

[| 일반 소스 공유 | >](#)

## (공유) 포커[텍사스 홀덤] v1.3 by seori



seori 챗봇 마스터 🏆 1:1 채팅

2020.01.23. 02:34 조회 68

💬 댓글 5 URL 복사 ⋮

📁 첨부파일 2

이설 -> seori로 닉네임을 변경하였습니다.

### 공유할 소스 용도

=>

텍사스 홀덤(포커)

나중에 세븐 카드 스터드 포커도 제작할 예정이라  
구분하려고 명령어를 /홀덤 + @ 로 통일했었지만  
귀찮으니 그냥 /포커 + @ 로 통일하겠습니다.

### 소스 코드

=>

```
const scriptName="포커.js"
var cards=["s 14", "s 2", "s 3", "s 4", "s 5", "s 6", "s 7", "s 8", "s 9", "s 10", "s 11", "s 12", "s 13", "d 14", "d 2", "d 3", "d 4",
"d 5", "d 6", "d 7", "d 8", "d 9", "d 10", "d 11", "d 12", "d 13", "h 14", "h 2", "h 3", "h 4", "h 5", "h 6", "h 7", "h 8", "h 9", "h
10", "h 11", "h 12", "h 13", "c 14", "c 2", "c 3", "c 4", "c 5", "c 6", "c 7", "c 8", "c 9", "c 10", "c 11", "c 12", "c 13"];
var poker=false;
var Gstart=false;
var player=[];
var player2=[];
var R;
var card1;
var card2;
var cardlist=[];
var common=[];
var coin=[];
var chips=[];
var basicchips=100;
var basicseed=1;
var money=0;
var pot=[];
var pot2=[];
var beting=[];
var call=0;
var m=0;
var pc="";
var isBet=false;
var status=0;
var x="";
var isFirstgame=true;
var allsee="Wu200b".repeat(500);
var nagagi=[];
var preChat=[];
var turnoff=false;
var onoff=[];
```

```
function seol(moyang){
    var mo=moyang.replace(/s/g,3).replace(/d/g,2).replace(/h/g,1).replace(/c/g,0);
```

```

return mo;
}

function lee(sutja){
  var isStraight=0;
  var sut=[];
  var ja=[];

  for (var i=0;i<sutja.length;i++){
    sutja[i]=Number(sutja[i]);
    sut.push(Number(sutja[i]));
    ja.push(Number(sutja[i]));
  }
  for (var i=0;i<sutja.filter(e => e == 14).length;i++){
    sutja.push(1);
    sut.push(1);
    ja.push(1);
  }

  for (var i=14;i>1;i--){
    if (sutja.filter(e => e == i).length==4){ return [9, i]; }
  }

  for (var i=14;i>1;i--){
    if (sutja.filter(e => e == i).length==3){
      for (k=0;k<3;k++){ sut.splice(sut.indexOf(i),1); }
      for (var j=14;j>1;j--){
        if (sut.filter(e => e == j).length>=2){ return [8,i+0.01*j]; }
      }
    }
  }

  for (var i=14;i>9;i--){
    if (sutja.filter(e => e == i).length>=1){
      isStraight+=1;
      if (isStraight==5){ return [6,0]; }
    } else {
      isStraight=0;
      break;
    }
  }

  for (var i=5;i>0;i--){
    if (sutja.filter(e => e == i).length>=1){
      isStraight+=1;
      if (isStraight==5){ return [5,0]; }
    } else {
      isStraight=0;
      break;
    }
  }

  for (var p=13; p>5; p--){
    for (var i=0;i<5;i++){
      if (sutja.filter(e => e == (p-i)).length >=1 ){
        isStraight+=1;
        if (isStraight==5){ return [4,p]; }
      } else { break;}
    }isStraight=0;
  }

  for (var i=14;i>1;i--){
    if (sutja.filter(e => e == i).length==3){ return [3,i]; }
  }
}

```

```

        for (var i=14;i>1;i--){
            if (sutja.filter(e => e == i).length==2){
                for (k=0;k<2;k++){ ja.splice(ja.indexOf(i),1); }
                for (var j=i-1;j>1;j--){
                    if (ja.filter(e => e == j).length==2){ return [2,i+0.01*j]; }
                }
                return [1,i];
            }
        }
    }

    var ans=[0,Math.max.apply(null, sutja)];
    return ans;
}

function grade(CARD){
    var Flushshape;
    var shape=[];
    var num=[];
    var Realgrade=[];
    var isFlush=false;
    var card=[];

    for (var i=0; i<CARD.length; i++){
        shape.push(seol(CARD[i].split(" ")[0]));
        num.push(CARD[i].split(" ")[1]);
        card.push(Number(num[i])*10+Number(shape[i]));
    }

    for (var i=0; i<4; i++){
        if (shape.filter(e => e == i).length>=5){
            isFlush=true;
            Flushshape=i;
            break;
        }
    }

    if (isFlush==true){
        for(var i=0;i<shape.length; i++){
            if (shape[i]!=Flushshape){
                card.splice(card[i],1);
                num.splice(num[i],1);
            }
        }
        if (lee(num)[0]==6){ Realgrade[0].push(12); }
        else if (lee(num)[0]==5){ Realgrade[0].push(11); }
        else if (lee(num)[0]==4){ Realgrade[0].push(10); }
        else { Realgrade[0].push(7); }
    } else { Realgrade[0].push(lee(num)[0]); }

    num.sort((a,b)=>b-a);
    var nnum=0;
    card.sort((a,b)=>b-a);
    var sshape=0;
    for (var i=0;i<num.length;i++){
        nnum+=Math.pow(0.01,i)*num[i];
        sshape+=Math.pow(0.001,i)*card[i];
    }
    Realgrade[0].push(lee(num)[1]);
    Realgrade.push(nnum);
    Realgrade.push(sshape);

    return Realgrade;
}

function sh(shs){
    chs=chs.replace(/s /g "▲?") replace(/d /g "??") replace(/h /g "??") replace(/c /g "▲?") replace(/14/g "A") replace(/13/g

```

```

shs=shs.replace(/s/g, " "); shs=shs.replace(/d/g, " "); shs=shs.replace(/n/g, " "); shs=shs.replace(/c/g, " "); shs=shs.replace(/t/g, " "); shs=shs.replace(/r/g, " "); shs=shs.replace(/l/g, " ");
"K").replace(/12/g, "Q").replace(/11/g, "J");
return shs;
}

```

```

function outgrade(put){
    var first=Number(put[1]).toFixed(0);
    var second=Math.round((Number(put[1])-first)*100);
    var third=Number(put[0]).toFixed(0).replace("12", "로티플").replace("11", "백티플").replace("10", "스티플").replace("9", "포카드").replace("8", "플하우스").replace("7", "플러쉬").replace("6", "마운틴").replace("5", "백스트레이트").replace("4", "스트레이트").replace("3", "트리플").replace("2", "투페어").replace("1", "원페어").replace("0", "탐");

    var ans=first+" "+second+" "+third;

    if (first==0){ return third; }
    else if (second==0){ return first+" "+third; }
    else{return ans;}
}

```

```

function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId){

```

```

    if (preChat[room]==msg){ return 0; }
    preChat[room]=msg;

```

```

        var cmd=msg.split(" ")[0];
        var data=msg.replace(cmd+" ", "");
        if (msg==" /포커 방만들기" && isGroupChat==true){
            if (poker==true){
                replier.reply("이미 생성된 방이 있습니다.");
            }else{
                poker=true;
                R=room;
                replier.reply("방이 생성되었습니다.~n저에게 랜톡으로 ' /포커 참가'라고 보내주시면 참가가 됩니다.~n5명까지 참가할 수 있습니다."+allsee+"~n~n[기본값]~n칩: 100개, 시드: 1개~n이는 /칩 (숫자), /시드 (숫자) 로 변경 가능합니다.~n~n조건 ~n0.2% ≤ 시드/칩 ≤ 5%~n50 ≤ 칩 ≤ 1000");
            }
        }
    }
}

```

```

        if (Gstart==false && poker==true && cmd==" /칩" && isNaN(data)==false){
            if ((data-Math.floor(data))==0){
                if (50<=data && data<=1000){
                    basicchips=Number(data);
                    replier.reply("기본 칩이 "+data+"개로 설정되었습니다.");
                    if (basicseed/basicchips<0.002){
                        basicseed=Math.cell(basicchips*0.002);
                        replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
                    } else if (basicseed/basicchips>0.05){
                        basicseed=Math.floor(basicchips*0.05);
                        replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
                    }
                } else { replier.reply("50에서 1000 사이의 자연수를 입력해주세요."); }
            } else { replier.reply("자연수를 입력해주세요."); }
        }
    }
    if (Gstart == false && poker==true && cmd==" /시드" && isNaN(data)==false){
        if ((data-Math.floor(data))==0){
            basicseed=Number(data);
            if (basicseed/basicchips<0.002){
                basicseed=Math.cell(basicchips*0.002);
                replier.reply("최저시드("+basicseed+"개)로 자동 설정되었습니다.");
            } else if (basicseed/basicchips>0.05){
                basicseed=Math.floor(basicchips*0.05);
                replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
            } else {
                replier.reply("시드가 "+data+"개로 설정되었습니다.");
            }
        } else { replier.reply("자연수를 입력해주세요."); }
    }
}

```

```

    }

    if (msg=="/포커 참가"){
        if (Gstart==false){
            if (poker==true){
                if (room==sender){
                    if (player.length<5){
                        if (player.indexOf(sender)==-1){
                            player.push(sender);
                            player2.push(sender);
                            coin.push(basicchips);
                            chips.push(basicchips);
                        }
                    }
                }
            }
        }
        if (isFirstgame==true){
            coin.fill(basicchips);
            chips.fill(basicchips);
        }

        pot.push(0);
        pot2.push(0);
        cardlist.push([]);
        replier.reply("정상 참가 처리 되었습니다.");
        replier.reply(R, "현재 참가자: "+player);
        } else {
            replier.reply(sender + "님은 이미 참가하셨습니다.");
        }
        } else {
            replier.reply("최대 인원은 5명입니다.");
        }
        } else {
            replier.reply("갯톡으로 보내주셔야 참가가 인정됩니다.");
        }
        } else {
            replier.reply("아직 방이 만들어지지 않았습니다.");
        }
        } else {
            replier.reply("이미 게임이 진행중입니다.");
        }
    }

    if (msg=="/포커 시작" && room==R && player.indexOf(sender)!=-1){
        if (player.length>=2){
            if (poker==true){
                if (Gstart==false){
                    Gstart=true;
                    if (isFirstgame==false && Math.floor(basicseed/basicchips)<0.05){
                        replier.reply("시드를 올리시겠습니까?Wn- /업 (숫자) 또는 /킵 으로 설정 가능Wn주의: 한 번 올린 시드는 되돌릴 수 없습니다.");
                    }
                    while (isFirstgame==false){}
                }
                cards.sort(function(a, b){return 0.5-Math.random()});
                replier.reply("게임을 시작합니다.");
                for(var i=0; i<=player.length-1; i++){
                    card1=cards[cards.length-1];
                    cards.pop();
                    card2=cards[cards.length-1];
                    cards.pop();
                    cardlist[i].push(card1);
                    cardlist[i].push(card2);
                    var ans="핸드 카드를 받았습니다.Wn"+cardlist[i]+"Wn현재 등급: "+outgrade(grade(cardlist[i]))[0]);
                    replier.reply(player[i],sh(ans));
                }
                java.lang.Thread.sleep(1000);
                replier.reply("처음 기본 칩은 "+basicchips+"개입니다. "+basicseed+"/"+basicseed*2+"$ 게임입니다.WnSB: "+player[0]+"님, "+basicseed+"개 지불WnBB: "+player[1]+"님, "+basicseed*2+"개 지불");
                coin[0]=coin[0]-basicseed;
                coin[1]=coin[1]-basicseed*2;
                chip[0]=chip[0]-basicseed;
                chip[1]=chip[1]-basicseed*2;
            }
        }
    }
}

```

```

        cnips[u]=cnips[u]-basicseed;
        chips[1]=chips[1]-basicseed*2;
        money=basicseed*3;
        pot[0]=basicseed;
        pot2[0]=basicseed;
        pot[1]=basicseed*2;
        pot2[1]=basicseed*2;
        if (player.length!=3){
            for (var p=0; p<=3*player.length-1;p++){
                beting.push(player[(3*player.length-1+p)%player.length]);
            }
        } else {
            for (var p=0; p<=3*player.length-1;p++){
                beting.push(player[(2+p)%player.length]);
            }
        }
        pc="";
        for (var k=0; k<=player.length-1;k++){
            pc=pc+player[k]+" : "+pot[k]+"개 배팅/"+chips[k]+"개 남음Wn";
        }
        replier.reply(beting[m]+"님부터 배팅을 시작합니다.Wn/콜, /폴드, /레이즈 (숫자)로 배팅이 가능합니
다.WnWn현재 콜: "+(Math.max.apply(null,pot)-pot[player.indexOf(beting[m]]))+"개Wn레이즈 최대치: "+(2*Math.max.apply(null,
pot)-pot[player.indexOf(beting[m]]))+"개WnWnPOT: "+money+"개WnWn"+pc.trim());
        isBet=true;
    } else {
        replier.reply("이미 게임이 시작되었습니다.");
    }
} else {
    replier.reply("아직 방이 만들어지지 않았습니다.");
}
} else {
    replier.reply("최소 참가자는 2명입니다.");
}
}

if (isFirstgame==false && cmd=="/업" && isNaN(data)==false){
    basicseed+=Math.floor(data);
    if (basicseed/basicchips>0.05){
        basicseed=Math.floor(basicchips*0.05);
        replier.reply("최고시드("+basicseed+"개)로 자동 설정되었습니다.");
        isFirstgame=true;
    } else {
        replier.reply("시드가 "+data+"개 올라 "+basicseed+"개가 되었습니다.");
        isFirstgame=true;
    }
} else if (isFirstgame==false && msg=="/킵"){
    isFirstgame=true;
}

if (isBet==true){
    if (msg == "/체크"){
        if (m==0 && sender==beting[m] && room==R && status>=1){
            replier.reply("체크! 다음 "+beting[m+1]+"님 배팅해주세요.Wn/콜, /폴드, /레이즈 (숫자)로 배팅이 가능합니다.WnWn현
재 콜: "+(Math.max.apply(null, pot)-pot[player.indexOf(beting[m+1]]))+"개Wn레이즈 최대치: "+(2*Math.max.apply(null, pot)-
pot[player.indexOf(beting[m+1]]))+"개WnWnPOT: "+money+"개WnWn"+pc);
            m+=1;
        }
    } else if (msg=="/콜" && beting[m]==sender && room==R){
        if (status==0 || m!=0){
            coin[player.indexOf(sender)]=coin[player.indexOf(sender)]-Math.max.apply(null,
pot)+pot[player.indexOf(sender)];
            chips[player2.indexOf(sender)]=chips[player2.indexOf(sender)]-Math.max.apply(null,
pot)+pot2[player2.indexOf(sender)];
            money = money+ Math.max.apply(null, pot)-pot[player.indexOf(sender)];
            pot[player.indexOf(sender)]=Math.max.apply(null, pot);
            pot2[player.indexOf(sender)]=Math.max.apply(null, pot2);

```

```

        pc="";
        for (var k=0; k<=player.length-1;k++){
            pc=pc+player[k]+": "+pot[k]+"개 배팅/"+coin[k]+"개 남은Wn";
        }
        pc=pc.trim();
        call +=1;
        if (call==player.length-1){
            replier.reply("배팅이 완료되었습니다.WnWnPOT: "+money+"개WnWn"+pc.trim());
            isBet=false;
            status+=1;
            call=0;
        } else {
            replier.reply("다음 "+betting[m+1]+"님 배팅해주세요.Wn/콜, /폴드, /레이즈 (숫자)로 배팅이 가능합니
다.WnWn현재 콜: "+(Math.max.apply(null, pot)-pot[player.indexOf(betting[m+1]]))+"개Wn레이즈 최대치: "+
(2*Math.max.apply(null, pot)-pot[player.indexOf(betting[m+1]]))+"개WnWnPOT: "+money+"개WnWn"+pc);
            m+=1;
        }
    }
} else if (msg==" /폴드" && sender==betting[m] && room==R){
    if (player.length != 2) {
        coin.splice(player.indexOf(sender),1);
        pot.splice(player.indexOf(sender),1);
        cardlist.splice(player.indexOf(sender),1);
        player.splice(player.indexOf(sender),1);
        for (var i=0;i<=2;i++){
            betting.splice(betting.indexOf(sender),1);
        }
        pc="";
        for (var k=0; k<=player.length-1;k++){
            pc=pc+player[k]+": "+pot[k]+"개 배팅/"+coin[k]+"개 남은Wn";
        }
        replier.reply(sender+"님 폴드(다이)하셨습니다.");
        if (call==player.length-1){
            replier.reply("배팅이 완료되었습니다.WnWnPOT: "+money+"개WnWn"+pc.trim());
            isBet=false;
            status+=1;
            call=0;
        } else {
            m=m-Math.floor(m/(player.length+1));
            replier.reply("다음 "+betting[m]+"님 배팅해주세요.Wn/콜, /폴드, /레이즈 (숫자)로 배팅이 가능합니
다.WnWn현재 콜: "+(Math.max.apply(null, pot)-pot[player.indexOf(betting[m]]))+"개Wn레이즈 최대치: "+(2*Math.max.apply(null,
pot)-pot[player.indexOf(betting[m]]))+"개WnWnPOT: "+money+"개WnWn"+pc.trim());
        }
    }
} else {
    player.splice(player.indexOf(sender),1);
    chips[player2.indexOf(player[0])]+=money;
    pc="현재 칩 보유량";
    for (var k=0; k<=player2.length-1;k++){
        pc=pc+"Wn"+player2[k]+": "+chips[k]+"개";
    }
    replier.reply("나머지 선수가 모두 폴드하여 "+player+"님 승리입니다!WnWn"+pc);
    if (nagagi.length!=0){
        for (i=0;i<nagagi.length;i++){
            player2.splice(player2.indexOf(nagagi[i]),1);
        }
        replier.reply(nagagi+"님이 나가셨습니다.Wn남은 참가자: "+player2);
    }

    cards=["s 14", "s 2", "s 3", "s 4", "s 5", "s 6", "s 7", "s 8", "s 9", "s 10", "s 11", "s 12", "s 13", "d 14", "d
2", "d 3", "d 4", "d 5", "d 6", "d 7", "d 8", "d 9", "d 10", "d 11", "d 12", "d 13", "h 14", "h 2", "h 3", "h 4", "h 5", "h 6", "h 7",
"h 8", "h 9", "h 10", "h 11", "h 12", "h 13", "c 14", "c 2", "c 3", "c 4", "c 5", "c 6", "c 7", "c 8", "c 9", "c 10", "c 11", "c 12", "c
13"];

    Gstart=false;
    coin=[];
    player=[];

    pot=[];

```

```

pot2=[];
cardlist=[];
    player2.unshift(player2[player2.length-1]);
    chips.unshift(chips[chips.length-1]);
    player2.pop();
    chips.pop();
        for ( var i=0; i<=player2.length-1;i++){
            coin.push(chips[i]);
            player.push(player2[i]);
        }
        money=0;
        for(var j=0; j<=player.length-1; j++){
            pot.push(0);
            pot2.push(0);
            cardlist.push([]);
        }
        beting=[];
        call=0;
        m=0;
        isBet=false;
        status=0;
        common=[];
        x="";
        isFirstgame=false;
nagagi=[];
    replier.reply("새로운 참가자가 저에게 갬톡으로 '/포커 참가'를 보내주시면 참가 처리됩니다.Wn'/포커 시작'을 입력하시
면 게임을 이어 하실 수 있습니다.");
    }
    } else if (cmd=="레이즈" && sender==beting[m] && room==R){
        if (isNaN(data) == false){
            if ((data-Math.floor(data))==0){
                if ((Math.max.apply(null, pot)-pot[player.indexOf(sender)])<data && data<=
(2*Math.max.apply(null, pot)-pot[player.indexOf(sender)])){
                    if (m<beting.length-player.length){
                        data=Number(data);
                        pot[player.indexOf(sender)]= pot[player.indexOf(sender)]+data;
                        pot2[player2.indexOf(sender)]= pot2[player2.indexOf(sender)]+data;
                        coin[player.indexOf(sender)]=coin[player.indexOf(sender)]-data;
                        chips[player2.indexOf(sender)]=chips[player2.indexOf(sender)]-data;
                        money +=data;
                        call=0;
                        pc="";
                        for (var k=0; k<=player.length-1;k++){
                            pc=pc+player[k]+": "+pot[k]+"개 배팅/"+coin[k]+"개 남음Wn";
                        }
                        replier.reply("다음 "+beting[m+1]+"님 배팅해주세요.Wn/콜, /폴드, /레이즈 (숫자)로 배팅
이 가능합니다.WnWn현재 콜: "+(Math.max.apply(null, pot)-pot[player.indexOf(beting[m+1]]))+"개Wn레이즈 최대치: "+
(2*Math.max.apply(null, pot)-pot[player.indexOf(beting[m+1]]))+"개WnPOT: "+money+"개WnWn"+pc.trim());
                        m+=1;
                    } else {
                        replier.reply("레이즈 횟수 최대치 달성으로 /콜만 가능합니다.");
                    }
                } else {
                    replier.reply("배팅금액이 콜 이하이거나 레이즈 최대치("+(2*Math.max.apply(null, pot)-
pot[player.indexOf(sender)]))+"를 초과하여 레이즈할 수 없습니다.");
                }
            } else {
                replier.reply("자연수만 입력해주세요.");
            }
        } else {
            replier.reply("숫자만 입력해주세요.");
        }
    }
}

if (status==1 && isBet==false){

```



```

    if (status==1 && isBet==false){
        m=0;
        beting=[];
        for (var i=0; i<3*player.length; i++){
            beting.push(player[i%player.length]);
        }
        for (var i=0; i<3; i++){
            common.push(cards[cards.length-1]);
            cards.pop();
        }
        for (var i=0; i<player.length; i++){
            for (var j=0; j<common.length; j++){
                cardlist[i].push(common[j]);
            }
        }
    }
    var ans="공통 카드 3장을 공개합니다.\n"+common;
    replier.reply(sh(ans));
    for (var i=0; i<player.length; i++){
        var ans="현재 나의 카드\n"+cardlist[i]+"현재 등급: "+outgrade(grade(cardlist[i])[0]);
        replier.reply(player[i],sh(ans));
    }
    replier.reply(beting[m]+"님부터 배팅을 시작합니다.\n/체크, /폴드, /레이즈 (숫자) 로 배팅이 가능합니다.\n\n현재 콜: "+
(Math.max.apply(null,pot)-pot[player.indexOf(beting[m]]))+"개\n레이즈 최대치: "+(2*Math.max.apply(null, pot)-
pot[player.indexOf(beting[m]]))+"개\n\nPOT: "+money+"개\n\n"+pc.trim());
    isBet=true;
}

if (status>=2 && status<=3 && isBet==false){
    m=0;
    if (status==3){x="마지막 ";}
    common.push(cards[cards.length-1]);
    cards.pop();
    for (var j=0; j<player.length; j++){
        cardlist[j].push(common[status+1]);
    }
    var ans=x+"추가 공통 카드 1장을 공개합니다: "+common[status+1]+"\n공통 카드: "+common;
    replier.reply(sh(ans));
    for (var i=0; i<player.length; i++){
        var ans="현재 나의 카드\n"+cardlist[i]+"현재 등급: "+outgrade(grade(cardlist[i])[0]);
        replier.reply(player[i],sh(ans));
    }
    replier.reply(beting[m]+"님부터 "+x+"배팅을 시작합니다.\n/체크, /폴드, /레이즈 (숫자) 로 배팅이 가능합니다.\n\n현재
콜: "+(Math.max.apply(null,pot)-pot[player.indexOf(beting[m]]))+"개\n레이즈 최대치: "+(2*Math.max.apply(null, pot)-
pot[player.indexOf(beting[m]]))+"개\n\nPOT: "+money+"개\n\n"+pc.trim());
    isBet=true;
}

if (status==4 && isBet==false){
    var gl=[];
    var gradelist=[];
    var gradelist2=[];
    var ult="공통 카드: "+common+"\n\n핸드 카드를 공개합니다.";
    var Res="";
    replier.reply("마지막 배팅이 종료되었습니다.");
    for (var i=0; i<player.length; i++){
        var gg=player[i]+" : "+outgrade(grade(cardlist[i])[0])+"\n";
        Res=Res+gg;
        gg=player[i]+" : "+cardlist[i][0]+" , "+cardlist[i][1];
        ult=ult+"\n"+gg;
    }
    for (var i=0; i<player.length; i++){
        gl.push(grade(cardlist[i]));
        gradelist.push(grade(cardlist[i]));
        gradelist2.push(grade(cardlist[i])[0][0]);
    }
    for (var i=0; i<gradelist2.length; i++){

```

```

        if (gradelist2[i]!==Math.max.apply(null, gradelist2)){
            gradelist2.splice(i, 1);
            gradelist.splice(i, 1);
        }
    }

    if (gradelist.length!=1){
        for (var i=0;i<gradelist.length;i++){
            gradelist2[i]=gradelist[i][0][1];
        }
        for (var i=0;i<gradelist2.length;i++){
            if (gradelist2[i]!==Math.max.apply(null, gradelist2)){
                gradelist2.splice(i, 1);
                gradelist.splice(i, 1);
            }
        }
    }
}

if (gradelist.length!=1){
    for (var i=0;i<gradelist.length;i++){
        gradelist2[i]=gradelist[i][1];
    }
    for (var i=0;i<gradelist2.length;i++){
        if (gradelist2[i]!==Math.max.apply(null, gradelist2)){
            gradelist2.splice(i, 1);
            gradelist.splice(i, 1);
        }
    }
}

if (gradelist.length!=1){
    for (var i=0;i<gradelist.length;i++){
        gradelist2[i]=gradelist[i][2];
    }
    for (var i=0;i<gradelist2.length;i++){
        if (gradelist2[i]!==Math.max.apply(null, gradelist2)){
            gradelist2.splice(i, 1);
            gradelist.splice(i, 1);
        }
    }
}

var winner=player[gl.map((s,i) => s.toSource() != gradelist[0].toSource() ? false : i).filter(s => s !== false)[0]];
replier.reply(sh(ult));
replier.reply(sh(Res.trim()));
    chips[player2.indexOf(winner)]+=money
    pc="현재 칩 보유량";
    for (var k=0; k<=player2.length-1;k++){
        pc=pc+"\n"+player2[k]+" : "+chips[k]+"개";
    }
    replier.reply("승리자는 "+winner+"님입니다!\n\n"+pc);

if (nagagi.length!=0){
    for (i=0;i<nagagi.length;i++){
        player2.splice(player2.indexOf(nagagi[i]),1);
    }
    replier.reply(nagagi+"님이 나가셨습니다.\n남은 참가자: "+player2);
}

    cards=["s 14", "s 2", "s 3", "s 4", "s 5", "s 6", "s 7", "s 8", "s 9", "s 10", "s 11", "s 12", "s 13", "d 14", "d 2", "d 3", "d 4", "d 5", "d 6", "d 7", "d 8", "d 9", "d 10", "d 11", "d 12", "d 13", "h 14", "h 2", "h 3", "h 4", "h 5", "h 6", "h 7", "h 8", "h 9", "h 10", "h 11", "h 12", "h 13", "c 14", "c 2", "c 3", "c 4", "c 5", "c 6", "c 7", "c 8", "c 9", "c 10", "c 11", "c 12", "c 13"];
    Gstart=false;
    coin=[];
    player=[];

pot=[];
pot2=[];
cardlist=[];
player2.unshift(player2[player2.length-1]);
chips.unshift(64);chips[player2.length-1]=0;

```

```

        chips.unshift(chips[chips.length-1]);
        player2.pop();
        chips.pop();
        for ( var i=0; i<=player2.length-1;i++){
            coin.push(chips[i]);
            player.push(player2[i]);
        }
        money=0;
        for(var j=0; j<=player.length-1; j++){
            pot.push(0);
            pot2.push(0);
            cardlist.push([]);
        }
        beting=[];
        call=0;
        m=0;
        isBet=false;
        status=0;
        common=[];
        x="";
        isFirstgame=false;
        nagagi=[];
        replier.reply("새로운 참가자가 저에게 랜톡으로 '/포커 참가'를 보내주시면 참가 처리됩니다.Wn '/포커 시작'을 입력하시면 게임을 이어하실 수 있습니다.");
    }

    if (msg=="/나가기" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)==-1){
        if (Gstart==true){
            nagagi.push(sender);
            replier.reply(R, sender+"님의 나가가 예약되었습니다. 한 게임이 끝나거나 폴드할 시 자동으로 나가집니다.");
        } else {
            player.splice(player.indexOf(sender),1);
            replier.reply(R, sender+"님이 나가셨습니다.Wn남은 참가자: "+player2);
        }
    }

    if (msg=="/나가기취소" && player2.indexOf(sender)!=-1 && nagagi.indexOf(sender)!=-1){
        nagagi.splice(nagagi.indexOf(sender),1);
        replier.reply(R, sender+"님 나가기 예약 취소");
    }

    if (Gstart==true && player2.length==1){
        replier.reply("참가자가 1명밖에 남지 않아 게임이 자동 종료됩니다.");
        Api.compile("포커.js");
    }

    if (msg=="/포커 종료" && player2.indexOf(sender)!=-1){
        turnoff=true;
        onoff.push(sender);
        replier.reply("/찬성 으로 종료 찬성이 가능합니다. 5초 이내 과반수 이상 찬성 시 게임이 종료됩니다.");
        java.lang.Thread.sleep(5000);
        if (onoff.length>=player2.length/2){
            replier.reply(onoff.length+"명의 찬성으로 게임이 종료됩니다.");
            Api.compile("포커.js");
        } else {
            replier.reply("과반수가 넘지 않아 게임이 계속 진행됩니다.");
        }
    }

    if (turnoff=true && onoff.indexOf(sender)==-1 && msg=="/찬성"){
        onoff.push(sender);
        replier.reply(sender+"님 찬성");
    }
}

```

ver 1.1 초기 칩과 시드를 설정 가능하게 하였습니다.

게임이 끝나고 새 게임을 시작하기 전 새로운 참가자를 받을 수 있도록 하였습니다.

게임이 끝나고 새 게임을 시작할 때 시드를 올릴 지 말 지 선택할 수 있도록 하였습니다.

ver 1.2 게임 나가기, 나가기 예약, 나가기 취소 기능을 추가하였습니다.

참가자들의 투표로 이루어지는 게임 종료 기능을 추가하였습니다.

ver 1.3 명령어가 한 번에 여러번 올라올 경우 한 번만 반응하도록 설정하였습니다.

게임이 끝나고 배팅액 리스트가 초기화되지 않는 오류를 수정하였습니다.

두 번째 이상의 배팅에서 처음에 콜이 되는 오류를 수정하였습니다.

게임이 끝나고 각 플레이어들의 족보리스트가 초기화되지 않는 오류를 수정하였습니다.

\*나가기 기능을 테스트하려면 많은 인원이 필요하여 아직 테스트되지 않았습니다.

\*오류 발견시 제보 바랍니다.

#### \*업데이트 예정

1. 첫 배팅의 최대 레이즈 횟수를 1회로 줄일 예정입니다. (현재 2회)

2. 올인과 사이드배팅 기능을 추가할 예정입니다.

3. '/포커' 입력 시 간단한 룰 설명과 명령어 설명이 나오도록 할 예정입니다.

( 말머리 등록 해주세요! )



seori님의 게시글 더보기 >

❤ 좋아요 0 💬 댓글 5

🔗 공유 | 신고

댓글 등록순 최신순 C

댓글알림 ☐



seori 작성자

열심히 쳐놓은 tab이 다 날아가서 들어쓰기가 뒤죽박죽됐넹

2020.01.23. 02:36 답글쓰기



AlphaDo

궁금한게 이써여  
몇줄이에요?

2020.01.23. 02:49 답글쓰기



seori 작성자

AlphaDo 지금은 육백오십육줄이래여

2020.01.23. 02:52 답글쓰기



알랑뽕까



2020.01.23. 08:34 답글쓰기



seori 작성자

폴드 시 나가기

2020.01.23. 16:21 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

## '| 일반 소스 공유 |' 게시판 글

이 게시판 새글 구독하기 ☐

끝말잇기 능력자 만들다가 [5]	구름	2020.01.24.
(공유) 포커[텍사스 홀덤] v1.4 by seori [6]	seori	2020.01.23.
(공유) 포커[텍사스 홀덤] v1.3 by seori [5]	seori	2020.01.23.
심심해서 만든 욕감지,인사봇 [5]	줄러	2020.01.23.
cpu 코어 개수 가져오기 [7]	쿨칠	2020.01.22.

1 2 3

전체보기

## 이 카페 인기글

path질문

젤리파덕  
♡0 💬2

[ 치\*\*, 한\*\* ] 님 강제 탈퇴

AlphaDo  
♡1 💬2

기나무 곡괭이[노말]의  
최악의 문들 중 하나를

5000분의 1을 얻은 사람

eval 질문

Kiri  
♡0 💬5

안녕하세요

KHP  
♡0 💬6

안녕하세요

ㅇㄷ  
♡0 💬5



틱택토 (Player vs Bot)

간편 차종응답과 한글코딩을 메인으로 제공  
하는 카톡봇앱 개발 예정

성빈  
♡1 💬16



틱택토 (카카오링크 적용)

1 2 3 4 5