# Treasure Hunt Game

## Description

You will build a small JavaScript game. The game represents a straight line of positions. A player moves left and right on this line trying to find a hidden treasure before running out of moves.

This exercise is meant to practice:

- functions
- loops
- arrays
- objects

Nothing more than that.

---

## General Idea

Imagine the map as numbered positions:

```
0 1 2 3 4 5 6
```

- The player starts at one position.
- A treasure is hidden at another position.
- On each move, the player goes either **left** or **right**.
- Every move uses one available move.

The game ends when:

- the player finds the treasure or
- the player runs out of moves.

---

## What You Need to Build

The game should be managed using **one main object** that represents the game state. This object should hold everything the game needs while it is running.

Your logic must be organized into **functions**, where each function has a clear responsibility.

You must also keep a **history of moves**, where every move is saved as an object inside an array.

---

## Player Movement

- The player can move only `"left"` or `"right"`.
- The player cannot move outside the map.
- Invalid moves should not crash the program.

- Every move must:
    - update the player position
    - decrease the remaining moves
    - be saved in the move history

---

## User Input

The user must be able to choose:

- the size of the map
- the number of allowed moves
- the starting position

User input must be validated. The program should handle invalid input safely.

You may use simple tools such as:

- `prompt()`
- number conversion
- basic number validation

---

## Treasure Placement

- The treasure position must be generated randomly.
- The treasure cannot start at the same position as the player.

---

## Game Execution

The game receives a list of directions and processes them one by one. After each move, a message should be printed explaining what happened.

If the game ends early (win or lose), no further moves should be processed.

---

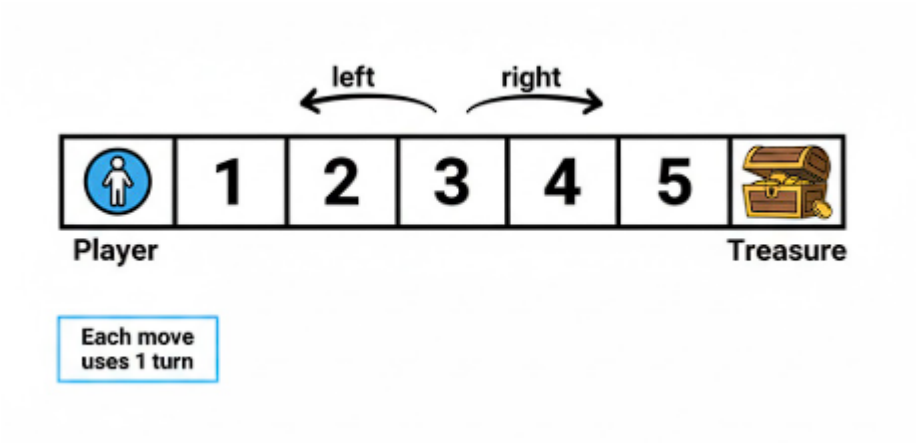
## Final Output

At the end of the game, the program should print:

- whether the player won or lost
- the treasure location (if it was not found)
- the full history of moves

All output should be shown in the **console**.

---


left
right
1
2
3
4
5
Player
Treasure
Each move uses 1 turn