

12 ஆம் வகுப்பு - கணினி அறிவியல்
பாடம் - 7
பைத்தான் செயற்கூறுகள்

Prepared by,

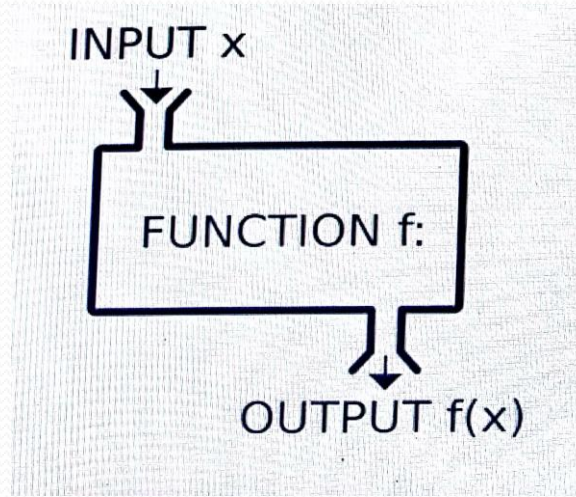
J. KAVITHA, B.Sc,B.Ed,M.C.A,M.Phil.,
Computer Instructor Gr - I,
GHSS, S.S.KULAM,
Coimbatore.

கற்றலின் நோக்கங்கள்

- செயற்கூறுகளின் கருத்து மற்றும் அதன் வகைகளைப் பற்றி அறிந்து கொள்ளுதல்.
- பயனர் வரையறுக்கும் செயற்கூறு மற்றும் உள்ளிணைந்த செயற்கூறுகளுக்கு இடையே உள்ள வேறுபாட்டை அறிந்து கொள்ளுதல்.
- ஒரு செயற்கூறினை எவ்வாறு அழைப்பது என்பதை பற்றி தெரிந்து கொள்ளுதல்.
- செயற்கூறின் அளபுருக்களை புரிந்து கொள்ளுதல்.
- செயற்கூறு செயலுருபுகள் மற்றும் மாறிகளின் வரையெல்லை ஆகியவற்றை பற்றி தெரிந்து கொள்ளுதல்.

செயற்கூறுகள் - அறிமுகம்

- ஒரு குறிப்பிட்ட செயலினை செய்வதற்காக வடிவமைக்கப்பட்டு பெயரிடப்பட்ட குறிமுறையின் தொகுதியே செயற்கூறு எனப்படும்.
- செயற்கூறானது பல வகை உள்ளீடுகளான மாறிகள் மற்றும் கோவைகள் மீது செயல்பட்டு நிலையான வெளியீட்டைத் தருகிறது.



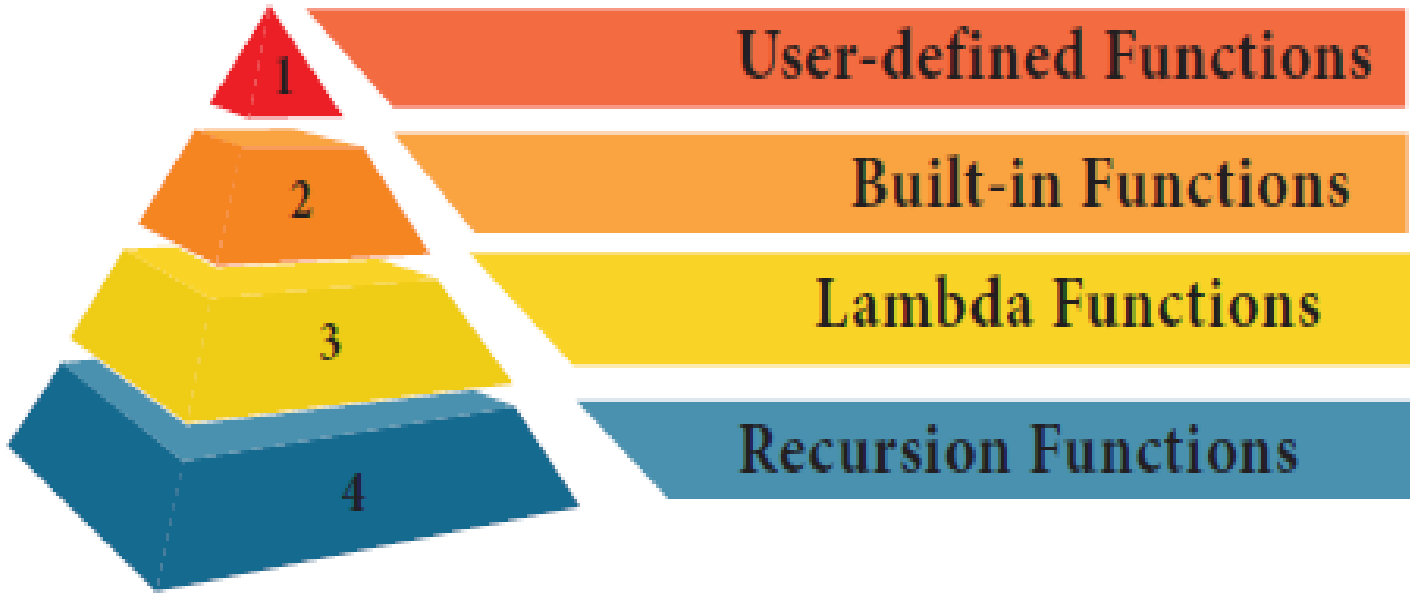
செயற்கூறுகள்

- நிரலில், ஒரே செயலை பல தடவை செய்ய நேரிட்டால் அதற்கான குறிமுறையை மீண்டும் எழுதத் தேவையில்லை. செயற்கூறுவை அழைத்தால் மட்டும் போதும், அந்த அழைப்பு பைத்தானின் செயற்கூற்றின் உள்ளேயுள்ள குறிமுறையை இயக்கச் செய்யும்.
- செயற்கூறுவின் பயன்பாடு ஒரு நிரலை, எழுத, படிக்க, சோதிக்க மற்றும் பிழை திருத்தும் வேலைகளை எளிதாக்குகிறது.

செயற்கூறுவின் நன்மைகள்

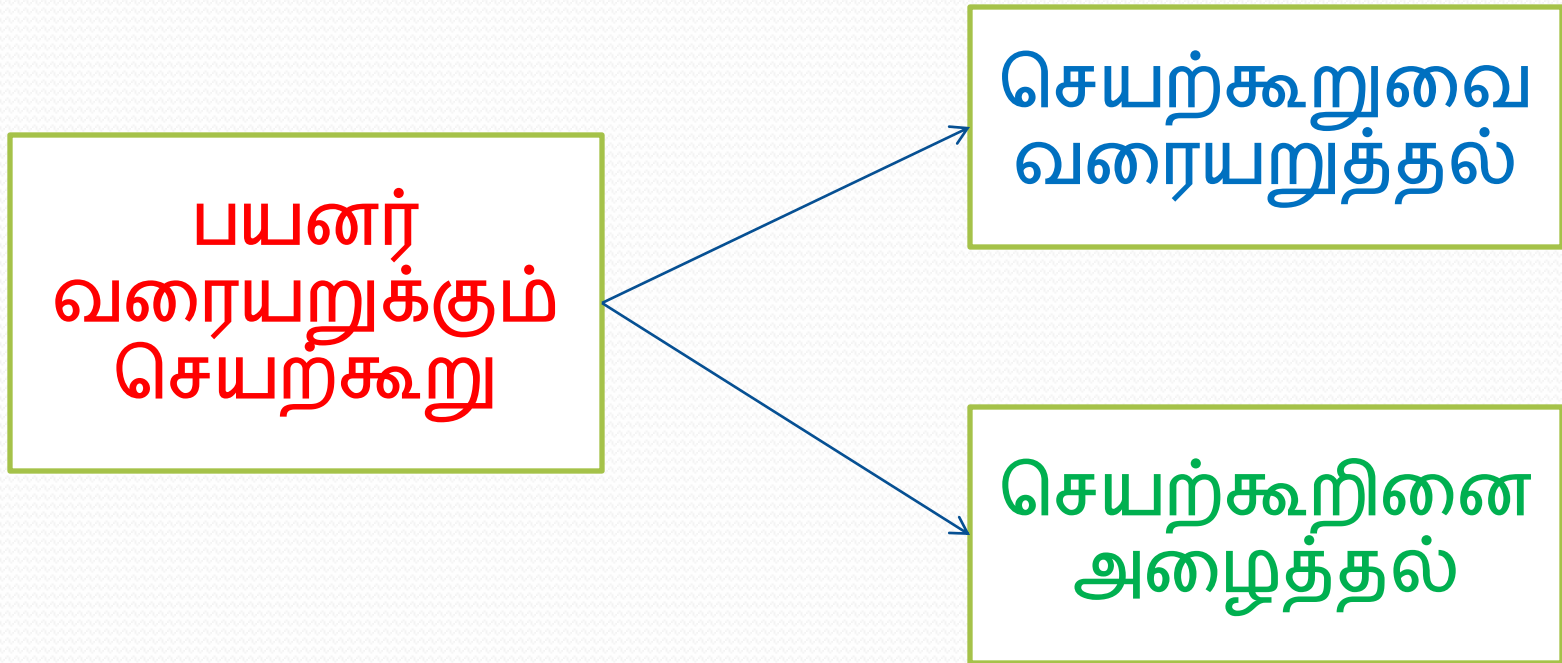
- நிரலில், குறிமுறையை மீண்டும் எழுதுவதை தவிர்த்து குறிமுறையின் மறு பயனாக்கத்திற்கு உதவுகிறது.
- நமது பயன்பாட்டிற்குச் சிறந்த கூறுநிலையை வழங்குகிறது.

செயற்கூறுகளின் வகைகள்



செயற்கூறுகள்	விளக்கம்
பயனர் வரையறுக்கும் செயற்கூறுகள்	<p>பயனர்கள் தாங்களாகவே வரையறுக்கும் செயற்கூறுகள் பயனர் வரையறுக்கும் செயற்கூறுகள் எனப்படும்</p>
உள்ளிணைந்த செயற்கூறுகள்	<p>பைத்தானில் உள்ளடக்கப்பட்ட செயற்கூறுகள் உள்ளிணைந்த செயற்கூறுகள் எனப்படும்</p>
லாம்டா செயற்கூறுகள்	<p>பைத்தானில், அனாமத்து வரையறுக்கப்படும் செயற்கூறுகள் பெயரில்லாத செயற்கூறுகள் அல்லது லாம்டா செயற்கூறுகள் எனப்படும்.</p>
தற்சுழற்சி செயற்கூறுகள்	<p>ஒரு செயற்கூறு தன்னைத்தானே அழைத்தால் அது தற்சுழற்சி செயற்கூறு எனப்படும்.</p>

பயனர் வரையறுக்கும் செயற்கூறுகள்



செயற்கூறுவை வரையறுத்தல்:

- ஒரு செயல்பாட்டை உருவாக்கி அதனை பயன்படுத்துவதற்கு செயற்கூறினை வரையறுக்க வேண்டும்.
- பைத்தான் மொழியில் பல உள்ளிணைந்த செயற்கூறுகள் உள்ளன.
- இருப்பினும் பயனர் அவருக்குத் தேவையான செயற்கூறினையும் வரையறுக்க முடியும்.

செயற்கூறினை வரையறுக்கும் போது நினைவில் கொள்ள வேண்டியவை:

- செயற்கூறு தொகுதி def என்ற சிறப்புச் சொல்லுடன் தொடங்கி செயற்கூறுவின் பெயர் மற்றும் () அடைப்புக்குறியுடன் முடிய வேண்டும்.
- ஏதேனும் உள்ளீட்டு செயலுருபுகள் அல்லது அளபுருக்கள் இருப்பின் அவற்றை வரையறுக்கும்போதே () என்ற அடைப்புக் குறிக்குள் கொடுக்க வேண்டும்.
- குறிமுறை தொகுதியானது எப்பொழுதும் முக்காற்புள்ளிக்கு பிறகு உள்தள்ளி வரவேண்டும்.
- return [கோவை] கூற்று செயற்கூறுவை முடித்து வைக்கும். விருப்பப்பட்டால் கோவையின் மதிப்பை அழைக்கும் கூற்றுக்கு திருப்பி அனுப்பும். செயலுருபுகள் இல்லாத return கூற்று, return none க்கு நிகரானது.

தொடரியல்

```
def <function_name ([parameter1, parameter2...] )> :  
    <Block of Statements>  
    return <expression / None>
```

- சதுர அடைப்புக்குறிக்குள் உள்ளவை அனைத்தும் விருப்பத்தேர்வாகும்.
- பைத்தான் சிறப்புச் சொற்கள் செயற்கூறுவின் பெயர்களாக பயன்படுத்தக் கூடாது.

செயற்கூறினை அழைத்தல்:

எடுத்துக்காட்டு:

```
def hello():  
    print ("hello - Python")  
    return  
hello()
```

- “hello()” செயற்கூறினை அழைக்கும் பொழுது நிரல் பின்வரும் சரத்தை வெளியீடுகிறது.

வெளியீடு:

hello - Python

- மாற்றாக, “hello()” செயற்கூறினை print() செயற்கூற்றிலிருந்தும் அழைக்கலாம்.

எடுத்துக்காட்டு:

```
def hello():  
    print (“hello - Python”)  
    return  
print (hello())
```

- return கூற்றில் எந்த அளபுருவும் இல்லையெனில் வெளியீட்டின் இறுதிக் கூற்றாக, “None” வெளியிடப்படும்.

வெளியீடு:

```
hello - Python  
None
```

செயற்கூறினுள் அளபுருக்களை அனுப்புதல்

- அளபுருக்கள் அல்லது செயலுருபுகளை செயற்கூறினுக்கு அனுப்பலாம்.

பொதுவடிவம்:

def function_name (parameter(s) separated by comma):

எடுத்துக்காட்டு : # assume $w = 3$ and $h = 5$

```
def area(w,h):  
    a = w*h  
    print(a)  
area(3,5)
```

வெளியீடு: 15

- மேலே உள்ள குறிமுறையில் அகலம் மற்றும் உயரத்தின் மதிப்புகள் முறையே w மற்றும் h அளபுருக்களில் இருத்தப்படுகிறது. அளபுருக்களுக்கு முறையே 3 மற்றும் 5 மதிப்புகளை அனுப்பினால் வெளியீடாக 15 என்ற மதிப்பதை திருப்பி அனுப்பும்.

பயனர் வரையறுக்கும் செயற்கூறுகளின் நன்மைகள்

- செயற்கூறுகள் ஒரு நிரலை சிறு தொகுதியாக பிரிக்க உதவுகிறது. இது குறிமுறையை எளிதாக கையாள உதவுகிறது.
- குறிமுறையின் மறுபயனாக்கத்திற்கு உதவுகிறது. ஒவ்வொரு முறையும் கூற்றுகளின் வரிசைகளை நிறைவேற்றும் போது, நாம் அந்த செயற்கூற்றினை அழைத்தால் போதுமானது.
- செயற்கூற்றின் செயல்பாடுகளை மாற்றம் செய்வது எளிதாகிறது. வெவ்வேறு நிரலர்கள் வெவ்வேறு செயற்கூற்றில் வேலை செய்ய முடியும்.

செயற்கூறு செயலுருபுக்கள்

- செயற்கூறை அழைப்பதற்கு செயலுருபுக்கள் பயன்படுகின்றன.
- 4 வகையான செயலுருபுக்களை செயற்கூறில் பயன்படுத்தலாம்.

1

தேவைப்படும் செயலுருபுக்கள்
(Required arguments)

2

சிறப்புச் சொல் செயலுருபுக்கள்
(Keyword arguments)

3

தானமைவு செயலுருபுக்கள்
(Default arguments)

4

மாறும்-நீள செயலுருபுக்கள்
(Variable-length arguments)

தேவைப்படும் செயலுருபுகள்

- செயலுருபுகளை செயற்கூற்றுக்கு செயலுருபுகள் ஆகும். சரியான அனுப்புவதே இடவரிசையில் தேவைப்படும்
- தொடரியல் வெளியீட்டை செயலுருபாவது தேவை. பிழையை தவிர்த்து, தேவையான பெறுவதற்கு குறைந்தது ஒரு

எடுத்துக்காட்டு

```
def printstring(str):  
    print ("Example - Required arguments ")  
    print (str)  
    return  
printstring("Welcome")
```

வெளியீடு

Example - Required arguments
Welcome

சிறப்புச்சொல் செயலுருபுக்கள்

- அளபுருக்களின் பெயரை அடையாளம் கண்ட பின்பு சிறப்புச்சொல் செயலுருபானது செயற்கூறினை அழைக்கிறது.

எடுத்துக்காட்டு

```
def printdata (name):  
    print ("Example-1 Keyword arguments")  
    print ("Name :",name)  
    return  
printdata(name = "Gshan")
```

வெளியீடு

```
Example-1 Keyword arguments  
Name :Gshan
```

தானமைவு செயலுருபுக்கள்

- பைத்தானில் செயற்கூறை அழைக்கும் போது எந்த மதிப்பும் கொடுக்கப்படவில்லை எனில், செயலுருபானது தானாகவே மதிப்பை எடுத்துக்கொள்ளும். இதுவே தானமைவு செயலுருபு ஆகும்.

எடுத்துக்காட்டு

```
def printinfo( name, salary = 3500):  
    print ("Name: ", name)  
    print ("Salary: ", salary)  
    return  
printinfo("Mani")
```

வெளியீடு

```
Name: Mani  
Salary: 3500
```

மாறும் நீள செயலுருபுக்கள்

- சில சமயங்களில், ஏற்கனவே குறிப்பிட்ட செயலுருபுக்களை விட அதிகமான செயலுருபுக்களை அனுப்ப வேண்டி இருக்கும். அதற்கு மாறும் நீள செயலுருபை பயன்படுத்தலாம்.
- * குறியீடு மாறும் நீள செயலுருபுகளை வரையறுக்க பயன்படுகிறது.

எடுத்துக்காட்டு

```
def printnos (*nos):  
    for n in nos:  
        print(n)  
    Return  
print ('Printing two values')  
printnos (1,2)  
print ('Printing three values')  
printnos (10,20,30)
```

வெளியீடு

```
Printing two values  
1  
2  
Printing three values  
10  
20  
30
```

உள்ளிணைந்த செயற்கூறுகள்

- பைத்தான் மொழியில் பல உள்ளிணைந்த செயற்கூறுகள் உள்ளன.

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
abs ()	எண்ணின் முழு எண்ணை திருப்பி அனுப்பும். செயலுருபானவது முழு எண்ணாகவோ அல்லது தசம எண்ணாகவோ இருக்கலாம்.	<pre>x=20 y=-23.2 print('x = ', abs(x)) print('y = ', abs(y)) வெளியீடு x = 20 y = 23.2</pre>
ord ()	கொடுக்கப்பட்ட யுனிக்கோடு எழுத்திற்கு ASCII மதிப்பை திருப்பி அனுப்பும்.	<pre>c= 'a' d= 'A' print ('c = ',ord (c)) print ('A = ',ord (d)) வெளியீடு c = 97 A = 65</pre>

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
chr ()	கொடுக்கப்பட்ட ASCII மதிப்பிற்கு யுனிக்கோடு எழுத்தை திருப்பி அனுப்பும்.	<pre>c=65 d=43 print (chr (c)) print (chr (d)) வெளியீடு A +</pre>
bin ()	கொடுக்கப்பட்ட முழு எண்ணிற்கு நிகரான இரும் எண்ணை “ob”யை முன்னொட்டாக கொண்டு திருப்பி அனுப்பும்.	<pre>x=15 y=101 print ('15 in binary : ',bin (x)) print ('101 in binary : ',bin (y)) வெளியீடு 15 in binary : 0b1111 101 in binary : 0b1100101</pre>

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
type ()	கொடுக்கப்பட்ட பொருளின் தரவின வகையை திருப்பி அனுப்பும்.	<pre>x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s))</pre> வெளியீடு <class 'float'> <class 'str'> <class 'bool'>
id ()	கொடுக்கப்பட்ட பொருளின் முகவரியை அனுப்பும். நினைவக திருப்பி	<pre>x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y))</pre> வெளியீடு address of x is : 1357486752 address of y is : 13480736

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
min ()	கொடுக்கப்பட்ட பட்டியலில் இருந்து மிகச்சிறிய மதிப்பைத் திருப்பி அனுப்பும்	<pre>MyList = [21,76,98,23] print ('Minimum of MyList :', min(MyList)) வெளியீடு Minimum of MyList : 21</pre>
max ()	கொடுக்கப்பட்ட பட்டியலில் இருந்து மிகப்பெரிய மதிப்பைத் திருப்பி அனுப்பும்	<pre>MyList = [21,76,98,23] print ('Maximum of MyList :', max(MyList)) வெளியீடு Maximum of MyList : 98</pre>
sum ()	கொடுக்கப்பட்ட பட்டியலில் உள்ள மதிப்புகளின் கூட்டுத் தொகையை திருப்பி அனுப்பும்	<pre>MyList = [21,76,98,23] print ('Sum of MyList :', sum(MyList)) வெளியீடு Sum of MyList : 218</pre>

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
format ()	கொடுக்கப்பட்ட எண்ணை வேறு எண்முறைக்கு மாற்றி திருப்பி அனுப்புகிறது	x= 14 y= 25 print ('x value in binary :',format(x,'b')) print ('y value in octal :',format(y,'o')) வெளியீடு x value in binary : 1110 y value in octal : 31

round ()	கொடுக்கப்பட்ட எண்ணிற்கு அருகே உள்ள முழு எண்ணாக மாற்றி திருப்பி அனுப்பும்.	x= 17.9 y= 22.2 print ('x value is rounded to', round(x)) print ('y value is rounded to', round(y)) வெளியீடு x value is rounded to 18 y value is rounded to 22
------------------	---	---

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
pow ()	கொடுக்கப்பட்ட எண்ணின் ab அடுக்கு பெருக்கத்தை திருப்பி அனுப்பும். (a**b) a ன் அடுக்கு b.	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) வெளியீடு 25 125.0
floor ()	x ஐ விடக் குறைவான அல்லது x-க்கு நிகரான பெரிய முழு எண்ணை திருப்பி அனுப்பும்.	import math x=26.7 y=-26.7 print (math.floor (x)) print (math.floor (y)) வெளியீடு 26 -27

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
ceil ()	x ஐ விட பெரிய அல்லது x-க்கு நிகரான சிறிய முழு எண்ணை திருப்பி அனுப்பும்.	<pre>import math x= 26.7 y= -26.7 print (math.ceil (x)) print (math.ceil (y))</pre> வெளியீடு 27 -26
sqrt ()	x ன் வர்க்கமூலத்தை திருப்பி அனுப்பும்.	<pre>import math a= 30 b= 49 print (math.sqrt (a)) print (math.sqrt (b))</pre> வெளியீடு 5.477225575051661 7.0

பெயரில்லாத செயற்கூறுகள்

- பைத்தானில், பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறுவுக்கு பெயரில்லாத செயற்கூறு என்று பெயர்.
- மற்ற சாதாரண செயற்கூறுகள் def என்ற சிறப்புச் சொல்லுடன் வரையறுக்கப்படுகிறது.
- பைத்தானில் அனாமத்து செயற்கூறுகள் லாம்ப்டா என்ற சிறப்புச் சொல்லுடன் வரையறுக்கப்படுகிறது. எனவே, அனாமத்து செயற்கூறுகளை லாம்ப்டா செயற்கூறுகள் என்றும் கூறலாம்.

பெயரில்லா செயற்கூறின் பயன்கள்:

- லாம்ப்டா செயற்கூறு பெரும்பாலும் சிறிய மற்றும் ஒருமுறை பெயரில்லாத செயற்கூறை உருவாக்க பயன்படுகிறது.
- filter(), map() மற்றும் reduce() போன்ற செயற்கூறுகளுடன் சேர்த்து லாம்ப்டா செயற்கூறுகளை பயன்படுத்தலாம்.

பொதுவடிவம்:

`lambda [argument(s)] :expression`

எடுத்துக்காட்டு:

```
sum = lambda arg1, arg2: arg1 + arg2  
print ('The Sum is :', sum(30,40))  
print ('The Sum is :', sum(-30,40))
```

வெளியீடு:

```
The Sum is : 70  
The Sum is : 10
```

return கூற்று

- return கூற்று செயற்கூறினை முடித்து வைத்து அழைப்புக் கூற்றுக்கு மதிப்பை திருப்பி அனுப்பும். பொதுவாக செயற்கூறின் நோக்கம் உள்ளீடைப் பெற்று ஏதேனும் ஒரு மதிப்பை திருப்பி அனுப்புவதாகும்.
- ஒரு செயற்கூறு மதிப்பை அழைப்புக்கூற்றுக்கு திருப்பி அனுப்ப தயாராக இருக்கும் போது return கூற்று பயன்படுத்தப்படுகிறது.
- செயற்கூறில் ஒன்றுக்கு மேற்பட்ட return கூற்றுகள் இருந்தாலும் ஒரே ஒரு return கூற்று மட்டுமே இயக்க நேரத்தில் இயக்கப்படும்.

return கூற்றின் பொது வடிவம்:

return [கோவைகளின் பட்டியல்]

தற்சுழற்சி செயற்கூறுகள்

- ஒரு செயற்கூறு தன்னைத்தானே அழைத்தால் அது தற்சுழற்சி செயற்கூறு எனப்படும். மடக்கினைப் போன்று தற்சுழற்சியும் செயல்படும்.
- ஒரு நிபந்தனையில் நிறுத்தப்படாவிட்டால், ஒரு செயல்முறை காலவரையின்றி செயல்படும்.
- தற்சுழற்சி செயற்கூறுவிற்கு கொடுக்கப்படும் நிபந்தனை அடிப்படை நிபந்தனை எனப்படும். அடிப்படை நிபந்தனை கொடுக்கப்படவில்லையெனில் மடக்கு காலவரையின்றி இயங்கும்.

தற்சுழற்சி செயற்கூறு இயங்கும் விதம்

- தற்சுழற்சி செயற்கூறு வெளிப்புற குறிமுறையிலிருந்து அழைக்கப்படும்.
- அடிப்படை நிபந்தனை நிறைவேற்றப்பட்டால் நிரலானது ஏற்ற வெளியீடு கொடுத்து வெளியேறும்.
- இல்லையென்றால் செயற்கூறானது தேவையான செயற்பாட்டை இயக்கும். மேலும் தற்சுழற்சி முறையில் தன்னைத்தானே அழைத்துக் கொள்ளும்.

எடுத்துக்காட்டு :

```
def fact(n):  
    if n == 0:  
        return 1  
    else:  
        return n * fact (n-1)  
print (fact (0))  
print (fact (5))
```

வெளியீடு:

1
120

மாறிகளின் வரையெல்லை

- மாறிகளின் வரையெல்லை நிரலின் அணுகக்கூடிய பகுதியைக் குறிப்பதாகும். அதாவது எந்த பகுதியில் மாறிகளைப் பயன்படுத்துகிறோமோ அதைக் குறிக்கிறது.
- வரையெல்லையானது நடப்பு மாறித் தொகுதிகள் மற்றும் அதன் மதிப்புகளைக் கொண்டிருக்கும்.
வகைகள்:

1. உள்ளமை வரையெல்லை
2. குளோபல் வரையெல்லை

உள்ளமை வரையெல்லை

- ஒரு செயற்கூறுவின் உடற்பகுதியின் உள்ளே ஒரு மாறியை அறிவிப்பது உள்ளமை மாறி எனப்படும்.

உள்ளமை மாறியின் விதிமுறைகள்

- உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே பயன்படுத்த முடியும்.
- செயற்கூறினுள் மாறி உருவாக்கப்படும் போது அது உள்ளமைவாக அமையும்.
- செயற்கூறு இயக்கப்படும் போது மட்டுமே உள்ளமை மாறிகள் உருவாக்கப்படும்.
- முறையான அளபுருக்களானது அச்செயற்கூறுக்கு உள்ளமை மாறியாக கருதப்படும்.

எடுத்துக்காட்டு : உள்ளமை மாறியை உருவாக்குதல்

```
def loc():  
    y=0                # local scope  
    print(y)  
loc()
```

வெளியீடு: 0

குளோபல் வரையெல்லை

- குளோபல் வரையெல்லை உடைய மாறியை நிரலில் எங்கு வேண்டுமானாலும் அணுக முடியும். எந்த ஒரு செயற்கூறு வரையெல்லைக்கு வெளியேயும் மாறியை வரையறுத்து உருவாக்க முடியும்.

குளோபல் வரையெல்லை சிறப்புச் சொல்லின் விதிமுறைகள்:

- செயற்கூறுக்கு வெளியே மாறியை அறிவிக்கும் போது அது தானமைவாக குளோபல் ஆகும் “global” என்ற சிறப்புச் சொல்லை பயன்படுத்த வேண்டியதில்லை.
- செயற்கூறின் முழுதளாவிய மாறியை படிக்க மற்றும் எழுத “global” சிறப்புச் சொல் பயன்படுத்த வேண்டும்.
- செயற்கூறுவிற்கு வெளியே “global” என்ற சிறப்புச் சொல் எந்த விளைவையும் ஏற்படுத்தாது.

எடுத்துக்காட்டு : செயற்கூறுவின் உள்ளிருந்து குளோபல் மாறியை அணுகுதல்

```
c = 1                                # global variable
def add():
    print(c)
add()
```

வெளியீடு: 1

செயற்கூறில் தொகுப்பு

- செயற்கூறு திருப்பி அனுப்பும் மற்றொரு செயற்கூறிற்கு செயலுருபாக, பின்னலான அமைப்பில் பயன்படுத்தினால் அதற்கு தொகுப்பு என்று பெயர்.

எ.கா: பயனரிடமிருந்து எண் மதிப்பை அல்லது கோவையை உள்ளீடாகப் பெற விரும்பினால், `input()` செயற்கூறு மூலம் பயனரிடமிருந்து சரத்தை உள்ளீடாகப் பெற்று `eval()` செயற்கூறு மூலம் அதன் மதிப்பை மதிப்பீடு செய்ய வேண்டும்.

இந்த நிரல் தொகுப்பை விளக்குகிறது.

```
>>> n1 = eval (input ("Enter a number: "))
Enter a number: 234
>>> n1
234
>>> n2 = eval (input ("Enter an arithmetic expression: "))
Enter an arithmetic expression: 12.0+13.0 * 2
>>> n2
38.0
```

மதிப்பீடு

1. ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக வடிவமைக்கப்பட்டு, பெயரிடப்பட்ட குறிமுறையின் தொகுதி
அ) மடக்கு ஆ) கிளைப்பிரிப்பு இ) செயற்கூறு ஈ) தொகுதி
செயற்கூறு
2. தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறை இவ்வாறு அழைப்பர்.
அ) உள்ளிணைந்த ஆ) தற்சுழற்சி இ) லாம்டா ஈ) return கூற்று
தற்சுழற்சி
3. எந்த செயற்கூறு பெயரில்லா செயற்கூறு என்று அழைக்கப்படுகிறது?
அ) லாம்டா ஆ) தற்சுழற்சி இ) செயற்கூறு ஈ) வரையறை
லாம்டா
4. செயற்கூறு தொகுதியை எந்த சிறப்புச்சொல் தொடங்கி வைக்கிறது?
அ) define ஆ) for இ) finally ஈ) def
def

மதிப்பீடு

5. எந்த சிறப்புச்சொல் செயற்கூறு தொகுதியை முடித்து வைக்கிறது?

அ) define ஆ) return இ) finally ஈ) def

return

6. செயற்கூறு வரையறையில் பின்வரும் எந்த குறியீடு பயன்படுத்தப்படுகிறது?

அ) ; (அரைப்புள்ளி) ஆ) . (புள்ளி) இ) : (முக்காற் புள்ளி) ஈ) \$ (டாலர்)
: (முக்காற் புள்ளி)

7. செயற்கூறுக்கு எந்த செயலுருபு சரியான இட வரிசையில் செயலுருபுகளை அனுப்பும்?

அ) தேவையான ஆ) சிறப்புச்சொல் இ) தானமைவு ஈ) மாறிநீளம்
தேவையான

8. testpython() செயற்கூறை வரையறுக்க பின்வரும் எந்த சிறப்புச் சொல் பயன்படுகிறது?

அ) define ஆ) pass இ) def ஈ) while

def

மதிப்பீடு

9. பின்வரும் கூற்றுகளைப் படித்து, சரியான கூற்றுகளை தேர்ந்து எடுக்கவும்.

- I. பைத்தானில், செயற்கூறை வரையறுக்கும் போது குறிப்பிட்ட தரவு வகைகளை குறிப்பிடத் தேவையில்லை.
- II. பைத்தான் சிறப்புச் சொற்களைச் செயற்கூறின் பெயராகப் பயன்படுத்தலாம்.

அ) I சரி மற்றும் II தவறு

ஆ) இரண்டுமே சரி

இ) I தவறு மற்றும் II சரி

ஈ) இரண்டுமே தவறு

I சரி மற்றும் II தவறு

10. கொடுக்கப்பட்ட கூற்றை வெற்றிகரமாக நிறைவேற்றுவதற்கு, பின்வருவனவற்றுள் சரியான ஒன்றைத் தேர்ந்தெடு

If__:

```
print(X,"is a leap year")
```

அ) $x \% 2 = 0$

ஆ) $x \% 4 == 0$

இ) $x / 4 = 0$

ஈ) $x \% 4 = 0$

$x \% 4 == 0$

முக்கிய வினாக்கள்

1. செயற்கூறு என்றால் என்ன?
2. செயற்கூறுவின் முக்கிய நன்மைகள் யாவை?
3. செயற்கூறுவின் வகைகளை எடுத்துக்காட்டுடன் விவரி.
4. செயற்கூறினை வரையறுக்கும் போது குறிப்பிடப்பட வேண்டிய குறிப்புகள் யாவை ?
5. மாறியின் வரையெல்லை என்றால் என்ன? அதன் வகைகளைக் குறிப்பிடுக.
6. தன்னைத் தானே அழைக்கும் செயற்கூறில் அடிப்படை நிபந்தனை என்றால் என்ன ?
7. உள்ளமை மாறிகளுக்கான விதிமுறைகளை எழுதுக.
8. பைத்தானிலுள்ள முழுதளாவிய சிறப்புச் சொல்லுக்கான அடிப்படை விதிமுறைகளை எழுதுக.
9. ceil() மற்றும் floor() செயற்கூறுகளை வேறுபடுத்துக.
10. பின்வரும் உள்ளிணைந்த செயற்கூறுகளை விளக்குக.
(a) id() (b) chr() (c) round() (d) type() (e) pow()

நன்றி!

கல்வி என்பது கடல்.
அதை கற்றுக்
கொடுப்பது தொழில்
அல்ல தவம்.
நம்பிக்கை கை
விட்டாலும் நீ கற்ற
கல்வி என்றும் உன்னை
கை விடாது.
அக்கல்வியைப் பெற்று
சிறப்போடு வாழ
வாழ்த்துக்கள்.



ஜெ. கவிதா B.Sc, B.Ed, M.C.A, M.Phil.,
கணிணி பயிற்றுநர் நிலை - I
அரசு மேல்நிலைப்பள்ளி,
சர்க்கார்சாமக்குளம்,
கோயம்புத்தூர் - 641107.