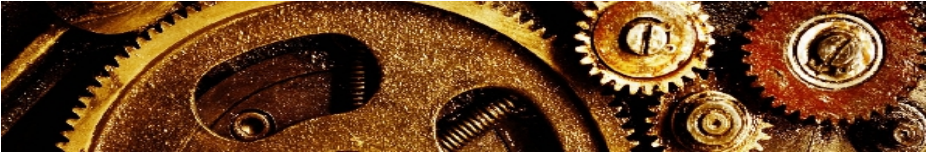


AlgoHack : පයිතන් #3



ශ්‍රීත හා ලැයිස්තු

කතුවරු
නිරංජන් මීගම්මන, නවින් තමිදු

සංස්කරණය
විශ්ව කුමාර



Google for Education



AlgoHack කුඩා අවදියේදී ප්‍රමුඛව පරිගණක විද්‍යාව සහ ක්‍රම ලේඛනය
ඉගැන්වීමට ගිණුම් සපුරා කල අත්මකායකි. එයට Google for Education සහ ශ්‍රී
ලංකා පරිගණක සංගමය සහය ලබා දේ

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/). Shilpa Sayura Foundation (shilpasayura.org)



1.0 පයිතන් ශ්‍රිත

ශ්‍රිතයක් යනු නිතර භාවිතා වන කේත කොටසකි.

ශ්‍රිත කේත ප්‍රමාණය අඩුකරන අතර කේත නඩත්තුවද පහසු කරයි.

විචල්‍යකට සේම ශ්‍රිතයකටද නාමයක් ඇත.

නාමකරණ නීතියද එසේම වේ.

`print(x)` යනු ප්‍රතිදාන ශ්‍රිතයකි.

ශ්‍රිත වලට ලබාදෙන ආදාන පරාමිති (parameters) නම් වේ.

ශ්‍රිත ආදාන ක්‍රියාවලියෙන් පසු ප්‍රතිදානයක් ලබාදේ.

පයිතන්හි සහජ ශ්‍රිත ඇති අතර, අපට කැමති ආකාරයේ ශ්‍රිත ද කේතනය කළ හැක.

පයිතන් ශ්‍රිතයක් අර්ථ දැක්වීම සඳහා `def` විධානය භාවිතා කරයි.

```
def greeting():  
    print ("Hello Friend!")  
    print ("Welcome to Python!")
```

#ශ්‍රිත කේතනයේ විවිධ තැන්වල යොදාගත හැක

```
greeting()  
greeting()  
greeting()
```

1.1 පරාමිති සහිත ශ්‍රිත

`str` විචල්‍යකි. එය `printme` ශ්‍රිතයට ලබාදෙන ආදානයකි.

```
def printme( str ):  
    print (str)  
    return; # කිසිවක් ප්‍රතිදානය නොකරයි
```

printme ශ්‍රිතය යොදාගැනීම

```
printme("1st call to printme function")
printme("2nd call to same function")
```

ශ්‍රිතයකින් අගයක් ප්‍රතිදානය

```
def addTwo( a,b ):
    c=a+b
    return c; # අගය ප්‍රතිදානය කරයි
```

ශ්‍රිතයක ප්‍රතිදාන අගයක් නැවත ආදානය

```
d=addTwo (1,3)
print (d)
e=addTwo (d,3)
print (d)
f= addTwo (d,e)
print(f)
```

මේ ක්‍රියාවලියේ අගයන් **2** සහ **5** වී නම් ප්‍රතිදාන මොනවාද?

ශ්‍රිතවල පරාමිති විචල්‍යයන් වේ. ඒවා ශ්‍රිතය තුලදී වෙනස් වූයේ නම් එය විචල්‍යයේ අගයට බලපෑමක් ඇති කරයි.

පහත කේතය සලකන්න

```
def appendMe( mylist ):
    mylist.append([1,2,3,4]);
    print "Values inside the function: ", mylist
    return
```

```
mylist = [10,20,30]
```

```
print ("Values: ", mylist)
appendMe( mylist )
print ("Values outside the function: ", mylist)
```

ඉහත කේතයේ ක්‍රියාවලිය සලකන්න

mylist ලැයිස්තුව මුද්‍රණය කරයි.
 Mylist appendMe() ශ්‍රිතයට පරාමිතියක් සේ ආදානය කරයි.
 appendMe() ශ්‍රිතය mylist ලැයිස්තුවට [1,2,3,4] අරුව නව
 අවයවයක් ලෙස එකතු කරයි.
 නව mylist ලැයිස්තුව මුද්‍රණය කරයි.
 ශ්‍රිතයෙන් ඉවත්වේ.
 නැවත mylist ලැයිස්තුව මුද්‍රණය කරයි.
 mylist අගය වෙනසක් වී ඇතිද?

1.2 ශ්‍රිත තුළ ශ්‍රිත

ශ්‍රිතයකට තවත් ශ්‍රිතයක් ඇමතීමට හැකිය.
 ගණකාභාසයක පරිමාව ලබාදෙන ශ්‍රිතය

```
def cube( y ):
    c= y * y * y
    return c

def doublecube ( z ):
    z=z * 2
    dc=cube( z) # cube ශ්‍රිතය ඇමතයි
    return dc
print ("Cubes")

for x in range(1,6):
    print (cube(x))
```

```
print ("1 to 5 doubled")
for x in range(1,6):
    print (doublecube(x))
```

මේ ක්‍රියාවලිය දැකීම සඳහා පයිතන් කේන් පියවර පෙන්වන
වැරදිසෙයනය (debugger) භාවිතා කරන්න.
ඒ මගින් කේතනය විච්ඡේද කරන්න.

1.3 බුලියානු

ග්‍රීක වලට බුලියානු අගයන් ප්‍රතිදානය කළ හැකිය.

සංඛ්‍යාවක් ධන අගයක්ද යන වග බැලීම.

```
def isPositive(x):
    if (x >= 0):
        return 1 # ජන්‍ය නම් 1
    else:
        return 0 # අජන්‍ය නම් 0
```

```
result = isPositive(1)
print (result)
result = isPositive(-1)
print (result)
```

1.4 සංඛ්‍යාවක් 0 න් බෙදීම වැළැක්වීම

```
def xdivision(x,y):
    if (y == 0):
        print ("division by zero not allowed")
        return
    else:
        return x / y
```

```
result = xdivision( 5.0 , 2 )
print (result)
result = xdivision( 5.0 , 0 )
print (result)
```

1.5 ගණිතමය ශ්‍රිත

ගණිතමය ශ්‍රිත භාවිතා කිරීමට ගණිත මොඩියුලය ආනයනය කළ යුතුය. එයට import math විධානය භාවිතා කරයි.

සෘණ සංඛ්‍යාවක නිරපේක්ෂ අගය

```
import math
n = -100.5
# නිරපේක්ෂ අගය abs() ශ්‍රිතයෙන් ලබාගනී
# abs() ශ්‍රිතය ගණිත මොඩියුලයේ අඩංගුය
print(abs(n))
```

භාගික සංඛ්‍යාවක නිරපේක්ෂ අගය

```
n = 100.7
# n හි පහළ පූර්ණ අගය ගන්න
print(math.floor(n)) # ප්‍රතිදානය 100.0
```

```
# n හි ඉහළ පූර්ණ අගය ගන්න
print(math.ceil(n)) # ප්‍රතිදානය 101.0
```

දශම සංඛ්‍යා වැටපීම

```
number = 1.23456
print(round(number)) # දශමස්ථාන නොමැත
print(round(number, 0))
print(round(number, 1)) # දශමස්ථාන 1 කට
print(round(number, 2))
print(round(number, 3)) # දශමස්ථාන 3 කට
```

ප්‍රතිදාන පරීක්ෂණ කරන්න

```
1      0 digits
1.0    0 digits
1.2    1 digit
1.23   2 digits
1.235  3 digits
```

1.6 ලැයිස්තු ශ්‍රිත

```
values = [0.9999999, 1, 2, 3]
r = sum(values)
print(r)
```

```
# fsum සමග අගයන් ඒකතු කිරීම
r = math.fsum(values)
print(r)
```

```
value1 = 123.45
truncate1 = math.trunc(value1)
print(truncate1) ප්‍රතිදානය කුමක්ද
```

```
sqrt වර්ගමූල ශ්‍රිතය
value1 = 4
print(math.sqrt(value1))
```

```
# සංඛ්‍යාවක බලයන් ලබාදෙන math.pow ශ්‍රිතය
a = math.pow(2, 3)
print(a)
```

```
** ක්‍රියාකාරකය
b = 2 ** 3
```

```
print(b) ප්‍රතිදානය කුමක්ද
```

```
# pi හි අගය ප්‍රතිදානය  
print(math.pi)
```

```
වචන හා ඉලක්කම් ප්‍රතිදානය
```

```
r = 10  
c = 75  
print ('Total is: ', r, 'Rupees and', c, ' cents')
```

```
# %d පූර්ණ සංඛ්‍යාවක් වෙ අදහස් කරයි  
x = 20  
y = 75  
print ('The sum of %d and %d is %d' % (x, y, x + y))
```

```
# %f දශම (float) සංඛ්‍යාවක් වෙ අදහස් කරයි  
x = 20.512  
y = 15.269  
print ('The sum of %f and %f is %f' % (x, y, x + y))
```

```
%s වචන (string) වෙ අදහස් කරයි.  
print "My name is %s and I am  %d years old!" % ('Hiruni', 19)
```

1.7 වචන ශ්‍රිත

```
#capitalize() - වචනයක පළමු අකුර කැපිටල් කරයි  
str = "this is python!";  
print (str.capitalize())  
upper() - ඉංග්‍රීසි වචන සම්පූර්ණයෙන්ම කැපිටල් කරයි  
print (str.upper())  
lower() - ඉංග්‍රීසි වචන සම්පූර්ණයෙන්ම සිම්පල් කරයි  
print (str.lower())
```


len(string) - වචනයේ අකුරු ගණන ලබා දෙයි.
print (len(str))

1.8 වචන වෙන්කිරීම

වචනයක් තනි අකුරු එකතු වී සෑදේ. එය අකුරු ලැයිස්තුවකි.
වචනයේ පළමු අකුරේ ස්ථානය 0 වේ.

```
s = "Niranjan Meegammana"
# පළමු අකුර 0 වෙනි ස්ථානය අත.
print(s[0]) # එමනිසු N ලබාදේ
print(s[1]) කුමන අකුර ලබාදේද?
# අවසන් අකුරේ ස්ථානය අකුරු ගණනට එකක් අඩුය
print(s[-1])
```

උප වචන ලබා ගැනීම

```
print(s[0:5])
# 5 නොමැතිව, 0 ස්ථානයේ සිට 5 දක්වා ප්‍රතිදානය කරයි
# 5 නොමැතිව, 1 සිට 5 දක්වා ස්ථානවල අකුරු ලබාදේ
print(s[1:5])
```

```
# 4 ස්ථානයේ සිට අවසානය දක්වා ප්‍රතිදානය කරයි.
print(s[4:])
print(s[:]) # whole string
# 4 නොමැතිව, 4 ස්ථානයේ දක්වා ප්‍රතිදානය කරයි.
print(s[:4])
# සම්පූර්ණ නම කොටස් දෙකකට බෙදේ
print(s[0:1])
```

isnumeric(x) - x හි අගය සංඛ්‍යාවක් නම් සත්‍ය වේ.
isdigit(y) - y හි අංක පමණක් අත්‍යන්ති සත්‍ය වේ.
isalpha(z) - z හි අකුරු පමණක් අත්‍යන්ති සත්‍ය වේ.

ප්‍රතිදාන විස්තර කරන්න

```
str1 = "this2016";  
str2 = "23443434";  
print (str1.isnumeric())  
print (str2.isnumeric())  
print (str1.isdigit())  
print (str2.isdigit())  
print (str1.isalpha())  
print (str2.isalpha())
```

split(str) - වචනයක් කැඩීම් වලට කැඩීම

```
x = 'blue,red,green'
```

y=x.split(",") # අංශ කොමාවෙන් වෙන්කර y අරුව සාදයි

පහත ප්‍රතිදාන විස්තර කරන්න

```
print (y)  
print (y[0])  
a,b,c=x.split(",")  
print(a)  
print(b,c)  
words = "This is some text for python"  
# හිඟින් වලින් කැඩීම  
words2 = words.split(" ")  
print(words2)
```

1.9 join(seq) - ලැයිස්තුවක අවයව එක්කිරීම

```
s = "";  
seq = ("a", "b", "c"); # වචන අරුවකි  
print s.join( seq )  
# ප්‍රතිදාන a*b*c වේ. අවයව අතරට s යොදා ඇත.
```

```

s = " ";
seq = ("a", "b", "c"); # වෙන අරුවකි
print s.join( seq )
ප්‍රතිදානය කුමක් විය හැකිද?
words = "This is some text"
# ජිනිතයක් අනුව කැඩීම
words2 = words.split(" ")
print(words2)

```

වෙනසක අති කණයක් වෙනුවට වෙනත් කණයක් ආදේශනය.
 replace(oldstr, newstr)

```

find(str) - වෙනසක යම් කණයක් අනිදැයි බැලීම
str = "Kandy is a cool town";
str1=str.replace("Kandy", "Jaffna")
#ප්‍රතිදානය Jaffna is a cool town වේ
print (str1)
str2=str.replace("Kandy", "Candy")
str3=str2.replace("cool", "tasty")
str4=str3.replace("town", "food")
print (str4) # ප්‍රතිදානය කුමක් විය හැකිද?

```

1.10 සෙවීම

```

str1 = "Google is a search engine!";
str2 = "search";
print str1.find(str2)
# කණය හමුවුවහොත් ආරම්භක ස්ථානය ලබාදේ
# සෙවුම අසාර්ථක වුවොත් -1 ලැබේ
print str1.find("Yahoo")
ප්‍රතිදානය කුමක් විය හැකිද?
print str1.find("a")

```

```
# "a" සෙවීම අරම්භ කළ යුත්තේ 16 වන ස්ථානයේ සිටය .
print str1.find("a",16)
# "e" සෙවීම නැවතිය යුත්තේ 20 නි.
print str1.find("e",16,20)
```

2.0 ලැයිස්තු නව දුරටත්

```
myList=[1,2,3,4,5,6] # අරාමක් ලැයිස්තුවකි.
for x in myList:
    print x
```

```
n=len(myList) # ලැයිස්තුවේ අවසාන ගණන
for i in range(n):
    print myList[i]
```

2.1 ලැයිස්තුවක් තුළ ලැයිස්තුවක් සෑදීම

ලැයිස්තුවක අවසානයක් තවත් ලැයිස්තුවක් විය හැක.
ඔබට ශ්‍රී ලංකාවේ සියලුම සිසුන්, දිස්ත්‍රික්ක, පාසල්, ශ්‍රේණි හා නම්
අනුව ලැයිස්තු ගතකිරීමට ලැයිස්තු ව්‍යුහයක් නිර්මාණය කරන්නේ
කෙසේද?

මුලින්ම ඔබට දිස්ත්‍රික්ක ලැයිස්තුවක් අවශ්‍ය වේ.
ඒ ඒ දිස්ත්‍රික්ක ලැයිස්තු තුළ පාසල් ලැයිස්තු අවශ්‍ය වේ.
ඒ ඒ පාසල් තුළ ශ්‍රේණි ලැයිස්තු අවශ්‍ය වේ
ඒ ඒ ශ්‍රේණි තුළ සිසුන්ගේ නම් ලැයිස්තු අවශ්‍ය වේ.

```
# එම ව්‍යුහය පහත අංකාරයෙන් සැලසුම් කළ හැක.
studentList=[district1 [school1 [class1[std1, std2,... ]]], ...]
අවබෝධ කරගැනීමට ඔබගේ පංතියේ ලමුන් නිදෙනකුගේ
ලැයිස්තුවක් සකසන්න
```

2.2 පයිතන් ලැයිස්තු ක්‍රම

append(object) - ලැයිස්තුවේ අගට අවයවයක් එක් කරයි.
 insert(index, object) - ස්ථානයකට අවයවයක් අතුල් කරයි.
 pop ([index])- ස්ථානයක ඇති අවයවයක් ඉවත් කර ලබාදේ
 remove (v) - පළමුවෙන් හමුවන v නම් අගය ඉවත් කරයි
 reverse() - අවයව ස්ථානයන් කණපිට හරවයි
 sort () - ලැයිස්තුවක අවයව අනුක්‍රමණය කරයි
 count (v) - ලැයිස්තුවක v අගයන් ගණන් කරයි .
 index (v) - ලැයිස්තුවේ v පළමුවෙන් හමුවන ස්ථානය ලබාදේ
 extend (list) - ලැයිස්තුවක් තවත් ලැයිස්තුවකට එකතු කරයි.

```

myList = [123, 'Niranjan', 8.2]
myList.append( 2016 )
print (myList)

```

```

# ප්‍රතිදානය වන්නේ [123,'Niranjan',8.2,2016]

```

```

myList=[] # හිස් ලැයිස්තුවක් සෑදීම
for i in range(10):
    myList.append(i)
print (myList)

```

```

myList = [1,2,3]
myList.insert(0,6) # 6 අංකය 0 ස්ථානයට අතුල් කරනවා
print (myList)

```

```

#ප්‍රතිදානය [6, 1, 2, 3] වේ

```

```

# ලැයිස්තුවේ අන්තිම අගය ඉවත් කර එය ප්‍රතිදානය කිරීම
n=myList.pop() ;
print (n) # ප්‍රතිදානය 3 වේ
print (myList) # ප්‍රතිදානය [6, 1, 2] වේ

```

```
#index 2 (3rd) හි ඇති අගය ඉවත් කර එය ප්‍රතිදානය කිරීම
n1=myList.pop(2);
print (n1) # ප්‍රතිදානය 2 වේ
print (myList) #ප්‍රතිදානය [6, 1] වේ
#ප්‍රතිදාන ලැබෙන ආකාරය අධ්‍යයනය කරන්න
```

2.3 අවයව ඉවත් කිරීම

```
remove (value) - v ට සමාන මූලික අගය ඉවත් කරයි.
myList = [1,2,3,4,5,"Sam", "Niranjan", "Sam"]
myList.remove(4);
print(myList)
ප්‍රතිදානය [1, 2, 3, 5, 'Sam', 'Niranjan', 'Sam'] වේ
```

```
myList.remove('Sam');
print(myList)
ප්‍රතිදානය [1, 2, 3, 5, 'Niranjan', 'Sam'] වන්නේ ඇයි ?
```

2.4 අවයව සැකසීම

```
sort () - ලයිස්තුවේ අගයන් පිළිවෙලට සකසයි
reverse() - ලයිස්තුවේ ඇති අගයන්වල පිළිවෙල කෙසිට හරවයි.
count (v) - v අගය ලයිස්තුවේ තිබෙන වාර ගණන ඉදිරිපත් කරයි.
myList = [1,2,3,4,5,"Sam"]
myList.reverse();
print(myList)
#ප්‍රතිදානය ['Sam', 5, 4, 3, 2, 1] වේ
myList.sort();
print(myList)
#ප්‍රතිදානය [1, 2, 3, 4, 5, 'Sam'] වේ
myList.append("Sam")
print(myList)
n=myList.count("Sam")
print(n) # ප්‍රතිදානය කුමක්ද?
```



කමරි විෂය
කමරි වෙලාවක
කමරි තැනක
නිදහසේ ඉගෙන ගන්න
පාඩම් සහ ප්‍රශ්න
Shilpa64.lk



AlgoHack කුඩා අවදියේදී ප්‍රමුඛව පරිගණක විද්‍යාව සහ ක්‍රම ලේඛනය
ඉගැන්වීමට ගිල්ප සපුරා කල අත්මකැයි. එයට Google for Education සහ ශ්‍රී
ලංකා පරිගණක සංගමය සහය ලබා දේ.

AlgoHack : පයිතන් #3



ශ්‍රීත හා ලැයිස්තු