

Built_in Functions:

```
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> help(abs)
>>> help(abs)
>>> help(all)
>>>
>>> x = abs(3.57)
>>> x
3.57
>>> x = abs(-3.65)
>>> x
3.65
>>> mylist = [0,1,1]
>>> x = all(mylist)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'builtin_function_or_method' object is not subscriptable
>>> x = all(mylist)
>>> x
False
>>> mytuple = (0,1,False)
>>> x = any(mytuple)
>>> x
True
>>> x = ascii("my name is ria")
>>> x
"my name is ria"
>>> x = bin(36)
>>> x
'0b100100'
>>> x = bool(1)
>>> x
True
>>> x = bytearray(4)
>>> x
bytearray(b'\x00\x00\x00\x00')
>>> x = bytes(4)
>>> x
b'\x00\x00\x00\x00'
>>> x = 5
>>> callable(x)
False
>>> x = chr(97)
>>> x
'a'
>>> x = compile('55', 'test', 'eval')
<code object <module> at 0x79901564b050, file "test", line 1>
>>> exec(x)
```

```
'a'
>>> x = compile('55', 'test', 'eval')
>>> exec(x)
<code object <module> at 0x79901564b050, file "test", line 1>
>>> exec(x)
55
>>> x = complex(3,5)
>>> x
(3+5j)
>>> class Person
  File "<stdin>", line 1
    class Person
  ^
SyntaxError: expected ':'
>>> age = 36
>>> delattr('age')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: delattr expected 2 arguments, got 1
>>> x = dict(name = "John", age = 36, country = "Norway")
>>> x
{'name': 'John', 'age': 36, 'country': 'Norway'}
>>> class Person:
...     name = "John"
...     age = 36
...     country = "Norway"
>>> print(dir(Person))
['_class_', '_delattr_', '_dict_', '_dir_', '_doc_', '_eq_', '_format_', '_ge_', '_getattr_', '_gt_', '_hash_', '_init_', '_init_subclass_', '_le_', '_lt_', '_module_', '_ne_', '_new_', '_reduce_', '_reduce_ex_', '_repr_', '_setattr_', '_sizeof_', '_str_', '_subclasshook_', '_weakref_', 'age', 'country', 'name']
>>> x = divmod(5, 2)
>>> x
(2, 1)
>>> x = ('apple', 'banana', 'cherry')
>>> y = enumerate(x)
>>> y
<enumerate object at 0x799015462c80>
>>> list(y)
[(0, 'apple'), (1, 'banana'), (2, 'cherry')]
>>> x = 'print(55)'
>>> eval(x)
55
>>> x = 'name = "John"\nprint(name)'
>>> exec(x)
John
>>> ages = [5, 12, 17, 18, 24, 32]
>>>
>>> def myFunc(x):
...     if x < 18:
...         return False
...     else:
...         return True
...
>>> adults = filter(myFunc, ages)
```

```
John
>>> ages = [5, 12, 17, 18, 24, 32]
>>>
>>> def myFunc(x):
...     if x < 18:
...         return False
...     else:
...         return True
...
>>> adults = filter(myFunc, ages)
>>>
>>> for x in adults:
...     print(x)
...
...
18
24
32
>>> x = float(3)
>>> x
3.0
>>> x = format(0.5, '%')
>>> x
's0.000000%'
>>> mylist = ['apple', 'banana', 'cherry']
>>> x = frozenset(mylist)
>>> x
frozenset({'banana', 'cherry', 'apple'})
>>> class Person:
...     name = "John"
...     age = 30
...     country = "Norway"
...
>>> x = getattr(Person, 'age')
>>>
>>> print(x)
30
>>>
>>> help(all)
>>>
>>> hash(all)
2045974022113
>>> x = hex(9)
>>> x
'0x9'
>>> x = id(4)
>>> x
133659762884944
>>> print("Enter your name:")
Enter your name:
>>> x = input()
hello
>>> x
'hello'
>>> x = int(3.5)
>>> x
3
```

```
>>> x = input()
hello
>>> x
'hello'
>>> x = int(3.5)
>>> x
3
>>> x = isinstance(5, int)
>>> x
True
>>> class myAge:
...     age = 30
...
>>> class myObj(myAge):
...     name = "John"
...     age = myAge
...
>>> x = isinstance(myObj, myAge)
>>> x
True
>>> x = iter(['apple', 'banana', 'cherry'])
>>> next(x)
'apple'
>>> next(x)
'banana'
>>> next(x)
'cherry'
>>> mylist = ['apple', 'banana', 'cherry']
>>> x = len(mylist)
>>> x
3
>>> x = list(['apple', 'banana', 'cherry'])
>>> x
['apple', 'banana', 'cherry']
>>> x = locals()
>>> x
{'_name_': 'main', '_doc_': None, '_package_': None, '_loader_': <class 'frozen.importlib.BuiltinImporter'>, '_spec_': None, '_annotations_': [], '_builtins_': <module 'builtins' (built-in)>, '__x__': {...}, 'mylist': ['apple', 'banana', 'cherry'], 'mytuple': (0, 1, False), 'age': 30, 'Person': <class '__main__.Person'>, 'y': <enumerate object at 0x799015652c80>, 'name': 'John', 'ages': [5, 12, 17, 18, 24, 32], 'myFunc': <function myFunc at 0x799016855e10>, 'adults': <filter object at 0x799015654ee0>, 'myAge': <class '__main__.myAge'>, 'myObj': <class '__main__.myObj'>}>
>>> def myFunc(n):
...     return len(n)
...
>>> x = map(myFunc, ('apple', 'banana', 'cherry'))
>>> x
<map object at 0x799015654d00>
>>> x = max(5, 10)
>>> x
10
>>> x = memoryview(b"Hello")
>>>
>>> print(x)
<memory at 0x7990157d3280>
>>>
>>> #return the Unicode of the first character
>>> print(x[0])
72
```

```
Activities Terminal Jul 25 23:56
Screenshot captured
You can paste the image from the clipboard.

...
>>> x = map(myfunc, ('apple', 'banana', 'cherry'))
>>> x
<map object at 0x799015654d08>
>>> x = max(5, 10)
>>> x
10
>>> x = memoryview(b"Hello")
>>>
>>> print(x)
<memory at 0x7990157d3280>
>>>
>>> #Return the Unicode of the first character
>>> print(x[0])
72
>>>
>>> #Return the Unicode of the second character
>>> print(x[1])
101
>>> x = oct(12)
>>> x
'014'
>>> f = open("demoFile.txt", "r")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or directory: 'demoFile.txt'
>>> print(f.read())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'f' is not defined
>>> x = ord('h')
>>> x
104
>>> x = pow(4, 3)
>>> x
64
>>> print("Hello World")
Hello World
>>> = range(6)
  File "<stdin>", line 1
    = range(6)
IndentationError: unexpected indent
>>> for n in x:
...     print(n)
...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not iterable
>>> x = range(6)
>>> for n in x:
...     print(n)
...
0
1
2
>>>
```

```
Activities Terminal Jul 25 23:56
Screenshot captured
You can paste the image from the clipboard.

>>> x = max(5, 10)
>>> x
10
>>> x = memoryview(b"Hello")
>>>
>>> print(x)
<memory at 0x7990157d3280>
>>>
>>> #Return the Unicode of the first character
>>> print(x[0])
72
>>>
>>> #Return the Unicode of the second character
>>> print(x[1])
101
>>> x = oct(12)
>>> x
'014'
>>> f = open("demoFile.txt", "r")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or directory: 'demoFile.txt'
>>> print(f.read())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'f' is not defined
>>> x = ord('h')
>>> x
104
>>> x = pow(4, 3)
>>> x
64
>>> print("Hello World")
Hello World
>>> = range(6)
  File "<stdin>", line 1
    = range(6)
IndentationError: unexpected indent
>>> for n in x:
...     print(n)
...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not iterable
>>> x = range(6)
>>> for n in x:
...     print(n)
...
0
1
2
3
4
5
>>>
```