باب- 10

ٹپل اور ڈکشنری
(Tuples and Dictionaries)

"کمپیوٹر کا کمپیوٹنگ سے وہی تعلق ہے جو آلات کا موسیقی سے ہے۔ سافٹ ویئر موسیقی کی اس تحریر کے مانند ہے جس کی تشریحات ہماری دسترس کو بڑھاتی ہے اور ہمارے جذبات کو رفعت بخشتی ہے۔
لیونارڈو دو ونسی کے مطابق موسیقی غیر مرئیات کی تصویر کشی ہے اور اس کا یہ مقولہ سافٹ ویئر کی تشریح کے متعلق نہایت موزوں ہے۔

—اے کے
(A. Kay)

اس باب میں

- ٹپل کا تعارف
- ٹپل آپریشن
- ٹپل میتھڈ اور بلٹ ان فنکشن
- ٹپل اسائنمنٹ
- نیسٹڈ ٹپل
- ٹپل کو استعمال کرنا
- ڈکشنری کا تعارف
- ڈکشنری قابل تغیر ہے
- ڈکشنری آپریشن
- ڈکشنری کا اعادہ
- ڈکشنری میتھڈ اور بلٹ ان فنکشن
- ڈکشنری میں ردوبدل کرنا

10.1 ٹپل کا تعارف (Introduction to Tuples)

ٹپل (Tuple) صحیح عدد (Integer)، فلوٹ، اسٹرنگ، لسٹ، یا ٹپل جیسے مختلف قسم کے اعداد و شمار کے عناصر کا ایک مرتب سلسلہ ہے۔ ٹپل کے عناصر کو قوسین (گول بریکٹ) کے اندر لکھا جاتا ہے اور یہ عناصر کوما کے ذریعے ایک دوسرے سے علاحدہ رہتے ہیں۔ لسٹ اور اسٹرنگ کی طرح، ٹپل کے عناصر کو 0 سے شروع ہونے والی انڈیکس ویلیو کا استعمال کرکے ایکسس کیا جا سکتا ہے۔

مثال 10.1

```
#tuple1 is the tuple of integers
>>> tuple1 = (1,2,3,4,5)
>>> tuple1
(1, 2, 3, 4, 5)

#tuple2 is the tuple of mixed data types
>>> tuple2 =('Economics',87,'Accountancy',89.6)
>>> tuple2
('Economics', 87, 'Accountancy', 89.6)

#tuple3 is the tuple with list as an element
>>> tuple3 = (10,20,30,[40,50])
>>> tuple3
(10, 20, 30, [40, 50])

#tuple4 is the tuple with tuple as an element
>>> tuple4 = (1,2,3,4,5,(10,20))
>>> tuple4
(1, 2, 3, 4, 5, (10, 20))
```

اگر ٹپل میں صرف ایک عنصر ہے تو عنصر کے بعد کوما لگانا چاہیے۔ اگر ہم کوما کے بغیر قدر (Value) تفویض کرتے ہیں تو اسے صحیح عدد تصور کیا جاتا ہے۔ یہ بات قابل غور ہے کہ بغیر قوسین والے سلسلے کو ٹپل تصور کر لیا جاتا ہے۔

```
#incorrect way of assigning single element to
#tuple
#tuple5 is assigned a single element
>>> tuple5 = (20)
```

© NCERT not to be republished

```
>>> tuple5
20
>>>type(tuple5)  #tuple5 is not of type tuple
<class 'int'>    #it is treated as integer

#Correct Way of assigning single element to
#tuple
#tuple5 is assigned a single element
>>> tuple5 = (20,) #element followed by comma
>>> tuple5
(20,)
>>>type(tuple5)    #tuple5 is of type tuple
<class 'tuple'>

#a sequence without parentheses is treated as
#tuple by default
>>> seq = 1,2,3      #comma separated elements
>>> type(seq)        #treated as tuple
<class 'tuple'>
>>> print(seq)       #seq is a tuple
(1, 2, 3)
```

ہم عام طور سے ایک جیسے ڈیٹا ٹائپ کے عناصر کو اسٹور کرنے کے لسٹ کا استعمال کرتے ہیں جب کہ مختلف ڈیٹا ٹائپ کے عناصر کو اسٹور کرنے کے لیے ٹپل کا استعمال کیا جاتا ہے۔

10.1.1 ٹپل میں عناصر تک دسترس حاصل کرنا

ٹپل میں عناصر تک دسترس بالکل اسی طرح حاصل کی جاتی ہے جس طرح لسٹ یا اسٹرنگ کے معاملے میں انڈیکسنگ اور سلائسنگ کا استعمال کرکے کی جاتی ہے۔

```
>>> tuple1 = (2,4,6,8,10,12)
#initializes a tuple tuple1
#returns the first element of tuple1
>>> tuple1[0]
2
#returns fourth element of tuple1
>>> tuple1[3]
8
#returns error as index is out of range
>>> tuple1[15]
IndexError: tuple index out of range
#an expression resulting in an integer index
>>> tuple1[1+4]
12
#returns first element from right
>>> tuple1[-1]
12
```

10.1.2 ٹپل ناقابل تغیر (Immutable) ہوتا ہے

ٹپل ایک غیر متغیر ڈیٹا ٹائپ ہے۔ اس کا مطلب ہے کہ ٹپل کی تشکیل کے بعد اس کے عناصر کو تبدیل نہیں کیا جاسکتا ہے۔ ایسا کرنے پر غلطی ظاہر ہوسکتی ہے۔

© NCERT not to be republished

√ لسٹ قابل تغیر ہیں لیکن ٹپل نا قابل تغیر ہیں چنانچہ لسٹ کے مقابلے ٹپل کا اعادہ تیز رفتار سے کیا جاسکتا ہے۔

√ اگر ہمارے پاس ایسا ڈیٹا ہے جسے تبدیل نہیں کیا جانا ہے تو اس ڈیٹا کو ٹپل میں اسٹور کرنے سے یہ بات یقینی ہے کہ یہ غیر ارادی طور پر تبدیل نہیں ہوگا۔

```
>>> tuple1[4] = 10
TypeError: 'tuple' object does not support
item assignment
```

تاہم ٹپل کا عنصر قابل تغیر (Mutable) قسم کا ہوسکتا ہے، مثلاً، ایک لسٹ

```
#4th element of the tuple2 is a list
>>> tuple2 = (1,2,3,[8,9])
#modify the list element of the tuple tuple2
>>> tuple2[3][1] = 10
#modification is reflected in tuple2
>>> tuple2
(1, 2, 3, [8, 10])
```

10.2 ٹپل آپریشنز (Tuple Operations)

10.2.1 کنکیٹینیشن (Concatenation)

پائتھن میں ہم + کے نشان سے ظاہر کیے جانے والے کنکیٹینیشن (Concatenation) آپریٹر کی مدد سے ٹپل کو یکجا کر سکتے ہیں۔ ہم ایک ایسے نئے ٹپل کی بھی تشکیل کر سکتے ہیں جو اس کنکیٹینیشن عمل کے نتیجے پر مشتمل ہوتا ہے۔

```
>>> tuple1 = (1,3,5,7,9)
>>> tuple2 = (2,4,6,8,10)
>>> tuple1 + tuple2
#concatenates two tuples
(1, 3, 5, 7, 9, 2, 4, 6, 8, 10)
>>> tuple3 = ('Red','Green','Blue')
>>> tuple4 = ('Cyan', 'Magenta', 'Yellow'
,'Black')
#tuple5 stores elements of tuple3 and tuple4
>>> tuple5 = tuple3 + tuple4
>>> tuple5
('Red','Green','Blue','Cyan','Magenta',
'Yellow','Black')
```

کنکیٹینیشن آپریٹر کا استعمال موجودہ ٹپل کی توسیع کے لیے بھی کیا جاسکتا ہے۔ جب ہم کنکیٹینیشن آپریٹر کی مدد سے کسی ٹپل کی توسیع کرتے ہیں تو ایک نئے ٹپل کی تشکیل ہوتی ہے۔

```
>>> tuple6 = (1,2,3,4,5)

#single element is appended to tuple6
>>> tuple6 = tuple6 + (6,)
>>> tuple6
(1, 2, 3, 4, 5, 6)

#more than one elements are appended
>>> tuple6 = tuple6 + (7,8,9)
```

© NCERT not to be republished

```
>>> tuple6
(1, 2, 3, 4, 5, 6, 7, 8, 9)
```

10.2.2 تکرار (Repetition)

تکراری عمل کو * کے نشان سے ظاہر کیا جاتا ہے۔اس کا استعمال ٹپل کے عناصر کی تکرار کے لیے کیا جاتا ہے۔ ہم ٹپل کے عناصر کو دوہرا سکتے ہیں۔

```
>>> tuple1 = ('Hello','World')
>>> tuple1 * 3
('Hello', 'World', 'Hello', 'World', 'Hello',
'World')
#tuple with single element
>>> tuple2 = ("Hello",)
>>> tuple2 * 4
('Hello', 'Hello', 'Hello', 'Hello')
```

10.2.3 ممبرشپ (Membership)

''in'' آپریٹر اس بات کی جانچ کرتا ہے کہ آیا کوئی عنصر ٹپل میں موجود ہے یا نہیں۔اگر موجود ہے تو نتیجے کے طور پر True ظاہر کردیتا ہے بصورت دیگر False ظاہر کرتا ہے۔

```
>>> tuple1 = ('Red','Green','Blue')
>>> 'Green' in tuple1
True
```

''not in'' آپریٹر کا نتیجہ اس وقت True کی شکل میں ظاہر ہوتا ہے جب ٹپل میں عنصر غیر موجود ہوتا ہے بصورت دیگر اس کا نتیجہ False کی شکل میں ظاہر ہوتا ہے۔

```
>>> tuple1 = ('Red','Green','Blue')
>>> 'Green' not in tuple1
False
```

10.2.4 سلائسنگ (Slicing)

اسٹرنگ اور لسٹ کی طرح ٹپل پر سلائسنگ کا بھی اطلاق کیا جا سکتا ہے۔

```
#tuple1 is a tuple
>>> tuple1 = (10,20,30,40,50,60,70,80)

#elements from index 2 to index 6
>>> tuple1[2:7]
(30, 40, 50, 60, 70)

#all elements of tuple are printed
>>> tuple1[0:len(tuple1)]
(10, 20, 30, 40, 50, 60, 70, 80)

#slice starts from zero index
>>> tuple1[:5]
(10, 20, 30, 40, 50)
```

© NCERT not to be republished

```
#slice is till end of the tuple
>>> tuple1[2:]
(30, 40, 50, 60, 70, 80)

#step size 2
>>> tuple1[0:len(tuple1):2]
(10, 30, 50, 70)
#negative indexing
>>> tuple1[-6:-4]
(30, 40)

#tuple is traversed in reverse order
>>> tuple1[::-1]
(80, 70, 60, 50, 40, 30, 20, 10)
```

10.3 ٹپل کے لیے میتھڈ اور بلٹ اِن فنکشن (Tuple Methods and Built-in Functions)

پائتھن ایسے کئی فنکشن مہیا کرتا ہے جو ٹپل کے ساتھ کام کرتے ہیں۔ جدول 10.1 میں عام طور سے استعمال ہونے والے ٹپل کے طریقے اور بلٹ اِن فنکشن دیے گئے ہیں۔

جدول 10.1 ٹپل کے لیے میتھڈ اور بلٹ اِن فنکشن

میتھڈ	وضاحت	مثال

len()	دلیل کے طور پر کامیاب ہونے والے ٹپل کے عناصر کی لمبائی یا تعداد کو ظاہر کرتا ہے	`>>> tuple1 = (10,20,30,40,50)`
		`>>> len(tuple1)`
		`5`

tuple()	اگر کوئی آرگیومینٹ پاس نہیں کیا جاتا ہے تو یہ خالی ٹپل کی تشکیل کرتا ہے
	
	اگر کوئی تسلسل آرگیومینٹ کے طور پر پاس ہو جاتا ہے تو یہ ٹپل کی تشکیل کرتا ہے
		`>>> tuple1 = tuple()`
		`>>> tuple1`
		`( )`
		`>>> tuple1 = tuple('aeiou')#string`
		`>>> tuple1`
		`('a', 'e', 'i', 'o', 'u')`
		`>>> tuple2 = tuple([1,2,3]) #list`
		`>>> tuple2`
		`(1, 2, 3)`
		`>>> tuple3 = tuple(range(5))`
		`>>> tuple3`
		`(0, 1, 2, 3, 4)`

count()	کسی ٹپل میں کوئی عنصر جتنی مرتبہ ظاہر ہوتا ہے اس تعداد کو بتاتا ہے	`>>> tuple1 = (10,20,30,10,40,10,50)`
		`>>> tuple1.count(10)`
		`3`
		`>>> tuple1.count(90)`
		`0`

© NCERT not to be republished

index()	دیے ہوئے ٹپل میں عنصر کے پہلے وقوع کے انڈیکس کو ظاہر کرتا ہے	`>>> tuple1 = (10,20,30,40,50)`
		`>>> tuple1.index(30)`
		`2`
		`>>> tuple1.index(90)`
		`ValueError: tuple.index(x): x not in tuple`

sorted()	ٹپل کے عناصر کو لے کر ایک نئی مرتب فہرست ظاہر کرتا ہے۔ یہ بات قابل غور ہے کہ sorted() اصل ٹپل میں کسی قسم کی تبدیلی کا موجب نہیں ہے	`>>> tuple1 = ("Rama","Heena","Raj", "Mohsin","Aditya")`
		`>>> sorted(tuple1)`
		`['Aditya', 'Heena', 'Mohsin', 'Raj', 'Rama']`

min()
max()
sum()
	ٹپل کے کمترین یا مختصر ترین عنصر کو ظاہر کرتا ہے
	ٹپل کے بیش ترین یا سب سے بڑے عنصر کو ظاہر کرتا ہے
	ٹپل کے عناصر کے حاصل جمع کو ظاہر کرتا ہے
		`>>> tuple1 = (19,12,56,18,9,87,34)`
		`>>> min(tuple1)`
		`9`
		`>>> max(tuple1)`
		`87`
		`>>> sum(tuple1)`
		`235`

10.4 ٹپل اسائنمینٹ (TUPLE ASSIGNMENT)

ٹپل کا اسائنمینٹ پائتھن میں ایک مفید اور کارگر خصوصیت ہے۔ یہ اسائنمینٹ آپریٹر کے بائیں جانب ٹپل متغیرات کو دائیں جانب والے ٹپل کی متعلقہ قدریں تفویض کرتا ہے۔ بائیں جانب متغیرات کی تعداد ٹپل میں موجود عناصر کی تعداد کے مساوی ہونی چاہیے۔

مثال 10.2

```
#The first element 10 is assigned to num1 and
#the second element 20 is assigned to num2.
>>> (num1,num2) = (10,20)
>>> print(num1)
10
>>> print(num2)
20
>>> record = ( "Pooja",40,"CS")
>>> (name,rollNo,subject) = record
>>> name
'Pooja'
>>> rollNo
40
>>> subject
'CS'
>>> (a,b,c,d) = (5,6,8)
```

© NCERT not to be republished

```
ValueError: not enough values to unpack
(expected 4, got 3)
```

اگر دائیں جانب ایک عبارت ہے تو پہلے اس عبارت کو حل کیا جاتا ہے اور نتیجہ ٹپل کو تفویض کر دیا جاتا ہے۔

مثال 10.3

```
#15 is assigned to num3 and
#25 is assigned to num4
>>> (num3,num4) = (10+5,20+5)
>>> print(num3)
15
>>> print(num4)
25
```

\ یہ ایسکیپ (escape) کیریکٹر ہے جس کا استعمال افقی ٹیب اسپیس شامل کرنے کے لیے کیا جاتا ہے۔ n\ ایک اور ایسا ایسکیپ کیریکٹر ہے جسے عام طور سے نئی سطر کو داخل کرنے کے لیے استعمال کیا جاتا ہے۔

10.5 نیسٹیڈ ٹپل (Nested Tuples)

کسی ٹپل کے اندر موجود دوسرے ٹپل کو نیسٹیڈ ٹپل کہا جاتا ہے۔ پروگرام 1-10 میں ایک طالب علم کے رول نمبر، نام اور مارکس (فیصد میں) کو ٹپل میں محفوظ کیا گیا ہے۔ اس قسم کے بہت سے طلبا کی تفصیلات کو جمع کرنے کے لیے ہم نیسٹیڈ ٹپل کی تشکیل کر سکتے ہیں۔

پروگرام 1-10 یہ ایک ایسا پروگرام ہے جس کے تحت طلبا کے رول نمبر، نام اور مارکس کو اسٹور کرنے کے لیے نیسٹیڈ ٹپل کی تشکیل کی گئی ہے۔

```
#Program 10-1
#To store records of students in tuple and print them
st=((101,"Aman",98),(102,"Geet",95),(103,"Sahil",87),(104,"Pawan",79))
print("S_No"," Roll_No","    Name"," Marks")
for i in range(0,len(st)):
    print((i+1),'\t',st[i][0],'\t',st[i][1],'\t',st[i][2])
```

نتیجہ:

```
S_No   Roll_No   Name       Marks
1      101       Aman        98
2      102       Geet        95
3      103       Sahil       87
4      104       Pawan       79
```

10.6 ٹپل کو استعمال کرنا (Tuple Handling)

پروگرام 2-10 عارضی متغیر کا استعمال کیے بغیر دو اعداد کو ایک دوسرے سے بدلنے کے لیے پروگرام لکھیے۔

not to be republished © NCERT

```
#Program 10-2
#Program to swap two numbers
num1 = int(input('Enter the first number: '))
num2 = int(input('Enter the second number: '))
print("\nNumbers before swapping:")
print("First Number:",num1)
print("Second Number:",num2)
(num1,num2) = (num2,num1)
print("\nNumbers after swapping:")
print("First Number:",num1)
print("Second Number:",num2)
```

نتیجہ:

```
Enter the first number: 5
Enter the second number: 10

Numbers before swapping:
First Number: 5
Second Number: 10

Numbers after swapping:
First Number: 10
Second Number: 5
```

پروگرام 10-3 فنکشن کا استعمال کر کے دائرہ کا رقبہ اور محیط معلوم کرنے کے لیے ایک پروگرام لکھیے۔

```
#Program 10-3
#Function to compute area and circumference of the circle.
def circle(r):
     area = 3.14*r*r
     circumference = 2*3.14*r
     #returns a tuple having two elements area and circumference
     return (area,circumference)
#end of function

radius = int(input('Enter radius of circle: '))
area,circumference = circle(radius)
print('Area of circle is:',area)
print('Circumference of circle is:',circumference)
```

```
Enter radius of circle: 5
Area of circle is: 78.5
Circumference of circle is: 31.400000000000002
```

© NCERT not to be republished

پروگرام 10-4 ایک پروگرام لکھیے جس میں استعمال کنندہ سے n اعداد کو داخل کرنے کے لیے کہا جائے۔ ان اعداد کو ٹپل میں اسٹور کیجیے۔ اس ٹپل سے سب سے بڑے اور سب سے چھوٹے عدد کو پرنٹ کیجیے۔

```
#Program 10-4
#Program to input n numbers from the user. Store these numbers
#in a tuple. Print the maximum and minimum number from this tuple.

numbers = tuple()                #create an empty tuple 'numbers'
n = int(input("How many numbers you want to enter?: "))
for i in range(0,n):
    num = int(input())
    #it will assign numbers entered by user to tuple 'numbers'
  numbers = numbers +(num,)
print('\nThe numbers in the tuple are:')
print(numbers)
print("\nThe maximum number is:")
print(max(numbers))
print("The minimum number is:")
print(min(numbers))
```

نتیجہ:

```
How many numbers do you want to enter?: 5
9
8
10
12
15

The numbers in the tuple are:
(9, 8, 10, 12, 15)

The maximum number is:
15
The minimum number is:
8
```

10.7 ڈکشنری کا تعارف (Introduction to Dictionaries)

ڈکشنری ایک ایسا ڈیٹا ٹائپ ہے جو میپنگ کے تحت آتا ہے۔ یہ کیز (Keys) کے سیٹ اور قدروں (Values) کے سیٹ کے درمیان میپنگ ہے۔ کی (Key) اور قدر (Value) کی جوڑی کو آئٹم (Item) کہتے ہیں۔ کی (Key) اور اس کی قدر کو ایک دوسرے سے کولن (:) کے ذریعے علاحدہ کیا جاتا ہے اور ایک آئٹم دوسرے آئٹم سے کوما کے ذریعے علاحدہ رہتا ہے۔ ڈکشنری میں آئٹم بے ترتیب ہوتے ہیں، لہٰذا یہ ممکن ہے کہ

© NCERT not to be republished

نوٹ

ہمیں ڈیٹا واپس اس ترتیب میں حاصل نہ ہو سکے جس ترتیب میں ہم نے اسے شروع میں ڈکشنری میں داخل کیا تھا۔

10.7.1 ڈکشنری تشکیل دینا (Creating a Dictionary)

ڈکشنری کی تشکیل کے لیے، داخل کیے جانے والے آئٹموں کو ایک دوسرے سے کوما کے ذریعے علاحدہ کیا جاتا ہے اور انھیں منجھلے بریکٹ (Curly braces) کے اندر لکھا جاتا ہے۔ ہر ایک آئٹم ایک کی (Key) ویلیو جوڑی ہے جو ایک دوسرے سے کولن (:) کے ذریعے علاحدہ رہتے ہیں۔ ڈکشنری میں کیز کو منفرد اور ناقابل تغیر ڈیٹا ٹائپ یعنی عدد، اسٹرنگ یا ٹپل ہونا چاہیے۔ قدروں (Values) کو دہرایا جاسکتا ہے اور یہ کوئی بھی ڈیٹا ٹائپ ہو سکتی ہیں۔

مثال 10.4

```
#dict1 is an empty Dictionary created
#curly braces are used for dictionary
>>> dict1 = {}
>>> dict1
{}
#dict2 is an empty dictionary created using
#built-in function

>>> dict2 = dict()
>>> dict2
{}
#dict3 is the dictionary that maps names
#of the students to respective marks in
#percentage
>>> dict3 = {'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
>>> dict3
{'Mohan': 95, 'Ram': 89, 'Suhel': 92,
'Sangeeta': 85}
```

10.7.2 ڈکشنری میں آئٹم کو ایکسس کرنا

ہم پہلے ہی یہ دیکھ چکے ہیں کہ تسلسل (اسٹرنگ، لسٹ اور ٹپل) کے آئٹموں کو انڈیکسنگ کی تکنیک کا استعمال کرکے ایکسس کیا جاتا ہے۔ ڈکشنری کے آئٹموں کو کیز کے نسبتی مقامات یا انڈیکس کے ذریعے ایکسس کرنے کے بجائے کیز کے ذریعے ایکسس کیا جاتا ہے۔ ہر ایک کی (Key) انڈیکس کے طور پر کام کرتی ہے اور ایک قدر کے ساتھ مربوط ہوتی ہے۔

مندرجہ ذیل مثال میں یہ دکھایا گیا ہے کہ ڈکشنری کس طرح دی ہوئی کی (Key) کی نظیری قدر کو ظاہر کرتی ہے:

© NCERT not to be republished

نوٹ

```
>>> dict3 = {'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
>>> dict3['Ram']
89
>>> dict3['Sangeeta']
85
#the key does not exist
>>> dict3['Shyam']
KeyError: 'Shyam'
```

مندرجہ بالا مثال میں کی 'Ram' ہمیشہ ہی قدر 89 کے ساتھ مربوط ہے اور کی 'Sangeeta' ہمیشہ ہی قدر 85 کے ساتھ مربوط ہے۔ لہٰذا آئٹموں کی ترتیب کی کوئی اہمیت نہیں ہے۔ اگر ڈکشنری میں کی موجود نہیں ہے تو ہمیں KeyError حاصل ہوگی۔

10.8 ڈکشنری قابل تغیر ہے (Dictionaries are Mutable)

ڈکشنری قابل تغیر ہوتی ہے جس کا مطلب یہ ہے کہ ڈکشنری کی تشکیل کے بعد اس کے مواد کو تبدیل کیا جاسکتا ہے۔

10.8.1 نئے آئٹم کو شامل کرنا

ہم نئے آئٹم کو ڈکشنری میں شامل کرسکتے ہیں جیسا کہ مندرجہ ذیل مثال میں دکھایا گیا ہے۔

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
>>> dict1['Meena'] = 78
>>> dict1
{'Mohan': 95, 'Ram': 89, 'Suhel': 92,
'Sangeeta': 85, 'Meena': 78}
```

10.8.2 موجودہ آئٹم میں ترمیم کرنا

صرف کی-ویلیو جوڑی کو از سر نو تحریر کرکے موجودہ ڈکشنری میں ترمیم کی جاسکتی ہے۔ ڈکشنری کے اندر کسی دیے ہوئے آئٹم میں ترمیم کے طریقے کو مندرجہ ذیل مثال میں دکھایا گیا ہے۔

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
#Marks of Suhel changed to 93.5
>>> dict1['Suhel'] = 93.5
>>> dict1
{'Mohan': 95, 'Ram': 89, 'Suhel': 93.5,
'Sangeeta': 85}
```

10.9 ڈکشنری آپریشن (Dictionary Operations)

10.9.1 ممبرشپ (Membership)

ممبرشپ آپریٹر in اس بات کی جانچ کرتا ہے کہ آیا ڈکشنری میں کی (key) موجود ہے یا نہیں۔ اگر موجود ہے تو یہ True ظاہر کرے گا اور عدم موجودگی کی صورت میں False ظاہر کرے گا۔

© NCERT not to be republished

نوٹ

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
>>> 'Suhel' in dict1
True
```

آپریٹر not in اس وقت True ظاہر کرتا ہے جب کی (key) ڈکشنری میں موجود نہیں ہوتی ہے بصورت دیگر یہ False ظاہر کرے گا۔

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
>>> 'Suhel' not in dict1
False
```

10.10 ڈکشنری کا اعادہ (Traversing a Dictionary)

ہم For لوپ کا استعمال کرکے ڈکشنری کے ہر ایک آئٹم کو ایکسس کرسکتے ہیں یا ڈکشنری کا اعادہ کرسکتے ہیں۔

```
>>> dict1 ={'Mohan':95,'Ram':89,'Suhel':92,
'Sangeeta':85}
```

طریقہ 1

```
>>> for key in dict1:
        print(key,':',dict1[key])
Mohan: 95
Ram: 89
Suhel: 92
Sangeeta: 85
```

طریقہ 2

```
>>> for key,value in dict1.items():
        print(key,':',value)
Mohan: 95
Ram: 89
Suhel: 92
Sangeeta: 85
```

10.11 ڈکشنری میتھڈ اور بلٹ اِن فنکشن (Dictionary Methods and Built-in Functions)

پائتھن ڈکشنری پر کام کرنے کے لیے کئی فنکشن فراہم کرتا ہے۔ جدول 10.2 میں عام طور سے استعمال ہونے والے ڈکشنری میتھڈ کی فہرست دی گئی ہے۔

© NCERT not to be republished

جدول 10.2 ڈکشنری کے لیے میتھڈ اور بلٹ اِن فنکشن

میتھڈ	وضاحت	مثال

len()	دلیل کے طور پر پاس ہونے والی ڈکشنری کی کی:ویلیو جوڑیوں کی تعداد کو ظاہر کرتا ہے۔	`>>> dict1 = {'Mohan':95,'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> len(dict1)`
		`4`

dict()	کی:ویلیو جوڑیوں کے تسلسل سے ڈکشنری کی تشکیل کرتا ہے۔	`pair1 = [('Mohan',95),('Ram',89), ('Suhel',92),('Sangeeta',85)]`
		`>>> pair1`
		`[('Mohan', 95), ('Ram', 89), ('Suhel', 92), ('Sangeeta', 85)]`
		`>>> dict1 = dict(pair1)`
		`>>> dict1`
		`{'Mohan': 95, 'Ram': 89, 'Suhel': 92, 'Sangeeta': 85}`

keys()	ڈکشنری میں موجود کیز کی فہرست کو ظاہر کرتا ہے۔	`>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> dict1.keys()`
		`dict_keys(['Mohan', 'Ram', 'Suhel', 'Sangeeta'])`

values()	ڈکشنری میں موجود ویلیو کی فہرست کو ظاہر کرتا ہے۔	`>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> dict1.values()`
		`dict_values([95, 89, 92, 85])`

items()	ٹپل (کی-ویلیو) کی فہرست کو ظاہر کرتا ہے۔	`>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> dict1.items()`
		`dict_items([( 'Mohan', 95), ('Ram', 89), ('Suhel', 92), ('Sangeeta', 85)])`

get()	آرگیومنٹ کے طور پر پاس ہونے والی کی سے متعلق قدر کو ظاہر کرتا ہے۔
	اگر ڈکشنری میں کی موجود نہیں ہے تو یہ None ظاہر کرتا ہے۔
		`>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> dict1.get('Sangeeta')`
		`85`
		`>>> dict1.get('Sohan')`
		`>>>`

© NCERT not to be republished

update()	آرگیومینٹ کے طور پر پاس ہونے والی کی-ویلیو جوڑی کو دی ہوئی ڈکشنری کی کی-ویلیو جوڑی میں شامل کرتا ہے۔	`>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> dict2 = {'Sohan':79,'Geeta':89}`
		`>>> dict1.update(dict2)`
		`>>> dict1`
		`{'Mohan': 95, 'Ram': 89, 'Suhel': 92, 'Sangeeta': 85, 'Sohan': 79, 'Geeta': 89}`
		`>>> dict2`
		`{'Sohan': 79, 'Geeta': 89}`

del()	دی ہوئی کی کے آئٹم کو حذف کرتا ہے۔
	میموری سے ڈکشنری کو حذف کرنے کے لیے ہم مندرجہ ذیل سنٹیکس کا استعمال کرتے ہیں۔
	del Dict_name
		`>>> dict1 = {'Mohan':95,'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> del dict1['Ram']`
		`>>> dict1`
		`{'Mohan':95,'Suhel':92,'Sangeeta': 85}`
		
		`>>> del dict1 ['Mohan']`
		`>>> dict1`
		`{'Suhel': 92, 'Sangeeta': 85}`
		`>>> del dict1`
		`>>> dict1`
		`NameError: name 'dict1' is not defined`

clear()	ڈکشنری کے سبھی آئٹموں کو حذف یا ختم کر دیتا ہے۔	`>>> dict1 = {'Mohan':95,'Ram':89, 'Suhel':92, 'Sangeeta':85}`
		`>>> dict1.clear()`
		`>>> dict1`
		`{ }`

10.12 ڈکشنری میں ردوبدل کرنا (MANIPULATING DICTIONARIES)

اس باب میں ہم یہ سیکھ چکے ہیں کہ ڈکشنری کی تشکیل کس طرح کی جاتی ہے اور اس میں ردوبدل کرنے کے لیے مختلف طریقوں (Methods) کا اطلاق کس طرح کیا جاتا ہے۔ مندرجہ ذیل پروگرام اس بات کو ظاہر کرتا ہے کہ ڈکشنری میں ردوبدل کرنے کے لیے مختلف میتھڈ کا اطلاق کس طرح کیا جاتا ہے۔

پروگرام 5–10 عدد 1 اور 10 کے درمیان موجود طاق اعداد کی ڈکشنری 'ODD' تشکیل دیجیے جس میں کی (key) اعشاریہ عدد ہے اور ویلیو نظیری عدد کی لفظی شکل ہے۔ اس ڈکشنری پر مندرجہ ذیل عمل انجام دیجیے۔

(a) کیز کو ظاہر کرنا
(b) ویلیو کو ظاہر کرنا
(c) آئٹم کو ظاہر کرنا

© NCERT not to be republished

(d) ڈکشنری کے آئٹموں کی تعداد کو معلوم کرنا
(e) اس بات کی جانچ کرنا کہ آیا 7 موجود ہے یا نہیں
(f) اس بات کی جانچ کرنا کہ آیا 2 موجود ہے یا نہیں
(g) 9 کی سے متعلق قدر (ویلیو) کو معلوم کرنا
(h) 9 کی سے متعلق آئٹم کو ڈکشنری سے حذف کرنا

```
>>> ODD = {1:'One',3:'Three',5:'Five',7:'Seven',9:'Nine'}
>>> ODD
{1: 'One', 3: 'Three', 5: 'Five', 7: 'Seven', 9: 'Nine'}
```

(a) کیز کو ظاہر کرنا

```
>>> ODD.keys()
dict_keys([1, 3, 5, 7, 9])
```

(b) ویلیو کو ظاہر کرنا

```
>>> ODD.values()
dict_values(['One', 'Three', 'Five', 'Seven', 'Nine'])
```

(c) آئٹم کو ظاہر کرنا

```
>>> ODD.items()
dict_items([(1, 'One'), (3, 'Three'), (5, 'Five'), (7, 'Seven'),
(9, 'Nine')])
```

(d) ڈکشنری کے آئٹموں کی تعداد کو معلوم کرنا

```
>>> len(ODD)
5
```

(e) اس بات کی جانچ کرنا کہ آیا 7 موجود ہے یا نہیں

```
>>> 7 in ODD
True
```

(f) اس بات کی جانچ کرنا کہ آیا 2 موجود ہے یا نہیں

```
>>> 2 in ODD
False
```

(g) 9 کی سے متعلق قدر (ویلیو) کو معلوم کرنا

```
>>> ODD.get(9)
'Nine'
```

(h) 9 کی سے متعلق آئٹم کو ڈکشنری سے حذف کرنا

```
>>> del ODD[9]
>>> ODD
{1: 'One', 3: 'Three', 5: 'Five', 7: 'Seven'}
```

پروگرام 6-10 ایک پروگرام لکھیے جس میں ملازمین کے نام اور ان کی تنخواہوں کو ان پُٹ کے طور پر داخل کیجیے اور انھیں ڈکشنری میں اسٹور کیجیے۔

© NCERT not to be republished

```
#Program 10-6
#Program to create a dictionary which stores names of the employee
#and their salary
num = int(input("Enter the number of employees whose data to be
stored: "))
count = 1
employee = dict()                  #create an empty dictionary
while count <= num:
     name = input("Enter the name of the Employee: ")
     salary = int(input("Enter the salary: "))
     employee[name] = salary
     count += 1
print("\n\nEMPLOYEE_NAME\tSALARY")
for k in employee:
     print(k,'\t\t',employee[k])
```

نتیجہ:

```
Enter the number of employees to be stored: 5
Enter the name of the Employee: 'Tarun'
Enter the salary: 12000
Enter the name of the Employee: 'Amina'
Enter the salary: 34000
Enter the name of the Employee: 'Joseph'
Enter the salary: 24000
Enter the name of the Employee: 'Rahul'
Enter the salary: 30000
Enter the name of the Employee: 'Zoya'
Enter the salary: 25000
EMPLOYEE_NAME    SALARY
'Tarun'          12000
'Amina'          34000
'Joseph'         24000
'Rahul'          30000
'Zoya'           25000
```

پروگرام 7–10 دی ہوئی اسٹرنگ میں کسی کیریکٹر (حرفی علامت) کے ظاہر ہونے کی تعداد کو شمار کرنے کے لیے ایک پروگرام لکھیے۔

```
#Program 10-7
#Count the number of times a character appears in a given string
st = input("Enter a string: ")
dic = {}               #creates an empty dictionary
for ch in st:
     if ch in dic:     #if next character is already in the dictionary
          dic[ch] += 1
     else:
          dic[ch] = 1  #if ch appears for the first time

for key in dic:
     print(key,':',dic[key])
```

© NCERT not to be republished

نتیجہ:

```
Enter a string: HelloWorld
H : 1
e : 1
l : 3
o : 2
W : 1
r : 1
d : 1
```

پروگرام 10-8 استعمال کنندہ (User) کے ذریعے داخل کیے گئے عدد کو لفظی شکل میں تبدیل کرنے کے لیے ایک فنکشن لکھیے۔ مثال کے طور پر اگر داخل کیا گیا عدد 876 ہے تو نتیجہ 'Eight Seven Six' ہوگا۔

```
# Program 10-8
# Write a function to convert number into corresponding number in
# words
def convert(num):
     #numberNames is a dictionary of digits and corresponding number
     #names
     numberNames = {0:'Zero',1:'One',2:'Two',3:'Three',4:'Four',\
                    5:'Five',6:'Six',7:'Seven',8:'Eight',9:'Nine'}

     result = ''
     for ch in num:
          key = int(ch)              #converts character to integer
          value = numberNames[key]
          result = result + ' ' + value
     return result

num = input("Enter any number: ")     #number is stored as string
result = convert(num)
print("The number is:",num)
print("The numberName is:",result)
```

نتیجہ:

```
Enter any number: 6512
The number is: 6512
The numberName is: Six Five One Two
```

خلاصہ

- ٹپل ناقابل تغیر تسلسل ہیں یعنی ہم ٹپل کی تشکیل کے بعد اس کے عناصر کو تبدیل نہیں کر سکتے ہیں۔
- ٹپل کے عناصر کو چھوٹے بریکٹ (Round Bracket) میں لکھا جاتا ہے اور یہ عناصر ایک دوسرے سے کوما کے ذریعے علاحدہ رہتے ہیں۔
- اگر کسی تسلسل کے عناصر ایک دوسرے سے کوما کے ذریعے علاحدہ ہیں اور انھیں بغیر قوسین کے

© NCERT not to be republished

نوٹ

لکھا گیا ہے تو اسے بھی ٹپل تصور کیا جائے گا۔

- ٹپل ایک مرتب تسلسل ہے کیوں کہ ہر ایک عنصر کا مقام متعین ہے۔
- ٹپل کے عناصر کو ایکسس کرنے کے لیے انڈیکسنگ کا استعمال کیا جاتا ہے۔ ڈکشنری میں انڈیکسنگ کے دو طریقے ہیں جیسا کہ اسٹرنگ اور لسٹ میں کیا جاتا ہے۔
- آپریٹر '+' ایک تسلسل (اسٹرنگ، لسٹ، ٹپل) کو دوسرے تسلسل کے ساتھ جوڑتا ہے۔
- آپریٹر '*' تسلسل (اسٹرنگ، لسٹ، ٹپل) کو دہراتا ہے۔
- ممبر شپ آپریٹر 'in' یہ بتاتا ہے کہ آیا کوئی عنصر تسلسل میں موجود ہے یا نہیں اور 'not in' آپریٹر اس کے برعکس کام کرتا ہے۔
- ٹپل میں ردو بدل کے فنکشن مندرجہ ذیل ہیں: len(), tuple(), count(), index(), sorted(), min(), max(), sum()
- ڈکشنری ایک میپنگ (غیر سمتی) ڈیٹا ٹائپ ہے۔ یہ کی-ویلیو جوڑی کا ایک غیر مرتب مجموعہ ہے۔ کی-ویلیو جوڑیوں کو منجھلے بریکٹ میں رکھا جاتا ہے۔
- ہر ایک کی (key) اور اس کی قدر کے درمیان کولن لگایا جاتا ہے۔
- کیز (Keys) منفرد نوعیت کی ہوتی ہیں اور انڈیکس کے طور پر کام کرتی ہیں۔
- کیز (Keys) ناقابل تغیر قسم کی ہوتی ہیں لیکن ویلیو قابل تغیر قسم کی ہوتی ہے۔

مشق

1۔ مندرجہ ذیل ٹپل، ٹپل 1 اور ٹپل 2 پر غور کیجیے۔

```
tuple1 = (23,1,45,67,45,9,55,45)
tuple2 = (100,200)
```

مندرجہ ذیل بیانات کا نتیجہ (آؤٹ پُٹ) معلوم کیجیے :

i. `print(tuple1.index(45))`
ii. `print(tuple1.count(45))`
iii. `print(tuple1 + tuple2)`
iv. `print(len(tuple2))`
v. `print(max(tuple1))`
vi `print(min(tuple1))`
vii. `print(sum(tuple2))`
viii. `print(sorted(tuple1))`
`print(tuple1)`

2۔ مندرجہ ذیل ڈکشنری StateCapital پر غور کیجیے۔

```
stateCapital = {"AndhraPradesh":"Hyderabad",
"Bihar":"Patna","Maharashtra":"Mumbai",
"Rajasthan":"Jaipur"}
```

مندرجہ ذیل بیانات کا نتیجہ (آؤٹ پُٹ) معلوم کیجیے :

© NCERT not to be republished

نوٹ

```
i.    print(stateCapital.get("Bihar"))
ii.   print(stateCapital.keys())
iii.  print(stateCapital.values())
iv.   print(stateCapital.items())
v.    print(len(stateCapital))
vi    print("Maharashtra" in stateCapital)
vii.  print(stateCapital.get("Assam"))
viii. del stateCapital["Rajasthan"]
      print(stateCapital)
```

3۔ ''لسٹ اور ٹپل مرتب قسم کے ہوتے ہیں'، وضاحت کیجیے۔

4۔ ایک مثال کی مدد سے یہ دکھائیے کہ آپ ایک فنکشن کی مدد سے ایک سے زیادہ ویلیوکس طرح ظاہر کرسکتے ہیں؟

5۔ لسٹ کے مقابلے ٹپل کا کیا فائدہ ہے؟

6۔ پائتھن میں ٹپل اور ڈکشنری کا استعمال کب کرتے ہیں؟ کچھ پروگراموں کی مثالیں پیش کرتے ہوئے ان کی افادیت کا ذکر کیجیے۔

7۔ ایک مثال کی مدد سے ثابت کیجیے کہ نا قابل تغیر ڈیٹا ٹائپ کے معاملے میں متغیر کو تبدیل کیا جاسکتا ہے۔

8۔ ذیل میں بیان 2 پر عمل درآمد کے دوران TypeError ظاہر ہوتی ہے۔ اس کی وجہ بتائیے۔ اسے کس طرح درست کیا جاسکتا ہے؟

پروگرامنگ سوالات (Programming Problems)

1۔ ایک پروگرام لکھیے جس میں n تعداد والے طلبا کی ای میل آئی ڈی کو پڑھ کر انھیں ایک ٹپل میں اسٹور کیجیے۔ دو نئے ٹپل تشکیل دیجیے۔ ان میں سے ایک ٹپل ای میل آئی ڈی سے صرف استعمال کنندہ کا نام (Username) اور دوسرا ای میل آئی ڈی سے ڈومین کا نام اسٹور کرنے کے لیے ہو۔ پروگرام کے آخر میں سبھی تینوں ٹپل کو پرنٹ کیجیے۔ [اشارہ: آپ split() فنکشن کا استعمال کر سکتے ہیں]

2۔ ایک پروگرام لکھیے جس میں n تعداد والے طلبا کے نام ان پُٹ کیجیے اور انھیں ایک ٹپل میں اسٹور کیجیے۔ استعمال کنندہ سے بھی ایک نام داخل کرنے کے لیے کہا جائے اور یہ معلوم کیجیے کہ آیا یہ نام ٹپل میں موجود ہے یا نہیں۔ ہم اس کام کو مندرجہ ذیل طریقے سے انجام دے سکتے ہیں:

(a) استعمال کنندہ کے ذریعے متعین کیے گئے فنکشن کا استعمال

(b) بلٹ اِن فنکشن کا استعمال

3۔ ایک ڈکشنری میں سب سے بڑی دو قدروں کو معلوم کرنے کے لیے ایک پائتھن پروگرام لکھیے۔

4۔ ایک اسٹرنگ سے ڈکشنری تشکیل دینے کے لیے پائتھن پروگرام لکھیے۔

نوٹ: اسٹرنگ سے حروف کی تعداد شمار کیجیے۔

سیمپل اسٹرنگ: 'w3resource'

متوقع نتیجہ: {'3': 1, 's': 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1,'e': 2, 'o': 1}

© NCERT not to be republished

نوٹ

5۔ ایک پروگرام لکھیے جس میں اپنے دوستوں کے نام اور ان کے فون نمبر کو داخل کیجیے اور انھیں کی(key) ویلیو جوڑی کے طور پر ڈکشنری میں اسٹور کیجیے۔ ڈکشنری پر مندرجہ ذیل عمل انجام دیجیے:

(a) اپنے سبھی دوستوں کے نام اور فون نمبر ڈسپلے کیجیے۔
(b) اس ڈکشنری میں ایک نئی کی ویلیو جوڑی کا اضافہ کیجیے اور ترمیم شدہ ڈکشنری کو ڈسپلے کیجیے۔
(c) کسی مخصوص دوست کو ڈکشنری سے حذف کیجیے۔
(d) ایک موجودہ دوست کے فون نمبر میں ترمیم کیجیے۔
(e) اس بات کی جانچ کیجیے کہ آیا کوئی دوست ڈکشنری میں موجود ہے یا نہیں۔
(f) ڈکشنری کو ناموں کی ترتیب کے اعتبار سے ڈسپلے کیجیے۔

نظیری مطالعہ سوال (Case Study-based Question)

باب 5 میں دیے گئے SMIS سسٹم کے لیے آیئے مندرجہ ذیل سرگرمی کو انجام دیں:

ایک پروگرام لکھیے جو دسویں کلاس کے رول نمبر، نام اور فیصد نمبروں کو حاصل کرتا ہے۔ مندرجہ ذیل کے لیے استعمال کنندہ کے ذریعے تعریف شدہ فنکشن لکھیے۔

- n طلبا کی تفصیلات کو قبول کرنا (n طلبا کی تعداد کو ظاہر کرتا ہے)
- رول نمبر کی بنیاد پر کسی مخصوص طالب علم کی تفصیلات کو تلاش کرنا اور نتیجے کو ظاہر کرنا
- سبھی طلبا کے نتیجہ کو ظاہر کرنا
- پہلا مقام حاصل کرنے والے طالب علم کا پتہ لگانا
- مضمون میں پہلا مقام حاصل کرنے والے طالب علم کا پتہ لگانا

(اشارہ: ڈکشنری کا استعمال کیجیے جس میں کی(Key)، رول نمبر اور ویلیو(Value) نام اور فیصد پر مشتمل ناقابل تغیر ڈیٹا ٹائپ ہوسکتا ہے۔)

آیئے باب 5 کے آخر میں ''دستاویز سازی کے لیے مشورہ'' کے تحت دیے گئے پیرامیٹر کی بنیاد پر دوسرے طلبا کے نظیری مطالعہ پر نظر ثانی کریں اور انھیں بازرسی فراہم کریں۔

نظیری مطالعہ سوال (Case Study-based Question)

1۔ بینک ایک مالیاتی ادارہ ہے جو قرض کے لین دین سے وابستہ کاموں کو انجام دیتا ہے۔ ٹیکنالوجی میں ہونے والی ترقی کی وجہ سے آن لائن بینکنگ جسے انٹرنیٹ بینکنگ بھی کہا جاتا ہے بینک کے گاہکوں کو کسی بھی وقت کہیں سے بھی بینک کی ویب سائٹ کے ذریعے مختلف قسم کے مالی لین دین کی سہولت فراہم کرتا ہے۔ ابتدائی تفتیش کے حصے کے طور پر آپ کو مندرجہ ذیل تجاویز پیش کی جاتی ہیں:

- بینک کا درخواست فارم حاصل کریں۔ فارم کا بغور تجزیہ کرنے کے بعد بچت کھاتہ کھولنے کے لیے مطلوب معلومات کی شناخت کریں۔ بچت کھاتے پر ملنے والے سود کی شرح کے بارے میں بھی دریافت کریں۔

© NCERT not to be republished

نوٹ

- کھاتے سے متعلق دو بنیادی عمل ہیں ایک کھاتے میں رقم جمع کرنا دوسرا کھاتے سے رقم کو نکالنا۔ مینو کی مدد سے چلنے والا ایک پروگرام لکھیے جو جمع اور نکاسی کے دو متبادلات میں سے کسی ایک کو قبول کرتا ہے اور پھر رقم کو قبول کرتا ہے، لین دین کے عمل کو انجام دیتا ہے اور حسب حال بقایا رقم کو ظاہر کرتا ہے۔ یاد رہے کہ ہر ایک بینک کھاتے میں کم سے کم بقایا رقم کی موجودگی ضروری ہے جسے رقم نکالنے کے عمل کے دوران ذہن میں رکھنا ہے۔ اپنے بینک میں کم سے کم بقایا رقم کے متعلق دریافت کریں۔
- بینک میں میعادی جمع اور بچت کھاتہ کھولنے کے لیے مختلف سلیب کے تحت سود کی شرحوں کے بارے میں دریافت کیجیے۔ یاد رہے کہ بزرگ شہریوں کے لیے سود کی شرحیں مختلف ہوسکتی ہیں۔ آخر میں، مینو کی مدد سے چلنے والا ایک پروگرام لکھیے جس میں مندرجہ ذیل متبادلات موجود ہوں۔ (فنکشن اور موزوں ڈیٹا ٹائپ کا استعمال کیجیے):
- بچت بینک کھاتہ کھولنا
- رقم جمع کرنا
- رقم نکالنا
- میعادی جمع کے لیے رقم اور مدت جیسی تفصیلات کو حاصل کیجیے اور کسی مخصوص گاہک کے لیے واجب الادا رقم کو ظاہر کیجیے۔

2۔ کوئز میں حصہ لینا ایک دلچسپ تجربہ ہے کیوں کہ یہ ایک مسابقتی عنصر فراہم کرتا ہے۔ کچھ تعلیمی ادارے اس کا استعمال عمومی سطح پر یا مطالعے کے کسی مخصوص شعبے میں اپنے طلبا کی علمی سطح، صلاحیتوں اور/یا مہارتوں کی پیائش کے آلے کے طور پر کرتے ہیں۔ کسی معروف کوئز شو کی شناخت اور اس کا تجزیہ کریں اور کوئز کی تشکیل کے لیے ایک ایسا پائتھن پروگرام تحریر کریں جس میں آپ کے تجزیہ کے نتیجے کے طور پر شناخت کی گئی فعالیت کے علاوہ مندرجہ ذیل بھی موجود ہوں۔

- کسی سوال کو شامل کرنے، ترمیم کرنے یا حذف کرنے کے لیے یوزر آئی ڈی اور پاس ورڈ تشکیل دیجیے۔
- کوئز میں حصہ لینے کی اجازت دینے سے پہلے طالب علم کو رجسٹر کریں۔
- مضمون سے متعلق زمرہ کے انتخاب کی اجازت دیں۔
- منتخب کیے گئے زمرہ کے مطابق سوالات کو ظاہر کریں۔
- حتمی اسکور کو ظاہر کریں۔

3۔ ہمارے آثار قدیمہ ہمارا اثاثہ ہیں۔ یہ ہمارے ثمر آور اور عظیم الشان ماضی کی عکاس نیز ہمارے مستقبل کے لیے قوت محرکہ ہیں۔ یونیسکو نے ہندوستان کے کچھ وراثتی مقامات کو عالمی ثقافتی ورثہ قرار دیا ہے۔ ان مقامات کے متعلق مندرجہ ذیل معلومات جمع کریں:

- مقام کا نام کیا ہے؟

© NCERT not to be republished

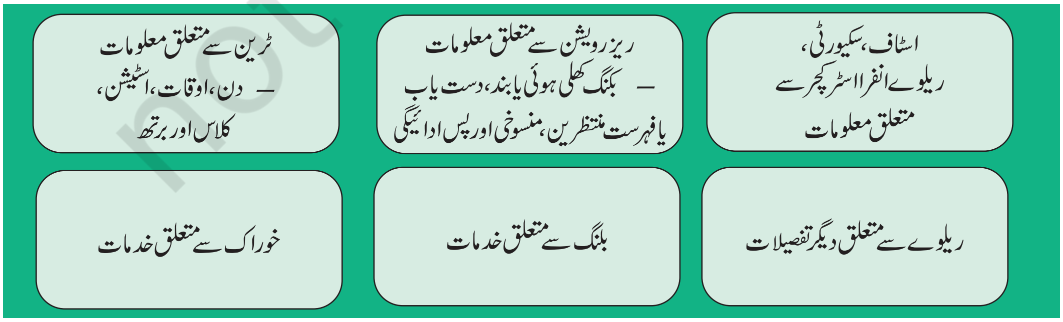
- یہ کہاں واقع ہے؟
  - ضلع
  - ریاست
- اسے کب تعمیر کیا گیا؟
- اس کی تعمیر کس نے کی؟
- اس کی تعمیر کیوں کی گئی؟
- ویب سائٹ لنک (اگر کوئی ہے)

مندرجہ ذیل کے لیے ایک پائتھن پروگرام لکھیے:

- مقامات کی فہرست میں وراثتی مقامات کو شامل کرنے، اس میں ترمیم کرنے یا اسے فہرست سے ہٹانے کے لیے یوزر آئی ڈی اور پاس ورڈ تشکیل دیجیے۔
- ہندوستان میں موجود عالمی ثقافتی ورثہ کی فہرست کو ظاہر کیجیے۔
- استعمال کنندہ کے ذریعے درج کیے گئے عالمی ثقافتی ورثہ سے متعلق معلومات کو تلاش کریں اور اور اسے ظاہر کریں۔
- استعمال کنندہ کے ذریعے داخل کیے گئے صوبے کی بنیاد پر عالمی ثقافتی ورثہ سے متعلق مقامات کے نام کو ظاہر کریں۔

4۔ نقل وحمل کا ہر ایک ذریعہ اپنے کام کو بے روک ٹوک اور موثر طور پر انجام دینے کے لیے ریزرویشن سسٹم کو استعمال کرتا ہے۔ اگر آپ تجزیہ کریں تو آپ کو بہت سی باتیں مشترک نظر آئیں گی۔ آپ کو نقل وحمل کے کسی ایک ذریعہ کی شناخت کرنی ہے اور اس کے لیے ایک ریزرویشن سسٹم تیار کرنا ہے۔ مثال کے طور پر آیئے ریلوے ریزرویشن سسٹم پر غور کریں جس کے بارے میں ہم نے پہلے بھی گفتگو کی تھی۔ ایک اچھے ریلوے ریزرویشن سسٹم کو ڈیزائن کرنے کے پیچیدہ کام کو سسٹم کے مختلف اجزا کو ڈیزائن کرنے اور پھر انھیں ایک دوسرے کے ساتھ کارگر انداز میں کام کرنے کے اہل بنانے کے طور پر دیکھا جاتا ہے۔ ممکنہ ذیلی نظاموں کو شکل 1 میں دکھایا گیا ہے۔ ان میں سے ہر ایک کو فنکشن کا استعمال کرکے تیار کیا جاسکتا ہے۔

نقل وحمل کے منتخب ذریعہ کی ریزرویشن ضروریات کو خودکار بنانے کے لیے پائتھن کوڈ لکھیے۔



شکل 1: ریلوے ریزرویشن سسٹم

© NCERT not to be republished