

JS script.js > ...

```

1 //Generate 2 random numbers: 1 for the user, and 1 for the computer
2 function generateRandomNumber()
3 {
4   const randomNumber = Math.floor(Math.random() * 6) + 1;
5   return randomNumber;
6 }
7 //Math.random() generates a random number between 0 (included) and 1 (not included)
8 //Multiplying by 6, makes the range go from 0 (included) to 6 (not included)
9 //Adding 1, makes the range go from 1 (included) to 7 (not included)
10 //Math.floor() rounds the number down to the nearest whole number
11
12 //I could also do Math.ceil() to round the number up to the nearest whole number without adding + 1 to the end of the equation
13
14 const userNumber = generateRandomNumber();
15 const computerNumber = generateRandomNumber();
16
17 // Create the <p> elements for user score, computer score, and score comparison
18 const userScoreP = document.createElement('p');
19 userScoreP.id = 'user-score';
20 const computerScoreP = document.createElement('p');
21 computerScoreP.id = 'computer-score';
22 const scoreComparisonP = document.createElement('p');
23 scoreComparisonP.id = 'score-comparison';
24
25 // Get the score-container div
26 const scoreContainer = document.querySelector('.score-container');
27
28 // Append the <p> elements to the container
29 scoreContainer.appendChild(userScoreP);
30 scoreContainer.appendChild(computerScoreP);
31 scoreContainer.appendChild(scoreComparisonP);
32
33 //Compare the 2 numbers
34 function compareNumbers(userNumber, computerNumber)
35 {
36   //Display the 2 numbers in the console
37   console.log("User number: " + userNumber);
38   console.log("Computer number: " + computerNumber);
39
40   //Display the 2 numbers in the DOM
41   userScoreP.innerHTML = "User number: " + userNumber;
42   computerScoreP.innerHTML = "Computer number: " + computerNumber;
43
44   if(userNumber > computerNumber)
45   {
46     const userScore = "You win!";
47     console.log(userScore);
48     scoreComparisonP.innerHTML = userScore;
49   }
50
51   else if(userNumber < computerNumber)
52   {
53     const computerScore = "You lose!";
54     console.log(computerScore);
55     scoreComparisonP.innerHTML = computerScore;
56   }
57
58   else if (userNumber === computerNumber)
59   {
60     const tieScore = "It's a tie!";
61     console.log(tieScore);
62     scoreComparisonP.innerHTML = tieScore;
63   }
64 }
65
66 //Display the results
67 compareNumbers(userNumber, computerNumber);
68
69 //Then, I need to ask the user for their email, check if the email address is on the list of those who can access, and print a message appropriate on the result of the check
70 const userEmail = prompt("Please enter your email address: ");
71 var emailFound = false;
72 //Create an array of email addresses
73 const emailAddresses = [
74   "matteo.did9@gmail.com",
75   "riccardo.petricca@gmail.com",
76   "stefano.cappellini@gmail.com",
77   "antonio.sagoleo@gmail.com",
78   "elisabetta.daho@gmail.com",
79 ];

```

Funzione che genera e restituisce un numero casuale

Crea due costanti: una per il numero dell'utente, una per il numero del computer

Crea un elemento p all'interno del documento

Crea una serie di elementi p, e do' a ciascuno di loro un "id"

querySelector() è un metodo che consente di selezionare e "prendere" il primo elemento che corrisponde ad uno specifico selettore CSS del DOM

appendChild() è un metodo che consente di aggiungere un nodo come figlio di un altro nodo all'interno del DOM.

A differenza di append(), non consente di aggiungere stringhe

Proprietà che consente di ottenere/modificare il contenuto HTML di un elemento

Restituisce una stringa che rappresenta il codice HTML all'interno dell'elemento

Consente di modificare il codice HTML all'interno dell'elemento

Qui invoca il metodo per comparare i numeri generati casualmente

Crea una variabile booleana e la inizializzo a false

Chiedo all'utente la sua mail

email Addresses

Crea un array

MATTEO	RICCARDO	STEFANO	ANTONIO	ELISABETTA
--------	----------	---------	---------	------------

```

82 // Create the <p> elements for user score, computer score, and score comparison
83 const userEmailP = document.createElement('p');
84 userEmailP.id = 'user-email';
85 const accessP = document.createElement('p');
86 accessP.id = 'access-paragraph';
87
88 // Get the score-container div
89 const emailAddressesContainer = document.querySelector('.email-addresses-container');
90
91 // Append the <p> elements to the container
92 emailAddressesContainer.appendChild(userEmailP);
93 emailAddressesContainer.appendChild(accessP);
94
95 //Display the user's email address in the document
96 userEmailP.innerHTML = "User email: " + userEmail;
97
98
99 //Check if the email address is on the list of those who can access
100 //To do so, I'll iterate through the array
101 for(var i = 0; i < emailAddresses.length; i++)
102 {
103     if(emailAddresses[i] === userEmail)
104     {
105         emailFound = true;
106         const accessGranted = "Access granted!";
107         console.log(accessGranted);
108         accessP.innerHTML = accessGranted;
109         break;
110     }
111 }
112
113 if(emailFound === false)
114 {
115     const accessDenied = "Access denied!";
116     console.log(accessDenied);
117     accessP.innerHTML = accessDenied;
118 }

```

Tramite ciclo for, itero all'interno dell'array

Se l'indirizzo email all'interno dell'array corrisponde a quello dato dall'utente:

- Booleano → Si imposta a **true**
- Creo costante accessGranted
- Poi, sovrascrivo accessP con accessGranted
- Rompo fuori dal ciclo

Se il valore booleano NON è stato impostato a true, hence rimane false

- Creo costante accessDenied
- Stampo su console
- Sovrascrivo accessP con accessDenied