```
Type "help", "copyright", "credits" or "license" for more information.
>>> s = "sggs It"
>>> s.removesuffix()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: str.removesuffix() takes exactly one argument (0 given)
>>> s.removesuffix("It")
'sggs '
>>> s = "sggsie&t"
>>> s.replace("ie&t")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: replace expected at least 2 arguments, got 1
>>> s.replace("ie&t","it")
'sggsit'
>>> s = "nanded"
>>> s.rfind('d')
5
>>> s = "hello"
>>> s.rindex()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: rindex() takes at least 1 argument (0 given)
>>> s.rindex(l)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'l' is not defined
>>> s.rindex(l'')
  File "<stdin>", line 1
    s.rindex(l'')
             ^^
SyntaxError: invalid syntax
>>> s.rindex('l')
3
>>> s.rjust(10)
'     hello'
>>>
```

```
Python 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = {"apple","banana"}
>>> x.add("cherry")
>>> x
{'banana', 'apple', 'cherry'}
>>> x.clear()
>>> x
set()
>>> s = [1,2,3]
>>> s.copy()
[1, 2, 3]
>>> set1 = {1,2,3,4}
>>> set2 = {3,4,5}
>>> set1.difference(set2)
{1, 2}
>>> set1.difference_update(set2)
>>> set1
{1, 2}
>>> s = {3,4,"It","pen"}
>>> s.discard(4)
>>> s
{'pen', 3, 'It'}
>>> x = {7,6,5}
>>> y = {4,5,6}
>>> x.intersection(y)
{5, 6}
>>> x.isdisjoint(y)
False
>>> m = {1,2,3}
>>> n = {1,2,3,4,5,6,7,8,9}
>>> m.issubset(n)
True
>>> m.issuperset(n)
False
```

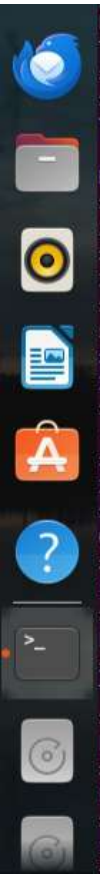
```
Python 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> s = 3.14
>>> s.as_integer_ratio()
(7070651414971679, 2251799813685248)
>>> print(s.as_integer_ratio())
(7070651414971679, 2251799813685248)
>>> s = 12
>>> s.as_integer_ratio()
(12, 1)
>>> s.bitcount()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'int' object has no attribute 'bitcount'. Did you mean: 'bit_count'?
>>> s = 6
>>> s.bitcount()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'int' object has no attribute 'bitcount'. Did you mean: 'bit_count'?
>>> s.bit_count()
2
>>> s.bit_length()
3
>>> s = 3+7j
>>> s.conjugate()
(3-7j)
```

```
Python 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> b = b\x00\x10
  File "<stdin>", line 1
    b = b\x00\x10
         ^
SyntaxError: unexpected character after line continuation character
>>> int.from_bytes(b'\x00\x10', 'big')  # Output: 16
16
>>> z = 3 + 4j
>>> z.imag
4.0
>>> (3.5).numerator
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'float' object has no attribute 'numerator'
>>> (3.5).numerator
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'float' object has no attribute 'numerator'
>>> (3.5).numerator
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'float' object has no attribute 'numerator'
>>> z.real
3.0
>>> (16).to_bytes(2, 'big')
b'\x00\x10'
>>>
```

```
  File "<stdin>", line 1
    dictionary{}
              ^
SyntaxError: invalid syntax
>>> s = "sggs"
>>> help(s.isalpha())

>>> s.isalpha()
True
>>> s.isascii()
True
>>> s.isdecimal()
False
>>> s = 46
>>> s.isdecimal()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'int' object has no attribute 'isdecimal'
>>> s = '46'
>>> s.isdecimal()
True
>>> s.isdigit()
True
>>> s.isidentifier()
False
>>> s = "sggs"
>>> s.isidentifier()
True
>>> s.islower()
True
>>> s.isnumeric()
False
>>> s = '45'
>>> s.isnumeric()
True
>>>
```

```
>>> a.isdigit()
True
>>> s.isidentifier()
True
>>> s = "sggs"
>>> s.islower()
True
>>> s = "nanded"
>>> s.isprintable()
True
>>> s.isspace()
False
>>> s.istitle()
False
>>> s.issuper()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'str' object has no attribute 'issuper'. Did you mean: 'isupper'?
>>> s.issupper()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'str' object has no attribute 'issupper'. Did you mean: 'isupper'?
>>> s.isupper()
False
>>> s ="'"
>>> s.join(["pallavi","more"])
"pallavi'more"
>>> a = "hello"
>>> a.ljust(5)
'hello'
>>> a.ljust(10)
'hello     '
>>> s = "SGGS"
>>> s.lower()
'sggs'
>>>
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> s = "world is cruel"
>>> s.rpartition(" ")
('world is', ' ', 'cruel')
>>> s.rsplit()
['world', 'is', 'cruel']
>>> s = "hello world"
>>> s.rstrip()
'hello world'
>>> s.rsplit(" ")
['hello', 'world']
>>> s = "everything\nis\nfake"
>>> s.splitlines()
['everything', 'is', 'fake']
>>> s.startswith('e')
True
>>> s = "       world"
>>> s.strip()
'world'
>>> s.swapcase()
'       WORLD'
>>> s = "world"
>>> s.title()
'World'
>>> s = "aeiou"
>>> n = '12345'
>>> x = str.maketrans(s,n)
>>> s.translate(x)
'12345'
>>> s = "sggs"
>>> s.upper()
'SGGS'
>>> s = "23"
>>> s.zfill()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```



```
Python 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 2345
>>> _
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name '_' is not defined
>>> x
2345
>>> _
2345
>>> x = 1000
>>> y = 20000
>>> x + y
21000
>>> x - y
-19000
>>> x*y
20000000
>>> x / y
0.05
>>> x % y
1000
>>> x == y
False
>>> x < y
True
>>> x > y
False
>>> x >= y
False
>>> x
1000
>>>
```

```
>>> dir(__builtins__)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BlockingIOError', 'BrokenPipeError', 'BufferError', 'BytesWarning',
'ChildProcessError', 'ConnectionAbortedError', 'ConnectionError', 'ConnectionRefusedError', 'ConnectionResetError', 'DeprecationWarning', 'EOF
Error', 'Ellipsis', 'EncodingWarning', 'EnvironmentError', 'Exception', 'False', 'FileExistsError', 'FileNotFoundError', 'FloatingPointError',
 'FutureWarning', 'GeneratorExit', 'IOError', 'ImportError', 'ImportWarning', 'IndentationError', 'IndexError', 'InterruptedError', 'IsADirect
oryError', 'KeyError', 'KeyboardInterrupt', 'LookupError', 'MemoryError', 'ModuleNotFoundError', 'NameError', 'None', 'NotADirectoryError', 'N
otImplemented', 'NotImplementedError', 'OSError', 'OverflowError', 'PendingDeprecationWarning', 'PermissionError', 'ProcessLookupError', 'Recu
rsionError', 'ReferenceError', 'ResourceWarning', 'RuntimeError', 'RuntimeWarning', 'StopAsyncIteration', 'StopIteration', 'SyntaxError', 'Syn
taxWarning', 'SystemError', 'SystemExit', 'TabError', 'TimeoutError', 'True', 'TypeError', 'UnboundLocalError', 'UnicodeDecodeError', 'Unicode
EncodeError', 'UnicodeError', 'UnicodeTranslateError', 'UnicodeWarning', 'UserWarning', 'ValueError', 'Warning', 'ZeroDivisionError', '_', '__
build_class__', '__debug__', '__doc__', '__import__', '__loader__', '__name__', '__package__', '__spec__', 'abs', 'aiter', 'all', 'anext', 'an
y', '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', 'ha
sattr', 'hash', 'help', 'hex', 'id', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len', 'license', 'list', 'locals', 'map', 'max', 'me
moryview', 'min', 'next', 'object', 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr', 'reversed', 'round', 'set', 'se
tattr', 'slice', 'sorted', 'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'vars', 'zip']
>>> help(abs)

>>> help(abs)

>>> abs(x,/)
  File "<stdin>", line 1
    abs(x,/)
          ^
SyntaxError: invalid syntax
>>> help(abs)

>>> abs(23)
23
>>> abs("23")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: bad operand type for abs(): 'str'
>>> help(aiter)

>>> help(all)

>>> x = [2,3,4,5]
```

```
Python 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = {"name":"seema","age":"12"}
>>> x.clear()
>>> x
{}
>>> x = {"name","age","bird"}
>>> x.copy()
{'age', 'name', 'bird'}
>>> x = {"name","age","bird"}
>>> d.fromkeys(x,"kind")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'd' is not defined. Did you mean: 'id'?
>>> x.fromkeys(x,"kind")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'set' object has no attribute 'fromkeys'
>>> x = {"name":"seema","age":"12"}
>>> x.get("name")
'seema'
>>> x = {"name":"seema","age":"12"}
>>> x.items()
dict_items([('name', 'seema'), ('age', '12')])
>>> x = {"name":"seema","age":"12"}
>>> x.keys()
dict_keys(['name', 'age'])
>>> x = {"name":"seema","age":"12"}
>>> x.pop("age")
'12'
>>> x = {"name":"seema","age":"12"}
>>> x.popitem()
('age', '12')
>>> x = {"college":"sggs","city":"nanded"}
>>> x.update{"branch": "It"}
```

```
  File "<stdin>", line 1, in <module>
AttributeError: 'str' object has no attribute 'expandtads'. Did you mean: 'expandtabs'?
>>> s.expandtabs(4)
'Engineering college                                    '
>>> help(s.find)

>>> s = "engineering"
>>> s.find(sub(2,5))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sub' is not defined. Did you mean: 'sum'?
>>> s.find(2,5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not int
>>> s.find(neer)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'neer' is not defined
>>> s = "sggs nanded"
>>> s.find("nanded")
5
>>> help(s.form)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'str' object has no attribute 'form'. Did you mean: 'format'?
>>> s = "sggs"
>>> s.count("gg",1,3)
1
>>> s = "sgggs"
>>> s.count("gg",1,3)
1
>>> s.count("gg")
1
>>> help(s.format)
```

```
', 'slice', 'sorted', 'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'vars', 'zip']
>>> help(bool)

>>> help(bin)

>>> bin(2345)
'0b100100101001'
>>> help(complex)

>>> complex(34)
(34+0j)
>>> help(slice)

>>> slice(29)
slice(None, 29, None)
>>> num = list(range(50))
>>> s = slice(29)
>>> result = num[s]
>>> print(result)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
>>> help(breakpoint)

>>> int x = 6
  File "<stdin>", line 1
    int x = 6
        ^
SyntaxError: invalid syntax
>>> x = 7
>>> y = 9
>>> breakpoint()
--Return--
> <stdin>(1)<module>()->None
(Pdb) print(x)
7
(Pdb) print(y)
9
(Pdb) c
>>>
```

```
1000
>>>
>>>
>>>
>>>
>>> x <= y
True
>>> x != y
True
>>> y != x
True
>>> x << y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: Exceeds the limit (4300) for integer string conversion; use sys.set_int_max_str
>>> x >> y
0
>>> y << x
214301721437253464189685009812000362112280962341106721488750077674070210224987224498639675
915514939714960786913554964846197084214921012474228375590836430609294996716388253479753511
7736744113361387520000
>>> x ** y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: Exceeds the limit (4300) for integer string conversion; use sys.set_int_max_str
>>> y ** x
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: Exceeds the limit (4300) for integer string conversion; use sys.set_int_max_str
>>> ~x
-1001
>>> y > x
True
>>> x : y ? 10:20
  File "<stdin>", line 1
    x : y ? 10:20
          ^
SyntaxError: invalid syntax
```

```
6
>>> tuple([1,2,3])
(1, 2, 3)
>>> type('hello')
<class 'str'>
>>> class Myclass
  File "<stdin>", line 1
    class Myclass
                ^
SyntaxError: expected ':'
>>> class Myclass:
... x = 10
  File "<stdin>", line 2
    x = 10
    ^
IndentationError: expected an indented block after class definition on line 1
>>> class Myclass:
... x = 2
  File "<stdin>", line 2
    x = 2
    ^
IndentationError: expected an indented block after class definition on line 1
>>> class Myclass:
...   x = 2
... vars(Myclass)
  File "<stdin>", line 3
    vars(Myclass)
    ^^^^
SyntaxError: invalid syntax
>>> class Myclass:
...   x = 2
...
>>> vars(Myclass)
mappingproxy({'__module__': '__main__', 'x': 2, '__dict__': <attribute '__dict__' of 'Myclass' objects>, '__weakref__': <attribute '__weakref_
_' of 'Myclass' objects>, '__doc__': None})
>>> list(zip([1,2,3],('a','b','c')))
[(1, 'a'), (2, 'b'), (3, 'c')]
>>>
```

```
>>> repr('hello')
"'hello'"
>>> list(reversed([1,2,3]))
[3, 2, 1]
>>> round(3.142345,3)
3.142
>>> set([1,2,3])
{1, 2, 3}
>>> class Myclass:
...
  File "<stdin>", line 2

    ^
IndentationError: expected an indented block after class definition on line 1
>>> class Myclass:
... obj = Myclass()
  File "<stdin>", line 2
    obj = Myclass()
    ^
IndentationError: expected an indented block after class definition on line 1
>>> setattr(obj,'name','Alice')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'obj' is not defined
>>> s = slice(5)
>>> list(range(10))[s]
[0, 1, 2, 3, 4]
>>> sorted([3,1,2])
[1, 2, 3]
>>> str(123)
'123'
>>> sum([1,2,3])
6
>>> tuple([1,2,3])
(1, 2, 3)
>>> type('hello')
<class 'str'>
>>>
```

```
>>> max([1,2,3])
3
>>> memoryview("hello")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: memoryview: a bytes-like object is required, not 'str'
>>> memoryview('hello')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: memoryview: a bytes-like object is required, not 'str'
>>> memoryview(b'hello')
<memory at 0x778874455900>
>>> it = iter([1,2,3])
>>> next(it)
1
>>> min([4,5,6])
4
>>> oct(8)
'0o10'
>>> ord('A')
65
>>> pow(2,3)
8
>>> print('Hello World!')
Hello World!
>>> property()
<property object at 0x7788756fcc20>
>>> range(5)
range(0, 5)
>>> list(range(5))
[0, 1, 2, 3, 4]
>>> repr('hello')
"'hello'"
>>> list(reversed([1,2,3]))
[3, 2, 1]
>>> round(3.142345,3)
3.142
>>>
```

```
    ^
IndentationError: expected an indented block after class definition on line 1
>>> Class A:
  File "<stdin>", line 1
    Class A:
          ^
SyntaxError: invalid syntax
>>> iter([1,2,3])
<list_iterator object at 0x7788756e1870>
>>> len([1,2,3])
3
>>> help(list)

>>> [1,2,3]
[1, 2, 3]
>>> locals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <class '_frozen_importlib.BuiltinImporter'>, '__spec__': None, '__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>}
>>> list(map(lambda x:x*2,[1,2,3]))
[2, 4, 6]
>>> max([1,2,3])
3
>>> memoryview("hello")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: memoryview: a bytes-like object is required, not 'str'
>>> memoryview('hello')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: memoryview: a bytes-like object is required, not 'str'
>>> memoryview(b'hello')
<memory at 0x778874455900>
>>> it = iter([1,2,3])
>>> next(it)
1
>>> min([4,5,6])
4
>>>
```

```
['__annotations__', '__builtins__', '__doc__', '__loader__', '__name__', '__package__', '__return__', '__spec__', 'a', 'ba', 'course', 'myDic
', 'x', 'y']
>>>
>>> # Now let's import two modules
>>> import random
>>> import math
>>>
>>> print(dir())
['__annotations__', '__builtins__', '__doc__', '__loader__', '__name__', '__package__', '__return__', '__spec__', 'a', 'ba', 'course', 'math'
 'myDict', 'random', 'x', 'y']
>>> print('(5, 4) = ', divmod(5, 4))
(5, 4) =  (1, 1)
>>> print('(10, 16) = ', divmod(10, 16))
(10, 16) =  (0, 10)
>>> fruits = ['apple', 'banana', 'cherry']
>>> enum_fruits = enumerate(fruits)
>>>
>>> next_element = next(enum_fruits)
>>> print(f"Next Element: {next_element}")
Next Element: (0, 'apple')
>>> x = 5
>>> print(eval('x == 4'))
False
>>>
>>> x = None
>>> print(eval('x is None'))
True
>>> prog = 'print("The sum of 5 and 10 is", (5+10))'
>>> exec(prog)
The sum of 5 and 10 is 15
>>> help(exit)
```

```
...     globals()['a'] = d
...     print (a)
...
... # Driver Code
... func()
  File "<stdin>", line 10
    func()
    ^^^^
SyntaxError: invalid syntax
>>> print(globals())
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <class '_frozen_importlib.BuiltinImporter'>, '__spec__': None, '
_annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'seq': [0, 1, 2, 3, 5, 8, 13], 'result': <filter object at 0x7606ba5677f
>, 'num': 10.0, 'txt': 'I have {an:.2f} Rupees!', 'nu': (), 'fnum': frozenset(), 'GfG': <class '__main__.GfG'>, 'obj': <__main__.GfG object a
 0x7606ba4e47c0>, 'a': 5}
>>> print("")

>>>
>>> p,q,r,s=10,100,1000,10000
>>>
>>> print(globals())
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <class '_frozen_importlib.BuiltinImporter'>, '__spec__': None, '
_annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, 'seq': [0, 1, 2, 3, 5, 8, 13], 'result': <filter object at 0x7606ba5677f
>, 'num': 10.0, 'txt': 'I have {an:.2f} Rupees!', 'nu': (), 'fnum': frozenset(), 'GfG': <class '__main__.GfG'>, 'obj': <__main__.GfG object a
 0x7606ba4e47c0>, 'a': 5, 'p': 10, 'q': 100, 'r': 1000, 's': 10000}
>>> # hash() for immutable tuple object
>>> var = ('G','E','E','K')
>>>
>>> print(hash(var))
-5052158403120423996
>>> x = 15
>>> print(hex(x))
0xf
>>> help(id)

>>> x = "sggs"
>>> id(x)
129771247164912
>>>
```

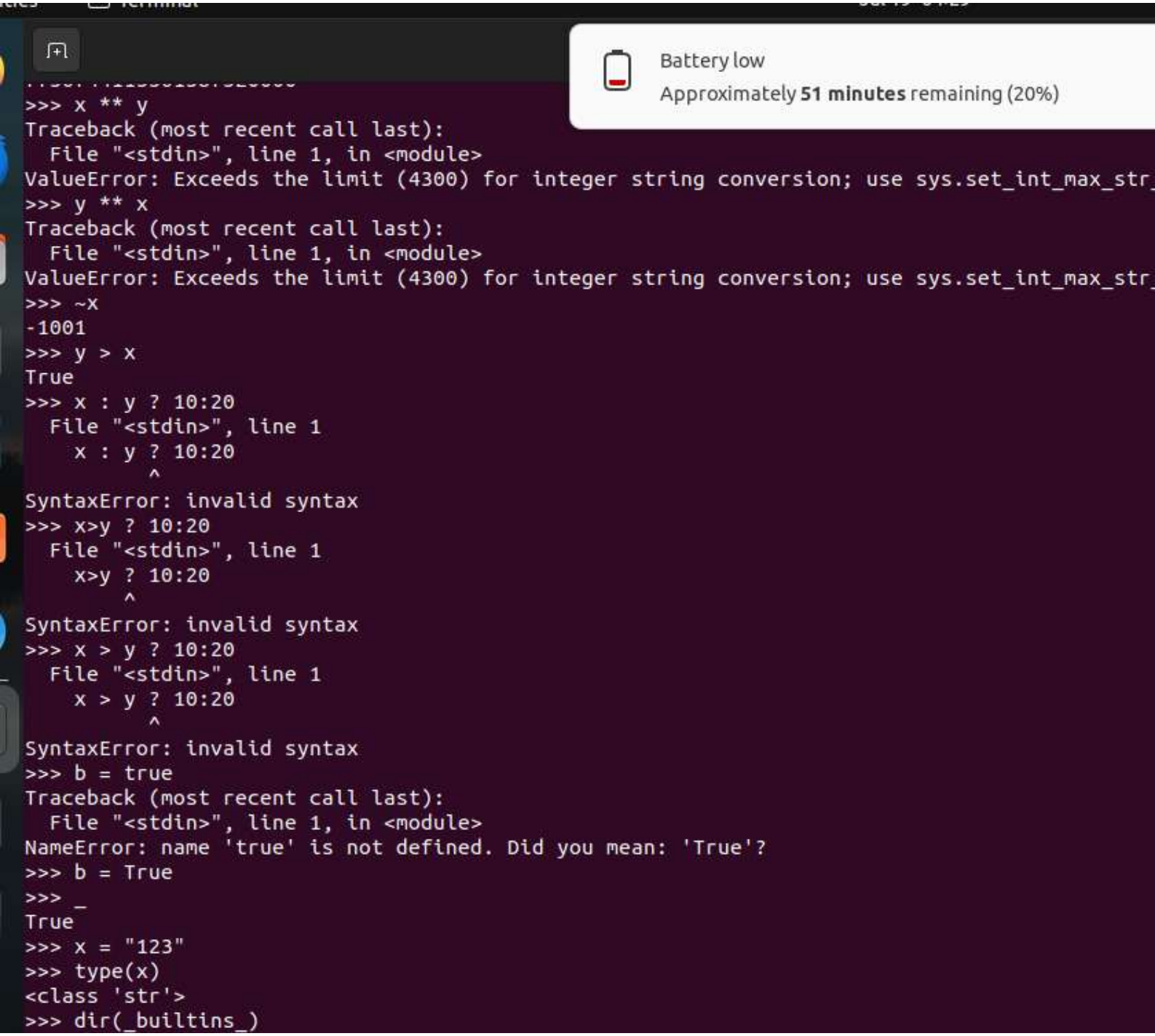
```
iew', 'min', 'next', 'object', 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr', 'reversed', 'round', 'set', 'setatt
', 'slice', 'sorted', 'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'vars', 'zip']
>>> seq = [0, 1, 2, 3, 5, 8, 13]
>>>
>>> # result contains odd numbers of the list
>>> result = filter(lambda x: x % 2 != 0, seq)
>>> print(list(result))
[1, 3, 5, 13]
>>> # convert integer value to float
>>> num = float(10)
>>> print(num)
10.0
>>> txt = "I have {an:.2f} Rupees!"
>>> print(txt.format(an = 4))
I have 4.00 Rupees!
>>> # passing an empty tuple
>>> nu = ()
>>>
>>> # converting tuple to frozenset
>>> fnum = frozenset(nu)
>>>
>>> # printing empty frozenset object
>>> print("frozenset Object is : ", fnum)
frozenset Object is :  frozenset()
>>> class GfG:
...     name = "GeeksforGeeks"
...     age = 24
... obj = GfG()
  File "<stdin>", line 4
    obj = GfG()
    ^^^
SyntaxError: invalid syntax
```

```
SyntaxError: invalid syntax
>>> class course:
...     name = "data structures using c++"
...     duration_months = 6
...     price = 20000
...     rating = 5
...
>>> # creating an object of course
>>> print(course.rating)
5
>>>
>>> # deleting the rating attribute from object
>>> delattr(course, 'rating')
>>>
>>> # checking if the rating attribute is there or not
>>> try:
...     print(course.rating)
... except Exception as e:
...     print(e)
...
type object 'course' has no attribute 'rating'
>>> # passing keyword arguments to dict() method
>>> myDict = dict(a=1, b=2, c=3, d=4)
>>>
>>> print(myDict)
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
>>> # Python3 code to demonstrate dir()
>>> # when no parameters are passed
>>> # Note that we have not imported any modules
>>> print(dir())
['__annotations__', '__builtins__', '__doc__', '__loader__', '__name__', '__package__', '__return__', '__spec__', 'a', 'ba', 'course', 'myDic
', 'x', 'y']
>>>
```

```
>>> x = 7
>>> y = 6
>>> breakpoint()
--Return--
> <stdin>(1)<module>()->None
(Pdb) x
7
(Pdb) y
6
(Pdb) q
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/python3.10/bdb.py", line 94, in trace_dispatch
    return self.dispatch_return(frame, arg)
  File "/usr/lib/python3.10/bdb.py", line 156, in dispatch_return
    if self.quitting: raise BdbQuit
bdb.BdbQuit
>>> bytearray
<class 'bytearray'>
>>> bytearray(34)
bytearray(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00')
>>> bytearray(6)
bytearray(b'\x00\x00\x00\x00\x00\x00')
>>> ba = bytearray([34,56,12])
>>> print(ba)
bytearray(b'"8\x0c')
>>> help(bytes)

>>> ba = bytes(4)
>>> print(ba)
b'\x00\x00\x00\x00'
>>> bytes(4)
b'\x00\x00\x00\x00'
>>> bytes([23,34,56])
b'\x17"8'
>>> help(callable)
```

```
>>>
>>> help(classmethod)

>>> classmethod(45)
<classmethod(45)>
>>> help(compile)

>>> x = 50
>>> a = compile('x', 'test', 'single')
>>> print(type(a))
<class 'code'>
>>> exec(a)
50
>>>
>>> help(complex)

>>> x = 45
>>> complex(x)
(45+0j)
>>> help(copyright)

>>> help(credits)

>>>
>>> help(credits)

>>> credits(5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: _Printer.__call__() takes 1 positional argument but 2 were given
>>> >>> import sys
  File "<stdin>", line 1
    >>> import sys
    ^^
SyntaxError: invalid syntax
>>> >>> print(sys.credits)
  File "<stdin>", line 1
```

```
(Pdb) c
>>> help(bytearray)

>>> bytearray(256)
bytearray(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00')
>>> ba = bytearray(256)
>>> ba[1] = 1
>>> print(ba)
bytearray(b'\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00')
>>> bytes(34)
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
>>> help(callable)

>>> callable(y)
False
>>>
```


Battery low
Approximately 51 minutes remaining (20%)
>>> x ** y
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: Exceeds the limit (4300) for integer string conversion; use sys.set_int_max_str
>>> y ** x
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: Exceeds the limit (4300) for integer string conversion; use sys.set_int_max_str
>>> ~x
-1001
>>> y > x
True
>>> x : y ? 10:20
File "<stdin>", line 1
x : y ? 10:20
^
SyntaxError: invalid syntax
>>> x>y ? 10:20
File "<stdin>", line 1
x>y ? 10:20
^
SyntaxError: invalid syntax
>>> x > y ? 10:20
File "<stdin>", line 1
x > y ? 10:20
^
SyntaxError: invalid syntax
>>> b = true
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'true' is not defined. Did you mean: 'True'?
>>> b = True
>>> _
True
>>> x = "123"
>>> type(x)
<class 'str'>
>>> dir(_builtins_)

```
0
>>> x.insert(2,9)
>>> x
[1, 3, 9, 1, 1, 6, 5, 6, 9]
>>> x.pop(6)
5
>>> x.remove(5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: list.remove(x): x not in list
>>> x = [1,2,3,4]
>>> x.remove(3)
>>> x
[1, 2, 4]
>>> x.reverse()
>>> x
[4, 2, 1]
>>> x = [6,7,5,4,3]
>>> x.sort(reverse= True)
>>> x
[7, 6, 5, 4, 3]
>>> z = ("sggs",67,"It")
>>> z.count(""It)
  File "<stdin>", line 1
    z.count(""It)
            ^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
>>> z.count("It")
1
>>> z.index("It")
2
>>>
```

```
>>> help(callable)

>>> def x()
  File "<stdin>", line 1
    def x()
           ^
SyntaxError: expected ':'
>>> def x():
... a = 5
  File "<stdin>", line 2
    a = 5
    ^
IndentationError: expected an indented block after function definition on line 1
>>> def x();
  File "<stdin>", line 1
    def x();
           ^
SyntaxError: expected ':'
>>> def x():
... a = 5
  File "<stdin>", line 2
    a = 5
    ^
IndentationError: expected an indented block after function definition on line 1
>>> def x():
...     a = 5
...
>>> callable(x)
True
>>> help
Type help() for interactive help, or help(object) for help about object.
>>> help(chr)

>>> chr(34)
'"'
>>> help(classmethod)

>>>
```

```
6
>>> tuple([1,2,3])
(1, 2, 3)
>>> type('hello')
<class 'str'>
>>> class Myclass
  File "<stdin>", line 1
    class Myclass
                ^
SyntaxError: expected ':'
>>> class Myclass:
... x = 10
  File "<stdin>", line 2
    x = 10
    ^
IndentationError: expected an indented block after class definition on line 1
>>> class Myclass:
... x = 2
  File "<stdin>", line 2
    x = 2
    ^
IndentationError: expected an indented block after class definition on line 1
>>> class Myclass:
...   x = 2
... vars(Myclass)
  File "<stdin>", line 3
    vars(Myclass)
    ^^^^
SyntaxError: invalid syntax
>>> class Myclass:
...    x = 2
...
>>> vars(Myclass)
mappingproxy({'__module__': '__main__', 'x': 2, '__dict__': <attribute '__dict__' of 'Myclass' objects>, '__weakref__': <attribute '__weakref_
_' of 'Myclass' objects>, '__doc__': None})
>>> list(zip([1,2,3],('a','b','c')))
[(1, 'a'), (2, 'b'), (3, 'c')]
>>>
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> x = [1,2,3,4]
>>> dir(x)
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__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']
>>> x = [4,5,6]
>>> x.append(8)
>>> x
[4, 5, 6, 8]
>>> x = [1,2,3]
>>> x.clear()
>>> x
[]
>>> x = [8,9,10]
>>> x.copy(0
...
...
... q
  File "<stdin>", line 1
    x.copy(0
           ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
>>> x.copy()
[8, 9, 10]
>>> x = [1,3,1,1,6,5]
>>> x.count(1)
3
>>> x.extend([6,9])
>>> x
[1, 3, 1, 1, 6, 5, 6, 9]
>>> x.index(1)
0
>>> x.insert(2,0)
```

```
False
>>> x = {1,2,3}
>>> s.remove(3)
>>> x.remove(3)
>>> x
{1, 2}
>>> x = {1,2,3}
>>> y = {2,3,4}
>>> x.symmetric_difference(y)
{1, 4}
>>> x.symmetric_difference_update(y)
>>> x
{1, 4}
>>> s = {1,2,3}
>>> m = {3,4,5}
>>> s.union(m)
{1, 2, 3, 4, 5}
>>> x = {1,2,3}
>>> y = {"sggs","pen","sim"}
>>> x.update(y)
>>> x
{1, 2, 3, 'pen', 'sggs', 'sim'}
>>>
```

```
'SGGS'
>>> s = "23"
>>> s.zfill()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: str.zfill() takes exactly one argument (0 given)
>>> s.zfill(3)
'023'
>>>
```