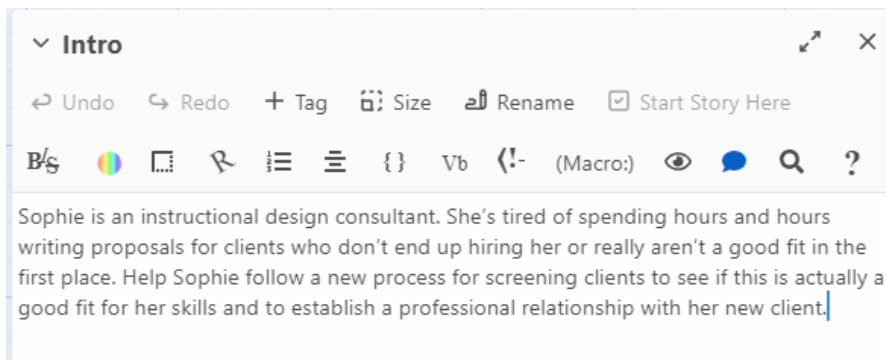
Double click the Untitled Passage to open it.



Click **Rename** to change the title from "Untitled Passage" to "Intro" or whatever you want.



Enter the introduction to the scenario in the body text section.



Scenario introduction

Copy and paste this text into your first passage.

You have recently been promoted to team lead in the call center for TaxLink, a tax preparation software company in San Diego. You're excited about the new role, but a little nervous about giving feedback to the customer service agents who now report to you.

Some recent audits of calls have uncovered issues you'll need to address with the team to improve their performance.

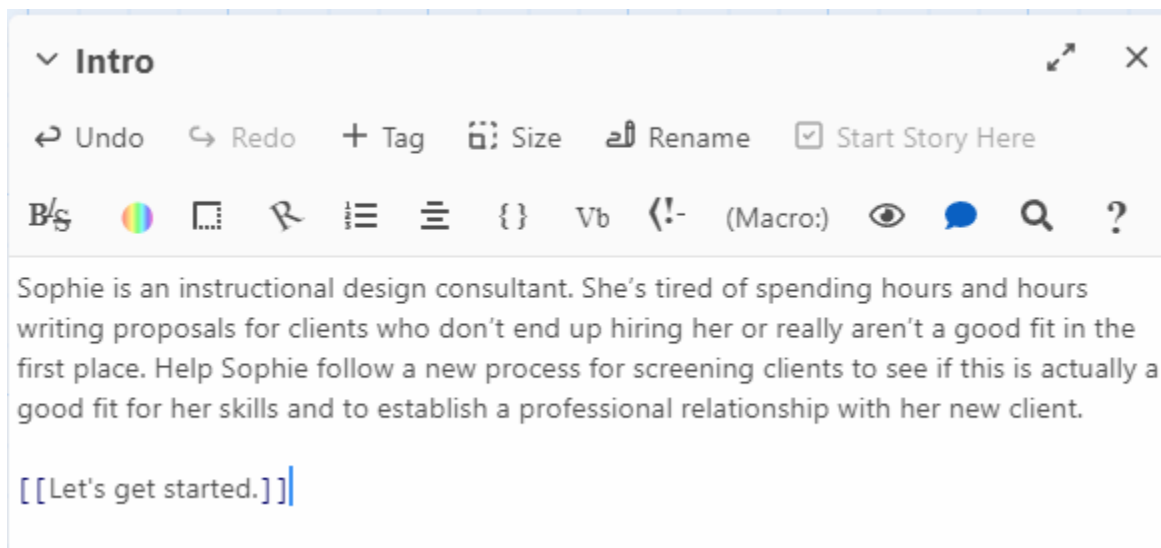
[[Let's get started.]]

Add and link passages to create the branching structure

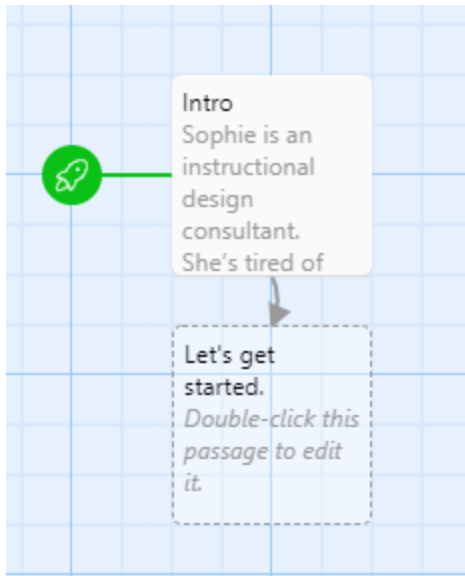
The power of Twine for branching scenarios is in the simplicity of creating additional passages with links between them. Unlike many other tools, Twine is built for nonlinear structures.

To create a link to a new passage, just type double brackets around the text of the link.

In the screenshot below, [[Let's get started]] is the new link.



When you close the first passage, you'll see a new passage with an arrow linking it to the first passage. The new passage is transparent and has a dotted outline because it's currently empty.



Repeat this process to add a question with multiple options.

The screenshot shows a story editor interface. At the top, there is a title bar with a dropdown arrow and the text 'Let's get started.'. Below the title bar is a toolbar with icons for Undo, Redo, Tag, Size, Rename, Start Story Here, Bold, Italic, Text Color, Background Color, Bulleted List, Numbered List, Link, Macro, Hide, Show, Search, and Help. The main text area contains the following content:

Sophie receives an email from a prospective client, Robert.

//Hello Sophie,

My company has 4 classroom training courses we'd like to convert to online. One of them is a half day course; the others range from one day to four days long. Can you please tell me what you would charge to convert these courses to online?

Regards,

Robert//

What should Sophie do?

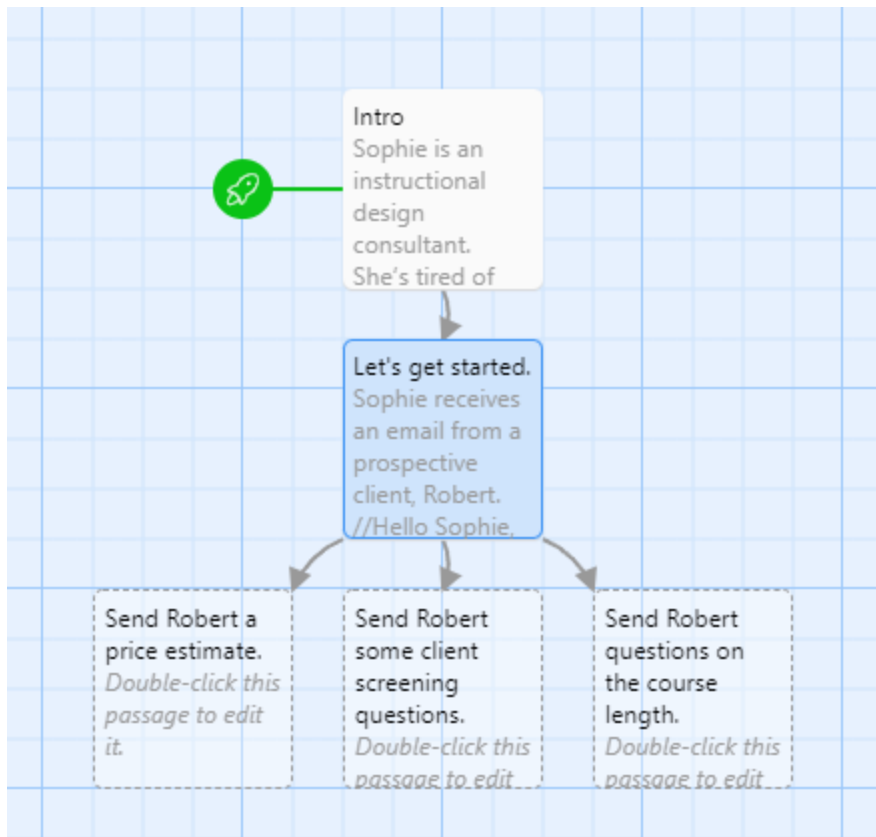
[[Send Robert a price estimate.]]

[[Send Robert some client screening questions.]]

[[Send Robert questions on the course length.]]

A blue callout box with the text '3 links to 3 new passages' has three blue arrows pointing to the three clickable options listed below.

Adding multiple links creates multiple new passages.



First decision point

Copy and paste this text to add your first decision point.

On a recent call, Mollie, a senior call center tech, was helping Jim. Jim was confused at the beginning of the call, and he was increasingly frustrated that his problem hadn't been solved even though he had called several times. Mollie started the call very cheerful, but quickly lost patience when she felt Jim's requests were unreasonable. Emotions escalated on the call until both Mollie and Jim were yelling, and Jim hung up.

How do you start your conversation with Mollie?

[[Do you have a few minutes to chat?]]

[[I scheduled a meeting so we can talk about your call with Jim.]]

[[We need to talk.]]

Different link text and passage name

The text shown in the link can be different from the passage name. You can use an arrow -> or a vertical pipe | to separate the displayed text from the passage name.

```
[[Try again->Let's get started]]
```

```
[[Try again|Let's get started]]
```

Both of the above links will display "Try again," but link to the passage "Let's get started."

Multiple links to a single passage

To link to a single passage more than once (like returning to the beginning), make sure your link target text exactly matches the passage name. Any difference (including punctuation or capitalization) will create a new passage.

Scenario information

Use this background information for drafting your branching scenario prototype.

Learning Objective	Team leads will provide constructive feedback to customer service agents to improve performance and reduce customer complaints.
Audience	New team leads in the inbound call center, especially those who have been internally promoted
Player Character	The player is a newly promoted team lead who previously worked as a customer service agent in this inbound call center. They have no prior management experience, but they do have good skills helping customers on the phone and chat.
Additional Character	Mollie: A senior call center agent who has been working in the call center for almost 15 years. Mollie is a senior call center tech. She has good troubleshooting skills, but she gets impatient and sometimes too short with customers when she feels their requests are unreasonable.
Context	This training is for a tax prep software company in San Diego, TaxLink. In this inbound call center, team leads audit call recordings to identify problems and areas of improvement.

Ideal Path

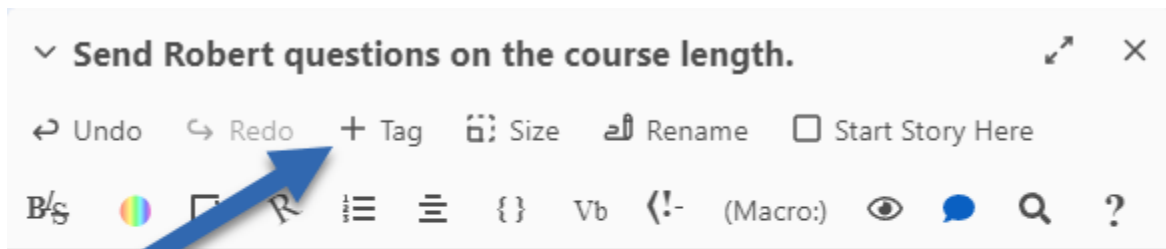
The "ideal path" is the process someone would take if everything went perfectly in the scenario.

1. Ask if the agent has time for a conversation.
2. Assure the agent that you're there to support them.
3. Ask the agent for their perspective on what happened.
4. Summarize what you heard from the agent.
5. Specifically identify the behaviors you want to change or stop, based on observations.
6. Specifically describe the desired behavior, with examples if appropriate.
7. Give the agent time to ask questions or make suggestions.
8. Thank the agent.

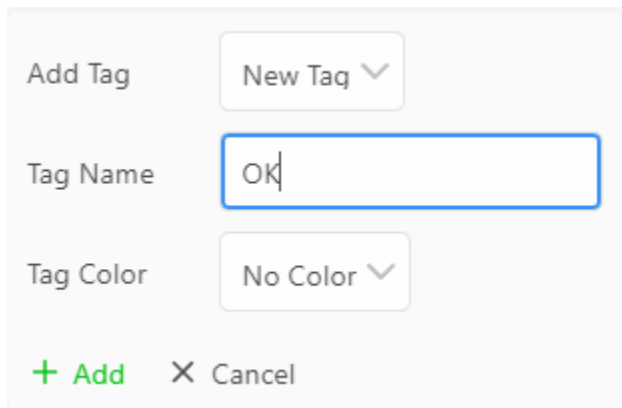
Mistakes	Consequences
Start the conversation too aggressively, making the agent immediately anxious	Agent resists feedback
After listening to the agent's description, don't acknowledge what they said	Agent doesn't feel listened to
Attack the person or their personal characteristics	Agent gets insulted, angry
Don't give the agent time to explain from their perspective.	Agent doesn't feel heard or valued, is less motivated to listen to feedback
Jump into talking about mistakes too early	Lack of trust; relationship not built; agent may resist feedback
Give vague feedback like, "That wasn't very good."	Agent is confused about what to do
Only talk about the negative behavior to stop or change, not what you want instead	Agent understands it's a problem, but isn't clear about what you want and therefore the problem continues
Rush through the feedback too quickly	Agent doesn't think the feedback is important, disregards it

Tags in Twine

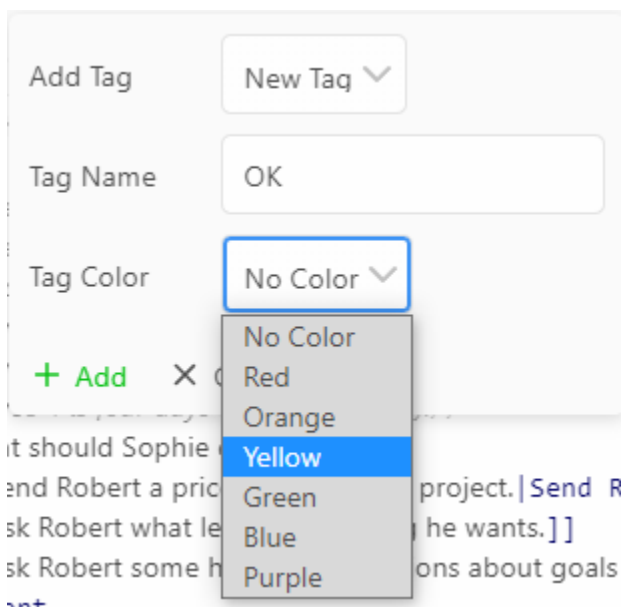
1. Click the **+Tag** button.



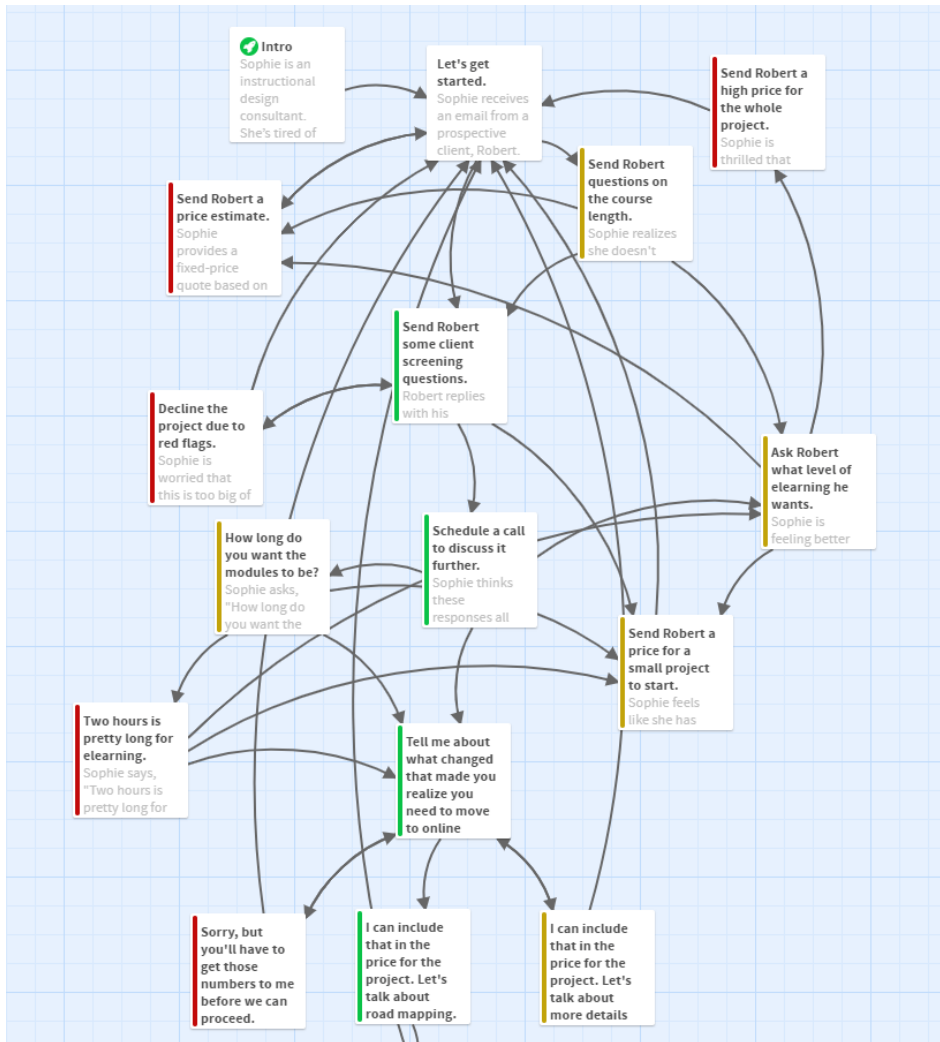
2. Type the name you want for the tag.



3. Assign a color to each tag.



This is especially helpful for the "Good," "OK," and "Bad" tags as you review the structure of the whole scenario later.



4. When you're finished, click **+Add**.

Story and proofing formats

Twine works a little differently from other elearning tools you might be familiar with. Tools like Storyline, Rise, and Captivate are mostly WYSIWYG (what you see is what you get). You work in a visual environment, and you change the layout and appearance directly along with the content.

Twine separates the content from the presentation, like using HTML to build a website. You work directly with the content, and then you apply a format on top of it. It's similar to creating a PowerPoint presentation on blank white slides with black text, and then applying a theme to automatically reformat the entire presentation.

Advantages

This gives us a couple of advantages for branching scenarios. First, when you're writing, you can focus just on the writing and the relationships between passages. You're not splitting your attention between writing and visual design.

Second, you can change the format for the entire story at once. That means you can create a basic design quickly, without having to design each and every passage.

Third, you can create different versions of your story by using different formats. For example, you can have an interactive version of the story for testing, but also a plain text export for reviewing and editing.

Story formats

Story formats control how the story is displayed. They also control functionality, so if you're using variables, macros, or more advanced features, it's best to pick a format at the beginning of development so you don't break anything by changing.

Harlowe

The default format for Twine is Harlowe. The Harlowe story format for Twine provides a number of built-in features to make it easier to enhance your stories, even if you have no experience with HTML, CSS, or Javascript. The toolbar makes these features easier to access, providing simple ways to format text, set conditional states, or manage variables. However, Harlowe has so many options that it has its own learning curve.

Consult the [full documentation for Harlowe](#) for details on what's possible. Note that the official documentation assumes some prior experience and background, rather than being a step-by-step guide (which is why I'm giving you some more detailed instructions here to get started).

All of the instructions in this workshop are based on Harlowe unless otherwise noted.

Sugarcube and Snowman

If you are comfortable working in CSS and Javascript, you may find Harlowe too limiting. Sugarcube has more flexibility if you're comfortable with programming. Snowman is another story format that relies heavily on Javascript knowledge.

While the directions in this workshop will all use Harlowe, you're welcome to use Sugarcube or Snowman if you have the skills to take advantage of their power.

Other formats

Twine also has a number of specialized story formats with different purposes and features. For example, botscripTEN and Trialogue are designed for creating chat simulations. Other Twine formats are meant for game design, with inventory and health tracking. While you probably don't need any of these to create your first branching scenario, you can check the catalog of Twine formats¹ for inspiration.

Proofing formats

Proofing formats are different from story formats. While story formats are designed for end users, proofing formats are designed for developers and reviewers.

One way I commonly use a proofing format is for sending a text-only version of a story for review. You can put this into a Word document so reviewers can use Track Changes.

Build and test a functional prototype

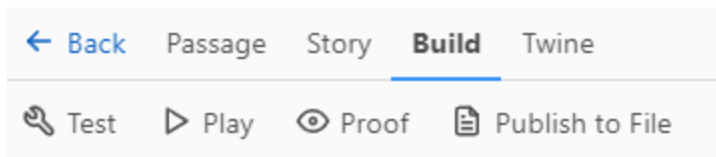
In addition to the ease of creating links and new passages, the other huge time saver with Twine is that those links are functional. You can preview and test your branching scenario.

Play mode is perfect to just click through your entire scenario. **Test** mode is better if you are troubleshooting specific issues. Play mode is usually fine for basic branching scenarios and when you're just getting started.

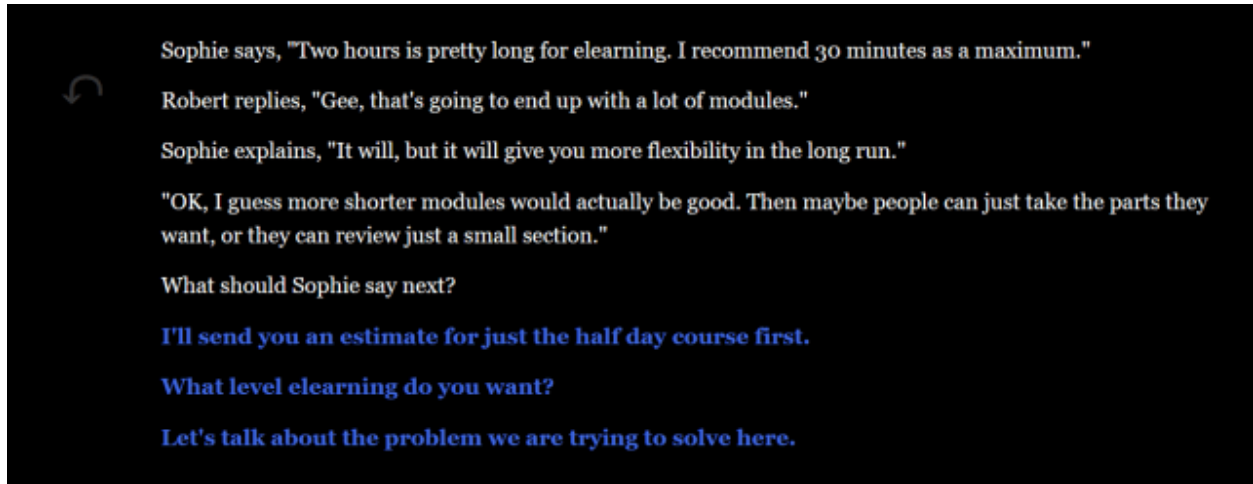
Play mode

To walk through your entire prototype so far, click **Build**, then **Play**.

¹ <http://mcdemarco.net/tools/hyperfic/twine/catalog/>

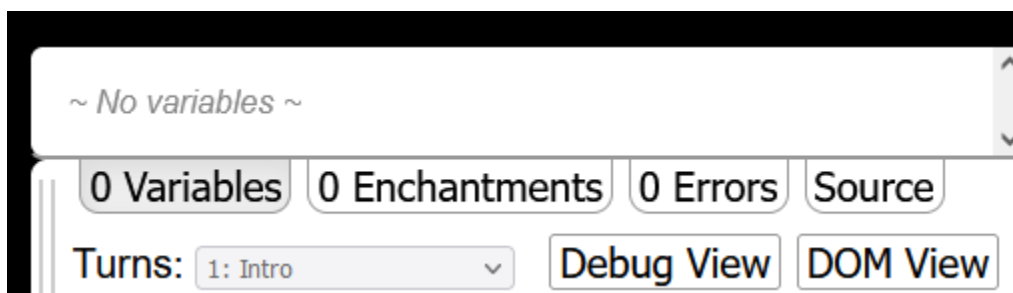


In Play mode, you will see a passage with text links, like this. By default, Twine uses a dark background and light text. (We'll change that later, but it's good enough for the first prototype.)



Test mode

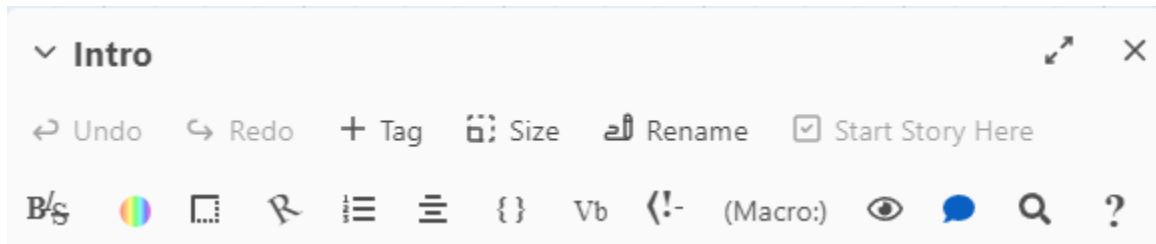
In the Build menu, click **Test** if you want to use debug mode to view where links go (very useful if the link text is different from the passage name) or are troubleshooting variables, macros, or other advanced features. Test mode will initially look the same as Play, but with additional information on variables and errors in the lower right. In test mode, you can turn on Debug view via a button in the lower right.



Visual Design in Twine

Harlowe Toolbar

The Harlowe toolbar gives you access to a number of features without having to know how to code them.

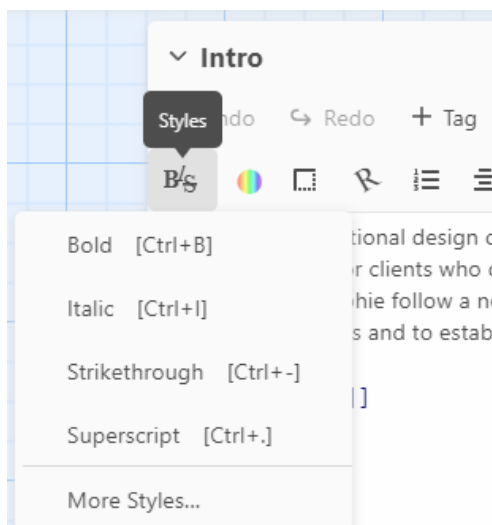


In past versions of Harlowe, a passage appears with a title, tag, and space for the passage text. If your version looks like this, you should download the update.



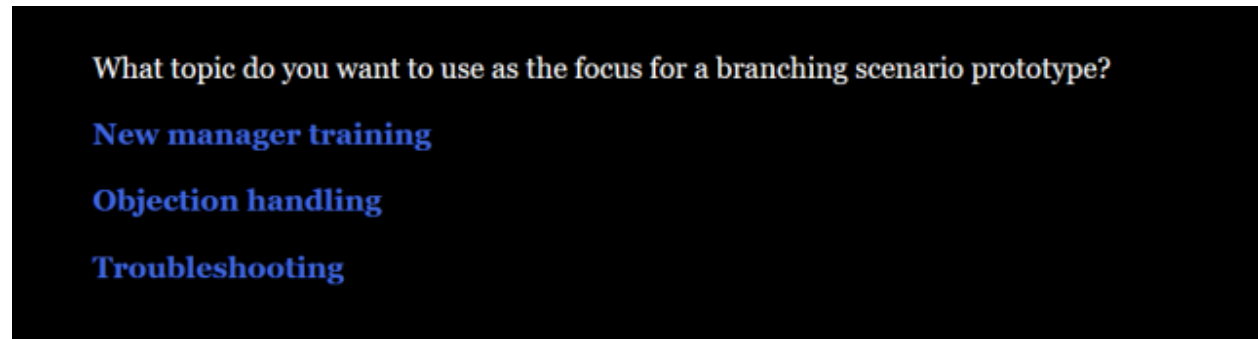
Formatting text

Formatting text with bold, italics, and strikethrough styles works like other text editing programs. You don't have to remember the markup language, but it will appear that way in the passage.

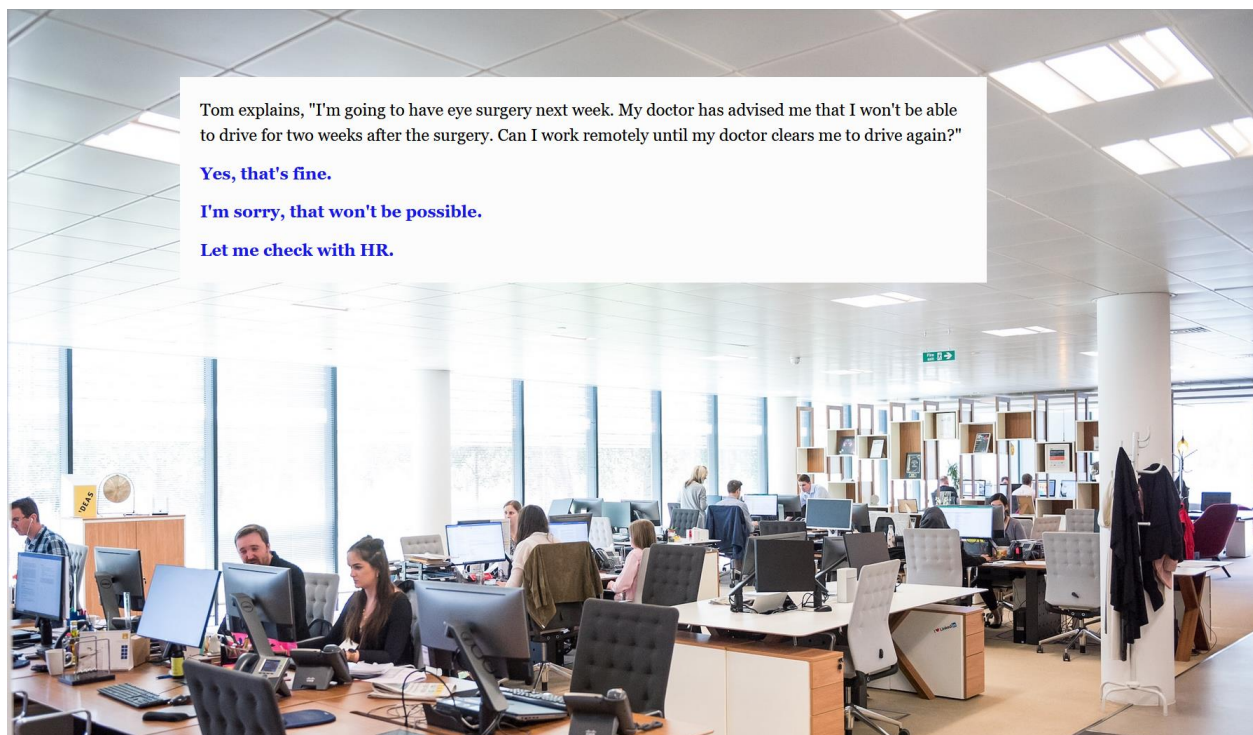


Look and Feel in Twine

If you want to change the overall look and feel of the whole story, it's a little more complicated than just formatting a short passage of text. For example, the default styling for Harlowe is a black background with white text. I don't personally like this, plus I think the blue links are hard to read.



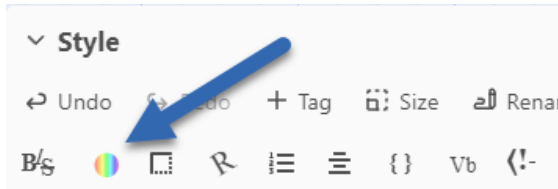
While I often use Twine for simple text-based prototypes of branching scenarios, it's also possible to create a more polished look and feel. With some work on the visual design, you can use Twine to create branching scenarios for end users without migrating the content to another tool for final development. This requires changing the formatting with "enchant" macros on three separate elements: the page, passage, and links.



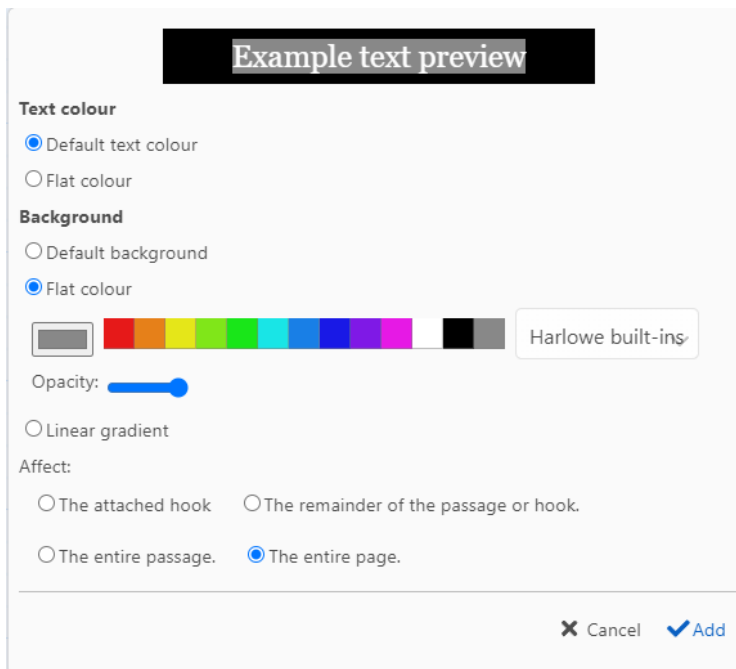
Page settings

The page settings affect the entire background of the scenario.

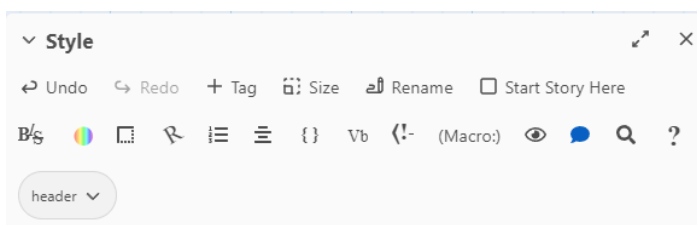
1. Create a new passage. This passage should not connect to any other passages.
2. Rename the passage "Style."
3. Select the **Colours** button on the toolbar.



4. Change the background to a flat gray color.
5. Set the color to affect the entire page.



6. Add the tag "header" to the passage. This tag adds the style to every passage in the story automatically. (Note: tags are case sensitive. "Header" is not the same as "header.")



Enchant macro

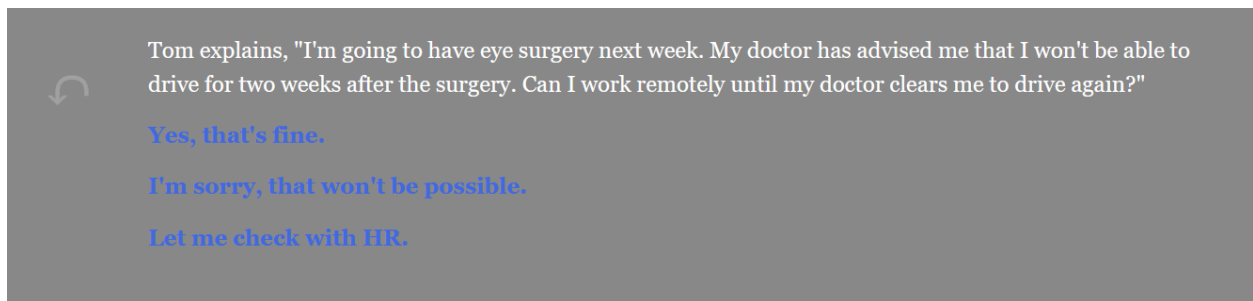
That generates the following code:

```
(enchant:?page, (bg:grey)) Your Text Here
```

Delete the placeholder "Your Text Here."

This code in Harlowe is called an *enchant*: macro. A macro is basically a little code snippet in a Twine passage. The "enchant" macro makes a persistent change to something in a passage (in this case, the background color).

The results of this enchant macro are a story with a gray background. It's not much better now, but this gives us a starting point. We'll also return to this in a little while to add an image to the background.



Could you accomplish the same thing without using the toolbar? Yes! You could enter the same "enchant" code shown below to make these adjustments. However, if you aren't used to using Twine, this format may be hard to remember. That's what Harlowe tries to accomplish; making it easier to generate these codes without consulting the documentation each time.

```
(enchant:?page, (bg:grey))
```

Passage Settings

In the passage itself, let's create a white background with black text so the links are more readable.

1. Add a new line in your Style passage.
2. Select the **Colours** button on the toolbar.
3. Change the text color and background. Use Harlowe's default black for the text and a slightly off white (#fcfcfc) for the background.

Note: To switch from RGB to Hex colors, click the small arrows next to the color details.



4. Set the **Affect** to "The entire passage."
5. Select **Add**.
6. In the resulting macro, type a + after the background color.
7. Enter this line to add some padding. This will give the text some space from the edge of the passage.

```
(css:"padding:1.2em")
```

This is the final code for the passage (with line breaks added to make it easier to read):

```
(enchant:?passage,  
  (text-colour:black)+  
  (bg:#fcfcfc)+  
  (css:"padding:1.2em")  
)
```

Link settings

To change the link color and hover, you need to add your own enchant macro.

1. Type this code.

```
(enchant:?link,  
  (text
```

2. As you start typing "text," a tool tip will appear to help you with your macro.



3. Press the down arrow until **(text-color)** is selected.
4. Press the **Tab** key to select the suggestion.
5. After the colon, inside the parenthesis, type "navy."
6. Type a + after the text color and start a new line.
7. Type (hover-style:). You can use the tool tip suggestion again.
8. The *hover-style* macro requires another macro embedded in it. In this case, we'll use text-color:blue.

```
(enchant:?link,  
  (text-color:navy) +  
  (hover-style:(text-color:blue))  
)
```

Background Image

For a branching scenario prototype, it may be easiest to skip images, or to only use a single image for the background. Unless an image is important to set the context or part of the scenario, consider skipping it.

We'll use a single photo as a background for this example.

1. Open your Style passage.
2. For the page enchant macro, replace (bg:grey) with this code.

```
(bg:"Insert Image URL here")
```

You can use relative links to a local file rather than absolute URLs but be sure to publish the image file(s) with your story file.

Hide the sidebar

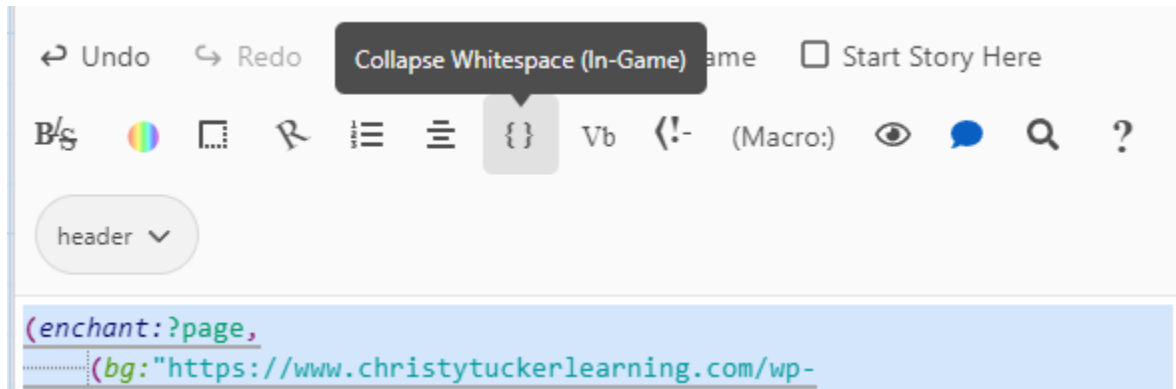
You can hide the sidebar to remove the undo and redo buttons and create a cleaner design. Use the hide macro to hide the sidebar.

```
(hide:?sidebar)
```

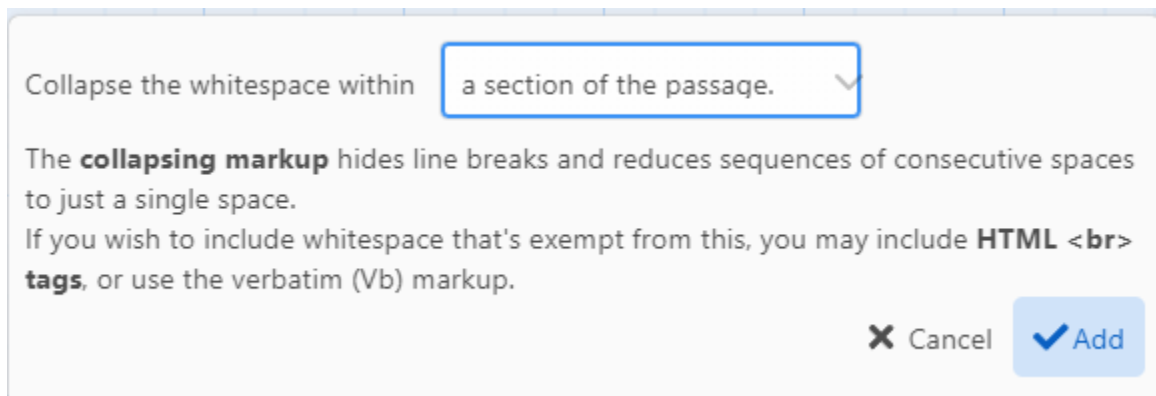
Collapse whitespace

If you have tested your scenario during this process, you may have noticed some extra white space at the top of the headers. Even though your enchant macros aren't visible to users, they still take up space in the header unless you collapse the whitespace. The extra line breaks make it easier to read your code, but they aren't so good for the final product.

1. Select all of the code in your Style passage.
2. Click **Collapse Whitespace (In-Game)**.



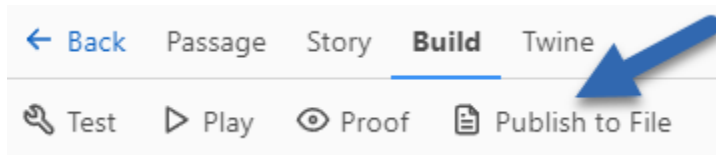
3. Leave the default selection area (you already selected the section of the passage).
4. Click **Add**.



Publish a Twine Story

A published Twine file is an HTML file that can be shared on a server for SMEs and stakeholders to review, or to an LMS that accepts HTML.

In the **Build** menu, click **Publish to file**.



You can host this html file on a server if you have your own website, or you can use a hosting service Amazon S3 or Google Cloud.

You can also share the html file in a shared drive like Google Drive, Dropbox, or Sharepoint. However, the story won't be playable if it's shared this way; users will have to download the html file and open it locally.

Share scenarios on itch.io

One popular platform for hosting Twine projects is itch.io, a platform for sharing independent games. If you don't have your own website where you can upload Twine samples for review, itch.io is a free option. You can use it for this course, or use it to send to stakeholders.

View an [example of a Twine scenario hosted on itch.io](#). Use the password "bybs" to access it.

Publish and prepare your story

1. Publish your story in Twine. If it has images, make sure that the published story and images are in the correct folder.
2. Open your story and take a screenshot. You can use this as the cover image later. Alternatively, you can create a cover image with the story title.
3. **If your story has images or other media:**
 1. Rename the story file to index.html.
 2. Create a zip file of all of your story files (i.e., the whole folder containing your story file and any media files).
4. **If your story does not have any other media**, you can leave the story file name as is.

Get started on itch.io

1. Go to <https://itch.io>.
2. The first time on the site, you need to register for an account. Check both boxes for your interests since you will be both playing and sharing content.

About you

I'm interested in playing or downloading games on itch.io

I'm interested in distributing content on itch.io

You can change your responses to these questions later, they are used to hint itch.io in how it should present itself to you.

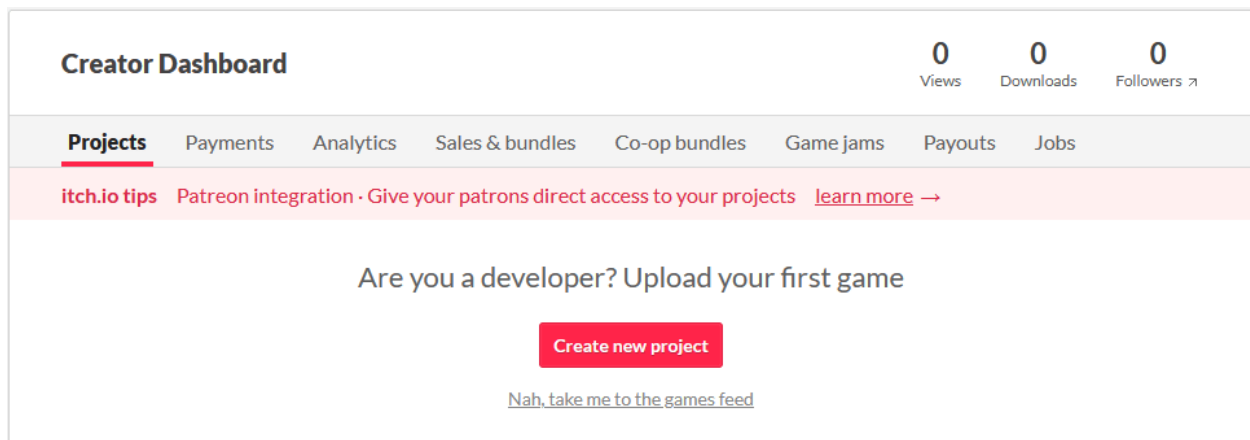
3. Despite the fact that itch.io will immediately direct you to upload your first game, you can't actually upload it until you verify your email address. If you don't get a verification email, you may have to manually request one.
 1. If you are asked about payout settings, select **Collected by itch.io, paid later**.
 2. In your **Account settings**, select **Email addresses**.
 3. Click **Send verification email**.
 4. Check your email and follow the instructions.
4. After the first time registering, you can log in with your existing account.

Upload your project on itch.io

Note: If you just registered, you will have to verify your email address prior to uploading a file. However, you can adjust other settings like the title and description while you wait for the email verification.

Check the [documentation on uploading projects](#) for additional information.

1. Click **Create new project**.



2. Add a title and short description (max 120 characters).
3. Adjust the project URL if you want.
4. Set the **Classification** to **Games**.
5. Set the **Kind of project** to **HTML**.
6. Set the **Release status** to **Prototype** (at least for your work in this course). You might select another option later in the process.
7. Set the **Pricing** to **No payments**.
8. Optional: Upload a cover image (a screenshot or title image).
9. Optional: Add other screenshots (probably not needed for showing prototypes, especially within this course).

Create a new project

You don't have payment configured If you set a minimum price above 0 no one will be able to download your project. [Edit account](#)

Make sure everyone can find your page

Review our [quality guidelines](#) before posting your project

Title

Project URL

Short description or tagline

Shown when we link to your project. Avoid duplicating your project's title

Classification

What are you uploading?

Kind of project

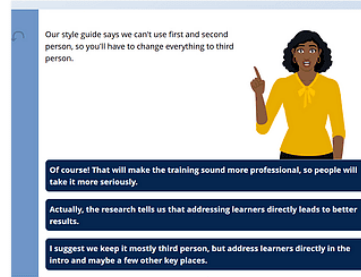
TIP You can add additional downloadable files for any of the types above

Release status

Pricing

 \$0 or donate Paid No payments

The project's files will be freely available and no donations can be made



The cover image is used whenever itch.io wants to link to your project from another part of the site. Required (Minimum: 315x250, Recommended: 630x500)

Gameplay video or trailer

Provide a link to YouTube or Vimeo.

Screenshots

Optional but highly recommended. Upload 3 to 5 for best results.

[Add screenshots](#)

10. Upload your file(s).

1. If you have media files, upload the zip file you created earlier.
2. If you have a single html file, upload that file.

11. Check this option: **This file will be played in the browser.**

12. For **Embed options**, select **Click to launch in fullscreen**. (If you know the size of your project, you can embed it instead.)

13. Add a **Description**.

14. For **Genre**, select **Educational** (or something else if that makes sense for your project).

15. Skip adding tags (we don't want people to find these prototypes).

16. If you want people to be able to add comments, select **Comments** under **Community**. Otherwise, select **Disabled**. For the course projects, it may be simpler to disable comments on itch.io and collect all feedback in Slack.

17. Set the **Visibility and access** to **Restricted**. This keeps the project unlisted.
18. Check the box to **Also allow a password to view page**.
19. Type the password. (Remember this password so you can share it with reviewers later.)
20. Click **Save and view page**.

Community

Build a community for your project by letting people post to your page.

- Disabled
- Comments – Add a nested comment thread to the bottom of the project page
- Discussion board – Add a dedicated community page with categories, threads, replies & more

Visibility & access

Use Draft to review your page before making it public. [Learn more about access modes](#)

- Draft – Only those who can edit the project can view the page
- Restricted – Only owners & authorized people can view the page
- Public – Anyone can view the page

Restricted access settings

Only people who own the project can view the page. You can give access by generating a download key. The page will be unlisted in browse and search.

Also allow a password to view page ?

••••

Save & view page

Additional Resources

Check out all of my blog posts on [storytelling and scenarios](#), including multiple posts on Twine with additional examples.

If you're interested in chatbots, these posts explain how I quickly built a chat simulation.

- [How I Built a Chat Simulation in 2 Hours](#)
- [Twine Chatbot Improvements](#)