

AlgoHack : පයිතන් #1



අදාන - ප්‍රතිදාන - විචල්‍ය - දත්ත - ගණිත කාර්ම

කතුවරු
නිරංජන් මීගම්මන, නවින් නමියි

සංස්කරණය
විශ්ව කුමාර



AlgoHack කුඩා අවදියේදී ළමුන්ට පරිගණක විද්‍යාව සහ ක්‍රම ලේඛනය ඉගැන්වීමට ශිල්ප සපුරා කල අරමුණකි. 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 (www.shilpasayura.org)





පයිතන් ඉතාම සරළ එහෙත් දෘඪ ක්‍රමලේඛනය සඳහා
සෞදාගත හැකි පරිගණක කේතන භාෂාවකි. එය 1991 දී
[Guido van Rossum](#) විසින් නිර්මාණය කර විවෘත
මුද්‍රාණාංගයක් ලෙස නිදහස් කරන ලදී.

පයිතන් කේතනය සඳහා අපි python IDLE භාවිතා කරමු.
පයිතන් IDLE [python.org](#) වෙතින් ලබාගත ඔබේ
පරිගණකයේ ස්ථාපනය කරගන්න. නැතහොත්
අන්තර්ජාලයේ ක්‍රමලේඛන අඩවියක් ([tutorialspoint.com](#))
භාවිතා කරන්න.

QPython පයිතන් කේතනය සඳහා වූ ජංගම දුරකථන
අංගයකි. එය [play.google.com](#) වෙතින් ලබාගත හැක.

[Python ගැන](#) □□ □□□□□□ □□□□□□□□ □□ □□□
පයිතන් ඉගැනුම් මුද්‍රාණාංගයකි.

<http://shilpasayura.com/dev/python/>

1.0 අදාන හා ප්‍රතිදාන (Input & Output)

පසිනන් අදාන හා ප්‍රතිදාන සිදුකිරීමි පහත පරිදි වේ.

```
name = input("Enter Your Name ")
print ("Your name is : ")
print (name)
```

\n යෙදීම මගින් අලුත් පේලියකට මුද්‍රණය යොමු කෙරේ.

```
age = input("Enter Your Age\n")
print ("Your age is: ")
print (age)
```

input() ශ්‍රිතය මගින් අංක, අකුරු වචන අනුලක්කල හැක.
a=b+c සමීකරණය සලකමු.

```
number1 = input( ) # enter 1
number2= input( ) # enter 2
number3=number1+ number2
print (number3)
```

මෙහි ප්‍රතිදානය **1+2=12** වෙන්නේ ඇයි?

දැන් 1.5 හා 2.7 අදාන වශයෙන් ලබාදී බලන්න.

ප්‍රතිදානය 1.52.7 වෙන්නේ ඇයි?

පසිනන් සියලු ප්‍රතිදානයන් වචන ලෙස හදුනා ගනී.

කේතනයේදී වචන හා අංක යනු වෙනස් දත්ත වර්ග වේ.

ඒ ඇයි?

හැකුරු සම ද්‍රව්‍යයකි ඇණි දියරයකි

මාන වෙනස්ය. හැසිරීම වෙනස්ය. එකිනෙකට හැරවිය හැකිය.

සංඛ්‍යා වචන (string) ලෙස ආදානය වුවත්, ආදානයෙන් පසුව එය සංඛ්‍යාවකට හරවා භාවිතයෙන් ඉහත ගැටළුව විසඳා ගත හැක.

```
number1 = input() # enter 1
n1=int(number1)
number2 = input() # enter 2
n2=int(number2)
n3= n1+ n2
print (n3)
```

ප්‍රතිදානය 3 වීමට පෙන්නුව කුමක්ද?
මෙහිදී int() ශ්‍රිතය ආදාන වචනය සංඛ්‍යාවකට හරවයි.

2.0 විචල්‍ය (Variables)

ක්‍රමලේඛන ක්‍රියාත්මක වන්නේ දත්ත මගිනි.
අගය වෙනස් වන දත්ත වලට විචල්‍ය කියවු.
ඈම විචල්‍යයකටම නාමයක් ඇත.
මෙම විචල්‍යයන් තුල අපි දත්ත ගබඩා කරමු.
විචල්‍යයන් හි සංඛ්‍යා, වචන, ලයිස්තු ගබඩා කළ හැකිය.
විචල්‍ය ක්‍රමලේඛන ක්‍රියාවලි සඳහා ලබාදිය හැක.

2.1 විචල්‍යයකට දත්ත අනුයුක්ත කිරීම (Assignment)

මේ සඳහා සමාන ලකුණ (=) භාවිතා කරයි
name="Ruvini" #
age=18
height=5.6

විචල්‍යකට දත්ත වෙනස් කිරීමටද සමාන ලකුණ (=) භාවිතා කරයි.

name="Sindi" # දැන් name විචල්‍යයේ අගය "Sindi" වේ.

age=19 # දැන් age විචල්‍යයේ අගය 19 වේ.

height=5.7 # දැන් height විචල්‍යයේ අගය 5.7 වේ.

2.1 බූලීය විචල්‍ය (Logical Variables)

බූලීය විදුලි බුබුලකට සවිකළ ස්ථාවයක් වැනිය

විදුලි බුබුල දැල්වෙන අවස්ථාව සත්‍ය (True) ලෙසද

එය නිවැරදි අවස්ථාව අසත්‍ය (False) ලෙසද ගත හැකිය.

Python හි සත්‍ය = 1 ලෙසත්, අසත්‍ය = 0 ලෙසත් භාවිතා වේ.

ප්‍රතිදානයේදී සත්‍ය (True) හා අසත්‍ය (False) ලෙස ලබාදේ.

බූලීය ප්‍රකාශණ සත්‍ය හෝ අසත්‍ය අගයක් ප්‍රතිදානය කරයි.

e=c < d ප්‍රකාශණයේ

c=3 d= 4 වූ විට e නම් ප්‍රතිදානය සත්‍ය වේ.

c=4 d= 3 වූ විට e නම් ප්‍රතිදානය අසත්‍ය වේ.

2.3 විචල්‍යයන් නම් කිරීමේ නීති රීති (Variable Names)

විචල්‍ය සඳහා සීමිත මෙතම කැපිටල් අකුරුද භාවිතා වේ.

විචල්‍ය නාම මුල් අකුර ඉංග්‍රීසි අකුරක් හෝ `_` සලකුණ විය යුතුය.

ඉතිරිය සඳහා ඉංග්‍රීසි අකුර, `_` සලකුණ හා සංඛ්‍යා යෙදිය හැකිය.

විචල්‍යය නාමයේ රටාව කුමලේඛනය සඳහා බලන්න.

“name” හා “Name” යනු විචල්‍යයන් දෙකකි.

name, Name, nAme, NaMe, nameE යනු විචල්‍ය 4 කි.

විචල්‍ය නීතියටත් යටත් විචල්‍ය නම්

i, `_my_name`, `name__23` , `a1b2_c3` and `r_r_c23`

නීති රීති වලට යටත් නොවන විචල්‍ය නම්
2things, *name, age+ වේ.

නොද විචල්‍ය නම් භාවිතා කිරීම කේතනයේදී වැදගත් වේ!
විචල්‍යයේ අගය විස්තර වන පරිදි නාමකරණය වඩාත්
සුදුසුය.

වඩා සුදුසු විචල්‍ය නාමය කුමක්ද?
numberOfCats, numbcats, ncats, num_cats

තෝරුම් ගැනීමට අමාරු විචල්‍ය නාම
ir = 0.12
b = 12123.34
i = ir * b

තෝරුම් ගැනීමට පහසු විචල්‍ය නාම
interest_rate = 0.12
account_balance = 12123.34
interest_amount = interest_rate * account_balance

විචල්‍යකට අගයක් අනුයුක්ත කළ පසුව නැවත වෙනස් කරන
තුර එම අගයම පවතී.

3.0 පයිතන්ගේ දත්ත වර්ග (Data Types)

විචල්‍යයන් තුළ විවිධ ආකාරයේ දත්ත ගබඩා කළ හැකිය.
එම දත්ත වර්ග Data Types ලෙස හදුන්වයි.

පරිගණක මතකයේ වයස සංඛ්‍යාමය අගයක් ලෙස ගබඩා
කරයි..

නමක් ගබඩා කරනු ලබන්නේ අකුරු සමූහයක් ලෙසය.

ලිපිනයක සංඛ්‍යාමය හා අකුරු එකතුවකි.

විචල්‍යවල දශම අගයන් තිබිය හැක.

පයිතන් කේතනයේදී දත්ත වර්ග 5 ක් භාවිතා වේ.

Numbers - සංඛ්‍යාමය දත්තයන්

String - අකුරු දත්තයන්

List - වෙනස්කල හැකි දත්ත ලැයිස්තුව

Tuple - වෙනස්කල නොහැකි දත්ත ලැයිස්තුව

Dictionary - ශබ්ද කෝෂ

3.1 සංඛ්‍යාමය දත්ත වර්ග

සංඛ්‍යාමය දත්ත වර්ගය තුළ ගබඩා කළ හැක්කේ ඉලක්කම් පමණි. ඒවා පූර්ණ හෝ දශම සංඛ්‍යා විය හැක.

```
var1 = 1
```

```
var2 = 10
```

```
var3=1.1
```

සංඛ්‍යා දත්ත තවත් උප දත්ත වර්ග වලට බෙදේ.

int : (පූර්ණ සංඛ්‍යා) : 7, -1, -240,

long (විශාල පූර්ණ සංඛ්‍යා) : 51924361L

float (දශම ස්ථාන සහිත සංඛ්‍යා) : 3.147

3.2 පයිතන් වචන (Strings)

"Kandy", 'Jaffna' වැනි පයිතන් වචන Strings ලෙස හඳුන්වයි.

ඒවා අර්ථ දැක්වීමට උඩු කොමා භාවිතා කරයි.

```
name="Navin"
```

(+) ලකුණ මගින් වචන එකතු කල හැකිය.

(*) ලකුණ වචනයක් පුනරාවර්තනය කරයි.

```
str = 'Hello Python!'
```

```
print str      # සම්පූර්ණ වචනය ප්‍රතිදානය කරයි
```

```
print str + "T" # එකතු කල වචනප්‍රතිදානය කරයි
```

```
print str * 2   # වචනය දෙවරක් ප්‍රතිදානය කරයි
```

```
print str[0]    # වචනයේ පළමු අකුර ප්‍රතිදානය කරයි
```

```
print str[2:5] # 3 සිට 5 වන අකුර දක්වා ප්‍රතිදානය කරයි
print str[2:] #3 වන අකුරේ සිට ඉතිරිය ප්‍රතිදානය කරයි
```

ඉහත කේතනයේ ප්‍රතිදානය බලන්න.

Hello python!

Hello python!T

Hello python!Hello python!

H

llo

llo python!

3.3 දත්ත වර්ග පරිවර්තනය (Data Type Conversion)

අපට දත්තයක් වෙනත් වර්ගයකට පරිවර්තනය කළ හැකිය.

මේ සඳහා දත්ත පරිවර්තන ශ්‍රිත භාවිතා කරයි

int(x) - x හි අගය සංඛ්‍යාවකට හරවයි

long(x) - x හි අගය දිගු සංඛ්‍යාවකට හරවයි

float(x) - x හි අගය දශම සංඛ්‍යාවකට හරවයි

str(x) - x වචනයක බවට හරවයි

chr(x) - x හි අගය අකුරක් බවට හරවයි

ord(x)- x අකුර එහි සංඛ්‍යාත්මක අගයට හරවයි

hex(x) - x සංඛ්‍යාව ඡායීදශමයට (hexadecimal) හරවයි

oct(x) - සංඛ්‍යාව අෂ්ටමය (octal) ගණයට හරවයි

unichr(x) - සංඛ්‍යාව යුනිකෝඩ් අක්ෂරයකට හරවයි.

3.4 පයිතන් අරුවන් (Arrays)

දත්ත සමූහයක් සහිත විචල්‍යක් අරුවක් වේ.

පයිතන් තුළ ඇති ලැයිස්තු (list) අරුවක් ලෙස භාවිතා වේ.

අරුවක විවිධ අගයන් සහිත අවයව ගණනක් තිබිය හැකිය.

පාඨයක සිසුන්ගේ නම් ලැයිස්තුව අරුවකි.

අගයන් n සංඛ්‍යාවක් ඇති අරාවක පළමු අවයව ස්ථානය 0 වේ.

අවසන් අවයවයේ ස්ථානය $n-1$ වේ.

myList අරාවේ අවයව හඳුනා ගන්න.

```
myList=[1,2,3,4,5,6]
```

මෙම ප්‍රකාශණයන් වල ප්‍රතිදානය කුමක්ද?

```
print(myList[0])
```

```
print(myList[5])
```

```
print(myList[6])
```

```
myList[0]=100
```

```
print(myList[0])
```

```
myList[1]="Navin"
```

```
print(myList[1])
```

3.4 පයිතන් ලයිස්තු (lists)

පයිතන් ලයිස්තුවක දත්ත කොමා මගින් වෙන් කරමු.

පයිතන් ලයිස්තුවක විවිධ වර්ගයේ දත්ත නිකිය හැක.

පහත ප්‍රතිදානයන් කුමක් විය හැකිද?

```
list1 = [1,23, 'Ruwini',7]
```

```
print list1      # සම්පූර්ණ ලයිස්තුව ප්‍රතිදානය කරයි
```

```
print list1[0]   # පළමු දත්තය ප්‍රතිදානය කරයි
```

```
print list1[1:3] # 2 සිට 3 වන දත්තය ප්‍රතිදානය කරයි
```

```
print list[2:]   # 3 වන දත්තයේ සිට ප්‍රතිදානය කරයි
```

3.5 පයිතන් ටියුපල් (Tuples)

ටියුපල් පයිතන් ලයිස්තුවලට සමාන වේ.

පයිතන් ලයිස්තු අර්ථ දැක්වීමට කොටු වරහන් [] යොදාගනී.

ටියුපල් අර්ථ දැක්වීම සඳහා සඳහා වරහන් () යොදාගනී.

ලැයිස්තුවල අවයව ප්‍රමාණය හා අගයන් වෙනස් කළ හැකිය.
ටිප්පල අවයව හා අගයන් වෙනස් කළ හැකිය.

පහත ප්‍රතිදානයන් කුමක් විය හැකිද?

```
tuple1 = (123, 'Niki', "Sugi",9)
print tuple1
print tuple1[0]
print tuple1[1:3]
print tuple1[2:]
```

3.6 පයිතන් ශබ්දකෝෂ (Dictionaries)

පයිතන් ශබ්දකෝෂ විශේෂ අරුවකි.

සෑම අවයවයකම යතුර-අගය (key-value) යුගලයක් පවතී.
අගයන් ඕනෑම දත්ත වර්ගයක් විය හැක.

ශබ්දකෝෂ අර්ථ දැක්වීමට සහල වරහන් { } භාවිතා කරයි.

අවයව හඳුනා ගැනීමට කොටු වරහන් [] භාවිතා කරයි.

හිස් ශබ්ද කෝෂයක් අර්ථ දැක්වී

```
dict1 = { }
```

'one' නැමති යතුරට "ONE" අගය අනුයුක්ත කරයි

```
dict1['one'] = "ONE"
```

'two' යතුරට "TWO" අගය අනුයුක්ත කරයි

```
dict1['2'] = "TWO"
```

ප්‍රතිදානයන් කුමක් විය හැකිද?

```
print dict1['one'] # 'one' යතුරේ අගය ප්‍රතිදානය කරයි
print dict1[2] # '2' යතුරේ අගය ප්‍රතිදානය කරයි
print dict1 # ශබ්දකෝෂයේ සියලු අවයව ප්‍රතිදානය කරයි
print dict1.keys() # යතුරු සියල්ල ප්‍රතිදානය කරයි
```

```
print dict1.values() # අගයන් සියල්ල ප්‍රතිදානය කරයි
```

```
පිළිතුරු
```

```
One
```

```
Two
```

```
{2: 'Two', 'one': 'One'}
```

```
[2, 'one']
```

```
['Two', 'One']
```

4.0 පයිතන් ගණිත කාර්ම

පයිතන් ගණිත කාර්ම ☐☐☐☐ ☐☐☐☐☐ ☐☐ ☐☐☐☐

රැසකි.

4.1 විෂ ගණිත කාර්ම

```
a=2+3 # සංඛ්‍යා එකතු කිරීම
```

```
b=9-2 # සංඛ්‍යා අඩු කිරීම
```

```
c= 108 * 0.5 # සංඛ්‍යා ගුණ කිරීම
```

```
c= 6/2 # සංඛ්‍යා බෙදීම
```

```
eight = 2 ** 3 # දර්ශක ශ්‍රිත
```

```
x=10 හා y=2 නම්
```

ඔබ x හා y අගයන් භාවිතයෙන් 100 ලබාගන්නේ කෙසේද?

```
z=x**y # එනම් xy වේ
```

4.2 අනුයුක්ත කාර්ම (Assignment)

අප මෙහිදී විචලක අර්ථ දැක්වූ ඊට අගයක් පවරමු.

```
c = a + b # a+c හි අගය c ට අනුයුක්ත කෙරේ  
කෙටි කාර්ම
```

```
c += a # c = c + a යන්න අදහස් කරයි
```

```
c -= a # c = c - a යන්න අදහස් කරයි
```

`c *= a # c = c * a යන්න අදහස් කරයි`

`c=1`

`c+=3`

`print(c) # ප්‍රතිදානය 4 වේ`

`c*=3`

`print(c) # ප්‍රතිදානය 12 වේ`

ප්‍රතිදානය කුමක් විය හැකිද?

`x = 4`

`y = x + 1`

`x = 2`

`print (x, y)`

ප්‍රතිදානය කුමක් විය හැකිද?

`a, b = 2, 3`

`print (a, b)`

`b, c = a, b + 1`

`print (a, b,c)`

4.2 බෙදීමේ ඉතිරිය (Modulo)

බෙදීමකදී ඉතිරිවන අගය ලෙස **Modulo** අර්ථ දැක්වේ.

`print (1% 3) # 1 ප්‍රතිදානය කරයි`

`print (2% 3) # 2 ප්‍රතිදානය කරයි`

`print (3% 3) # 0 ප්‍රතිදානය කරයි`

`print (4% 3) # 1 ප්‍රතිදානය කරයි`

4.3 කුඩා පූර්ණ බෙදීම (Floor Division)

බෙදීමේ පිළිතුර කුඩාම පූර්ණ අගයට වටයයි.

`print (9//2) # 4 ප්‍රතිදානය කරයි`

`print (-11//3) # -4 ප්‍රතිදානය කරයි`

4.4 සංඛ්‍යා සංසන්දනය (Comparison)

A හා B සංඛ්‍යා විචල්‍ය නම්

$A > B$ A අගය B ට වඩා විශාල බව අදහස් කරයි

$A < B$ A අගය B ට වඩා කුඩා බව අදහස් කරයි

$A == B$ A අගය B ට සමාන බව අදහස් කරයි

$A <= B$ A අගය B ට වඩා කුඩා හෝ සමාන බව අදහස් කරයි

$A >= B$ A අගය B ට වඩා විශාල හෝ සමාන බව අදහස් කරයි

$A != B$ A අගය B ට සමාන නොවන බව අදහස් කරයි

සංඛ්‍යා සංසන්දනයේදී ලැබෙන ප්‍රතිදානය බූලියානු අගයකි.
එනම් ප්‍රතිදාන අගය සත්‍ය හෝ අසත්‍ය වේ. මේ ප්‍රතිදාන
සලකන්න

```
print (7 > 10) # අසත්‍ය වේ
```

```
print (4 < 16) # සත්‍ය වේ
```

```
print (4 == 4) # සත්‍ය වේ
```

```
print (4 <= 4) # සත්‍ය වේ
```

```
print (4 >= 4) # සත්‍ය වේ
```

```
print (4 != 4) # අසත්‍ය වේ
```

මේ ප්‍රතිදාන අගයන්න.

```
print (7 > 10)
```

```
print (4 < 16)
```

```
print (4 == 4)
```

```
print (4 <= 4)
```

```
print (4 >= 4)
```

```
print (4 != 4)
```

අගය සත්‍යය ද? අසත්‍යයද?

```
number = 10
```

```
isPositive = (number > 0)
```

print (isPositive)

4.4 තාර්කික කාර්ම (logical operators)

and සහ: කොළඹ සහ නුවර යමු.

or හෝ: කොළඹ හෝ නුවර යමු

not නොවේ: කොළඹ නොවේ නුවර යමු.

බුලීය ප්‍රකාශණ ගොඩනැගීමට and, or, not භාවිතා කරයි.

බුලීය ප්‍රකාශණ වම් පස සිට ඇගයේ.

එක් බුලීය ප්‍රකාශනයක් කුඩා බුලීය පද ගණනක සංයුක්ත ප්‍රතිදානය මත රඳා පවතිය හැකිය. සෑම බුලීය පදයක්ම වරහන් තුළ දැක්වීම කරයි.

And භාවිතා වන මේ ප්‍රකාශණයේ x හි අගය 0 ට වැඩි හා 10 ට අඩු නම් පමණක් ප්‍රතිදානය සත්‍ය වේ

(x > 0) and (x < 10)

ප්‍රකාශණ එකක් සත්‍ය වේනම් ප්‍රතිදානය සත්‍ය වේ.

මේ ප්‍රකාශණය සත්‍ය වන්නේ n නම් අගය 2 න් හෝ 3 න් බෙදිය හැකිනම් පමණි

(n % 2 == 0) or (n % 3 == 0)

n අගය 2, 3, 4, 5 වන විට කුමක් වේද?

not මගින් බුලීය ප්‍රකාශණයක අගය කණපිට හැරවිය හැකිය.

එනම් සත්‍ය not මගින් අසත්‍ය කෙරේ.

true=not(false)

පහත ප්‍රකාශණයේ x හි අගය y ට වඩා කුඩා හෝ සමාන නම් පමණක් සත්‍ය වේ.

not (4 > 3) # අසත්‍ය

4.5 කෙටි අගයීම

or භාවිතා වන ප්‍රකාශණ වල විට වම් පසින් අඟි පදය මුලින්ම අගයයි. එහිදී ප්‍රතිපලය සත්‍ය වේනම් අනෙක් පද නොසලකා හරී.

$a=((7 < 9) \text{ or } (3 < 7))$

and භාවිතා වන ප්‍රකාශණ වල පලමු පදය අසත්‍ය වේනම් අනෙක් පද නොසලකා හරී.

$a=((7 < 6) \text{ and } (3 < 7))$

පහත බූලීය ප්‍රකාශණ වල ප්‍රතිදානය අගයන්න.

$b=((6 < 5) \text{ and } (5 > 4))$

$c=((((3 < 5) \text{ and } (2 > 1)) \text{ or } ((5 < 8) \text{ and } (2 > 1))))$

$d=\text{not}(3>2)$

$c=((((3 < 5) \text{ and } (2 > 1)) \text{ or } ((5 >= 8) \text{ and } (2 > 1)) \text{ or } (4==4)))$



කමෙහි විෂය
කමෙහි වෙලාවක
කමෙහි තැනක
නිදහසේ ඉගෙන ගන්න
පාඩම් සහ ප්‍රශ්න

Shilpa64.lk



AlgoHack කුඩා අවදියේදී ළමුන්ට පරිගණක විද්‍යාව සහ ක්‍රම ලේඛනය ඉගැන්වීමට ගිල්ප සයුර කළ ආරම්භයකි. එයට Google for Education සහ ශ්‍රී ලංකා පරිගණක සංගමය සහය ලබා දේ

AlgoHack : පයිතන් #1



ආදාන - ප්‍රතිදාන - විචල්‍ය - දත්ත - ගණිත කාර්ම