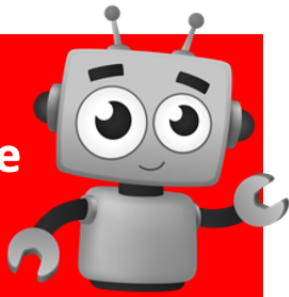


Artificial Intelligence



Adventures in HTML

- Login to replit.com with the login provided.

- **Objects** can be picked up and carried.

1. Load a **script** at the top of each page:

```
<script src="script.js"></script>
```

2. Add a couple of lines to look around and see what objects are in the room.

Use **look()** with the name of your room.

```
<div id="ROOM_NAME"></div>
```

```
<script>look("ROOM_NAME")</script>
```

3. Let your teacher know what objects you need so they can be added to objects.json and start.json.

This creates a button for every object you can see in the room (you can pick it up) or are carrying (you can drop it).

You see a key, pick it up.

You see a power gem, pick it up.

- **Lock links to other rooms.**

1. You already have **anchors** (**a** for "anchor") to connected rooms including a hypertext reference (**href**) for the room.

e.g. `go north`

2. To unlock the link when you are carrying something like a key, add

must_have(this, KEY_OBJECT); return false

e.g. `<a onclick= 'must_have(this, "a key"); return false'`

`href="room.html">go north`

- You can also make the key something that's in the room, with:

`must_have(this, KEY_OBJECT, ROOM); return false`

- You can also make the key something that's **NOT** in the room:

`must_not_have(this, KEY_OBJECT, ROOM); return false`

For example, you might not be able to pass a pit of lava.

- If you must be carry many objects at the same time (like Marvel’s infinity stones), use a comma-separated list with square brackets: `["a time gem", "a power gem", "an eternity gem"]`
- **Fixtures** *can’t be carried around, but can be changed by an object.*
 1. Add a button to show the fixture. The example fixture is ‘**lava**’.
`<button id="lava">There is lava here, you cannot pass</button>`
 2. In this example we can use ‘a power gem’ to solidify the lava. We can **use** ‘a power gem’ when we click the button.
`onclick = 'use("ROOM","a power gem","lava","a lava bridge","The lava solidifies and you are able to cross the bridge")'`
 3. Hide the button when the lava fixture disappears with **show_if**.
On the next line put:
`<script>show_if("ROOM","lava")</script>`
 4. Let your teacher know what fixtures you need so they can be added to fixtures.json and start.json.

There is lava here

```
<button id="lava" onclick='use("lava pit","a power gem","lava","a lava bridge","The lava solidifies and you are able to cross the bridge")'>There is lava here, you cannot pass</button>
<script>show_if("lava pit","lava")</script>
```

- **Properties** of the player might be “hungry” or “dead”. You can change these properties by eating or touching a fixture.
 1. Add a button to show what you can do with a “popcorn” fixture:
`<button id= "popcorn">The popcorn looks tempting</button>`
 2. Your properties **change** when you click the button. This example changes you from hungry to dead (you can leave out either “hungry” or “dead” and replace with **null**): `onclick='change("popcorn","hungry","dead","You eat the tasty popcorn and feel ill")'`
 3. Hide the button if the fixture (popcorn) isn’t in the room with **show_if**.
 4. Let your teacher know what properties you need so they can be added to properties.json and start.json.

```
<button id="popcorn" onclick='change("popcorn","hungry","dead","You eat the tasty popcorn and feel ill")'>The popcorn looks tempting</button>
<script>show_if("cinema","popcorn")</script>
```