

Quick references for Twine, from basic links to coding

- Twine works both as a web site and a full app you can install
 - Let’s use Twine in our Chrome web browser: <http://twinery.org/2/#!/stories>
- Twine can do a lot but getting started we can do a lot with the basics!
 - **Story List**, you see all your stories
 - **Story Map**, you see all the passages in your story
 - **A passage**, you can edit each portion of your story.
- Notice **links**...[[do something]] or [[open door 1->door1]] or [[sit back and wait->wait]] or [[[look up|another passage name]]
- macros... use parenthesis, for example:
 - (click:)
 - (if:) and (else:)
 - (link:)
 - (print:)
- hooks... use square brackets “[...]”
 - You’ll often be using macros and hooks together
- variables... use \$ and a name/label
 - (set: \$hasBook to true)
- named hooks... use |> or |) with a word inside like |reason> or |rhyme)... named hooks mean you can refer to a hook/block of text elsewhere
 - “>” means that hook is visible to the reader
 - “)” means that hook is hidden from the reader
- [Twine documentation](#) overview page, some handy quick references within:
 - [twine2 guide](#)
 - [harlowe reference](#)
- Using visual styles. CSS can change the look/style to your story background, text, links and more. Use your story’s main menu to **Edit Story Stylesheet** for example to change the font and overall colors of your story:

```
@import url('https://fonts.googleapis.com/css?family=Walter+Turncoat');

body, tw-story{
    font-family: 'Walter Turncoat', cursive;
    font-size: 18px;
    color:#666666;
    background-color:#ddd;
}
```

- Using media. Note with images and audio media you’ll need to save/publish your story to your computer’s disk and open your story in a web browser to preview your media. Only exception is if your files are hosted on a web site as is the case in the following examples.
 - Any media you work with is external though, not part of the all-in-one-HTML story file that Twine exports.
 - To work with media you’ll need to test your story by publishing to a file and opening that file.

- Use the tag to add an image for example:

```

```

- Use the tag <audio> to add sound for example:

```
<audio src="/location/of/your/sound.mp3">
```

- And an example from the Hunt the Wumpus story:

```
<audio id="roar" autoplay src="http://soundbible.com/mp3/Tyrannosaurus%20Rex%20Roar-SoundBible.com-807702404.mp3">
```

- (ds:) is a list <http://twinery.org/wiki/harlowe:ds>

```
(set: $itemInventory to (ds: "keys", "wallet", "flashlight"))
You walk into a room and say: (if: $itemInventory contains "keys")["Glad I have my keys."]
```

- (dm:) is a set of name-value pairs <http://twinery.org/wiki/harlowe:dm>

```
(set: $keys to (dm: "redkey", false, "bluekey", false, "greenkey", true))

Turns out I (if: $keys's "redkey" is false)[don't](else:)[do] have a red key,
(if: $keys's "bluekey" is false)[don't](else:)[do] have a blue key,
and (if: $keys's "greenkey" is false)[don't](else:)[do] have a green key.
```

- both (ds:) and (dm:) can help with inventories and lock/key mechanics: passages only available with the reader holding a key either via single variable or the dataset or datamap