

Adding More Scenes to Your Text Adventure

Goal: Turn your 3-scene story into a branching adventure with as many scenes as you like — including multiple endings — using a reusable scene template.

1. The Scene Template (copy this for every new scene)

Every scene in your game is just a *div* with a **unique *id***, the shared *class* *story-box*, and one or more buttons that call *goToScene()*.

The ***html*** below is the basic template, using *scene4* as an example:

```
<!-- SCENE 4 (Initially Hidden) -->
<div id="scene4" class="story-box hidden">
  <p class="story-text">
    Your story text goes here. What does the player see, hear, and smell?
  </p>
  <button class="choice-btn" onclick="goToScene('scene4', 'scene5')">Go deeper into the
  cave</button>
  <button class="choice-btn" onclick="goToScene('scene4', 'scene1')">Return to the
  entrance</button>
</div>
```

Paste that code block **inside** *<body>*, after your last scene and **before** *<script src="script.js"></script>*.

Important: The *.hidden* class is what makes a scene invisible at the start. Only *scene1* should be visible when the page loads — every other scene needs hidden.

2. Explanation: What Each Part Does (and What to Rename)

Part of the code	What it is	What you should change
<code><div id="scene4"></code>	The unique name of this scene	Must be unique. Never use an id more than once or the app will not work. Use <i>scene4</i> , <i>scene5</i> , <i>cave_scene</i> , <i>boss_fight</i> , etc. Do not use spacebar to put spaces in your <i>id</i> names.
<code>class="story-box hidden"</code>	Styling + starts hidden	Do not change this. These classes are needed for your CSS styling and to make the show/hide <i>JavaScript</i> work properly.
<code><p class="story-text"></code>	The narration paragraph	Do not change this but change the words between the tags (i.e. between <code><p class="story-text"></code> and <code></p></code> . The words and images between those tags makes your story.
<code>onclick="goToScene('scene4', 'scene5')"</code>	First argument ('scene4') = the scene you are leaving Second argument ('scene5') = the scene you are going to	The first argument must match this div's id. The second argument must match the id of the destination div – the new scene.
<code>class="choice-btn"</code>	Button styling	Don't change this. It allows you to style your buttons using CSS.

Button Explanation:

`<button class="choice-btn" onclick="goToScene('This is your First Argument – it should be the same as the current div's id', 'This is your Second Argument – it should be the same as the target div's id')">This is what is written on your button so change it to whatever you like.</button>`

The golden rule of *goToScene*

goToScene('WHERE_I_AM_NOW', 'WHERE_I_WANT_TO_GO')

If a button doesn't work, **99% of the time one of these two names doesn't exactly match a *div id*** (check spelling, capitals, and quotes, and don't use blank spaces).

3. Worked Example: Scene 4 → two different endings

You can add the following to your *index.html* file:

```
<!-- SCENE 4 -->
<div id="scene4" class="story-box hidden">
  <p class="story-text">
    The Robo-Mushroom's visor flickers... and it offers you a glowing spore key.
  </p>
  <button class="choice-btn" onclick="goToScene('scene4', 'scene5')">Take the key</button>
  <button class="choice-btn" onclick="goToScene('scene4', 'scene6')">Refuse and
run</button>
</div>

<!-- SCENE 5 — GOOD ENDING -->
<div id="scene5" class="story-box hidden">
  <p class="story-text">The key opens the Citadel gates. You are the Hero of the Spore
Citadel!</p>
  <button class="choice-btn" onclick="goToScene('scene5', 'scene1')">Play again</button>
</div>

<!-- SCENE 6 — BAD ENDING -->
<div id="scene6" class="story-box hidden">
  <p class="story-text">You flee into the fog... and are never seen again. The Shadow Fungus
wins.</p>
```

```
<button class="choice-btn" onclick="goToScene('scene6', 'scene1')">Try again</button>
</div>
```

Nothing will happen yet. You need to connect this new code to scene 2 or 3. Let's **connect Scene 3 to Scene 4**: find the button in scene3 and change its second argument from 'scene1' to 'scene4'.

```
<!--BEFORE – change this -->
```

```
<button class="choice-btn" onclick="goToScene('scene3', 'scene1')">Run Back
Outside!</button>
```

```
<!--AFTER – to this -->
```

```
<button class="choice-btn" onclick="goToScene('scene3', 'scene4')">Touch the robot's
visor.</button>
```

4. Checklist for Each New Scene

- ☐ New div has a **brand-new id**
- ☐ It has class="story-box hidden" (so it starts invisible)
- ☐ Every button's **first** argument = this scene's id
- ☐ Every button's **second** argument = a different scene id that **exists**
- ☐ At least one other scene has a button that **leads into** this scene (otherwise players can never reach it!)
- ☐ Test in the browser: click through every path

5. Challenges

1. **Add 3 more scenes** so your story has at least 6. At least two should be endings.
2. **Add a "lose" scene** reachable from two different scenes (multiple buttons can point to the same scene id).
3. **Add an image per scene** with `` inside a scene div.

4. **Colour-code endings** by adding CSS styles. For example, the following adds a coloured border to *scene5*'s story-box:

```
#scene5 .story-box {  
  border-color: #2ecc71;  
}
```

That is a green border for good events; you can use red #e74c3c for bad.

5. **Design challenge:** sketch your full story map (boxes = scenes, arrows = buttons) on paper first. Every arrow must have a matching button in the code. This way, you can plan a long and more complicated role-playing adventure story.