PROMINEO TECH

Intro to JavaScript Week 6 Coding Assignment

Points possible: 100

URL to GitHub Repository:

https://github.com/nmgolz/week6.git

URL to Your Coding Assignment Video:

https://www.dropbox.com/s/19rxr18ynh6gmqo/week6_video.mp4?dl=0

Instructions: In Visual Studio Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

For the final project you will be creating an automated version of the classic card game *WAR*. You do not need to accept any user input, when you run your code, the entire game should play out instantly without any user input.

There are many versions of the game *WAR,* but in this version there are only 2 players and you don't need to do anything special when there is a tie on a round.

Think about how you would build this project and write your plan down. Consider classes such as Card, Deck, and Player and what fields and methods they might each have. You can implement the game however you'd like (i.e. printing to the console, using alert, or some other way). The completed project should, when run, do the following:

- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card
- The Player who played the higher card is awarded a point
    - ○ Ties result in zero points for both Players

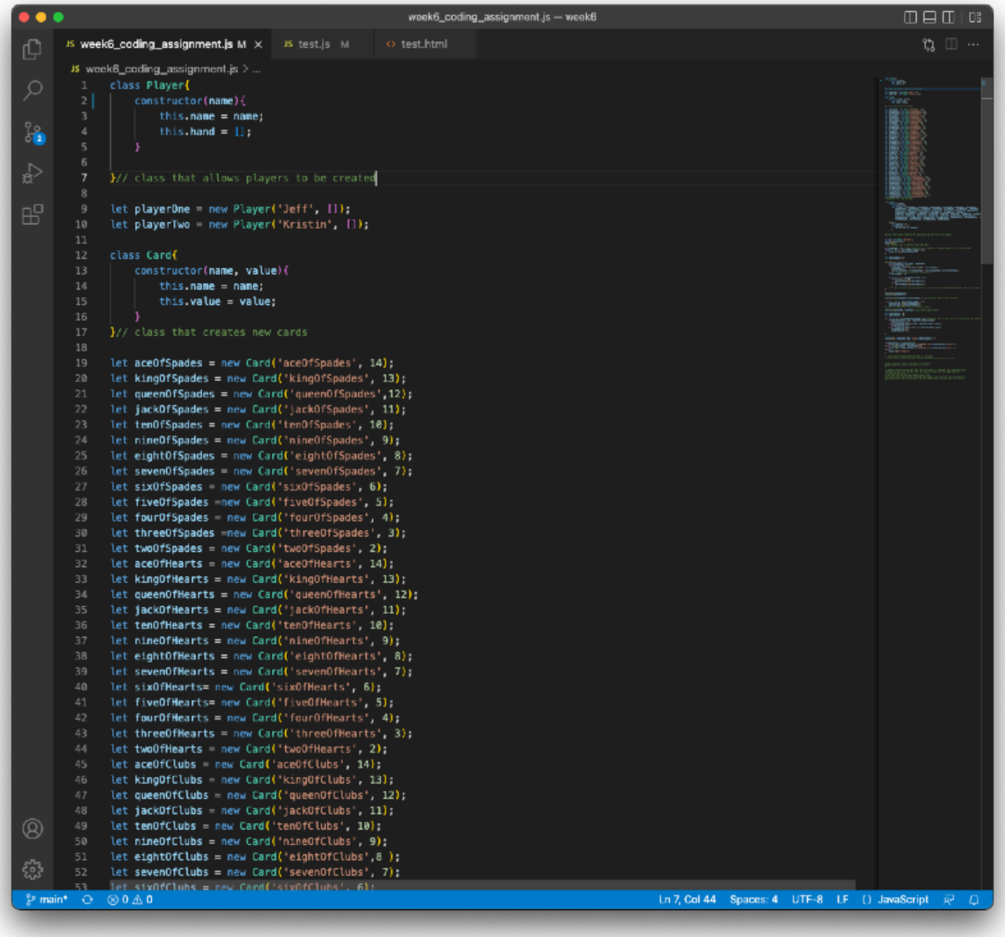
- After all cards have been played, display the score and declare the winner.

Write a Unit Test using Mocha and Chai for at least one of the functions you write.

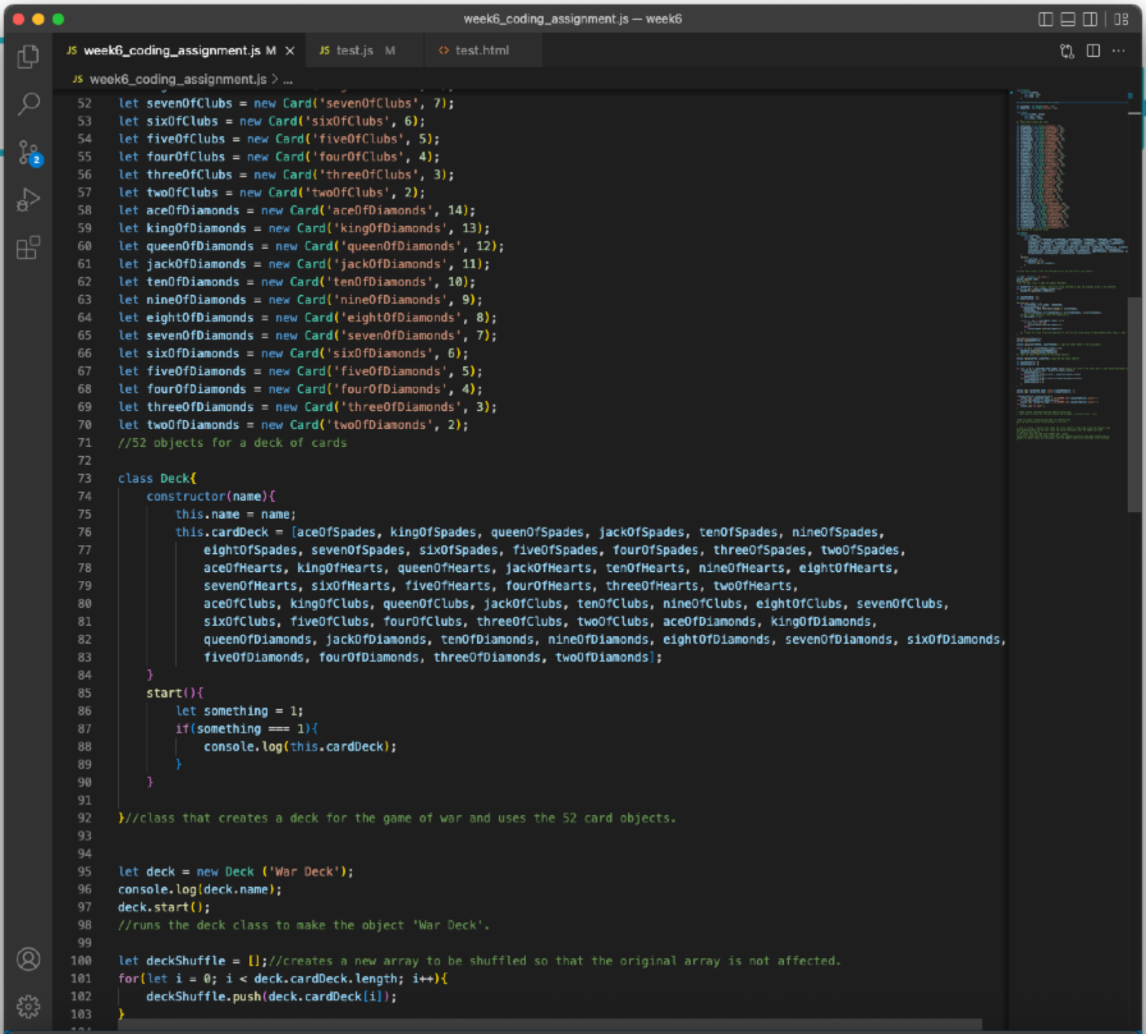
Video Steps:

Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed. This video should be done using screen share and voice over. This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend. You can create a new meeting, start screen sharing, and start recording. This will create a video recording on your computer. This should then be uploaded to a publicly accessible site, such as YouTube, Dropbox, or Google Drive. MAKE SURE THE LINK YOU SHARE IS PUBLIC or UNLISTED. If it is not accessible by your grader, your project will be graded based on what they can access. The link should be pasted in the submission text box after the GitHub repo link. REQUIRED: PUBLIC link to video, and GitHub repo link with everything listed above!

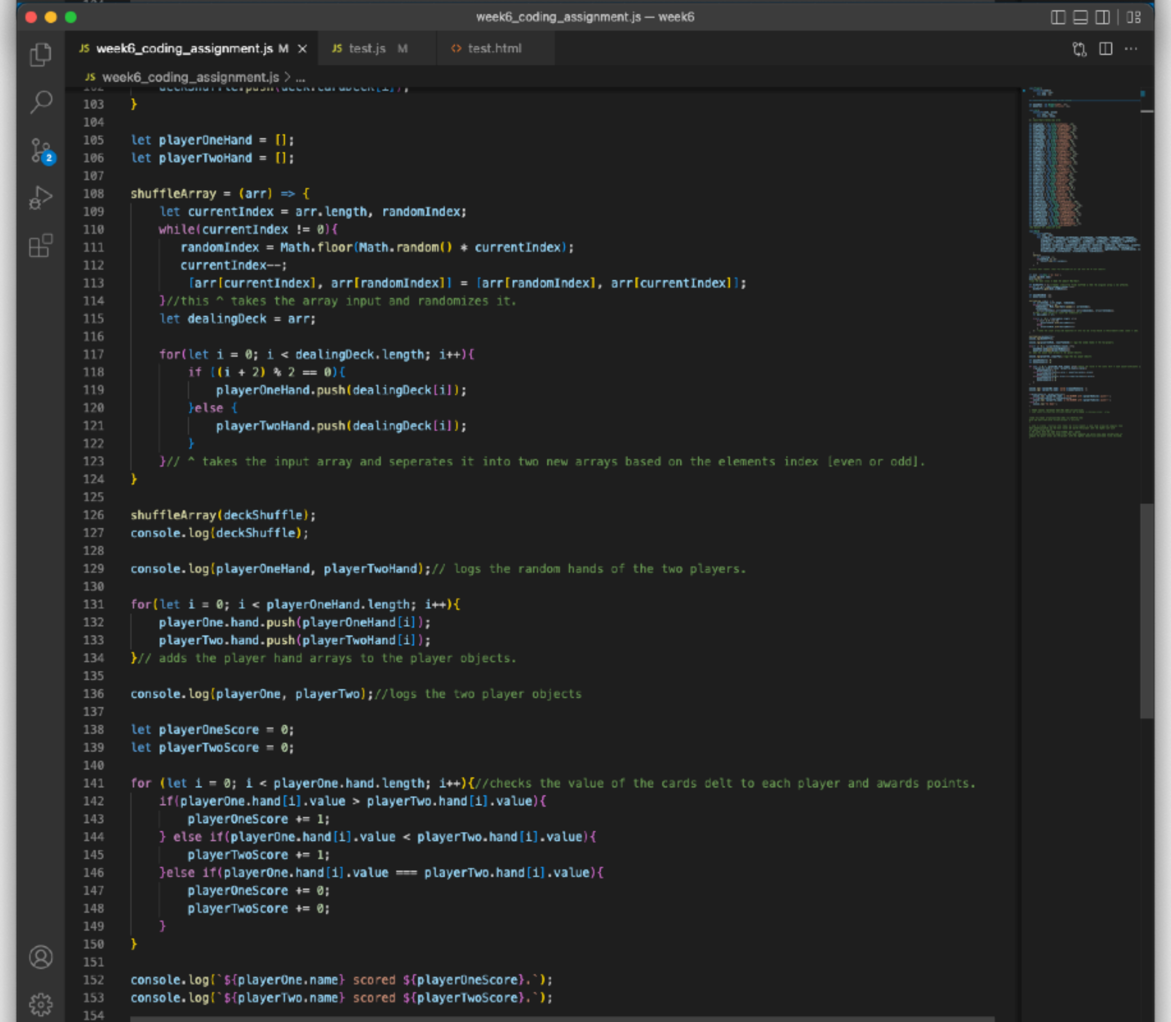
Screenshots of Code:


week6_coding_assignment.js — week6
week6_coding_assignment.js M ×
test.js M
test.html
week6_coding_assignment.js > ...
main*
⊗ 0 ⚠ 0
Ln 7, Col 44
Spaces: 4
UTF-8
LF
{} JavaScript

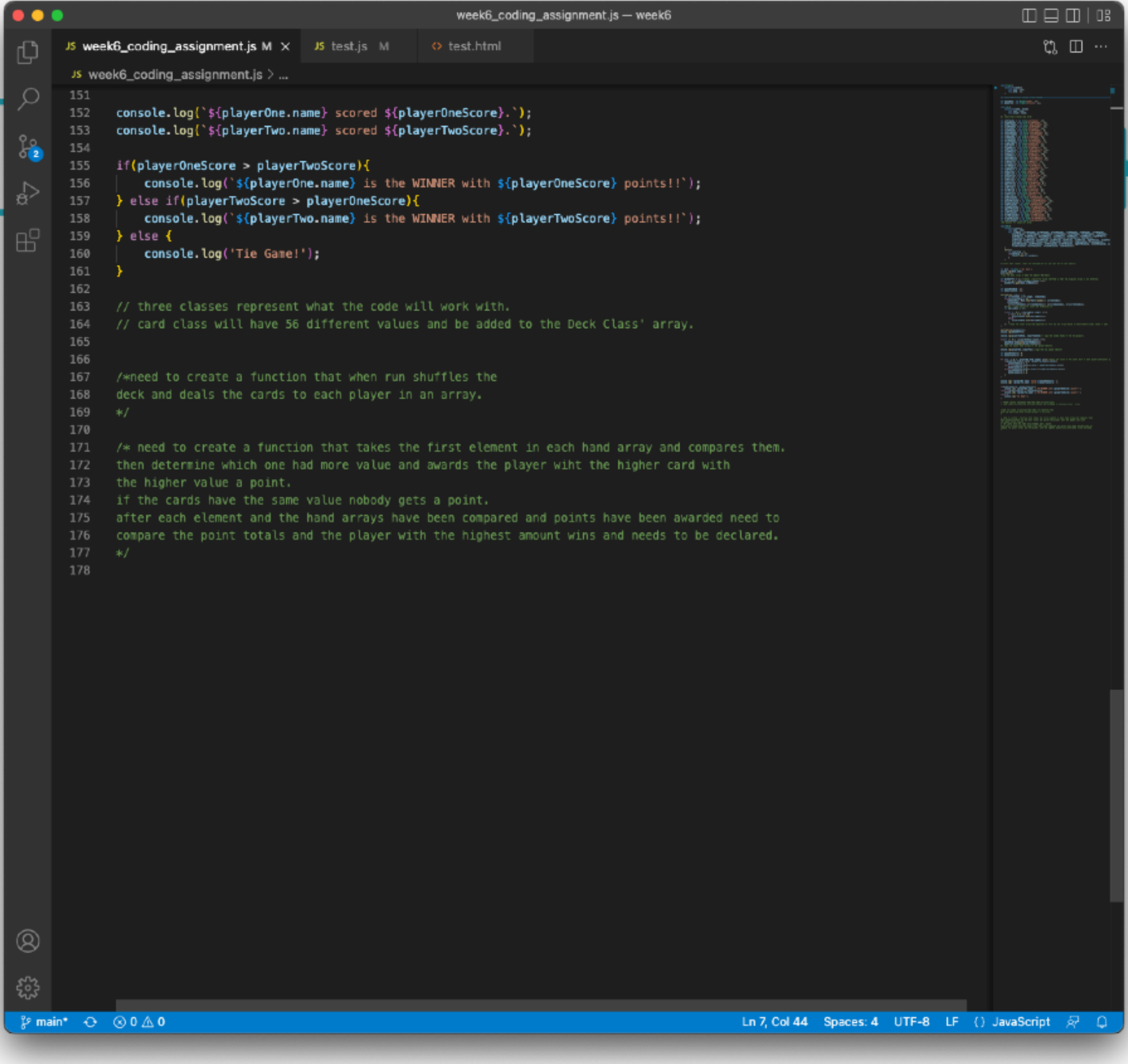
```javascript
class Player{
    constructor(name){
        this.name = name;
        this.hand = [];
    }

}// class that allows players to be created

let playerOne = new Player('Jeff', []);
let playerTwo = new Player('Kristin', []);

class Card{
    constructor(name, value){
        this.name = name;
        this.value = value;
    }
}// class that creates new cards

let aceOfSpades = new Card('aceOfSpades', 14);
let kingOfSpades = new Card('kingOfSpades', 13);
let queenOfSpades = new Card('queenOfSpades',12);
let jackOfSpades = new Card('jackOfSpades', 11);
let tenOfSpades = new Card('tenOfSpades', 10);
let nineOfSpades = new Card('nineOfSpades', 9);
let eightOfSpades = new Card('eightOfSpades', 8);
let sevenOfSpades = new Card('sevenOfSpades', 7);
let sixOfSpades = new Card('sixOfSpades', 6);
let fiveOfSpades =new Card('fiveOfSpades', 5);
let fourOfSpades = new Card('fourOfSpades', 4);
let threeOfSpades =new Card('threeOfSpades', 3);
let twoOfSpades = new Card('twoOfSpades', 2);
let aceOfHearts = new Card('aceOfHearts', 14);
let kingOfHearts = new Card('kingOfHearts', 13);
let queenOfHearts = new Card('queenOfHearts', 12);
let jackOfHearts = new Card('jackOfHearts', 11);
let tenOfHearts = new Card('tenOfHearts', 10);
let nineOfHearts = new Card('nineOfHearts', 9);
let eightOfHearts = new Card('eightOfHearts', 8);
let sevenOfHearts = new Card('sevenOfHearts', 7);
let sixOfHearts= new Card('sixOfHearts', 6);
let fiveOfHearts= new Card('fiveOfHearts', 5);
let fourOfHearts = new Card('fourOfHearts', 4);
let threeOfHearts = new Card('threeOfHearts', 3);
let twoOfHearts = new Card('twoOfHearts', 2);
let aceOfClubs = new Card('aceOfClubs', 14);
let kingOfClubs = new Card('kingOfClubs', 13);
let queenOfClubs = new Card('queenOfClubs', 12);
let jackOfClubs = new Card('jackOfClubs', 11);
let tenOfClubs = new Card('tenOfClubs', 10);
let nineOfClubs = new Card('nineOfClubs', 9);
let eightOfClubs = new Card('eightOfClubs',8 );
let sevenOfClubs = new Card('sevenOfClubs', 7);
let sixOfClubs = new Card('sixOfClubs', 6);
```


week6_coding_assignment.js — week6
week6_coding_assignment.js M
test.js M
test.html
week6_coding_assignment.js > ...

```javascript
let sevenOfClubs = new Card('sevenOfClubs', 7);
let sixOfClubs = new Card('sixOfClubs', 6);
let fiveOfClubs = new Card('fiveOfClubs', 5);
let fourOfClubs = new Card('fourOfClubs', 4);
let threeOfClubs = new Card('threeOfClubs', 3);
let twoOfClubs = new Card('twoOfClubs', 2);
let aceOfDiamonds = new Card('aceOfDiamonds', 14);
let kingOfDiamonds = new Card('kingOfDiamonds', 13);
let queenOfDiamonds = new Card('queenOfDiamonds', 12);
let jackOfDiamonds = new Card('jackOfDiamonds', 11);
let tenOfDiamonds = new Card('tenOfDiamonds', 10);
let nineOfDiamonds = new Card('nineOfDiamonds', 9);
let eightOfDiamonds = new Card('eightOfDiamonds', 8);
let sevenOfDiamonds = new Card('sevenOfDiamonds', 7);
let sixOfDiamonds = new Card('sixOfDiamonds', 6);
let fiveOfDiamonds = new Card('fiveOfDiamonds', 5);
let fourOfDiamonds = new Card('fourOfDiamonds', 4);
let threeOfDiamonds = new Card('threeOfDiamonds', 3);
let twoOfDiamonds = new Card('twoOfDiamonds', 2);
//52 objects for a deck of cards

class Deck{
    constructor(name){
        this.name = name;
        this.cardDeck = [aceOfSpades, kingOfSpades, queenOfSpades, jackOfSpades, tenOfSpades, nineOfSpades,
            eightOfSpades, sevenOfSpades, sixOfSpades, fiveOfSpades, fourOfSpades, threeOfSpades, twoOfSpades,
            aceOfHearts, kingOfHearts, queenOfHearts, jackOfHearts, tenOfHearts, nineOfHearts, eightOfHearts,
            sevenOfHearts, sixOfHearts, fiveOfHearts, fourOfHearts, threeOfHearts, twoOfHearts,
            aceOfClubs, kingOfClubs, queenOfClubs, jackOfClubs, tenOfClubs, nineOfClubs, eightOfClubs, sevenOfClubs,
            sixOfClubs, fiveOfClubs, fourOfClubs, threeOfClubs, twoOfClubs, aceOfDiamonds, kingOfDiamonds,
            queenOfDiamonds, jackOfDiamonds, tenOfDiamonds, nineOfDiamonds, eightOfDiamonds, sevenOfDiamonds, sixOfDiamonds,
            fiveOfDiamonds, fourOfDiamonds, threeOfDiamonds, twoOfDiamonds];
    }
    start(){
        let something = 1;
        if(something === 1){
            console.log(this.cardDeck);
        }
    }

}//class that creates a deck for the game of war and uses the 52 card objects.


let deck = new Deck ('War Deck');
console.log(deck.name);
deck.start();
//runs the deck class to make the object 'War Deck'.

let deckShuffle = [];//creates a new array to be shuffled so that the original array is not affected.
for(let i = 0; i < deck.cardDeck.length; i++){
    deckShuffle.push(deck.cardDeck[i]);
}
```


week6_coding_assignment.js — week6
week6_coding_assignment.js M
test.js M
test.html
week6_coding_assignment.js > ...

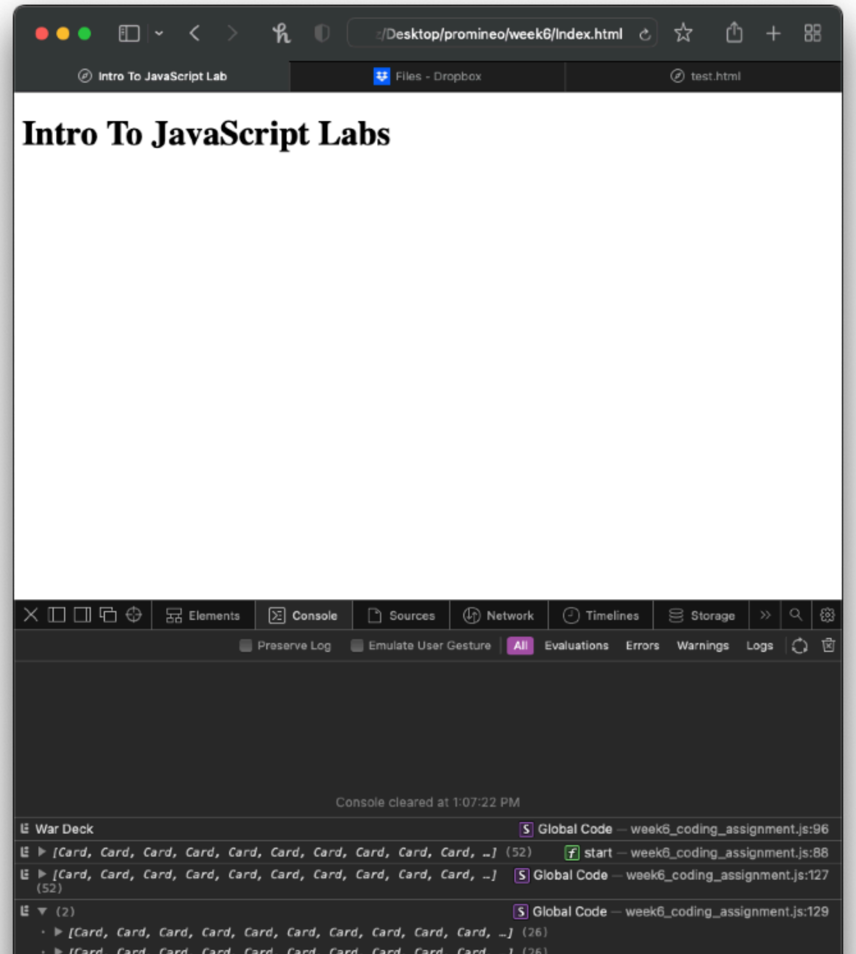
```javascript
}

let playerOneHand = [];
let playerTwoHand = [];

shuffleArray = (arr) => {
    let currentIndex = arr.length, randomIndex;
    while(currentIndex != 0){
       randomIndex = Math.floor(Math.random() * currentIndex);
       currentIndex--;
        [arr[currentIndex], arr[randomIndex]] = [arr[randomIndex], arr[currentIndex]];
    }//this ^ takes the array input and randomizes it.
    let dealingDeck = arr;

    for(let i = 0; i < dealingDeck.length; i++){
        if ((i + 2) % 2 == 0){
            playerOneHand.push(dealingDeck[i]);
        }else {
            playerTwoHand.push(dealingDeck[i]);
        }
    }// ^ takes the input array and seperates it into two new arrays based on the elements index [even or odd].
}

shuffleArray(deckShuffle);
console.log(deckShuffle);

console.log(playerOneHand, playerTwoHand);// logs the random hands of the two players.

for(let i = 0; i < playerOneHand.length; i++){
    playerOne.hand.push(playerOneHand[i]);
    playerTwo.hand.push(playerTwoHand[i]);
}// adds the player hand arrays to the player objects.

console.log(playerOne, playerTwo);//logs the two player objects

let playerOneScore = 0;
let playerTwoScore = 0;

for (let i = 0; i < playerOne.hand.length; i++){//checks the value of the cards delt to each player and awards points.
    if(playerOne.hand[i].value > playerTwo.hand[i].value){
        playerOneScore += 1;
    } else if(playerOne.hand[i].value < playerTwo.hand[i].value){
        playerTwoScore += 1;
    }else if(playerOne.hand[i].value === playerTwo.hand[i].value){
        playerOneScore += 0;
        playerTwoScore += 0;
    }
}

console.log(`${playerOne.name} scored ${playerOneScore}.`);
console.log(`${playerTwo.name} scored ${playerTwoScore}.`);
```



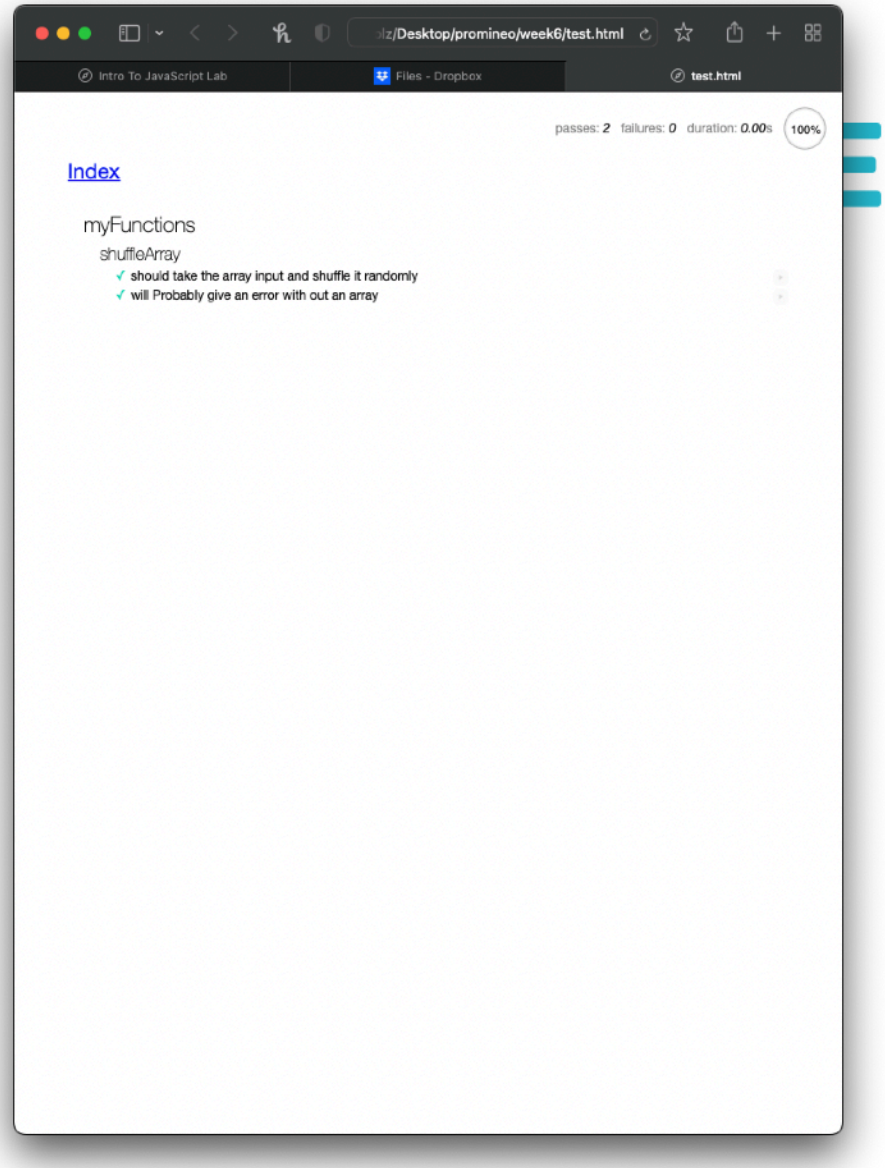
week6_coding_assignment.js — week6
week6_coding_assignment.js M
test.js M
test.html
week6_coding_assignment.js > ...
main*
0 0
Ln 7, Col 44
Spaces: 4
UTF-8
LF
JavaScript

```javascript

console.log(`${playerOne.name} scored ${playerOneScore}.`);
console.log(`${playerTwo.name} scored ${playerTwoScore}.`);

if(playerOneScore > playerTwoScore){
    console.log(`${playerOne.name} is the WINNER with ${playerOneScore} points!!`);
} else if(playerTwoScore > playerOneScore){
    console.log(`${playerTwo.name} is the WINNER with ${playerTwoScore} points!!`);
} else {
    console.log('Tie Game!');
}

// three classes represent what the code will work with.
// card class will have 56 different values and be added to the Deck Class' array.


/*need to create a function that when run shuffles the
deck and deals the cards to each player in an array.
*/

/* need to create a function that takes the first element in each hand array and compares them.
then determine which one had more value and awards the player wiht the higher card with
the higher value a point.
if the cards have the same value nobody gets a point.
after each element and the hand arrays have been compared and points have been awarded need to
compare the point totals and the player with the highest amount wins and needs to be declared.
*/

```

Screenshots of Running Application:



```
Console cleared at 1:07:22 PM
War Deck                                                      Global Code — week6_coding_assignment.js:96
▶ [Card, Card, Card, Card, Card, Card, Card, Card, Card, Card, …] (52)    start — week6_coding_assignment.js:88
▶ [Card, Card, Card, Card, Card, Card, Card, Card, Card, Card, …]  Global Code — week6_coding_assignment.js:127
(52)
▼ (2)                                                         Global Code — week6_coding_assignment.js:129
  ▶ [Card, Card, Card, Card, Card, Card, Card, Card, Card, Card, …] (26)
  ▶ [Card, Card, Card, Card, Card, Card, Card, Card, Card, Card, …] (26)
```



```
passes: 2  failures: 0  duration: 0.00s  100%

Index

myFunctions
  shuffleArray
    ✓ should take the array input and shuffle it randomly
    ✓ will Probably give an error with out an array
```