

Guess Who Number

creado por

CASTILLO ESPIONOSA
MAURICIO EMMANUEL

DISEÑO Y MAQUETACION:

CASTILLO ESPIONOSA
MAURICIO EMMANUEL

TRADUCIDO POR
ALEXANDRA DIAZ

Introduction

"Guess the Number" is an interactive application developed using HTML, CSS, and JavaScript. Its goal is for the player to guess a random number between 1 and 100 that the system automatically generates. Each attempt receives feedback that guides the player, indicating if the number is too high, too low, or correct.

This manual explains in detail what each part of the code does, its purpose, and how the files that make the game work are connected.

Documents Used

This project uses two main files:

- `index.html`: Contains the game's structure (what the user sees and interacts with).
- `styles.css`: Defines the game's visual appearance (colors, fonts, alignment, and buttons).

File 1: index.html

This file contains the skeleton of the game and the JavaScript code that provides its functionality.. Below is its content and an explanation of each part:

1. General HTML Structure

```
<!DOCTYPE html>
<html lang="es">
<head> ... </head>
<body> ... </body>
</html>
```

The `<!DOCTYPE html>` declaration defines the document as an HTML page.

The `<head>` section is where configurations are added (like the title and the link to the stylesheet), and the `<body>` section holds the visible content of the game.

2. Game Header

```
<h1>Adivina el Número</h1>
<p>Estoy pensando en un número entre 1 y 100.</p>
```

Displays the name of the game and the instructions for the player. The text briefly explains the goal: guessing a number from 1 to 100.

3. Input Field and Button

```
<input type="number" id="guess" placeholder="Introduce tu intento">
<button onclick="checkGuess()">Adivinar</button>
```

- The `<input>` element allows the player to type their number.
- The button executes the `checkGuess()` function when clicked.
This function, located in the JavaScript code, contains the game's logic.

4. Result Message

```
<p id="message"></p>
```

This paragraph element is where the game's messages will appear.

- Whether the number is too high or too low.
- If the player has guessed correctly.

5. JavaScript Code

```
<script>
  let randomNumber = Math.floor(Math.random() * 100) + 1;
  let attempts = 0;

  function checkGuess() {
    let guess = document.getElementById("guess").value;
    attempts++;

    if (guess < randomNumber) {
      document.getElementById("message").textContent = "Demasiado bajo."
    } else if (guess > randomNumber) {
      document.getElementById("message").textContent = "Demasiado alto."
    } else {
      document.getElementById("message").textContent = `¡Felicidades! Adivinaste el número secreto.`
    }
  }
</script>
```

The JavaScript code gives the game its logic.

- `let randomNumber = Math.floor(Math.random() * 100) + 1;`

Generates a random number between 1 and 100. This is the secret number the player must guess.

- `let attempts = 0;`

Initializes a counter that increases every time the player attempts to guess.

- `function checkGuess() { ... }`

Defines the function that runs when the "Guess" button is clicked.

Inside the function:

- `let guess = document.getElementById("guess").value;` ; Retrieves the number the user typed.
- `attempts++;` ; Adds one attempt each time the player guesses.
- The if, else if, else conditions compare the player's number with the secret number:
 - If it's smaller, it displays "Demasiado bajo" (Too low).
 - If it's greater, it displays "Demasiado alto" (Too high).
 - If it's equal, it displays a victory message along with the number of attempts made.

Detailed Explanation of File 2: styles.css

The styles.css file defines the game's visual appearance. While the HTML file organizes the content, the CSS gives it color, shape, size, and style. Thanks to this file, the game looks appealing, centered, and organized, rather than just plain text.

1. body { ... }

```
body {  
  background-color: #e0f7fa;  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
  margin: 0;  
  color: #333;  
}
```

Defines the background, font type, and general alignment of the game.

- background-color: #e0f7fa;

Provides the background of a light blue, giving a fresh and nice sensation for the player.

- font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

Sets a light blue background color, giving a fresh and pleasant feel

- display: flex;

Changes the typeface to a modern, easy-to-read font.

- flex-direction: column;

Activates flexbox mode to easily center elements both vertically and horizontally.

- align-items: center;` and `justify-content: center;

Organizes the elements one below the other (like a column)

- justify-content: center;

Centers the items vertically within the screen/container.

- height: 100vh;

Center the elements horizontally and vertically, respectively, on the screen

- margin: 0;

Removes the browser's default margins.

- color: #333;

Defines the text color as dark gray (softer than black).

The body is fully centered with a soft background and a clean design.

2. .container { ... }

```
.container {  
  background-color: white;  
  padding: 30px;  
  border-radius: 10px;  
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
  text-align: center;  
  width: 80%;  
  max-width: 500px;  
}
```

Defines the white box where the game is displayed (text, input field, and button)

Explanation:

- background-color: white;

Sets the box's background to white, contrasting with the blue background.

- padding: 30px;

Adds internal spacing inside the box, preventing text from touching the edges.

- border-radius: 10px;

Rounds the corners of the container, giving it a modern style

- box-shadow: ...;

Adds a soft shadow underneath the box, creating a "floating" effect.

- width: 80%;` and `max-width: 500px;

Makes the box "responsive": It takes up 80% of the screen but never more than 500 pixels

wide, adapting to both large and small screens.

3. h1 { ... }

```
h1 {  
    color: #009688;  
    margin-bottom: 20px;  
}
```

Styles the main title of the game

- color: #009688;

Gives it a teal/cyan color (a characteristic color of the theme).

- margin-bottom: 20px;

Adds space below the title to separate it from the text. The title “Guess the Number” is prominently displayed at the top with a vibrant color and adequate spacing.

4. p { ... }

```
p {  
  font-size: 16px;  
  margin-bottom: 20px;  
}
```

Controls the size and spacing of paragraphs

- font-size: 16px;

Makes the text comfortable to read.

- margin-bottom: 20px;

Leaves space between paragraphs to prevent crowding.

5.input[type="number"] { ... } styles code]

```
input[type="number"] {  
  padding: 12px;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
  margin-bottom: 20px;  
  width: 70%;  
  font-size: 16px;  
}
```

Designs the field where the player types their number.

- padding: 12px;: Adds internal space within the text box.
- border: 1px solid #ccc;: Gives it a clean, subtle light gray border.
- border-radius: 4px; : Rounds the corners slightly
- margin-bottom: 20px; : Adds space between the field and the button
- width: 70%;: Ensures the field is an appropriate size relative to the container.
- font-size: 16px; : Enlarges the text inside the input to make it more visible.

The number field looks sleek, easy to use, and is aligned with the overall design.

6. button { ... } and button:hover { ... }

```
button {  
  padding: 14px 28px;  
  background-color: #009688;  
  color: white;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
  font-size: 16px;  
  transition: background-color 0.3s ease;  
}  
  
button:hover {  
  background-color: #00796b;  
}
```

Defines the button's appearance and its hover effect

Explanation:

- padding: 14px 28px;

Makes the button larger and easier to press.

- background-color: #009688;

Teal background color that matches the title.

- cursor: pointer;

Changes the cursor when hovering, indicating it is clickable.

- transition: ...;

Makes the color change smoothly when the cursor passes over it.

- button:hover { ... }

and the block:

```
button:hover {  
    background-color: #00796b;  
}
```

Darkens the button when the player hovers over it, providing visual interactivity

7. #message { ... }

```
#message {  
    font-size: 18px;  
    font-weight: bold;  
    margin-top: 20px;  
}
```

Defines how the game's result text will look (messages like "Too low," "Too high," or "Congratulations!")

Explanation

- font-size: 18px;

Enlarges the text so it stands out.

- font-size: 18px; and font-weight: bold;

Enlarges and bolds the text so it stands out.

- margin-top: 20px;

- Adds space above the element to separate it from the button.

Conclusion

"Guess the Number" is an excellent example of how the three fundamental languages of web development HTML, CSS, and JavaScript can be integrated to create an interactive, functional, and visually appealing application.

- HTML builds the game's structure and the elements the user sees and interacts with.
 - CSS provides the visual design, achieving a clean, centered interface with pleasant colors.
- JavaScript provides the game's logic, allowing the system to generate a random number, receive user attempts, and display dynamic messages based on the result.

Together, these three components ensure the game operates in an organized, fluid, and understandable manner. This project also reinforces knowledge of DOM manipulation, value comparison, event interaction, and the application of modern styles.