

Day4_Datatypes in Python

Categories of Datatypes

1. **Numeric Category**
 - int
 - float
 - complex
2. **Boolean Category**
 - boolean
3. **None Category**
 - NoneType
4. **Sequential Category**
 - str
 - list
 - tuple
 - range
5. **Binary Category**
 - bytes
 - bytearray
6. **Set Category**
 - set
 - frozenset
7. **Map Category**
 - dict

1. Numeric Category

int
float
complex

int : 1. Represents the integers

2. This datatype is used for having the countable quantity of data
3. any other datatype element can be converted into int type using int()

float: 1. Represents the float point numbers

2. This datatype is used for having the measurable quantity of data
3. any other datatype element can be converted into float type using float()

complex : 1. Represents the complex numbers

2. any other datatype element can be converted into complex type using complex()

Conversions of Numeric Category

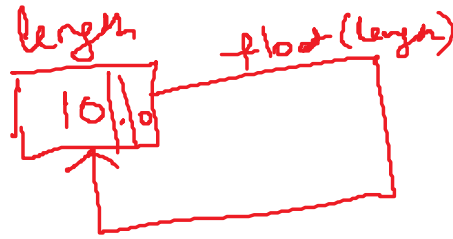
1. int to float :
length = 10



1. int to float :
length = 10

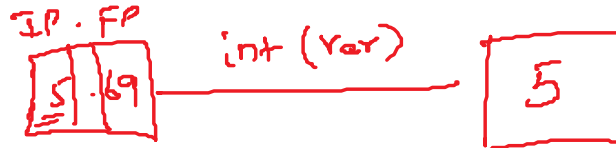


float(length)
#10.0



length = float(length)
print(length)
#10.0

2. float to int :



var = 5.69
int(var)

var = int(var)
print(var)

var = 5.69
dg = int(var)
dg

3. string to int : The numeric compatible strings can be converted into int type

st = "7854"
int(st)

v = int(st)
print(v)

st="Hello@123"
int(st)
#ValueError: invalid literal for int() with base 10: 'Hello@123'

4. int to complex and float to complex

var1 = 14
var2 = 5.63

complex(var1)
#(14+0j)

complex(var2)
#(5.63+0j)

complex(real, imaginary)

```
complex(4,8)
#(4+8j)
```

2. Boolean Category

bool

```
b1 = True
b1
#True
```

```
b2 = False
b2
#False
```

```
bool(5)
#True
```

```
bool(0)
#False
```

```
bool(5.32)
#True
```

Note: Anything other than 0 is True

3. None Category

NoneType
A None means nothing

```
n1 = None
n1
```

```
print(n1)
#None
```

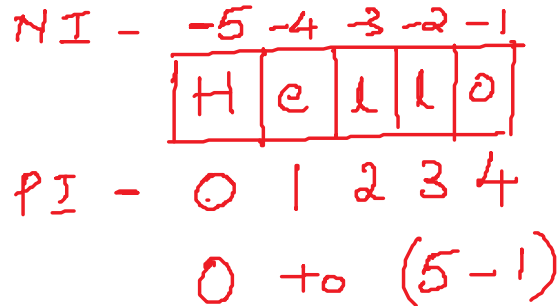
```
type(n1)
NoneType
```

```
NoneType(9)
#NameError: name 'NoneType' is not defined
```

4. Sequential Category : 1. The collection of elements
2. They follow an order or arrangement of elements

1. str : 1. str represents a String, a string is a collection of one more characters enclosed in Quotes
2. str follows indexing , an indexing is a linear arrangement of elements using integers
3. str follows 2 types of indexing
 - Positive Indexing : The index values will be taken from 0 to stringlength -1
 - Negative Indexing: The index values will be taken from -stringlength to -1
4. str is immutable i.e., once defined it cannot be grown or shrunk or updated

Ex:
s1 = "Hello"



Accessing single elements

Stringname[indexno]

s1[0]

s1[0]

s1[-1]

Strings are immutable

s1[6] = 'K'

#TypeError: 'str' object does not support item assignment

s1[2] = 'U'

#TypeError: 'str' object does not support item assignment

s1[3] = "

#TypeError: 'str' object does not support item assignment