
variable

this notes is made by- [Rehyan yadav](#)

1.A variable is considered as tag that is tied to some value. python

considered value as a objects.

2.variable values can be changed.

in python if the values of the variable is same thn it prints

the value same so, it is very useful for us beacuse it saves time and

memory.

**3)for find the locationa nd size of variable type
id(variable_name)**

Rules

1) Every variable should start with a letter or underscore(_)

2) No spaces are allowed in variable declaration.

3) Except underscore(_) no other special symbols are allowed in the middle of the variable declaration.

4) A variable is written with a combination of letters, numbers and special character _ (underscore).

5) No reserved keywords.

example-

do

don't

1) A

1) and

2) a

2) if

3) name

3) 15name

4) name15

4) #city

5) _city

5) full\$name

6) full_name

6) full name

7) fullname

for example - A = 10

**A is box in which 10 is stored,
10 is store in A**

**the memory is generated for a
values of the variable can be changed.**

question 1) what is the maximum possible of an integers in python?

answer: code

#a python code to demostated that wee can store

#large number in python

```
x= 100000000000000000000000000000000;
```

x = x + 1

```
print(x)
```

question 2) how to find the type?

answer: code

x= 10

```
print(type(x))
```

Types of conversion in python

there are two types of conversion:

1)Implicit types conversion 2)explit types conversion

***Implicts type conversion:it automatically converts one data types to another without any user involent**

code:

```
x = 10
```

```
print("x is of type:",type(x))
```

```
y = 10.6
```

```
print("y is of type:",type(y))
```

```
x = x + y
```

```
print(x)
```

```
print("x is of type:",type(x))
```

***Explicit Type Conversion:the data type is manually changed by the user as per their requirement.**

1. int(a, base): This function converts any data type to integer. 'Base' specifies the base in which string is if the data type is a string.

2. float(): This function is used to convert any data type to a floating-point number

code:

```
# Python code to demonstrate Type conversion
```

```
# using int(), float()
```

```
# initializing string
```

```
s = "10010"
```

```
# printing string converting to int base 2
```

```
c = int(s,2)
```

```
print ("After converting to integer base 2 : ", end="")
```

```
print (c)
```

```
# printing string converting to float
```

```
e = float(s)
```

```
print ("After converting to float : ", end="")
```

```
print (e)
```

3. ord() : This function is used to convert a character to integer.

4. hex() : This function is to convert integer to hexadecimal string.

5. oct() : This function is to convert integer to octal string.

code:

```
# Python code to demonstrate Type conversion
```

```
# using ord(), hex(), oct()
```

```
# initializing integer
```

```
s = '4'
```

```
# printing character converting to integer
```

```
c = ord(s)
```

```
print ("After converting character to integer : ",end="")
```

```
print (c)
```

```
# printing integer converting to hexadecimal string
```

```
c = hex(56)
```

```
print ("After converting 56 to hexadecimal string : ",end="")
```

```
print (c)
```

```
# printing integer converting to octal string
```

```
c = oct(56)
```

```
print ("After converting 56 to octal string : ",end="")
```

```
print (c)
```

6. tuple() : This function is used to convert to a tuple.

7. set() : This function returns the type after converting to set.

8. list() : This function is used to convert any data type to a list type.

code:

```
# Python code to demonstrate Type conversion
```

```
# using tuple(), set(), list()
```

```
# initializing string
```

```
s = 'geeks'
```

```
# printing string converting to tuple
```

```
c = tuple(s)
```

```
print ("After converting string to tuple : ",end="")
```

```
print (c)
```

```
# printing string converting to set
```



```
c = set(s)
```

```
print ("After converting string to set : ",end="")
```

```
print (c)
```

printing string converting to list

```
c = list(s)
```

```
print ("After converting string to list : ",end="")
```

```
print (c)
```

9. dict() : This function is used to convert a tuple of order (key,value) into a dictionary.

10. str() : Used to convert integer into a string.

11. complex(real,imag) : This function converts real numbers to complex(real,imag) number.

code:

```
# Python code to demonstrate Type conversion
```

using dict(), complex(), str()

initializing integers

a = 1

b = 2

initializing tuple

tup = (('a', 1), ('f', 2), ('g', 3))

printing integer converting to complex number

c = complex(1,2)

print ("After converting integer to complex number : ",end="")

print (c)

printing integer converting to string

c = str(a)

print ("After converting integer to string : ",end="")

print (c)

printing tuple converting to expression dictionary

c = dict(tup)

print ("After converting tuple to dictionary : ",end="")

print (c)

12. chr(number) : : This function converts number to its corresponding ASCII character.

code:

Convert ASCII value to characters

a = chr(76)

b = chr(77)

print(a)

print(b)

Bytes object vs String

***bytes**

1)Byte objects are sequence of Bytes, whereas Strings are sequence of characters.

2)Byte objects are in machine readable form internally, Strings are only in human readable form.

3)Since Byte objects are machine readable, they can be directly stored on the disk. Whereas, Strings need encoding before which they can be stored on disk.

Swap variable

Swap two variables in one line.

Python program to swap two variables in a single line

x = 5

y = 10

x, y = y, x

print("After Swapping values of x and y are", x, y)

Alternate Solutions :

C++ also provides a library function swap()

b = (a + b) – (a = b); [Thanks to Rajat Mishra for this]

a += b – (b = a); [Thanks to Zoran Davidovi? for this]

a = a * b / (b = a)[Thanks to kongasricharan for this]

Private Variables in Python

1)Underscore in Python

code:

Python code to illustrate how mangling works

class Map:

def __init__(self, iterate):

self.list = []

self.__geek(iterate)

def geek(self, iterate):

for item in iterate:

self.list.append(item)

private copy of original geek() method

__geek = geek

```
class MapSubclass(Map):
```

```
    # provides new signature for geek() but
```

```
    # does not break __init__()
```

```
    def geek(self, key, value):
```

```
        for i in zip(keys, value):
```

```
            self.list.append(i)
```

2)_Single Leading Underscores

code:

```
# Python code to illustrate
```

```
# how single underscore works
```

```
def _get_errors(self):
```

```
    if self._errors is None:
```

```
        self.full_clean()
```

```
    return self._errors
```

```
errors = property(_get_errors)
```

3) __Double Leading Underscores

code:

Python code to illustrate how double

underscore at the beginning works

class Geek:

def _single_method(self):

pass

def __double_method(self): # for mangling

pass

class Pyth(Geek):

def __double_method(self): # for mangling

pass

4) __Double leading and Double trailing underscores__

code:

Python code to illustrate double leading and

double trailing underscore works

class Geek:

'__init__' for initializing, this is a

special method

```
def __init__(self, ab):
```

```
    self.ab = ab
```

custom special method. try not to use it

```
def __custom__(self):
```

```
    pass
```

this notes is made by- [Rehyan yadav](#)

contact number - 8102448912

this is gmail- ritulyadav1984@gmail.com

this number is for asking any doubt.....

**if you want to add more information ask
me or contact me.**

then i will provide you the .txt file or .rtf file