Python - танымал, кеңінен қолданылатын бағдарламалау тілі және оның қысқа синтаксисі мен оқылатындығына байланысты жаңадан бастаушылар үшін оңай үйренуге болады, ал python тілінде кодтауды үйрену қазіргі уақытта көптеген мүмкіндіктер тудырады. Python тіліндегі бағдарламаларды үйрену сізге тіл мен оның синтаксисімен көбірек танысуға көмектеседі, бұл болашақта кодты жазуды және түсінуді жеңілдетеді.

Python тіліндегі 60 іргелі бағдарламалар

Бұл мақалада біз Python тіліндегі 60 іргелі бағдарламаларды, яғни сіз үйренуге және шешуге болатын Python бағдарламалау мысалдарын көреміз, өйткені олар сіздің проблемаларды шешу және сыни ойлау дағдыларын жақсартуға көмектеседі және күрделі мәселелерді қалай шешу керектігі туралы түсінік береді. кішірек, басқарылатын бөліктерге бөліңіз, содан кейін әрбір бөлік үшін шешім табыңыз.

1. Элементтердің қосындысын табуға арналған Python бағдарламасы

Python тіліндегі тізім кез келген басқа бағдарламалау тіліндегі массив сияқты. Біз тізімнің барлық элементтерінің қосындысын табу үшін Python бағдарламасын жазамыз.

def arraySum(arr):

sum = 0

for i in arr:

sum += i

return sum

print(arraySum([1, 2, 3, 4, 5]))

Шығару:

15

2. Санның жұп немесе тақ екенін тексеруге арналған Python бағдарламасы

Жұп сандар 2-ге бөлінеді, ал тақ сандар 2-ге бөлінбейді. Санның 2-ге толық бөлінетінін немесе бөлінбейтінін тексеру үшін модуль операторын қолданамыз.

def is\_even(n):

if n%2==0:

return True

else:

return False

print(is\_even(3))

**Шығару:**

False, True

3. Барлық элементтердің орташа мәнін табуға арналған Python бағдарламасы

Тізімді алып, ондағы барлық элементтердің орташа мәнін басып шығарыңыз. Сандардың орташа мәні барлық сандарды қосып, содан кейін сол сандардың санына бөлу арқылы есептеледі.

def average(arr):

sum = 0

for i in arr:

sum += i

return sum / len(arr)

print(average([1, 2, 3, 4, 5]))

**Шығару:**

3.0

4. Жолдағы дауысты дыбыстардың санын санауға арналған Python бағдарламасы

Дауысты дыбыстар – «а, е, и, о, у». Жолдағы дауысты дыбыстардың санын тексеру үшін питон бағдарламасын жазамыз.

def countVowels(str):

vowels = ['a', 'e', 'i', 'o', 'u']

count = 0

for i in str:

if i in vowels:

count += 1

return count

print('number of vowels in hello is '+str(countVowels('hello')))

**Шығару:**

number of vowels in hello is 2

5. Жолдың палиндром екенін тексеруге арналған Python бағдарламасы

Палиндрома – алға және артқа бірдей оқылатын сөз. Мысалы – «Деңгей», «Нун»

my\_str = 'malayalam'

print(my\_str==my\_str[::-1])

**Шығару:**

True

6. Тізімді кері айналдыруға арналған Python бағдарламасы

Біз екі нүктелі тәсілді қолданып, python тілінде тізімнің ретін өзгерту бағдарламасын жазамыз.

def reverse\_list(arr):

left = 0

right = len(arr)-1

while (left < right):

temp = arr[left]

arr[left] = arr[right]

arr[right] = temp

left += 1

right -= 1

return arr

arr = [1, 2, 3, 4, 5, 6, 7]

print(reverse\_list(arr))

# another and short approach

print(arr[::-1])

**Шығару:**

[7, 6, 5, 4, 3, 2, 1]

7. Санның факториалын табуға арналған Python бағдарламасы

Санның факториалы 1-ден осы санға дейінгі барлық натурал сандардың көбейтіндісі болып табылады. Мысалы, 5-тің факториалы 5-ке тең! = 5\*4\*3\*2\*1 = 120

def factorial(num):

if num == 1:

return 1

else:

return num \* factorial(num - 1)

print(factorial(5))

**Шығару:**

120

8. Тізімдегі барлық сөздерді біріктіруге арналған Python бағдарламасы

arr = ['hello', 'world']

st = ''

for i in arr:

st += i

print(st)

**Шығару:**

helloworld

9. Жолдағы барлық дауысты дыбыстарды жоюға арналған Python бағдарламасы

s = 'hello'

vowels = ['a', 'e', 'i', 'o', 'u']

newStr = ''

for i in s:

if i not in vowels:

newStr += i

print(newStr)

**Шығару:**

hll

10. Python тіліндегі Армстронг сандар бағдарламасы

Армстронг саны - бұл цифрлар санының дәрежесіне көтерілген өз цифрларының қосындысына тең сан. Мысалы, 371 - Армстронг саны, себебі 3^3 + 7^3 + 1^3 = 371.

n = 371

num\_digits = len(str(n))

sum = 0

for digit in str(n):

sum += int(digit)\*\*num\_digits

print(sum == n)

**Шығару:**

True

11. Тізімнен көшірмелерді жоюға арналған Python бағдарламасы

Жиын тек бірегей элементтерді қамтитын деректер құрылымы болып табылады. Біз барлық қайталанатын элементтерді жою үшін python жиынтығын қолданамыз.

my\_list = [1, 2, 3, 2, 1]

new\_list = list(set(my\_list))

print(new\_list)

**Шығару:**

[1, 2, 3]

12. Fibonacci Series бағдарламасы

Фибоначчи сериясы - әрбір сан алдыңғы екі санның қосындысы болатын сандар тізбегі. Мысалы, Фибоначчи тізбегіндегі алғашқы бірнеше сандар 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, т.б.

n = 10

a, b = 0, 1

print(a)

print(b)

for i in range(2, n):

c = a + b

print(c)

a, b = b, c

**Шығару:**

0 1 1 2 3 5 8 13 21 34

13. Тізімдегі барлық тақ элементтерді алып тастаңыз

arr = [1, 2, 3, 4, 5]

newArr = []

for i in arr:

if i % 2 == 0:

newArr.append(i)

print(newArr)

**Шығару:**

[2, 4]

14. Python тіліндегі «Кібісе жыл» бағдарламасы

Кібісе жыл - қосымша күні бар жыл (29 ақпан). Кібісе жылы 366 күннен тұрады.

year = 2000

if year % 400 == 0:

print('leap year')

elif year % 4 == 0:

print('leap year')

elif year % 100 == 0:

print('not a leap year')

else:

print('not a leap year')

**Шығару:**

leap year

15. Python тіліндегі қарапайым сандар бағдарламасы

Жай сан деп екі кіші натурал санның көбейтіндісі болып табылмайтын 1-ден үлкен натурал санды айтады. Алғашқы бірнеше жай сандар 2, 3, 5, 7 және 11.

n = 7

is\_prime = True

if n < 2:

is\_prime = False

else:

for i in range(2, n):

if n % i == 0:

is\_prime = False

break

if is\_prime:

print("Prime")

else:

print("Not prime")

**Шығару:**

Prime

16. Python тіліндегі ЕКОЕ бағдарламасы

Екі немесе одан да көп сандардың ең кіші ортақ еселігі (ЕКОЕ) барлық сандардың еселігі болып табылатын ең кіші сан болып табылады.

a = 6

b = 9

lcm = a

while lcm % b != 0:

lcm += a

print(f"LCM of {a} and {b} is {lcm}.")

**Шығару:**

LCM of 6 and 9 is 18.

17. Python тіліндегі ЕҮОБ бағдарламасы

Екі немесе одан да көп сандардың ең үлкен ортақ бөлгіш (ЕҮОБ) барлық сандарды дәл бөлетін ең үлкен сан болып табылады. Мысалы, 15 пен 20-ның HCF 5-ке тең, өйткені 5 - 15 пен 20-ны дәл бөлетін ең үлкен сан.

a = 15

b = 20

if a > b:

smaller = b

else:

smaller = a

for i in range(1, smaller+1):

if (a % i == 0) and (b % i == 0):

hcf = i

print(f"HCF of {a} and {b} is {hcf}")

**Шығару:**

HCF of 15 and 20 is 5

18. Екі санды ауыстыруға арналған Python бағдарламасы

first = 27

second = 4

temp = first

first = second

second = temp

print(f"After swapping, the first number is {first} and the second number is {second}")

**Шығару:**

After swapping, the first number is 4 and the second number is 27.

19. n-ге дейінгі барлық жай сандарды басып шығаруға арналған Python бағдарламасы

n = 10

for i in range(2, n+1):

is\_prime = True

for j in range(2, i):

if i % j == 0:

is\_prime = False

break

if is\_prime:

print(i)

**Шығару:**

2 3 5 7

20. Үш санның ең үлкенін табуға арналған Python бағдарламасы

num1 = 10

num2 = 14

num3 = 12

if (num1 >= num2) and (num1 >= num3):

largest = num1

elif (num2 >= num1) and (num2 >= num3):

largest = num2

else:

largest = num3

print(largest)

**Шығару:**

14

21. Көбейту кестесінің бағдарламасы

num = 19

for i in range(1, 11):

print(num\*i)

**Шығару:**

19 38 57 76 95 114 133 152 171 190

22. Массивті (тізімді) сұрыптауға арналған Python бағдарламасы

numbers = [2, 3, 4, 1]

numbers.sort()

print(numbers)

**Шығару:**

[1, 2, 3, 4]

23. Барлық символдарды бас әріпке түрлендіруге арналған Python бағдарламасы

s = 'follow us'

newStr = ''

for i in s:

newStr += i.upper()

print(newStr)

**Шығару:**

FOLLOW US

24. Әрбір сөздің бірінші әрпін бас әріппен жазуға арналған Python бағдарламасы

s = 'copy assignment'

newStr = ''

for i in range(len(s)):

if i == 0 or s[i - 1] == ' ':

newStr += s[i].upper()

else:

newStr += s[i]

print(newStr)

**Шығару:**

Copy Assignment

25. Барлық таңбаларды кіші әріптерге түрлендіруге арналған Python бағдарламасы

s = 'COPYASSIGNMENT'

newStr = ''

for i in s:

newStr += i.lower()

print(newStr)

**Шығару:**

copyassignment

26. Жолды кері айналдыруға арналған Python бағдарламасы

s = 'tnemngissAypoC'

newStr = ""

for i in s:

newStr = i + newStr

print(newStr)

**Шығару:**

CopyAssignment

27. Python тіліндегі калькулятор бағдарламасы

Біз қосу, азайту, көбейту және бөлуге болатын негізгі калькулятор бағдарламасын жазамыз.

def add(a,b):

return a+b

def sub(a,b):

return a-b

def multi(a,b):

return a \* b

def div(a,b):

return a / b

print(add(10,15))

print(sub(13,10))

**Шығару:**

25 3

28. Пирамида үлгісіне арналған Python бағдарламасы

n=5

for i in range(0, n):

for j in range(0, i+1):

print("\* ",end="")

print("\r")

**Шығару:**



29. Кері пирамида үлгісі

n=5

k = 2\*n - 2

for i in range(0, n):

for j in range(0, k):

print(end=" ")

k = k - 2

for j in range(0, i+1):

print("\* ", end="")

print("\r")

**Шығару:**



30. Python тіліндегі Continuous Character pattern бағдарламасы

n = 5

num = 65

for i in range(0, n):

for j in range(0, i+1):

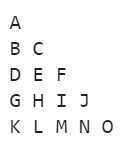
ch = chr(num)

print(ch, end=" ")

num = num + 1

print("\r")

**Шығару:**



31. Triangle басып шығаруға арналған Python бағдарламасы

n=5

k = n - 1

for i in range(0, n):

for j in range(0, k):

print(end=" ")

k = k - 1

for j in range(0, i+1):

print("\* ", end="")

print("\r")

**Шығару:**



32. Кері үшбұрыш бағдарламасы

rows = 4

k = 2 \* rows - 2

for i in range(rows, -1, -1):

for j in range(k, 0, -1):

print(end=" ")

k = k + 1

for j in range(0, i + 1):

print("\*", end=" ")

print("")

**Шығару:**



33. Алмаз пішінін басып шығаруға арналған Python бағдарламасы

Алмаз пішінін басып шығару үшін алдымен үшбұрыштың жоғарғы жартысын, содан кейін төменгі жарты үшбұрышты басып шығарамыз.

n = 4

k = 2 \* n - 2

for i in range(0, n):

for j in range(0, k):

print(end=" ")

k = k - 1

for j in range(0, i + 1):

print("\* ", end="")

print("")

k = n - 2

for i in range(n, -1, -1):

for j in range(k, 0, -1):

print(end=" ")

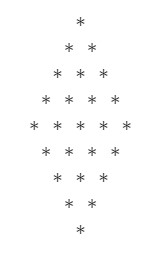
k = k + 1

for j in range(0, i + 1):

print("\* ", end="")

print("")

**Шығару:**



34. Print Hourglass Pattern бағдарламасы

Құм сағатының пішінін басып шығару үшін алдымен төмен қарай үшбұрышты, сосын жоғары қарай үшбұрышты басып шығарамыз.

row = 5

for i in range(row, 0, -1):

for j in range(row-i):

print(" ", end="")

for j in range(1, 2\*i):

print("\*", end="")

print()

for i in range(2, row+1):

for j in range(row-i):

print(" ", end="")

for j in range(1, 2\*i):

print("\*", end="")

print()

**Шығару:**



35. Print Butterfly Pattern бағдарламасы

«Көбелек» үлгісін басып шығару үшін алдымен жоғарғы жартысын, содан кейін төменгі жартысын басып шығарудың бірдей тәсілін орындаймыз.

row = 8

n = row//2

for i in range(1,n+1):

for j in range(1, 2\*n+1):

if j>i and j< 2\*n+1-i:

print(" ", end="")

else:

print("\* ", end="")

print()

for i in range(n,0,-1):

for j in range(2\*n,0,-1):

if j>i and j< 2\*n+1-i:

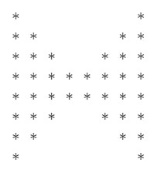
print(" ", end="")

else:

print("\* ", end="")

print()

**Шығару:**



36. Python бағдарламасы арқылы шырша үлгісін басып шығарыңыз

Шырша үлгісін басып шығару үшін оны үш бөлікке бөлуге болады. Біріншісі - жоғарғы үшбұрыш, екіншісі - орта үшбұрыш, үшінші - тік төртбұрыш.

n = 5

#upper triangle

for i in range(n):

for j in range(n-i):

print(' ', end=' ')

for k in range(2\*i+1):

print('\*',end=' ')

print()

#mid triangle

for i in range(n):

for j in range(n-i):

print(' ', end=' ')

for k in range(2\*i+1):

print('\*',end=' ')

print()

#pole shape

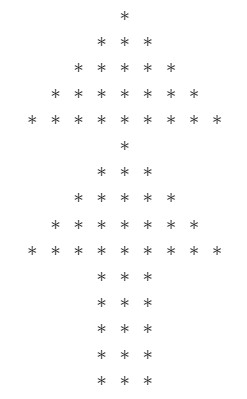
for i in range(n):

for j in range(n-1):

print(' ', end=' ')

print('\* \* \*')

**Шығару:**



37. Жолдағы сөздерді кері айналдыруға арналған Python бағдарламасы

string = "more for copyassignment visit"

s = string.split()[::-1]

l = []

for i in s:

l.append(i)

print(" ".join(l))

**Шығару:**

visit copyassignment for more

38. Екі жолдың анаграмма екенін тексеруге арналған Python бағдарламасы

Анаграмма – бір сөздің әріптерін қайта орналастыру арқылы жасалған сөз. Мысалы, «тыңдау» және «үнсіз» - анаграммалар, өйткені олардың құрамында бірдей әріптер бар, бірақ басқаша реттелген.

s1='listen'

s2='silent'

if(sorted(s1)== sorted(s2)):

print("The strings are anagrams")

else:

print("The strings aren't anagrams")

**Шығару:**

The strings are anagrams

39. Python бағдарламасы берілген диапазонда 3-ке немесе 5-ке бөлінетін сандарды басып шығаруға арналған

n = 30

for i in range(1, n):

if i % 3 == 0 or i % 5 == 0:

print(i)

**Шығару:**

3 5 6 9 10 12 15 18 20 21 24 25 27

40. Барлық сөздерді алфавит бойынша сұрыптауға арналған Python бағдарламасы

arr = ['hello', 'world', 'its', 'copyassignment', 'website']

for i in range(len(arr)):

for j in range(i + 1, len(arr)):

if arr[i] > arr[j]:

arr[i], arr[j] = arr[j], arr[i]

print(arr)

**Шығару:**

['copyassignment', 'hello', 'its', 'website', 'world']

41. Екі қосынды есебіне арналған Python бағдарламасы

Сандар тізімін ескере отырып, қосындысы берілген мақсатқа тең екі санды табуымыз керек

def twoSum(num\_arr, pair\_sum):

for i in range(len(num\_arr) - 1):

for j in range(i + 1, len(num\_arr)):

if num\_arr[i] + num\_arr[j] == pair\_sum:

print("Pair with sum", pair\_sum,"is: ", num\_arr[i],",",num\_arr[j])

num\_arr = [3, 5, 2, 4, 8, 1]

pair\_sum = 4

twoSum(num\_arr, pair\_sum)

**Шығару:**

Pair with sum 4 is: 3 , 1

42. Perfect санға арналған Python бағдарламасы

Мінсіз сан - бұл санның өзін есептемегенде оның тиісті оң бөлгіштерінің қосындысына тең натурал сан. Мысалы, 6 санының оң бөлгіштері 1, 2 және 3, олардың қосындысы 6-ға жетеді. Демек, 6 - тамаша сан.

n = 6

sum1 = 0

for i in range(1, n):

if(n % i == 0):

sum1 = sum1 + i

if (sum1 == n):

print("Perfect number")

else:

print("Not a Perfect number")

**Шығару:**

Perfect number

43. Күшті санға арналған Python бағдарламасы

Күшті сан деп оның цифрларының факториалдарының қосындысы санның өзіне тең болатын санды айтады. Мысалы, 145 - күшті сан, өйткені 1! + 4! + 5! = 1 + 24 + 120 = 145.

sum=0

num=145

temp=num

while(num):

i=1

fact=1

r=num%10

while(i<=r):

fact=fact\*i

i=i+1

sum=sum+fact

num=num//10

if(sum==temp):

print("Strong number")

else:

print("Not a Strong number")

**Шығару:**

Strong number

44. Санның жай көбейткіштерін табуға арналған Python бағдарламасы

Санның жай көбейткіштері - бұл санды толық бөлетін жай сандар. Мысалы, 15 санының жай көбейткіштері 3 және 5, өйткені 3 және 5 жай сандар, ал 3 х 5 = 15.

n=15

i=1

print("Factors:")

while(i<=n):

k=0

if(n%i==0):

j=1

while(j<=i):

if(i%j==0):

k=k+1

j=j+1

if(k==2):

print(i)

i=i+1

**Шығару:**

Factors: 3 5

45. Python тілінде санды кері айналдыру бағдарламасы

Санды кері қайтару үшін біз мына қадамдарды орындаймыз.

Модуль(%) операторы арқылы санның соңғы цифрын алыңыз.

Осы соңғы санды бір жерде жаңа айнымалыда сақтаңыз.

Ағымдағы саннан соңғы санды 10-ға (/) басу арқылы алып тастаңыз.

Ағымдағы нөмір бос болғанша жоғарыдағы қадамдарды қайталаңыз.

n= 123456

reversed=0

while(n>0):

dig=n%10

reversed=reversed\*10+dig

n=n//10

print("Reversed number is:",reversed)

**Шығару:**

Reversed number is: 654321

46. Сандағы барлық цифрлардың қосындысын табуға арналған Python бағдарламасы

Санның барлық цифрларын қосу үшін біз аздаған өзгерісі бар санды кері қайтару сияқты қадамдарды орындаймыз. Нәтижелік санды сақтаудың орнына, біз бұл сандарды қосынды айнымалысына қосуды жалғастырамыз.

n=12345

total=0

while(n>0):

lastDigit=n%10

total=total+lastDigit

n=n//10

print("Total sum of digits in number:",total)

**Шығару:**

Total sum of digits in number: 15

47. Достық сандарды тексеруге арналған Python бағдарламасы

Бірінші санның бөлгіштерінің қосындысы екінші санға, ал екінші санның бөлгіштерінің қосындысы бірінші санға тең болса, екі сан достық сандар болып табылады. Мысалы, 220 және 284 - достық сандар.

a=220

b=284

sum1=0

sum2=0

for i in range(1,a):

if a%i==0:

sum1+=i

for j in range(1,b):

if b%j==0:

sum2+=j

if(sum1==b and sum2==a):

print('Amicable')

else:

print('Not Amicable')

**Шығару:**

Amicable

48. Ауқымдағы Perfect Squares басып шығаруға арналған Python бағдарламасы

start = int(input("Enter the Lower bound"))

end = int(input("Enter the upper bound"))

for n in range(start, end+1):

sum = 0

for i in range(1, n):

if n%i == 0:

sum += i

if n == sum:

print(n)

**Шығару:**

Enter the Lower bound 1

Enter the upper bound 50

6 28

49. Identity Matrix басып шығаруға арналған Python бағдарламасы

Сәйкестік матрицасы – диагональында 1 саны (жоғарғы сол жақтан төменгі оңға қарай) және басқа жерде 0 болатын шаршы матрица.

rows=3

for i in range(0,rows):

for j in range(0,rows):

if(i==j):

print("1",sep=" ",end=" ")

else:

print("0",sep=" ",end=" ")

print()

**Шығару:**



50. Үшінші айнымалыны қолданбай екі санды ауыстыруға арналған Python бағдарламасы

Бұл сіз бір рет үйренуіңіз керек әйгілі сұхбат сұрағы. Үшінші айнымалысыз екі санды ауыстыру үшін біз мына қадамдарды орындаймыз

Екі санды қосып, оны бірінші нөмірге тағайындаңыз

Екі санды да азайтып, екінші санға тағайындаңыз

Қайтадан екі санды да алып, бірінші санға тағайындаңыз

Екі санды дәл осылай ауыстыру үшін бөлу және көбейту амалдарын да пайдалана аламыз.

a=4

b=10

a=a+b

b=a-b

a=a-b

print("a is",a,"and b is",b)

**Шығару:**

a is 10 and b is 4

51. Циклдерді қолданбай ауқымдағы сандарды басып шығару

Біз 1-ден 10-ға дейінгі аралықтағы сандарды циклсыз басып шығару үшін қарапайым рекурсивті бағдарламаны қолданамыз.

def printno(upper):

if(upper>0):

printno(upper-1)

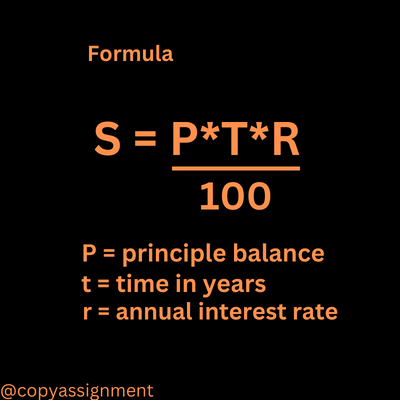
print(upper)

printno(10)

**Шығару:**

1 2 3 4 5 6 7 8 9 10

52. Қарапайым қызығушылықты табуға арналған Python бағдарламасы



principle=float(input("Principle amount:"))

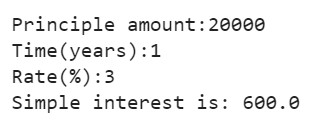
time=int(input("Time(years):"))

rate=float(input("Rate(%):"))

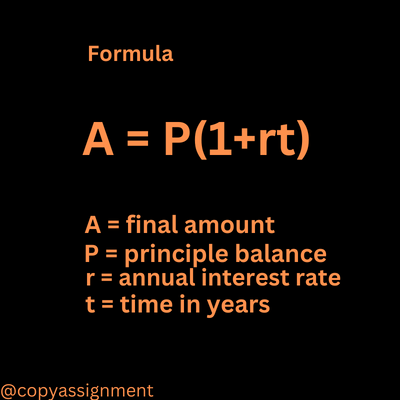
simple\_interest=(principle\*time\*rate)/100

print("Simple interest is:",simple\_interest)

**Шығару:**



53. Құрама қызығушылықты табуға арналған Python бағдарламасы



principal = float(input("Principle amount:"))

time=int(input("Time(years):"))

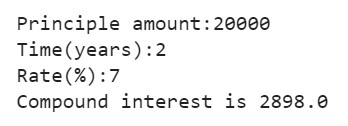
rate=float(input("Rate(%):"))

Amount = principal \* (pow((1 + rate / 100), time))

CI = Amount - principal

print("Compound interest is", CI)

**Шығару:**



54. Цельсийді Фаренгейтке түрлендіруге арналған Python бағдарламасы

celsius=32

f=(celsius\*1.8)+32

print("Temperature in farenheit is:",f)

**Шығару:**

Temperature in farenheit is: 89.6

55. Оқушының бағасын есептеуге арналған Python бағдарламасы

eng=int(input("Enter marks of English: "))

hindi=int(input("Enter marks of Hindi: "))

math=int(input("Enter marks of Math: "))

sci=int(input("Enter marks of Science: "))

sanskrit=int(input("Enter marks of588 Sanskrit: "))

avg=(eng+hindi+math+sci+sanskrit)/5

if(avg>=90):

print("Grade: A")

elif(avg>=80 and avg<90):

print("Grade: B")

elif(avg>=70 and avg<80):

print("Grade: C")

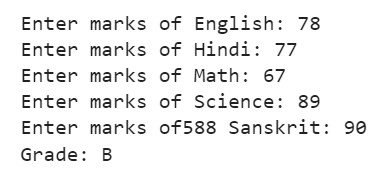
elif(avg>=60 and avg<70):

print("Grade: D")

else:

print("Grade: F")

**Шығару:**



56. Массивтің(тізімнің) ең үлкен элементін табыңыз.

list = [2, 4, 1, 16, 25, 3]

max = list[0]

for x in list:

if x > max:

max = x

print(max)

**Шығару:**

25

57. Python тіліндегі екі тізімнің қиылысуын табуға арналған Python бағдарламасы

Екі нысанның қиылысы екі нысанда да ортақ элементтерді ғана қосуды білдіреді.

list1 = [4, 9, 1, 17, 11, 26, 28, 54, 69]

list2 = [9, 9, 74, 21, 45, 11, 63, 28, 26]

list3 = [value for value in list1 if value in list2]

print(list3)

**Шығару:**

[9, 11, 26, 28]

58. Паскаль үшбұрышын басып шығаруға арналған Python бағдарламасы

n=5

a=[]

for i in range(n):

a.append([])

a[i].append(1)

for j in range(1,i):

a[i].append(a[i-1][j-1]+a[i-1][j])

if(n!=0):

a[i].append(1)

for i in range(n):

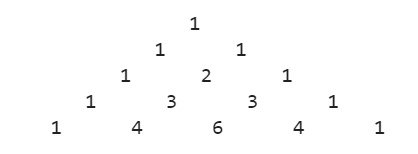
print(" "\*(n-i),end=" ",sep=" ")

for j in range(0,i+1):

print('{0:6}'.format(a[i][j]),end=" ",sep=" ")

print()

**Шығару:**



59. Quotient және Remainder басып шығаруға арналған Python бағдарламасы

a=325

b=15

print("Ондық бөлік:",a//b)

print("Қалдық бөлік:",a%b)

**Шығару:**

Quotient: 21 Remainder: 10

60. Бірінші n санның қосындысын басып шығаруға арналған Python бағдарламасы

n = int(input("Enter a number: "))

sum = 0

while(n > 0):

sum += n

n -= 1

print("Total sum is",sum)

**Шығару:**

Enter a number: 10 Total sum is 55

Қорытынды

Бұл мақалада біз 60 Python бағдарламалау мысалдарын және Python тіліндегі әртүрлі іргелі бағдарламаларды әртүрлі питон тақырыптарында қарастырдық, бұл сізге осы ұғымдарды көптеген әртүрлі мәселелерге үйренуге және қолдануға көмектеседі. Питон тілінде мәселені шешуді жақсы меңгеру үшін осы мақаладағы барлық сұрақтарды шешуді бастауды ұсынамыз.

Python бағдарламаларының тізімі сізге ұнады деп үміттенеміз.

1. Değişkenler ve Veri Tipleri:

* Tam sayılar (int), ondalıklı sayılar (float), metinsel veriler (str), diziler (list) ve sözlükler (dictionary) gibi veri tiplerini öğrenin.
* Değişkenleri tanımlayın ve veri tiplerini kullanın.

1. Kontrol Yapıları:

* Koşullu ifadeleri (if-else) öğrenin ve uygulayın.
* For ve while döngüleri ile program akışını kontrol edin.

1. Fonksiyonlar:

* Kendi fonksiyonlarınızı tanımlayın ve kullanın.
* Fonksiyonlar aracılığıyla kod tekrarını azaltın ve daha okunabilir kod oluşturun.

1. Dosya İşlemleri:

* Dosyalar üzerinde okuma ve yazma işlemleri yapın.
* Dosyalardaki verileri okuyun ve değiştirin veya dosyalara veri yazın.

1. Modüller ve Paketler:

* Python'daki standart modülleri ve diğer ek paketleri keşfedin.
* Modüller ve paketler aracılığıyla daha fazla fonksiyonalite ekleyin.

1. Hata Yakalama:

* Try-except blokları ile hataları yakalayın ve ele alın.
* Programınızın istediğiniz şekilde çalışmasını sağlayın.

1. Veri Yapıları:

* Listeler, sözlükler, kümeler ve diğer veri yapılarını öğrenin.
* Veri yapılarını kullanarak daha etkili ve verimli programlar oluşturun.

1. Nesne Yönelimli Programlama:

* Sınıflar ve nesne tanımlamalarını öğrenin.
* Nesnelerin özelliklerini ve metodlarını kullanarak daha etkili programlar oluşturun.

1. Veri Analizi ve Görselleştirme:

* Veri setleri üzerinde işlem yapmak için kütüphaneleri öğrenin.
* Veri analizi ve görselleştirme yapmak için güçlü araçlar edinin.

1. Uygulama Geliştirme:

* Python ile GUI uygulamaları ve web uygulamaları geliştirin.
* Python ile yapılabilecek olan birçok farklı türde uygulama geliştirin, örneğin masaüstü uygulamaları, veritabanı uygulamaları ve veri analitik uygulamaları.

Bu başlıklar, Python programlama dilini öğrenme yolculuğunuzun temel adımlarını içermektedir. İleri seviyede, bu konuları daha derinlemesine araştırarak ve farklı uygulamalar üzerinde çalışarak becerilerinizi ve bilginizi geliştirebilirsiniz.

İleri Seviye Python Öğrenme Başlıkları:

1. Modüller ve Paketler: Python'daki farklı modülleri (örneğin, NumPy, Pandas, Matplotlib vb.) ve paketleri nasıl kullanabileceğinizi öğrenin.
2. Dosya İşlemleri: Dosyaları açmak, okumak, yazmak ve değiştirmek için Python fonksiyonlarını kullanın.
3. Veri Yapıları: Listeler, sözlükler, kümeler ve eşleştiriler gibi farklı veri yapılarını öğrenin ve kullanın.
4. Nesne Yönelimli Programlama: Python'daki nesne yönelimli programlama kavramlarını (örneğin, sınıflar, nesneler, kalıtım) öğrenin ve kullanın.
5. İşletim Sistemi Erişimi: Python ile işletim sistemi komutlarını nasıl çalıştırabileceğinizi öğrenin.
6. Veri Analitik: Verileri okumak, analiz etmek ve görselleştirmek için Python paketleri kullanın (örneğin, NumPy, Pandas, Matplotlib vb.).
7. Veri Madenciliği: Verileri toplamak, temizlemek, analiz etmek ve öğrenme algoritmaları (örneğin, regresyon, sınıflandırma) kullanarak model oluşturmak için Python kütüphanelerini kullanın.
8. Web Uygulamaları: Python ile web uygulamaları geliştirmek için farklı kütüphaneleri (örneğin, Flask, Django) kullanın.
9. Veri Güvenliği: Verilerin güvenliğini ve gizliliğini koruyan fonksiyonları (örneğin, şifreleme, doğrulama) öğrenin.

Bu başlıklar size Python'u daha da derinlemesine öğrenmenize ve uygulamalar geliştirmenize yardımcı olacaktır.