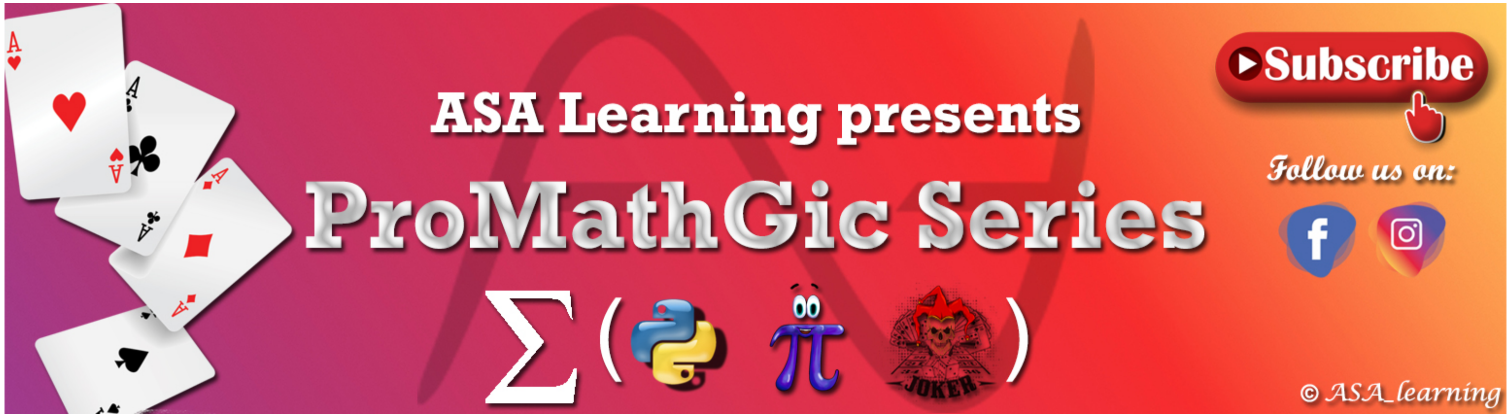

ProMathGic Series #1

Reveal of the card trick explained using Python !

This notebook will use the below basic topics:

- Data types
    - Numbers
    - Strings
    - Lists
- if Statement
- for Loop
- range()
- list comprehension
- Functions

Like | Share | Subscribe to : [ASA Learning (https://www.youtube.com/channel/UCY94xPO3da1xtmKgWQriuPw?sub_confirmation=1)](https://www.youtube.com/channel/UCY94xPO3da1xtmKgWQriuPw?sub_confirmation=1) | [ASA Learning Tamil (https://www.youtube.com/channel/UCRulm3NMzupmtUuX8DlasvQ?sub_confirmation=1)](https://www.youtube.com/channel/UCRulm3NMzupmtUuX8DlasvQ?sub_confirmation=1) | [ASA Learning Kannada (https://www.youtube.com/channel/UCi-eMzVBm4J606TF_jdPL6g?sub_confirmation=1)](https://www.youtube.com/channel/UCi-eMzVBm4J606TF_jdPL6g?sub_confirmation=1)

In [1]:
```
# Create deck of 52 cards
values = ['2','3','4','5','6','7','8','9','10','Jack','Queen','King','Ace']
suites = ['♥', '♣', '♢', '♠']
deck_of_cards = [v + ' of ' + s for s in suites for v in values]
```

In [2]:
```
print(deck_of_cards, 'and total no. of cards is',len(deck_of_cards))
```

```
['2 of ♥', '3 of ♥', '4 of ♥', '5 of ♥', '6 of ♥', '7 of ♥', '8 of ♥', '9 of ♥', '10 of ♥', 'Jack of ♥', 'Queen of ♥', 'King of ♥', 'Ace of ♥', '2 of ♣', '3 of ♣', '4 of ♣', '5 of ♣', '6 o
f ♣', '7 of ♣', '8 of ♣', '9 of ♣', '10 of ♣', 'Jack of ♣', 'Queen of ♣', 'King of ♣', 'Ace of ♣', '2 of ♢', '3 of ♢', '4 of ♢', '5 of ♢', '6 of ♢', '7 of ♢', '8 of ♢', '9 of ♢', '10 of
♢', 'Jack of ♢', 'Queen of ♢', 'King of ♢', 'Ace of ♢', '2 of ♠', '3 of ♠', '4 of ♠', '5 of ♠', '6 of ♠', '7 of ♠', '8 of ♠', '9 of ♠', '10 of ♠', 'Jack of ♠', 'Queen of ♠', 'King of
♠', 'Ace of ♠'] and total no. of cards is 52
```

In [3]:
```
# Split into two equal halves
first_half_deck=deck_of_cards[:26]
```

In [4]:
```
print(first_half_deck)
```

```
['2 of ♥', '3 of ♥', '4 of ♥', '5 of ♥', '6 of ♥', '7 of ♥', '8 of ♥', '9 of ♥', '10 of ♥', 'Jack of ♥', 'Queen of ♥', 'King of ♥', 'Ace of ♥', '2 of ♣', '3 of ♣', '4 of ♣', '5 of ♣', '6 o
f ♣', '7 of ♣', '8 of ♣', '9 of ♣', '10 of ♣', 'Jack of ♣', 'Queen of ♣', 'King of ♣', 'Ace of ♣']
```

In [5]:
```
second_half_deck=deck_of_cards[26:]
```

In [6]:
```
print(second_half_deck)
```

```
['2 of ♢', '3 of ♢', '4 of ♢', '5 of ♢', '6 of ♢', '7 of ♢', '8 of ♢', '9 of ♢', '10 of ♢', 'Jack of ♢', 'Queen of ♢', 'King of ♢', 'Ace of ♢', '2 of ♠', '3 of ♠', '4 of ♠', '5 of ♠', '6
of ♠', '7 of ♠', '8 of ♠', '9 of ♠', '10 of ♠', 'Jack of ♠', 'Queen of ♠', 'King of ♠', 'Ace of ♠']
```

In [7]:
```
def spectator_cut(n,l):
    return(l[n:])
```

In [8]:
```
# Let's say spectator chooses the first half and cuts it to some random number of cards

rem_cards=spectator_cut(10,first_half_deck)
print(rem_cards,'and no. of cards are',len(rem_cards))
```

```
['Queen of ♥', 'King of ♥', 'Ace of ♥', '2 of ♣', '3 of ♣', '4 of ♣', '5 of ♣', '6 of ♣', '7 of ♣', '8 of ♣', '9 of ♣', '10 of ♣', 'Jack of ♣', 'Queen of ♣', 'King of ♣', 'Ace of
♣'] and no. of cards are 16
```

In [9]:

```
rem_cards_2=[i for i in first_half_deck]
for a in rem_cards:
    if a in first_half_deck:
        rem_cards_2.remove(a)
print(rem_cards_2, 'and no. of cards are', len(rem_cards_2))
```

```
['2 of ♥', '3 of ♥', '4 of ♥', '5 of ♥', '6 of ♥', '7 of ♥', '8 of ♥', '9 of ♥', '10 of ♥', 'Jack of ♥'] and no. of cards are 10
```

In [10]:

```
print(second_half_deck)
```

```
['2 of ♢', '3 of ♢', '4 of ♢', '5 of ♢', '6 of ♢', '7 of ♢', '8 of ♢', '9 of ♢', '10 of ♢', 'Jack of ♢', 'Queen of ♢', 'King of ♢', 'Ace of ♢', '2 of ♠', '3 of ♠', '4 of ♠', '5 of ♠', '6
of ♠', '7 of ♠', '8 of ♠', '9 of ♠', '10 of ♠', 'Jack of ♠', 'Queen of ♠', 'King of ♠', 'Ace of ♠']
```

In [11]:

```
print(second_half_deck.index('King of ♠'))# choosen card
```

```
24
```

In [12]:

```
second_half_deck.insert(0, second_half_deck.pop(second_half_deck.index('King of ♠')))
```

In [13]:

```
print(second_half_deck)
```

```
['King of ♠', '2 of ♢', '3 of ♢', '4 of ♢', '5 of ♢', '6 of ♢', '7 of ♢', '8 of ♢', '9 of ♢', '10 of ♢', 'Jack of ♢', 'Queen of ♢', 'King of ♢', 'Ace of ♢', '2 of ♠', '3 of ♠', '4 of
♠', '5 of ♠', '6 of ♠', '7 of ♠', '8 of ♠', '9 of ♠', '10 of ♠', 'Jack of ♠', 'Queen of ♠', 'Ace of ♠']
```

In [14]:

```
combined_cards=rem_cards_2+second_half_deck
print(combined_cards, 'and no of cards are', len(combined_cards))
```

```
['2 of ♥', '3 of ♥', '4 of ♥', '5 of ♥', '6 of ♥', '7 of ♥', '8 of ♥', '9 of ♥', '10 of ♥', 'Jack of ♥', 'King of ♠', '2 of ♢', '3 of ♢', '4 of ♢', '5 of ♢', '6 of ♢', '7 of ♢', '8 of ♢', '9
of ♢', '10 of ♢', 'Jack of ♢', 'Queen of ♢', 'King of ♢', 'Ace of ♢', '2 of ♠', '3 of ♠', '4 of ♠', '5 of ♠', '6 of ♠', '7 of ♠', '8 of ♠', '9 of ♠', '10 of ♠', 'Jack of ♠', 'Queen of
♠', 'Ace of ♠'] and no of cards are 36
```

In [15]:

```
# Combine one from top and one from bottom
for i in range(1,len(combined_cards)+1,2):
    combined_cards.insert(i, combined_cards.pop(35))

print(combined_cards)
```

```
['2 of ♥', 'Ace of ♠', '3 of ♥', 'Queen of ♠', '4 of ♥', 'Jack of ♠', '5 of ♥', '10 of ♠', '6 of ♥', '9 of ♠', '7 of ♥', '8 of ♠', '8 of ♥', '7 of ♠', '9 of ♥', '6 of ♠', '10 of ♥', '5 of ♠',
'Jack of ♥', '4 of ♠', 'King of ♠', '3 of ♠', '2 of ◊', '2 of ♠', '3 of ◊', 'Ace of ◊', '4 of ◊', 'King of ◊', '5 of ◊', 'Queen of ◊', '6 of ◊', 'Jack of ◊', '7 of ◊', '10 of ◊', '8 of
◊', '9 of ◊']
```

In [16]:

```
print(len(rem_cards))
```

```
16
```

In [17]:

```
print(combined_cards[:len(combined_cards)-len(rem_cards)+1][-1])
```

```
King of ♠
```