

CaChiPun

March 7, 2018

1 Ejercicio - Cachipún

Vamos a iniciar esta ronda de ejercicios con un juego simple para practicar programación en Javascript.

1.1 Instrucciones

El juego consiste en lo siguiente:

- Para esta ocasión, sólo podrá ser jugado por 2 personas
- Cada jugador escoge una opción entre Piedra, Papel o Tijeras
- Al mismo tiempo, ambos jugadores muestran su elección. El ganador se determina por:
- Papel gana a Piedra
- Piedra gana a Tijera
- Tijera gana a Papel
- Si ambos eligen la misma opción entonces hay un empate

1.2 ¡Comencemos!

1.2.1 Inicializando las elecciones

Para facilitar las elecciones, crearemos un objeto que contendrá las diferentes alternativas:

```
In [ ]: const OPTIONS = {  
        paper: 'Papel',  
        rock: 'Piedra',  
        scissors: 'Tijera',  
      };  
  
      const OPTIONS_VALUES = Object.values(OPTIONS);
```

1.2.2 Generando una opción

Vamos a generar una opción aleatoria entre las que generamos anteriormente. Algo que te puede ser útil:

- Para generar un número aleatorio entre min y max podrías usar:

Math.round(Math.random() * (max - min) + min);
Aquí estamos usando la librería Math de Javascript.

- Recuerda que puedes inicializar variables usando const y let.

```
In [ ]: /**
        * Generates a random option.
        * @param {array} optionsArray - Array with options values.
        * @return {string} Random option.
        */
        function generateRandomOption(optionsArray) {
            // BEGIN SOLUTION

            // END SOLUTION
        }
```

¡Probemos el código anterior!

```
In [ ]: var newOptions = { rat: 'Rata', leon: 'Leon', cat: 'Gato'};
        var newOptionsValues = Object.values(newOptions);
        var randomOption = generateRandomOption(newOptionsValues);
        newOptionsValues.indexOf(randomOption) > -1
```

Valor Esperado: true

1.2.3 Eligiendo a un ganador

Ya que nuestros jugadores virtuales pueden escoger una opción, ahora tenemos que ver cuál es el que gana. Para eso debes completar la siguiente función:

```
In [ ]: /**
        * Returns winner based on preferences.
        * @param {object} options - Object with preferences.
        * @param {string} preference1 - Preference for first user
        * @param {string} preference2 - Preference for second user
        * @return {number} Winner, 0 for first user and 1 for second user and -1 for a tie.
        */
        function selectWinner(options, preference1, preference2) {
            // BEGIN SOLUTION

            // END SOLUTION
        }
```

Probemos el resultado

```
In [ ]: var userOneWinner = selectWinner(OPTIONS, OPTIONS.paper, OPTIONS.rock) === 0
        && selectWinner(OPTIONS, OPTIONS.rock, OPTIONS.scissors) === 0
        && selectWinner(OPTIONS, OPTIONS.scissors, OPTIONS.paper) === 0;
```

```

var userTwoWinner = selectWinner(OPTIONS, OPTIONS.rock, OPTIONS.paper) === 1
&& selectWinner(OPTIONS, OPTIONS.scissors, OPTIONS.rock) === 1
&& selectWinner(OPTIONS, OPTIONS.paper, OPTIONS.scissors) === 1;

var tie = selectWinner(OPTIONS, OPTIONS.paper, OPTIONS.paper) === -1;

userOneWinner && userTwoWinner && tie;

```

Valor Esperado: true

1.2.4 ¿Ahora juntemos todo!

Nuestros jugadores virtuales ya pueden escoger una opcion y además podemos saber quién gana
 ¿Ahora es momento de juntar todas las piezas!

```

In [ ]: /**
        * Simulates a Game
        * @param {object} options - Object with preferences.
        */
function simulateGame(options) {
    // BEGIN SOLUTION
    // Generate preferences
    const preferenceFirstUser = undefined;
    const preferenceSecondUser = undefined;

    // Select winner
    const winner = undefined;
    // END SOLUTION

    if(winner === -1)
        console.log('It\'s a Tie!');
    else
        console.log(`Player ${winner + 1} wins!`);
}

```

Veamos como funciona:

```

In [ ]: simulateGame(OPTIONS);

```

1.3 ¿A la tercera!

Hasta ahora podemos generar solo un turno de nuestros queridos jugadores virtuales, pero ¿y si queremos jugar más turnos? ¿Qué podemos hacer?

```

In [ ]: /**
        * Simulates a game with turns
        * @param {object} options - Object with preferences.
        * @param {number} turns.
        */

```

```

function simulateGameWithTurns(options, turns) {

    const winners = [];

    // BEGIN SOLUTION
    // Validate turns value

    // loop - implements turns

    // Generate preferences
    const preferenceFirstUser = undefined;
    const preferenceSecondUser = undefined;

    // Select winner
    const winner = undefined;

    if(winner === -1)
        console.log('It\'s a Tie!');
    else
        console.log(`Player ${winner + 1} wins!`);

    // Add winner to winners (avoid tie case)


    // Determine de winner and print in console (with console.log)

    // END SOLUTION
}

```

¶Veamos como funciona!

```

In [ ]: var turns2 = 5;
        simulateGameWithTurns(OPTIONS, turns2);

```

1.4 Resumen

En este ejercicio abordamos algunos elementos de Javascript. Esto fueron abordados de forma práctica para que vean cómo se ocupan realmente. Algunas cosas que vimos fueron:

- Definición de variables (const, let)
- Control de flujo (if/else if/else, for/while)
- Uso de librerías del lenguaje (Math en este caso)
- Salidas a consola (console.log)
- Operadores (comparación, incremento, etc)
- Entre otros

1.4.1 Para profundizar

- Fundamentos Javascript - <https://www.codecademy.com/es/tracks/javascript-traduccion-al-espanol-america-latina-clone>
- Arrays Javascript - <https://tech.io/playgrounds/6181/javascript-arrays---tips-tricks-and-examples>