Supplementary notes PYP1-MAY24

PYP1-MAY24

>>

INTERACTIVE HELP:

```
>>> dir(__builtins__)
>>> help(str)
>>> q
```

Yields:

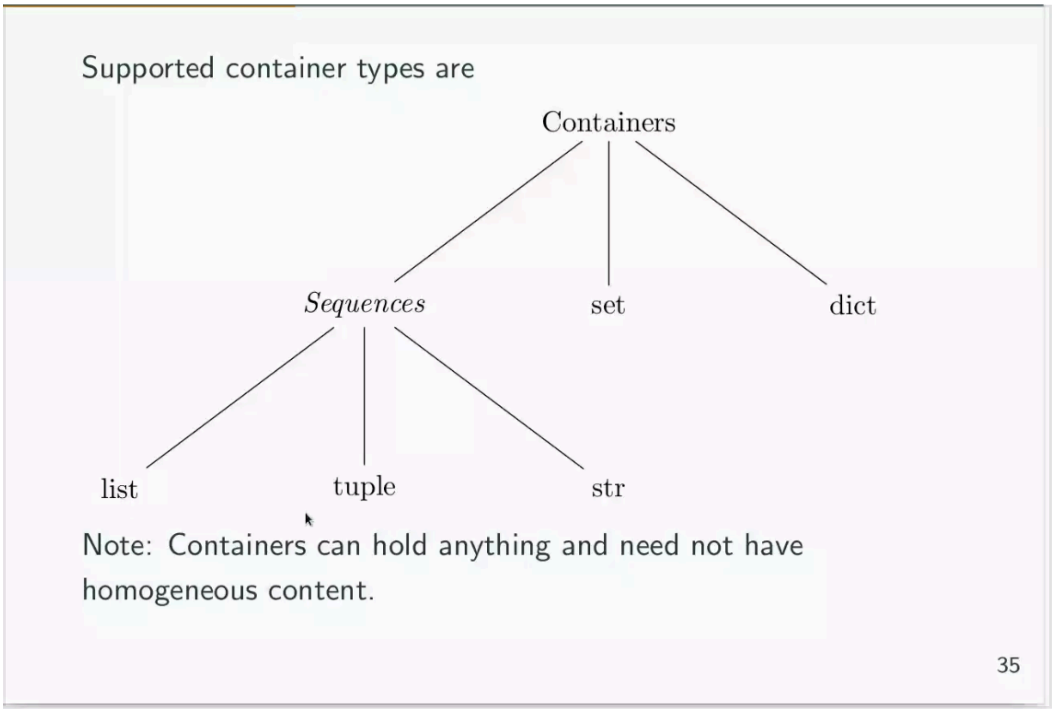
```
>>> dir(__builtins__)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException',
'BaseExceptionGroup', 'BlockingIOError', 'BrokenPipeError',
'BufferError', 'BytesWarning', 'ChildProcessError',
'ConnectionAbortedError', 'ConnectionError',
'ConnectionRefusedError', 'ConnectionResetError',
'DeprecationWarning', 'EOFError', 'Ellipsis', 'EncodingWarning',
'EnvironmentError', 'Exception', 'ExceptionGroup', 'False',
'FileExistsError', 'FileNotFoundError', 'FloatingPointError',
'FutureWarning', 'GeneratorExit', 'IOError', 'ImportError',
'ImportWarning', 'IndentationError', 'IndexError', 'InterruptedError',
'IsADirectoryError', 'KeyError', 'KeyboardInterrupt', 'LookupError',
'MemoryError', 'ModuleNotFoundError', 'NameError', 'None',
'NotADirectoryError', 'NotImplemented', 'NotImplementedError',
'OSError', 'OverflowError', 'PendingDeprecationWarning',
'PermissionError', 'ProcessLookupError', 'RecursionError',
'ReferenceError', 'ResourceWarning', 'RuntimeError',
'RuntimeWarning', 'StopAsyncIteration', 'StopIteration', 'SyntaxError',
'SyntaxWarning', 'SystemError', 'SystemExit', 'TabError', 'TimeoutError',
'True', 'TypeError', 'UnboundLocalError', 'UnicodeDecodeError',
'UnicodeEncodeError', 'UnicodeError', 'UnicodeTranslateError',
'UnicodeWarning', 'UserWarning', 'ValueError', 'Warning',
'ZeroDivisionError', '__build_class__', '__debug__', '__doc__',
'__import__', '__loader__', '__name__', '__package__', '__spec__', 'abs',
'aiter', 'all', 'anext', 'any', 'ascii', 'bin', 'bool', 'breakpoint', 'bytearray',
'bytes', 'callable', 'chr', 'classmethod', 'compile', 'complex', 'copyright',
'credits', 'delattr', 'dict', 'dir', 'divmod', 'enumerate', 'eval', 'exec', 'exit',
'filter', 'float', 'format', 'frozenset', 'getattr', 'globals', 'hasattr', 'hash',
'help', 'hex', 'id', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len',
'license', 'list', 'locals', 'map', 'max', 'memoryview', 'min', 'next', 'object',
'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr',
'reversed', 'round', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str',
'sum', 'super', 'tuple', 'type', 'vars', 'zip']
```

```
>>> dir(str)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__getnewargs__', '__getstate__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__',
'__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__',
'__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index',
'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower',
'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower',
'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace',
'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines',
'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

Qualify the individual function:

```
>>> help(str.upper)

Help on method_descriptor:

upper(self, /)
    Return a copy of the string converted to uppercase.

q # for quit help

# only return builtins not errors as well
# >>> dir (__builtins__)[dir(__builtins__).index('abs'):]
```

>>



Supported container types are

Containers

Sequences

set

dict

list

tuple

str

Note: Containers can hold anything and need not have homogeneous content.

35

>>

Post course readme links

For more info on extending classes:
https://www.geeksforgeeks.org/inheritance-and-composition-in-python/

https://www.geeksforgeeks.org/multiple-inheritance-in-python/

For more info on creating custom exceptions by extending base class Exception:
https://www.geeksforgeeks.org/user-defined-exceptions-python-examples/

For more info on list comprehensions:
https://www.geeksforgeeks.org/python-list-comprehension/

>>

RECOMMENDED READING:

TOP LEVEL TRAIL STARTS:

W3 Schools

https://www.w3schools.com/python/

GeeksForGeeks

https://www.geeksforgeeks.org/python-programming-language/?ref=lbp

The Python Docs

https://www.python.org/doc/

RealPython

https://realpython.com/

Benjamin Bennett Alexander on Twitter/X

https://twitter.com/RealBenjizo

EXAMPLE SEARCHES COMPARISON:

Sample Docs lookup (in ascending order of complexity):

On W3 Schools (excellent for a start or quick knowledge check)

https://www.w3schools.com/python/python_strings.asp

On GeeksForGeeks (I find them just the right level of detail while working)

https://www.geeksforgeeks.org/python-string/

https://www.geeksforgeeks.org/python-string/#accessing-characters-in-python-string

On The official docs (can be quite hard to read but very comprehensive)
https://docs.python.org/3/library/string.html

Miscellaneous references

For the spec on operator precedence rules:
https://docs.python.org/3/reference/expressions.html#operator-precedence

For more info on extending classes:
https://www.geeksforgeeks.org/inheritance-and-composition-in-python/

https://www.geeksforgeeks.org/multiple-inheritance-in-python/

For more info on creating custom exceptions by extending base class Exception:
https://www.geeksforgeeks.org/user-defined-exceptions-python-examples/

For more info on list comprehensions:
https://www.geeksforgeeks.org/python-list-comprehension/

Happy Hacking!

Alan