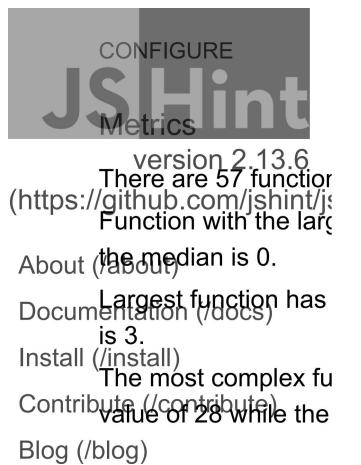


```

1  // Hello.
2  //
3  // This is JSHint, a tool that helps to detect errors and potential
4  // problems in your JavaScript code.
5  //
6  // To start, simply enter some JavaScript anywhere on this page. Your
7  // report will appear on the right side.
8  //
9  // Additionally, you can toggle specific options in the Configure
10 // menu.
11 let playerHand = [];
12
13 let dealerHand = [];
14
15 let deck = [];
16
17 let pot = [];
18
19 let splitHand = [];
20
21 let hiddenCard = [];
22
23 let score = 1000;
24
25 const resetDeck = [
26   {
27     suit: "hearts",
28     name: "ace",
29     value: 11,
30     frontend: "A",
31     id: "1"
32   },
33
34   {
35     suit: "hearts",
36     name: "two",
37     value: 2,
38     frontend: "2",
39     id: "2"
40   },
41
42   {
43     suit: "hearts",
44     name: "three",
45     value: 3,
46     frontend: "3",
47     id: "3"
48   },
49
50   {
51     suit: "hearts",
52     name: "four",
53     value: 4,
54     frontend: "4",
55     id: "4"
56   },
57
58   {
59     suit: "hearts",
60     name: "five",
61     value: 5,
62     frontend: "5",
63     id: "5"
64   },
65
66   {
67     suit: "hearts",
68     name: "six",
69     value: 6,
70     frontend: "6",

```



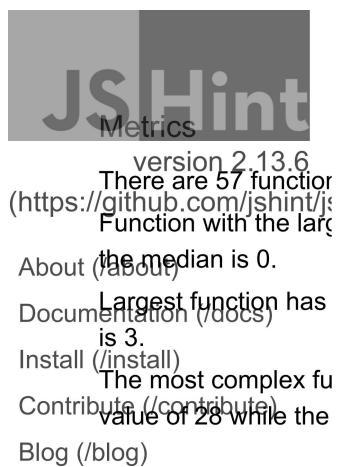
47 warnings

- 11 'let' is available JS extensions (
- 13 'let' is available JS extensions (
- 15 'let' is available JS extensions (
- 17 'let' is available JS extensions (
- 19 'let' is available JS extensions (
- 21 'let' is available JS extensions (
- 23 'let' is available JS extensions (
- 25 'const' is available Mozilla JS exte
- 443 'let' is available JS extensions (
- 444 'let' is available JS extensions (
- 445 'let' is available JS extensions (
- 446 'let' is available JS extensions (
- 447 'let' is available JS extensions (
- 448 'let' is available JS extensions (
- 449 'let' is available JS extensions (
- 450 'let' is available JS extensions (
- 451 'let' is available JS extensions (

```

71         id: "6"
72     },
73
74     {
75         suit: "hearts",
76         name: "seven",
77         value: 7,
78         frontend: "7",
79         id: "7"
80     },
81
82     {
83         suit: "hearts",
84         name: "eight",
85         value: 8,
86         frontend: "8",
87         id: "8"
88     },
89
90     {
91         suit: "hearts",
92         name: "nine",
93         value: 9,
94         frontend: "9",
95         id: "9"
96     },
97
98     {
99         suit: "hearts",
100        name: "ten",
101        value: 10,
102        frontend: "10",
103        id: "10"
104    },
105
106    {
107        suit: "hearts",
108        name: "jack",
109        value: 10,
110        frontend: "J",
111        id: "11"
112    },
113
114    {
115        suit: "hearts",
116        name: "queen",
117        value: 10,
118        frontend: "Q",
119        id: "12"
120    },
121
122    {
123        suit: "hearts",
124        name: "king",
125        value: 10,
126        frontend: "K",
127        id: "13"
128    },
129
130    {
131        suit: "diamonds",
132        name: "ace",
133        value: 11,
134        frontend: "A",
135        id: "14"
136    },
137
138    {
139        suit: "diamonds",
140        name: "two",
141        value: 2,
142        frontend: "2",

```



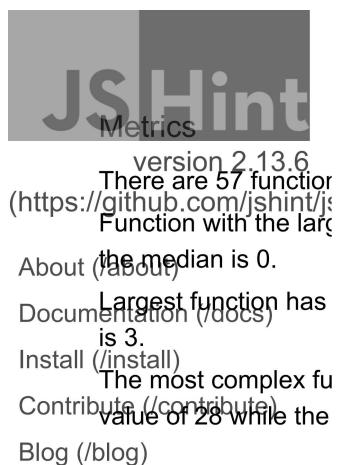
47 warnings

- 11 'let' is available JS extensions ([/extensions/let](#))
- 13 'let' is available JS extensions ([/extensions/let](#))
- 15 'let' is available JS extensions ([/extensions/let](#))
- 17 'let' is available JS extensions ([/extensions/let](#))
- 19 'let' is available JS extensions ([/extensions/let](#))
- 21 'let' is available JS extensions ([/extensions/let](#))
- 23 'let' is available JS extensions ([/extensions/let](#))
- 25 'const' is available Mozilla JS extensions ([/extensions/const](#))
- 443 'let' is available JS extensions ([/extensions/let](#))
- 444 'let' is available JS extensions ([/extensions/let](#))
- 445 'let' is available JS extensions ([/extensions/let](#))
- 446 'let' is available JS extensions ([/extensions/let](#))
- 447 'let' is available JS extensions ([/extensions/let](#))
- 448 'let' is available JS extensions ([/extensions/let](#))
- 449 'let' is available JS extensions ([/extensions/let](#))
- 450 'let' is available JS extensions ([/extensions/let](#))
- 451 'let' is available JS extensions ([/extensions/let](#))

```

143         id: "15"
144     },
145
146     {
147         suit: "diamonds",
148         name: "three",
149         value: 3,
150         frontend: "3",
151         id: "16"
152     },
153
154     {
155         suit: "diamonds",
156         name: "four",
157         value: 4,
158         frontend: "4",
159         id: "17"
160     },
161
162     {
163         suit: "diamonds",
164         name: "five",
165         value: 5,
166         frontend: "5",
167         id: "18"
168     },
169
170     {
171         suit: "diamonds",
172         name: "six",
173         value: 6,
174         frontend: "6",
175         id: "19"
176     },
177
178     {
179         suit: "diamonds",
180         name: "seven",
181         value: 7,
182         frontend: "7",
183         id: "20"
184     },
185
186     {
187         suit: "diamonds",
188         name: "eight",
189         value: 8,
190         frontend: "8",
191         id: "21"
192     },
193
194     {
195         suit: "diamonds",
196         name: "nine",
197         value: 9,
198         frontend: "9",
199         id: "22"
200     },
201
202     {
203         suit: "diamonds",
204         name: "ten",
205         value: 10,
206         frontend: "10",
207         id: "23"
208     },
209
210     {
211         suit: "diamonds",
212         name: "jack",
213         value: 10,
214         frontend: "J",

```



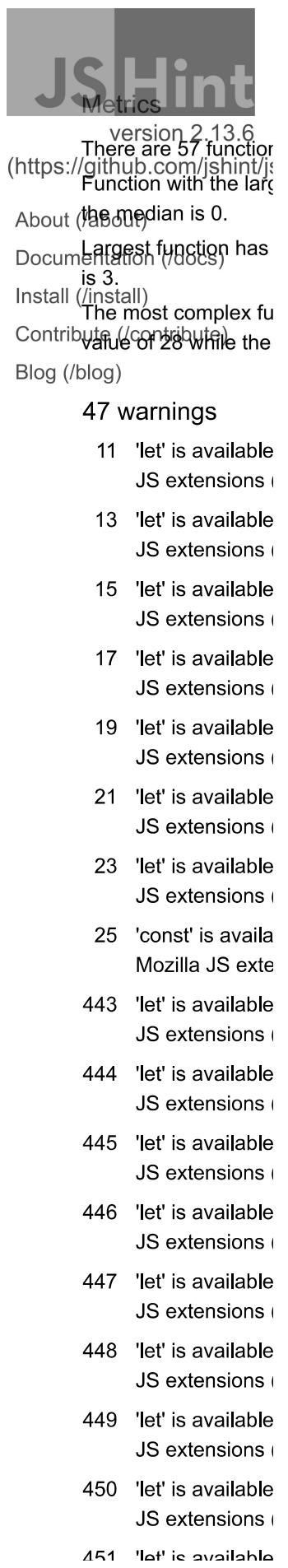
47 warnings

- 11 'let' is available JS extensions |
- 13 'let' is available JS extensions |
- 15 'let' is available JS extensions |
- 17 'let' is available JS extensions |
- 19 'let' is available JS extensions |
- 21 'let' is available JS extensions |
- 23 'let' is available JS extensions |
- 25 'const' is available Mozilla JS exte
- 443 'let' is available JS extensions |
- 444 'let' is available JS extensions |
- 445 'let' is available JS extensions |
- 446 'let' is available JS extensions |
- 447 'let' is available JS extensions |
- 448 'let' is available JS extensions |
- 449 'let' is available JS extensions |
- 450 'let' is available JS extensions |
- 451 'let' is available JS extensions |

```

215         id: "24"
216     },
217
218     {
219         suit: "diamonds",
220         name: "queen",
221         value: 10,
222         frontend: "Q",
223         id: "25"
224     },
225
226     {
227         suit: "diamonds",
228         name: "king",
229         value: 10,
230         frontend: "K",
231         id: "26"
232     },
233
234     {
235         suit: "spades",
236         name: "ace",
237         value: 11,
238         frontend: "A",
239         id: "27"
240     },
241
242     {
243         suit: "spades",
244         name: "two",
245         value: 2,
246         frontend: "2",
247         id: "28"
248     },
249
250     {
251         suit: "spades",
252         name: "three",
253         value: 3,
254         frontend: "3",
255         id: "29"
256     },
257
258     {
259         suit: "spades",
260         name: "four",
261         value: 4,
262         frontend: "4",
263         id: "30"
264     },
265
266     {
267         suit: "spades",
268         name: "five",
269         value: 5,
270         frontend: "5",
271         id: "31"
272     },
273
274     {
275         suit: "spades",
276         name: "six",
277         value: 6,
278         frontend: "6",
279         id: "32"
280     },
281
282     {
283         suit: "spades",
284         name: "seven",
285         value: 7,
286         frontend: "7",

```



```

287         id: "33"
288     },
289
290     {
291         suit: "spades",
292         name: "eight",
293         value: 8,
294         frontend: "8",
295         id: "34"
296     },
297
298     {
299         suit: "spades",
300         name: "nine",
301         value: 9,
302         frontend: "9",
303         id: "35"
304     },
305
306     {
307         suit: "spades",
308         name: "ten",
309         value: 10,
310         frontend: "10",
311         id: "36"
312     },
313
314     {
315         suit: "spades",
316         name: "jack",
317         value: 10,
318         frontend: "J",
319         id: "37"
320     },
321
322     {
323         suit: "spades",
324         name: "queen",
325         value: 10,
326         frontend: "Q",
327         id: "38"
328     },
329
330     {
331         suit: "spades",
332         name: "king",
333         value: 10,
334         frontend: "K",
335         id: "39"
336     },
337
338     {
339         suit: "clubs",
340         name: "ace",
341         value: 11,
342         frontend: "A",
343         id: "40"
344     },
345
346     {
347         suit: "clubs",
348         name: "two",
349         value: 2,
350         frontend: "2",
351         id: "41"
352     },
353
354     {
355         suit: "clubs",
356         name: "three",
357         value: 3,
358         frontend: "3",

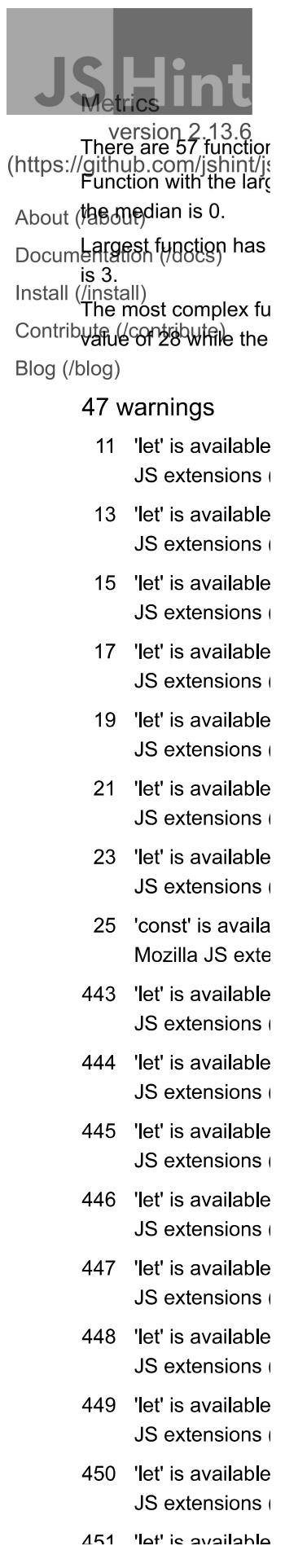
```



```

359         id: "42"
360     },
361     {
362         suit: "clubs",
363         name: "four",
364         value: 4,
365         frontend: "4",
366         id: "43"
367     },
368     {
369         suit: "clubs",
370         name: "five",
371         value: 5,
372         frontend: "5",
373         id: "44"
374     },
375     {
376         suit: "clubs",
377         name: "six",
378         value: 6,
379         frontend: "6",
380         id: "45"
381     },
382     {
383         suit: "clubs",
384         name: "seven",
385         value: 7,
386         frontend: "7",
387         id: "46"
388     },
389     {
390         suit: "clubs",
391         name: "eight",
392         value: 8,
393         frontend: "8",
394         id: "47"
395     },
396     {
397         suit: "clubs",
398         name: "nine",
399         value: 9,
400         frontend: "9",
401         id: "48"
402     },
403     {
404         suit: "clubs",
405         name: "ten",
406         value: 10,
407         frontend: "10",
408         id: "49"
409     },
410     {
411         suit: "clubs",
412         name: "jack",
413         value: 10,
414         frontend: "J",
415         id: "50"
416     },
417     {
418         suit: "clubs",
419         name: "queen",
420         value: 10,
421         frontend: "Q",
422         id: "51"
423     },
424     {
425         suit: "clubs",
426         name: "king",
427         value: 10,
428         frontend: "K",
429         id: "52"
430     }

```



```

431         id: "51"
432     },
433     {
434         suit: "clubs",
435         name: "king",
436         value: 10,
437         frontend: "K",
438         id: "52"
439     },
440 ],
441 ];
442
443 let playerTurn = false;
444 let dealerTurn = false;
445 let allowDoubleBet = true;
446 let betStage = true;
447 let allowSplitGame = true;
448 let firstHand = true;
449 let splitGame = false;
450 let useSplitHand = false;
451 let buttonPress = false;
452
453 /**
454  * Wait for dom to load before starting game
455 */
456 document.addEventListener("DOMContentLoaded", function() {
457     buttonPress = false;
458     let buttons = document.getElementsByTagName("button");
459     for (let button of buttons) {
460         button.addEventListener("click", function() {
461             buttonPress = true;
462             if (this.getAttribute("data-type") === "hit" && playerTurn) {
463                 hit(playerHand);
464                 toggleSplitHide();
465                 toggleDoubleHide();
466                 toggleBetHide();
467                 dealCards(playerHand, 'player');
468                 checkPlayerScore();
469             }
470             if (this.getAttribute("data-type") === "split-hit" && playerTurn) {
471                 hit(splitHand);
472                 dealCards(splitHand, 'player-split');
473                 checkSplitScore();
474             }
475             if (this.getAttribute("data-type") === "double" && playerTurn && allowDoubleB
476                 doubleBet();
477                 toggleBetHide();
478                 toggleSplitHide();
479             }
480             if (this.getAttribute("data-type") === "split" && playerTurn && allowSplitGam
481                 doubleBet();
482                 splitPlayerHand();
483                 toggleSplitConShow();
484                 toggleSplitHide();
485                 toggleSplitNone();
486                 toggleSplitTrackerShow();
487             }
488             if (this.getAttribute("data-type") === "stand" && playerTurn) {
489                 playerStand();
490                 toggleSplitHide();
491             }
492             if (this.getAttribute("data-type") === "new-game") {
493                 newGame();
494                 toggleNewGameHide();
495             }
496             if (this.getAttribute("data-type") === "bet10" && betStage) {
497                 placeBet(10);
498                 togglePotShow();
499             }
500             if (this.getAttribute("data-type") === "bet50" && betStage) {
501                 placeBet(50);
502                 togglePotShow();
503             }
504         });
505     });
506 });

```



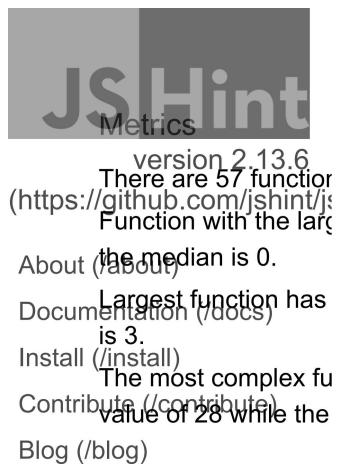
47 warnings

- 411 'let' is available JS extensions (
- 413 'let' is available JS extensions (
- 415 'let' is available JS extensions (
- 417 'let' is available JS extensions (
- 419 'let' is available JS extensions (
- 421 'let' is available JS extensions (
- 423 'let' is available JS extensions (
- 425 'const' is avail Mozilla JS exte
- 443 'let' is available JS extensions (
- 444 'let' is available JS extensions (
- 445 'let' is available JS extensions (
- 446 'let' is available JS extensions (
- 447 'let' is available JS extensions (
- 448 'let' is available JS extensions (
- 449 'let' is available JS extensions (
- 450 'let' is available JS extensions (
- 451 'let' is available JS extensions (

```

503         }
504         if (this.getAttribute("data-type") === "bet100" && betStage) {
505             placeBet(100);
506             togglePotShow();
507         }
508         if (this.getAttribute("data-type") === "bet500" && betStage) {
509             placeBet(500);
510             togglePotShow();
511         }
512         if (this.getAttribute("data-type") === "bet1000" && betStage) {
513             placeBet(1000);
514             togglePotShow();
515         }
516         if (this.getAttribute("data-type") === "lock-bet" && betStage) {
517             runGame("blackjack");
518             if (score >= getPotValue() && allowDoubleBet) {
519                 toggleDoubleShow();
520             }
521         }
522     });
523 }
524
525 checkPlayerScore();
526 checkDealerScore();
527 toggleBetShow();
528 toggleConHide();
529 toggleNewGameHide();
530 toggleSplitConHide();
531 toggleSplitRevert();
532 togglePlayerTrackerHide();
533 toggleSplitTrackerHide();
534 toggleDealerTrackerHide();
535 updateScore();
536
537 document.addEventListener("click", function() {
538     if (!buttonPress) {
539         hideAlert();
540     } else {
541         buttonPress = false;
542     }
543 });
544
545 */
546 /**
547 * runs blackjack game with two cards dealt to player and two cards dealt to dealer
548 */
549 function runGame(gameType) {
550     betStage = false;
551     playerTurn = true;
552     toggleDoubleShow();
553     toggleBetHide();
554     toggleConShow();
555     deck = [...resetDeck];
556     hit(playerHand);
557     hit(playerHand);
558     hit(dealerHand);
559     hit(dealerHand);
560     hideDealerCard();
561     dealCards(playerHand, 'player');
562     dealCards(dealerHand, 'dealer');
563     checkPlayerScore();
564     checkDealerScore();
565     togglePlayerTrackerShow();
566     toggleDealerTrackerShow();
567     let cost = getPotValue();
568     if (cost > 0) {
569         document.getElementById('double-cost').innerHTML = cost;
570         document.getElementById('split-cost').innerHTML = cost;
571     }
572     if (playerHand[0].value === playerHand[1].value && allowSplitGame) {
573         toggleSplitShow();
574     }

```



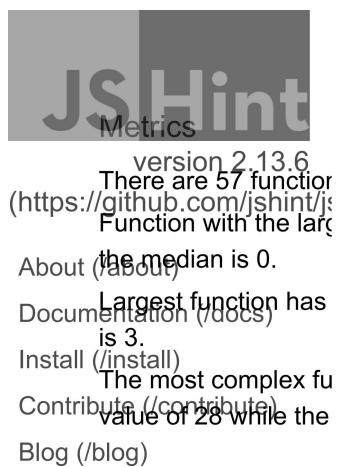
47 warnings

- 11 'let' is available JS extensions ([details](#))
- 13 'let' is available JS extensions ([details](#))
- 15 'let' is available JS extensions ([details](#))
- 17 'let' is available JS extensions ([details](#))
- 19 'let' is available JS extensions ([details](#))
- 21 'let' is available JS extensions ([details](#))
- 23 'let' is available JS extensions ([details](#))
- 25 'const' is available Mozilla JS extensions ([details](#))
- 443 'let' is available JS extensions ([details](#))
- 444 'let' is available JS extensions ([details](#))
- 445 'let' is available JS extensions ([details](#))
- 446 'let' is available JS extensions ([details](#))
- 447 'let' is available JS extensions ([details](#))
- 448 'let' is available JS extensions ([details](#))
- 449 'let' is available JS extensions ([details](#))
- 450 'let' is available JS extensions ([details](#))
- 451 'let' is available JS extensions ([details](#))

```

575     } else {
576         toggleSplitHide();
577     }
578 }
579 /**
580 * Gets card from deck
581 */
582 function getCardFromDeck(deck) {
583     var newCard = deck[Math.floor(Math.random() * deck.length)];
584     return newCard;
585 }
586 /**
587 * Removes from deck
588 */
589 function removeCardFromDeck(deck, cardToRemove) {
590     deck = deck.filter((card) => card.id !== cardToRemove.id);
591     return deck;
592 }
593 /**
594 * Sums the hand and adds logic if ace is in hand this can be treated as 11 or 1
595 */
596 function getHandValue(hand) {
597     let sum = 0;
598     for (let i = 0 ; i < hand.length ; i++) {
599         sum += hand[i].value;
600     }
601
602     if (sum > 21 && doesHandContainAce(hand)) {
603         sum -= 10;
604     }
605     return sum;
606 }
607 /**
608 * finds out if the hand contains an ace
609 */
610 function doesHandContainAce(hand) {
611     for (let i = 0 ; i < hand.length ; i++) {
612         if (hand[i].name === 'ace') {
613             return true;
614         }
615     }
616 }
617 /**
618 * adds card to the hand
619 */
620 function hit(hand) {
621     let cardHit = getCardFromDeck(deck);
622     toggleDoubleHide();
623     deck = removeCardFromDeck(deck, cardHit);
624     hand.push (cardHit);
625 }
626 /**
627 * doubles the bet placed into the pot
628 */
629 function doubleBet() {
630     let sum = getPotValue();
631     pot.push(sum);
632     score -= sum;
633     document.getElementById('pot').innerHTML = getPotValue();
634     document.getElementById('score').innerHTML = score;
635     toggleDoubleHide();
636 }
637 /**
638 * splits the hand allowing the player to play twice for the pot
639 */
640
641
642
643
644
645
646

```



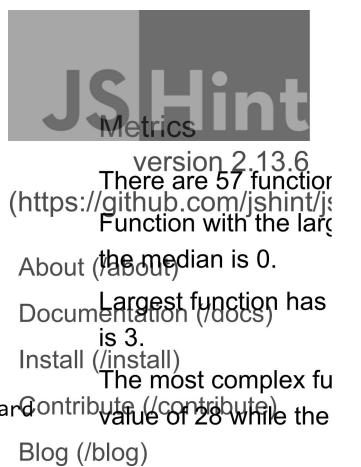
47 warnings

- 11 'let' is available JS extensions (
- 13 'let' is available JS extensions (
- 15 'let' is available JS extensions (
- 17 'let' is available JS extensions (
- 19 'let' is available JS extensions (
- 21 'let' is available JS extensions (
- 23 'let' is available JS extensions (
- 25 'const' is available Mozilla JS exte
- 443 'let' is available JS extensions (
- 444 'let' is available JS extensions (
- 445 'let' is available JS extensions (
- 446 'let' is available JS extensions (
- 447 'let' is available JS extensions (
- 448 'let' is available JS extensions (
- 449 'let' is available JS extensions (
- 450 'let' is available JS extensions (
- 451 'let' is available JS extensions (

```

647     function splitPlayerHand() {
648         allowDoubleBet = false;
649         allowSplitGame = false;
650         splitGame = true;
651         storeSecondCard();
652         removeSecondCard();
653         hit(playerHand);
654         hit(splitHand);
655         dealCards(playerHand, 'player');
656         dealCards(splitHand, 'player-split');
657         checkPlayerScore();
658         checkSplitScore();
659     }
660
661 /**
662  * stores the second card so that the first hand can be set to only have the first card
663  */
664 function storeSecondCard() {
665     splitHand.push (playerHand[1]);
666 }
667
668 /**
669  * removes the second card from the players hand
670 */
671 function removeSecondCard() {
672     playerHand.splice(1);
673 }
674
675 /**
676  * deals the cards to the html
677 */
678 function dealCards(hand, idString) {
679     let htmlHand = "";
680     for (let i = 0 ; i < hand.length ; i++) {
681         var cardName = hand[i].name;
682         var cardSuit = hand[i].suit;
683         var cardFrontend = hand[i].frontend;
684         var htmlValue = `<div class="card ${cardName} ${cardSuit}">${cardFrontend}</div>`;
685         htmlHand += htmlValue;
686     }
687     document.getElementById(idString).innerHTML = htmlHand;
688 }
689
690 /**
691  * treats the dealer card as if "face-down"
692 */
693 function hideDealerCard () {
694     hiddenCard = dealerHand.pop();
695 }
696
697 /**
698  * reveals the "face-down" dealer card
699 */
700 function revealDealerCard () {
701     dealerHand.push(hiddenCard);
702 }
703
704 /**
705  * ends the player turn and initiates dealer turn
706 */
707 function playerStand() {
708     playerTurn = false;
709     toggleDoubleHide();
710     toggleConHide();
711     playDealerTurn();
712 }
713
714 /**
715  * ends dealer turn
716 */
717 function dealerStand() {
718     dealerTurn = false;

```

**47 warnings**

- 11 'let' is available JS extensions ([details](#))
- 13 'let' is available JS extensions ([details](#))
- 15 'let' is available JS extensions ([details](#))
- 17 'let' is available JS extensions ([details](#))
- 19 'let' is available JS extensions ([details](#))
- 21 'let' is available JS extensions ([details](#))
- 23 'let' is available JS extensions ([details](#))
- 25 'const' is available Mozilla JS extensions ([details](#))
- 443 'let' is available JS extensions ([details](#))
- 444 'let' is available JS extensions ([details](#))
- 445 'let' is available JS extensions ([details](#))
- 446 'let' is available JS extensions ([details](#))
- 447 'let' is available JS extensions ([details](#))
- 448 'let' is available JS extensions ([details](#))
- 449 'let' is available JS extensions ([details](#))
- 450 'let' is available JS extensions ([details](#))
- 451 'let' is available JS extensions ([details](#))

```

719     }
720
721     /**
722      * checks player score and alerts player when bust - places sum into tracker
723      */
724     function checkPlayerScore() {
725         let playerScore = getHandValue(playerHand);
726         document.getElementById('tracker-player').innerHTML = playerScore;
727         if (playerScore > 21) {
728             if (!splitGame) {
729                 playerLost();
730             } else {
731                 firstHand = false;
732                 checkSplitScore();
733             }
734         }
735     }
736
737     /**
738      * checks split player score and alerts player when bust - places sum into tracker
739      */
740     function checkSplitScore() {
741         let splitScore = getHandValue(splitHand);
742         document.getElementById('tracker-split').innerHTML = splitScore;
743         if (splitScore > 21) {
744             if (!firstHand) {
745                 playerLost();
746             } else {
747                 splitGame = false;
748                 checkPlayerScore();
749             }
750         }
751     }
752
753     /**
754      * checks dealer score and alerts player if dealer bust - places sum into tracker
755      */
756     function checkDealerScore() {
757         let dealerScore = getHandValue(dealerHand);
758         document.getElementById('tracker-dealer').innerHTML = dealerScore;
759         if (dealerScore > 21) {
760             dealerTurn = false;
761         }
762     }
763
764     /**
765      * plays the dealers turn
766      */
767     function playDealerTurn() {
768         dealerTurn = true;
769         let dealerScore;
770         let betterScore;
771         revealDealerCard();
772         dealCards(dealerHand, 'dealer');
773
774         while (dealerTurn) {
775             dealerScore = getHandValue(dealerHand);
776             betterScore = getHighestPlayerScore();
777
778             if (dealerScore < betterScore && betterScore <= 21) {
779                 // dealer will hit when less than playerScore
780                 hit(dealerHand);
781                 dealCards(dealerHand, 'dealer');
782                 checkDealerScore();
783             } else if (dealerScore === betterScore && dealerScore <= 15) {
784                 // dealer will hit to try and beat player
785                 hit(dealerHand);
786                 dealCards(dealerHand, 'dealer');
787                 checkDealerScore();
788             } else {
789                 // dealer will stand to draw/win
790                 dealerStand();

```



47 warnings

- 11 'let' is available JS extensions (
- 13 'let' is available JS extensions (
- 15 'let' is available JS extensions (
- 17 'let' is available JS extensions (
- 19 'let' is available JS extensions (
- 21 'let' is available JS extensions (
- 23 'let' is available JS extensions (
- 25 'const' is availa Mozilla JS exte
- 443 'let' is available JS extensions (
- 444 'let' is available JS extensions (
- 445 'let' is available JS extensions (
- 446 'let' is available JS extensions (
- 447 'let' is available JS extensions (
- 448 'let' is available JS extensions (
- 449 'let' is available JS extensions (
- 450 'let' is available JS extensions (
- 451 'let' is available JS extensions (

```

791             checkDealerScore();
792         }
793     }
794     endDealerTurn();
795 }
796 /**
797  * ends dealer turn and resolves game
798  */
799 function endDealerTurn() {
800     let dealerScore = getHandValue(dealerHand);
801     let betterScore = getHighestPlayerScore(playerHand);
802
803     if (dealerScore > betterScore && dealerScore <= 21) {
804         playerLost();
805     } else if (dealerScore === betterScore && dealerScore <= 21) {
806         draw();
807     } else {
808         playerWon();
809     }
810 }
811 /**
812  * gets the highest player score that is equal/less than 21
813  */
814 function getHighestPlayerScore() {
815     let firstScore = getHandValue(playerHand);
816     let secondScore = getHandValue(splitHand);
817     let highestScore;
818
819     if (firstScore >= secondScore && firstScore <= 21) {
820         highestScore = firstScore;
821     } else {
822         highestScore = secondScore;
823     }
824
825     return highestScore;
826 }
827 /**
828  * places the chip from the score into the pot and settles the balances
829  */
830 function placeBet(chip) {
831     if (score >= chip) {
832         pot.push(chip);
833         let sum = getPotValue();
834         score -= chip;
835         document.getElementById('pot').innerHTML = sum;
836         document.getElementById('score').innerHTML = score;
837     }
838 }
839 /**
840  * gets value of the pot for the above function so that the pot is added correctly
841  */
842 function getPotValue() {
843     let sum = 0;
844     for (let i = 0 ; i < pot.length ; i++) {
845         var chipValue = pot[i];
846         sum += chipValue;
847     }
848     return sum;
849 }
850 /**
851  * resets pot to 0
852  */
853 function resetPot() {
854     pot = [];
855     document.getElementById('pot').innerHTML = 0;
856 }
857 
```



version 2.13.6

(https://github.com/jshint/jshint)

Function with the larg

the median is 0.

About (/about)

Largest function has

is 3.

Install (/install)

The most complex fu

Contribute (/contribute)

value of 28 while the

Blog (/blog)

47 warnings

11 'let' is available

JS extensions (

13 'let' is available

JS extensions (

15 'let' is available

JS extensions (

17 'let' is available

JS extensions (

19 'let' is available

JS extensions (

21 'let' is available

JS extensions (

23 'let' is available

JS extensions (

25 'const' is availa

Mozilla JS exte

443 'let' is available

JS extensions (

444 'let' is available

JS extensions (

445 'let' is available

JS extensions (

446 'let' is available

JS extensions (

447 'let' is available

JS extensions (

448 'let' is available

JS extensions (

449 'let' is available

JS extensions (

450 'let' is available

JS extensions (

451 'let' is available

```

863     }
864
865     /**
866      * updates score sent to html
867      */
868     function updateScore() {
869         document.getElementById('score').innerHTML = score;
870     }
871
872     /**
873      * displays player won, awards them value of the pot before allowing player to reset by playing new game
874      */
875     function playerWon() {
876         let sum = getPotValue();
877         score += sum*2;
878         document.getElementById('score').innerHTML = score;
879         playerTurn = false;
880         dealerTurn = false;
881         betStage = false;
882         displayAlert("You Won!");
883         toggleNewGameShow();
884         toggleConHide();
885         toggleSplitConHide();
886         toggleSplitRevert();
887     }
888
889     /**
890      * displays player lost, removes functions before allowing player to reset by playing new game
891      */
892     function playerLost() {
893         playerTurn = false;
894         dealerTurn = false;
895         betStage = false;
896         displayAlert("You Lost!");
897         toggleNewGameShow();
898         toggleConHide();
899         toggleSplitConHide();
900         toggleSplitRevert();
901     }
902
903     /**
904      * displays a draw, removes functions before allowing player to continue to play a new game
905      */
906     function draw() {
907         let sum = getPotValue();
908         score += sum;
909         document.getElementById('score').innerHTML = score;
910         playerTurn = false;
911         dealerTurn = false;
912         betStage = false;
913         displayAlert("You Draw!");
914         toggleNewGameShow();
915         toggleConHide();
916         toggleSplitConHide();
917         toggleSplitRevert();
918     }
919
920     /**
921      * resets all the arrays and clears the stages
922      */
923     function newGame() {
924         betStage = true;
925         allowDoubleBet = true;
926         splitGame = false;
927         useSplitHand = false;
928         playerHand = [];
929         splitHand = [];
930         dealerHand = [];
931         deck = [];
932         dealCards(playerHand, 'player');
933         dealCards(playerHand, 'player-split');
934         dealCards(dealerHand, 'dealer');

```



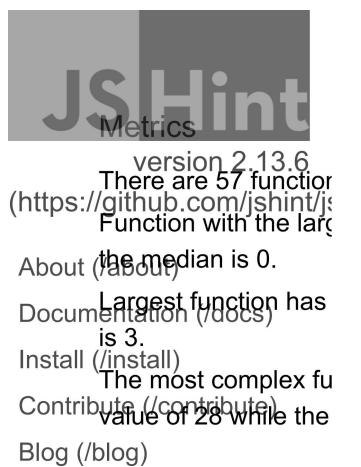
47 warnings

- 11 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 13 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 15 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 17 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 19 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 21 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 23 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 25 'const' is available in Mozilla JS extensions (https://github.com/jshint/jshint#const)
- 443 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 444 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 445 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 446 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 447 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 448 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 449 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 450 'let' is available in JS extensions (https://github.com/jshint/jshint#let)
- 451 'let' is available in JS extensions (https://github.com/jshint/jshint#let)

```

935     checkPlayerScore();
936     checkSplitScore();
937     checkDealerScore();
938     resetPot();
939     toggleBetShow();
940     toggleConHide();
941     toggleSplitConHide();
942     toggleSplitRevert();
943     toggleSplitTrackerHide();
944     toggleDealerTrackerHide();
945     togglePlayerTrackerHide();
946     togglePotHide();
947     hideAlert();
948     placeBet();
949 }
950 /**
951 * toggles the betting button on when the player is in the betting stage
952 */
953 function toggleBetShow() {
954     var div = document.getElementsByClassName('toggle-bet');
955     for (var i = 0; i < div.length; i++) {
956         div[i].style.display = "block";
957     }
958 }
959 /**
960 * toggles the betting button off when the player finishes the betting stage
961 */
962 function toggleBetHide() {
963     var div = document.getElementsByClassName('toggle-bet');
964     for (var i = 0; i < div.length; i++) {
965         div[i].style.display = 'none';
966     }
967 }
968 /**
969 * toggles the control buttons on
970 */
971 function toggleConShow() {
972     var div = document.getElementsByClassName('toggle-con');
973     for (var i = 0; i < div.length; i++) {
974         div[i].style.display = 'flex';
975     }
976 }
977 /**
978 * toggles the control buttons off
979 */
980 function toggleConHide() {
981     var div = document.getElementsByClassName('toggle-con');
982     for (var i = 0; i < div.length; i++) {
983         div[i].style.display = 'none';
984     }
985 }
986 /**
987 * toggles the double button on when it can be played
988 */
989 function toggleDoubleShow() {
990     var div = document.getElementsByClassName('double');
991     for (var i = 0; i < div.length; i++) {
992         div[i].style.display = 'block';
993     }
994 }
995 /**
996 * hides the double button when it cannot be played
997 */
998 function toggleDoubleHide() {
999     var div = document.getElementsByClassName('double');
1000 }
1001 /**
1002 * hides the double button when it cannot be played
1003 */
1004 function toggleDoubleHide() {
1005     var div = document.getElementsByClassName('double');
1006 }

```



47 warnings

- 11 'let' is available JS extensions (
- 13 'let' is available JS extensions (
- 15 'let' is available JS extensions (
- 17 'let' is available JS extensions (
- 19 'let' is available JS extensions (
- 21 'let' is available JS extensions (
- 23 'let' is available JS extensions (
- 25 'const' is availa Mozilla JS exte
- 443 'let' is available JS extensions (
- 444 'let' is available JS extensions (
- 445 'let' is available JS extensions (
- 446 'let' is available JS extensions (
- 447 'let' is available JS extensions (
- 448 'let' is available JS extensions (
- 449 'let' is available JS extensions (
- 450 'let' is available JS extensions (
- 451 'let' is available JS extensions (

```

1007     for (var i = 0; i < div.length; i++) {
1008         div[i].style.display = 'none';
1009     }
1010 }
1011 /**
1012 * shows the split button when it can be played
1013 // */
1014 function toggleSplitShow() {
1015     var div = document.getElementsByClassName('split');
1016     for (var i = 0; i < div.length; i++) {
1017         div[i].style.display = 'block';
1018     }
1019 }
1020 }
1021 /**
1022 * hides the split button when it cannot be played
1023 */
1024 function toggleSplitHide() {
1025     var div = document.getElementsByClassName('split');
1026     for (var i = 0; i < div.length; i++) {
1027         div[i].style.display = 'none';
1028     }
1029 }
1030 }
1031 /**
1032 * hides the split and double button when a split game is run by display none rather than
1033 */
1034 function toggleSplitNone() {
1035     var double = document.getElementsByClassName('double');
1036     var split = document.getElementsByClassName('split');
1037     for (var allDouble = 0; allDouble < double.length; allDouble++) {
1038         double[allDouble].style.display = 'none';
1039     }
1040     for (var allSplit = 0; allSplit < split.length; allSplit++) {
1041         split[allSplit].style.display = 'none';
1042     }
1043 }
1044 }
1045 /**
1046 * shows the split and double button when split game ends
1047 */
1048 function toggleSplitRevert() {
1049     var double = document.getElementsByClassName('double');
1050     var split = document.getElementsByClassName('split');
1051     for (var allDouble = 0; allDouble < double.length; allDouble++) {
1052         double[allDouble].style.display = '';
1053     }
1054     for (var allSplit = 0; allSplit < split.length; allSplit++) {
1055         split[allSplit].style.display = '';
1056     }
1057 }
1058 }
1059 /**
1060 * shows the new game button when it can be played
1061 */
1062 function toggleNewGameShow() {
1063     var div = document.getElementsByClassName('new-game');
1064     for (var i = 0; i < div.length; i++) {
1065         div[i].style.display = 'block';
1066     }
1067 }
1068 }
1069 /**
1070 * hides the new game button when it cannot be played
1071 */
1072 function toggleNewGameHide() {
1073     var div = document.getElementsByClassName('new-game');
1074     for (var i = 0; i < div.length; i++) {
1075         div[i].style.display = 'none';
1076     }
1077 }
1078 }

```



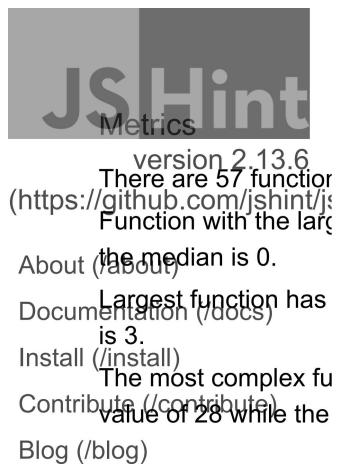
47 warnings

- 11 'let' is available JS extensions (
- 13 'let' is available JS extensions (
- 15 'let' is available JS extensions (
- 17 'let' is available JS extensions (
- 19 'let' is available JS extensions (
- 21 'let' is available JS extensions (
- 23 'let' is available JS extensions (
- 25 'const' is available Mozilla JS exte
- 443 'let' is available JS extensions (
- 444 'let' is available JS extensions (
- 445 'let' is available JS extensions (
- 446 'let' is available JS extensions (
- 447 'let' is available JS extensions (
- 448 'let' is available JS extensions (
- 449 'let' is available JS extensions (
- 450 'let' is available JS extensions (
- 451 'let' is available JS extensions (

```

1079 /**
1080  * shows the hit/stand buttons when a split game is played
1081 */
1082 function toggleSplitConShow() {
1083     var div = document.getElementsByClassName('toggle-con-split');
1084     for (var i = 0; i < div.length; i++) {
1085         div[i].style.display = 'block';
1086     }
1087 }
1088 }
1089 /**
1090  * shows the hit/stand buttons when a split game is played
1091 */
1092 function toggleSplitConHide() {
1093     var div = document.getElementsByClassName('toggle-con-split');
1094     for (var i = 0; i < div.length; i++) {
1095         div[i].style.display = 'none';
1096     }
1097 }
1098 }
1099 /**
1100  * toggles split tracker off
1101 */
1102 function toggleSplitTrackerShow() {
1103     var div = document.getElementById('tracker-split');
1104     div.style.display = 'block';
1105     div.style.visibility = 'visible';
1106 }
1107 }
1108 /**
1109  * toggles split tracker off
1110 */
1111 function toggleSplitTrackerHide() {
1112     var div = document.getElementById('tracker-split');
1113     div.style.display = 'none';
1114     div.style.visibility = 'hidden';
1115 }
1116 }
1117 /**
1118  * toggles player tracker on
1119 */
1120 function togglePlayerTrackerShow() {
1121     var div = document.getElementById('tracker-player');
1122     div.style.display = 'block';
1123     div.style.visibility = 'visible';
1124 }
1125 }
1126 /**
1127  * toggles player tracker off
1128 */
1129 function togglePlayerTrackerHide() {
1130     var div = document.getElementById('tracker-player');
1131     div.style.display = 'none';
1132     div.style.visibility = 'hidden';
1133 }
1134 }
1135 /**
1136  * toggles dealer tracker on
1137 */
1138 function toggleDealerTrackerShow() {
1139     var div = document.getElementById('tracker-dealer');
1140     div.style.display = 'block';
1141 }
1142 }
1143 /**
1144  * toggles dealer tracker off
1145 */
1146 function toggleDealerTrackerHide() {
1147     var div = document.getElementById('tracker-dealer');
1148     div.style.display = 'none';
1149 }
1150 }

```



47 warnings

- 11 'let' is available JS extensions (Mozilla JS exten...)
- 13 'let' is available JS extensions (Mozilla JS exten...)
- 15 'let' is available JS extensions (Mozilla JS exten...)
- 17 'let' is available JS extensions (Mozilla JS exten...)
- 19 'let' is available JS extensions (Mozilla JS exten...)
- 21 'let' is available JS extensions (Mozilla JS exten...)
- 23 'let' is available JS extensions (Mozilla JS exten...)
- 25 'const' is available Mozilla JS exten... (Mozilla JS exten...)
- 443 'let' is available JS extensions (Mozilla JS exten...)
- 444 'let' is available JS extensions (Mozilla JS exten...)
- 445 'let' is available JS extensions (Mozilla JS exten...)
- 446 'let' is available JS extensions (Mozilla JS exten...)
- 447 'let' is available JS extensions (Mozilla JS exten...)
- 448 'let' is available JS extensions (Mozilla JS exten...)
- 449 'let' is available JS extensions (Mozilla JS exten...)
- 450 'let' is available JS extensions (Mozilla JS exten...)
- 451 'let' is available JS extensions (Mozilla JS exten...)

```

1151
1152  /**
1153   * toggles pot on
1154   */
1155 function togglePotShow() {
1156     var div = document.getElementById('pot');
1157     div.style.display = 'block';
1158 }
1159
1160 /**
1161  * toggles pot off
1162  */
1163 function togglePotHide() {
1164     var div = document.getElementById('pot');
1165     div.style.display = 'none';
1166 }
1167
1168 /**
1169  * alert message shown
1170 */
1171 function displayAlert(message) {
1172     var div = document.getElementById('alert');
1173     div.style.display = 'flex';
1174     div.innerText = message;
1175 }
1176
1177 /**
1178  * alert message hide
1179 */
1180 function hideAlert() {
1181     var div = document.getElementById('alert');
1182     div.style.display = 'none';
1183 }
```

