Python Data Structures by Mrittika

List

```
In Python, a list is a built-in data structure that allows you to store
and organize a collection of items. These items can be of any data type,
including other lists. Lists are one of the most commonly used data
structures in Python due to their versatility and ease of use.

Lists are defined using square brackets [], and the items within the
list are separated by commas. Here's an example of how you can create a
simple list:

my_list = [1, 2, 3, 4, 5]


Lists can contain elements of different types, such as integers,
strings, floats, and even other lists:

mixed_list = [1, "hello", 3.14, [5, 6, 7]]

You can access individual elements of a list using indexing. Python uses
zero-based indexing, so the first element is at index 0, the second
element is at index 1, and so on. For example:

print(my_list[0])  # Output: 1
print(mixed_list[1])  # Output: "hello"
print(mixed_list[3][0])  # Output: 5 (accessing the first element of the
nested list)

Lists are mutable, which means you can modify them by changing, adding,
or removing elements:

my_list[2] = 10  # Change the third element to 10
my_list.append(6)  # Add an element to the end of the list
my_list.pop(1)  # Remove the element at index 1

You can also perform various operations on lists, such as slicing
(extracting a sublist), concatenating (joining two or more lists),
sorting, and more.
```

```
List : To store collection of objects in one variable
heterogenous : To store mixed datatypes in the same variable
mutable : Values can be changed
```

```
Subsetting or slicing a list in Python involves extracting a portion of
the list based on specified indices.

The slicing syntax is as follows:

new_list = original_list[start:end:step]

start: The index of the first element you want in the new list.
```

```
end: The index up to which you want to slice (exclusive, i.e., the
element at this index will not be included).
step: The interval between elements to include in the new list
(optional).
```

In [1]:
```
student_names=["Robert","John","Skeen Lee","Mike","Josh"]
```

In [2]:
```
student_names
```

Out[2]:
```
['Robert', 'John', 'Skeen Lee', 'Mike', 'Josh']
```

In [3]:
```
l2=["Robert",25,78.45,True]
```

In [4]:
```
l2
```

Out[4]:
```
['Robert', 25, 78.45, True]
```

In [5]:
```
l2[0]
```

Out[5]:
```
'Robert'
```

In [6]:
```
l2[1]
```

Out[6]:
```
25
```

In [7]:
```
l2[2]=81
```

In [8]:
```
l2
```

Out[8]:
```
['Robert', 25, 81, True]
```

```
Append: append() adds an element at the end of the list
```

In [9]:
```
sample = [1, 2, 3, 4, 5, 6]
sample.append(5)
sample.append([7, 8, 9])
sample.extend([6, 7, 8])
print(sample)
```

```
[1, 2, 3, 4, 5, 6, 5, [7, 8, 9], 6, 7, 8]
```

```
Slicing
```

In [10]:
```
list = [1, 2, 3, 4, 5, 6, 7]
print(list[0:4])
print(list[::])
print(list[::-1])
print(list[-1::])
```

```
[1, 2, 3, 4]
[1, 2, 3, 4, 5, 6, 7]
[7, 6, 5, 4, 3, 2, 1]
[7]
```

```
Deleting elements in list
```

In [11]:
```
list = [1, 2, 3, 4, 5, 6, 7]
print(list.pop(2))
print(list)
list.remove(4)
print(list)
list.clear()
print(list)
```

```
3
[1, 2, 4, 5, 6, 7]
[1, 2, 5, 6, 7]
[]
```

Tuple

```
Collection of items/objects in the same variable
hetergenous : It can mixed datatypes as well
immutable
```

In [12]:
```
t1=("Robert",34,60.67)
```

In [13]:
```
t1
```

Out[13]:
```
('Robert', 34, 60.67)
```

In [14]:
```
type(t1)
```

Out[14]:
```
tuple
```

In [15]:
```
t1[0]
```

Out[15]:
```
'Robert'
```

In [16]:
```
t1[2]=65
```

```
---------------------------------------------------------------------------
--
TypeError                                 Traceback (most recent call las
t)
Cell In[16], line 1
----> 1 t1[2]=65

TypeError: 'tuple' object does not support item assignment
```

```
Slicing
```

In [17]:
```
# accessing tuple
tuple = (1,2,3,4,5,6,7,8,9)

print(tuple[1:4])

print(tuple[:-7])

print(tuple[7:])

print(tuple[:])
```

```
(2, 3, 4)
(1, 2)
(8, 9)
(1, 2, 3, 4, 5, 6, 7, 8, 9)
```

Dictionary

```
They store key value pairs
mutable
```

In [18]:
```
student={"Name":"john","Age":34,"Height":156,"Grade":"A"}
```

In [19]:
```
student["Name"]
```

Out[19]:
```
'john'
```

In [20]:
```
student["Height"]
```

Out[20]:
```
156
```

In [21]:
```
student["Age"]=35
```

In [22]:
```
student
```

Out[22]:
```
{'Name': 'john', 'Age': 35, 'Height': 156, 'Grade': 'A'}
```

In [23]:
```
student["Gender"]="Male"
```

In [24]:
```
student
```

Out[24]:
```
{'Name': 'john', 'Age': 35, 'Height': 156, 'Grade': 'A', 'Gender': 'Mal
e'}
```

Sets

```
Empty curly braces { } will make an empty dictionary in Python.
```

In [25]:
```
num = {2, 4, 6, 6, 2, 8}
print(num)
```

```
{8, 2, 4, 6}
```

In [26]:
```
num = {45, 39, 30, 75}

print('before:',num)
num.add(32)
print('after:', num)
```

```
before: {75, 45, 30, 39}
after: {32, 39, 75, 45, 30}
```

In [27]:
```
languages = {'React', 'Java', 'Python'}

print('before:',languages)
removedValue = languages.discard('Java')
print('after:', languages)
```

```
before: {'Python', 'Java', 'React'}
after: {'Python', 'React'}
```

In [28]:
```
num = {2, 4, 6, 6, 2, 8}
print(len(num))
```

```
4
```

Set Intersection

In [29]:
```
A = {1, 3, 5}
B = {1, 2, 3}
print('using &:', A & B)
print('using intersection():', A.intersection(B))
```

```
using &: {1, 3}
using intersection(): {1, 3}
```

In [30]:
```
print('using -:', A - B)
print('using intersection():', A.difference(B))
```

```
using -: {5}
using intersection(): {5}
```

Methods

In [31]:
```
str1="John"
```

In [32]:
```
type(str1)
```

Out[32]:
```
str
```

In [33]:
```
print(dir(str1))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc
__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__getnewargs__', '__getstate__', '__gt__', '__hash__', '__init__', '__in
it_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__
mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subcla
sshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswi
th', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'i
salpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'i
snumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljus
t', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removes
uffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'r
strip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'titl
e', 'translate', 'upper', 'zfill']
```

In [34]:
```
str1.upper()
```

Out[34]:
```
'JOHN'
```

In [35]:
```
str1.isnumeric()
```

Out[35]:
```
False
```

In [36]:
```
str1.isdigit()
```

Out[36]:
```
False
```

In [37]:
```
str1.isalpha()
```

Out[37]:
```
True
```

In [38]:
```
str1.replace("J","W")
```

Out[38]:
```
'Wohn'
```

In [39]:
```
print(dir(l2))
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr_
_', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge_
_', '__getattribute__', '__getitem__', '__getstate__', '__gt__', '__hash_
_', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__',
'__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce_
_', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr_
_', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append',
'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove',
'reverse', 'sort']
```

In [40]:
```
l2.append("Male")
```

In [41]:
```
l2
```

Out[41]:
```
['Robert', 25, 81, True, 'Male']
```

In [42]:
```
l2.pop()
```

Out[42]:
```
'Male'
```

In [43]:
```
l2
```

Out[43]:
```
['Robert', 25, 81, True]
```

In [44]:
```
l2.remove("Robert")
```

In [45]:
```
l2
```

Out[45]:
```
[25, 81, True]
```

Package

In [46]:
```
import numpy as np
```

In [47]:
```
print(dir(np))
```

['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'Data
Source', 'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT',
'ERR_RAISE', 'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'F
PE_INVALID', 'FPE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinit
y', 'MAXDIMS', 'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXACT', 'ModuleDeprecationW
arning', 'NAN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAISE', 'RankWa
rning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_U
NDERFLOW', 'ScalarType', 'Tester', 'TooHardError', 'True_', 'UFUNC_BUFSIZ
E_DEFAULT', 'UFUNC_PYVALS_NAME', 'VisibleDeprecationWarning', 'WRAP', '_C
opyMode', '_NoValue', '_UFUNC_API', '__NUMPY_SETUP__', '__all__', '__buil
tins__', '__cached__', '__config__', '__deprecated_attrs__', '__dir__',
'__doc__', '__expired_functions__', '__file__', '__former_attrs__', '__fu
ture_scalars__', '__getattr__', '__git_version__', '__loader__', '__mkl_v
ersion__', '__name__', '__package__', '__path__', '__spec__', '__version_
_', '_add_newdoc_ufunc', '_builtins', '_distributor_init', '_financial_na
mes', '_get_promotion_state', '_globals', '_int_extended_msg', '_mat', '_
no_nep50_warning', '_pyinstaller_hooks_dir', '_pytesttester', '_set_promo
tion_state', '_specific_msg', '_version', 'abs', 'absolute', 'add', 'add_
docstring', 'add_newdoc', 'add_newdoc_ufunc', 'all', 'allclose', 'alltru
e', 'amax', 'amin', 'angle', 'any', 'append', 'apply_along_axis', 'apply_
over_axes', 'arange', 'arccos', 'arccosh', 'arcsin', 'arcsinh', 'arctan',
'arctan2', 'arctanh', 'argmax', 'argmin', 'argpartition', 'argsort', 'arg
where', 'around', 'array', 'array2string', 'array_equal', 'array_equiv',
'array_repr', 'array_split', 'array_str', 'asanyarray', 'asarray', 'asarr
ay_chkfinite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asmatr
ix', 'atleast_1d', 'atleast_2d', 'atleast_3d', 'average', 'bartlett', 'ba
se_repr', 'binary_repr', 'bincount', 'bitwise_and', 'bitwise_not', 'bitwi
se_or', 'bitwise_xor', 'blackman', 'block', 'bmat', 'bool_', 'broadcast',
'broadcast_arrays', 'broadcast_shapes', 'broadcast_to', 'busday_count',
'busday_offset', 'busdaycalendar', 'byte', 'byte_bounds', 'bytes_', 'c_',
'can_cast', 'cast', 'cbrt', 'cdouble', 'ceil', 'cfloat', 'char', 'charact
er', 'chararray', 'choose', 'clip', 'clongdouble', 'clongfloat', 'column_
stack', 'common_type', 'compare_chararrays', 'compat', 'complex128', 'com
plex64', 'complex_', 'complexfloating', 'compress', 'concatenate', 'con
j', 'conjugate', 'convolve', 'copy', 'copysign', 'copyto', 'corrcoef', 'c
orrelate', 'cos', 'cosh', 'count_nonzero', 'cov', 'cross', 'csingle', 'ct
ypeslib', 'cumprod', 'cumproduct', 'cumsum', 'datetime64', 'datetime_as_s
tring', 'datetime_data', 'deg2rad', 'degrees', 'delete', 'deprecate', 'de
precate_with_doc', 'diag', 'diag_indices', 'diag_indices_from', 'diagfla
t', 'diagonal', 'diff', 'digitize', 'disp', 'divide', 'divmod', 'dot', 'd
ouble', 'dsplit', 'dstack', 'dtype', 'e', 'ediff1d', 'einsum', 'einsum_pa
th', 'emath', 'empty', 'empty_like', 'equal', 'errstate', 'euler_gamma',
'exp', 'exp2', 'expand_dims', 'expm1', 'extract', 'eye', 'fabs', 'fastCop
yAndTranspose', 'fft', 'fill_diagonal', 'find_common_type', 'finfo', 'fi
x', 'flatiter', 'flatnonzero', 'flexible', 'flip', 'fliplr', 'flipud', 'f
loat16', 'float32', 'float64', 'float_', 'float_power', 'floating', 'floo
r', 'floor_divide', 'fmax', 'fmin', 'fmod', 'format_float_positional', 'f
ormat_float_scientific', 'format_parser', 'frexp', 'from_dlpack', 'frombu
ffer', 'fromfile', 'fromfunction', 'fromiter', 'frompyfunc', 'fromregex',
'fromstring', 'full', 'full_like', 'gcd', 'generic', 'genfromtxt', 'geoms
pace', 'get_array_wrap', 'get_include', 'get_printoptions', 'getbufsize',
'geterr', 'geterrcall', 'geterrobj', 'gradient', 'greater', 'greater_equa
l', 'half', 'hamming', 'hanning', 'heaviside', 'histogram', 'histogram2
d', 'histogram_bin_edges', 'histogramdd', 'hsplit', 'hstack', 'hypot', 'i
0', 'identity', 'iinfo', 'imag', 'in1d', 'index_exp', 'indices', 'inexac
t', 'inf', 'info', 'infty', 'inner', 'insert', 'int16', 'int32', 'int64',
'int8', 'int_', 'intc', 'integer', 'interp', 'intersect1d', 'intp', 'inve
rt', 'is_busday', 'isclose', 'iscomplex', 'iscomplexobj', 'isfinite', 'is
fortran', 'isin', 'isinf', 'isnan', 'isnat', 'isneginf', 'isposinf', 'isr
eal', 'isrealobj', 'isscalar', 'issctype', 'issubclass_', 'issubdtype',
'issubsctype', 'iterable', 'ix_', 'kaiser', 'kron', 'lcm', 'ldexp', 'left

```
_shift', 'less', 'less_equal', 'lexsort', 'lib', 'linalg', 'linspace', 'l
ittle_endian', 'load', 'loadtxt', 'log', 'log10', 'log1p', 'log2', 'logad
dexp', 'logaddexp2', 'logical_and', 'logical_not', 'logical_or', 'logical
_xor', 'logspace', 'longcomplex', 'longdouble', 'longfloat', 'longlong',
'lookfor', 'ma', 'mask_indices', 'mat', 'math', 'matmul', 'matrix', 'ma
x', 'maximum', 'maximum_sctype', 'may_share_memory', 'mean', 'median', 'm
emmap', 'meshgrid', 'mgrid', 'min', 'min_scalar_type', 'minimum', 'mintyp
ecode', 'mkl', 'mod', 'modf', 'moveaxis', 'msort', 'multiply', 'nan', 'na
n_to_num', 'nanargmax', 'nanargmin', 'nancumprod', 'nancumsum', 'nanmax',
'nanmean', 'nanmedian', 'nanmin', 'nanpercentile', 'nanprod', 'nanquantil
e', 'nanstd', 'nansum', 'nanvar', 'nbytes', 'ndarray', 'ndenumerate', 'nd
im', 'ndindex', 'nditer', 'negative', 'nested_iters', 'newaxis', 'nextaft
er', 'nonzero', 'not_equal', 'numarray', 'number', 'obj2sctype', 'object
_', 'ogrid', 'oldnumeric', 'ones', 'ones_like', 'outer', 'packbits', 'pa
d', 'partition', 'percentile', 'pi', 'piecewise', 'place', 'poly', 'poly1
d', 'polyadd', 'polyder', 'polydiv', 'polyfit', 'polyint', 'polymul', 'po
lynomial', 'polysub', 'polyval', 'positive', 'power', 'printoptions', 'pr
od', 'product', 'promote_types', 'ptp', 'put', 'put_along_axis', 'putmas
k', 'quantile', 'r_', 'rad2deg', 'radians', 'random', 'ravel', 'ravel_mul
ti_index', 'real', 'real_if_close', 'rec', 'recarray', 'recfromcsv', 'rec
fromtxt', 'reciprocal', 'record', 'remainder', 'repeat', 'require', 'resh
ape', 'resize', 'result_type', 'right_shift', 'rint', 'roll', 'rollaxis',
'roots', 'rot90', 'round', 'round_', 'row_stack', 's_', 'safe_eval', 'sav
e', 'savetxt', 'savez', 'savez_compressed', 'sctype2char', 'sctypeDict',
'sctypes', 'searchsorted', 'select', 'set_numeric_ops', 'set_printoption
s', 'set_string_function', 'setbufsize', 'setdiff1d', 'seterr', 'seterrca
ll', 'seterrobj', 'setxor1d', 'shape', 'shares_memory', 'short', 'show_co
nfig', 'show_runtime', 'sign', 'signbit', 'signedinteger', 'sin', 'sinc',
'single', 'singlecomplex', 'sinh', 'size', 'sometrue', 'sort', 'sort_comp
lex', 'source', 'spacing', 'split', 'sqrt', 'square', 'squeeze', 'stack',
'std', 'str_', 'string_', 'subtract', 'sum', 'swapaxes', 'take', 'take_al
ong_axis', 'tan', 'tanh', 'tensordot', 'test', 'testing', 'tile', 'timede
lta64', 'trace', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tri
l', 'tril_indices', 'tril_indices_from', 'trim_zeros', 'triu', 'triu_indi
ces', 'triu_indices_from', 'true_divide', 'trunc', 'typecodes', 'typenam
e', 'ubyte', 'ufunc', 'uint', 'uint16', 'uint32', 'uint64', 'uint8', 'uin
tc', 'uintp', 'ulonglong', 'unicode_', 'union1d', 'unique', 'unpackbits',
'unravel_index', 'unsignedinteger', 'unwrap', 'use_hugepage', 'ushort',
'vander', 'var', 'vdot', 'vectorize', 'version', 'void', 'vsplit', 'vstac
k', 'where', 'who', 'zeros', 'zeros_like']
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: