

Lists Review

Write a module that can be used to play card games. Include the following functions in your module.

- `make_deck`: creates a list containing the [52 cards](#) (4 suits of 13 cards each) if a standard deck. You should choose a method of storing both the rank (e.g. Jack) and suit (e.g. Hearts) of each card. Popular methods include strings, such as "Jack of Hearts" or "JH", or tuples, such as ("Jack", "Hearts") or (11, 2). Use some form of loop to build the deck, rather than specifying all cards manually. Return the deck of cards.
- `shuffle_deck`: randomly arranges the cards in the deck. As an exercise, try to do this without using the `random` module's `shuffle` method. Return the shuffled deck.
- `deal_card`: remove one card from the top of the deck. Return that card, so it can be used in the main program.
- `make_hand`: remove n cards from the top of the deck and create a list representing a player's hand. Try to use your `deal_card` function to reduce code. Return the hand.
- `play_card`: remove one card from a player's hand. Return that card, so it can be used in the main program. You will need to pass arguments to this function that will be able to identify a card that is in the player's hand. Do not ask for the user to select the card in this function.
- `get_card_name`: obtain the name of a card. For example, if a card is represented as "JH", your function should return "Jack of Hearts". Return the card as a string.
- `print_cards`: displays the names of all cards in a given hand or deck. Call your `get_card_name` function here. This function does not need to return a value.

For your main program, create the deck and deal two hands of n cards each. Alternate between these two hands, either dealing cards from the deck and adding them to a hand, or removing cards from a hand and placing them on a "discard pile" (a list). When removing a card, use a method of your choice to select a valid card from the player's hand. Be sure to display the hands, and the deck, when appropriate (e.g. for testing purposes).

Extension: make an actual game using this module, like *Go Fish* or *Hearts*. Add additional functions as required. Note that this is a complicated undertaking, and will require some time and planning on your part, but you have all of the necessary skills to do this.