```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello world!")
hello world!
>>> 2+3
5
>>> 3-2
1
>>> 2*3
6
>>> 10/5
2.0
>>> 11/2
5.5
>>> 11//2
5
>>> 10//2
5
>>> 10 % 2
0
>>> 11%2
1
>>> 2**2
4
>>> 2**3
8
>>> x = 10
>>> type(x)
<class 'int'>
>>> x = 'hello'
>>> type(x)
<class 'str'>
>>> name  = 'shivank'
>>> type(name)
<class 'str'>
>>> help()

Welcome to Python 3.8's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.8/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics".  Each module also comes
with a one-line summary of what it does; to list the modules whose name
```

```
or summary contain a given string such as "spam", type "modules spam".

help> print
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.

help>

You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)".  Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>> x = 10
>>> y = 20
>>> print(x, y)
10 20
>>> print(x, y, sep = '\n')
10
20
>>> print(x,y, sep='')
1020
>>> help()

Welcome to Python 3.8's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.8/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics".  Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.
```

```
False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not

help>

You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)".  Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>> import builtins
>>> print(dir(builtins))
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException',
'BlockingIOError', 'BrokenPipeError', 'BufferError', 'BytesWarning',
'ChildProcessError', 'ConnectionAbortedError', 'ConnectionError',
'ConnectionRefusedError', 'ConnectionResetError', 'DeprecationWarning', 'EOFError',
'Ellipsis', 'EnvironmentError', 'Exception', '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', 'WindowsError', 'ZeroDivisionError', '_',
'__build_class__', '__debug__', '__doc__', '__import__', '__loader__', '__name__',
'__package__', '__spec__', 'abs', 'all', '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']
>>> print([f for f in dir(builtins) if '__' not in f and f[0].islower()])
['abs', 'all', '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']
>>> x = 10
>>> print(x, type(x))
10 <class 'int'>
>>> float(x)
10.0
>>> y = 100.0
>>> int(x)
10
>>> int(y)
100
>>> var = '1.2'
>>> type(var)
<class 'str'>
>>> float(var)
1.2
>>> z = float(var)
>>> type(z)
<class 'float'>
>>> type(True)
<class 'bool'>
>>> int(True)
1
>>> int(false)
Traceback (most recent call last):
  File "<pyshell#41>", line 1, in <module>
    int(false)
NameError: name 'false' is not defined
>>> int(False)
0
>>> float(true)
Traceback (most recent call last):
  File "<pyshell#43>", line 1, in <module>
    float(true)
NameError: name 'true' is not defined
>>> float(True)
1.0
>>> type('hello world!')
<class 'str'>
>>> type("hello world!")
<class 'str'>
>>> fname = 'shivank'
>>> lname = 'singh'
>>> fname + lname
'shivanksingh'
```

```
>>> greet = 'Hello World!'
>>> greet[0]
'H'
>>> greet[2]
'l'
>>> greet[5]
' '
>>> greet[12]
Traceback (most recent call last):
  File "<pyshell#54>", line 1, in <module>
    greet[12]
IndexError: string index out of range
>>> greet[11]
'!'
>>> greet[-1]
'!'
>>> print(fname, lname)
shivank singh
>>> fname+ " " + lname
'shivank singh'
>>> greet
'Hello World!'
>>> print(greet[0:5])
Hello
>>> print(greet[4:9])
o Wor
>>> greet[8]
'r'
>>> greet[:5]
'Hello'
>>> greet[:]
'Hello World!'
>>> print(greet[:])
Hello World!
>>> print(greet[4:3])

>>> greet[-1: -7]
''
>>> greet[0:5:2]
'Hlo'
>>> greet[0:9:3]
'HlW'
>>> greet[0:11:3]
'HlWl'
>>> len(greet)
12
>>> greet[-12:-1]
'Hello World'
>>> greet()
Traceback (most recent call last):
  File "<pyshell#73>", line 1, in <module>
    greet()
TypeError: 'str' object is not callable
>>> greet
'Hello World!'
>>> greet.lower()
'hello world!'
>>> greet.upper()
'HELLO WORLD!'
>>> greet
'Hello World!'
>>> greet.split()
['Hello', 'World!']
>>> newVar = greet.upper()
>>> newVar
'HELLO WORLD!'
>>> greet
'Hello World!'
>>> x = "hello my name is shivank"
>>> print(x)
hello my name is shivank
>>> type(x)
<class 'str'>
>>> x.split()
['hello', 'my', 'name', 'is', 'shivank']
>>> x.split('a')
['hello my n', 'me is shiv', 'nk']
>>> fname
'shivank'
>>> x = 10
>>> printf("%d is my variable" , x)
Traceback (most recent call last):
  File "<pyshell#89>", line 1, in <module>
    printf("%d is my variable" , x)
NameError: name 'printf' is not defined
>>> .format
SyntaxError: invalid syntax
>>> fname
'shivank'
>>> print(f"this is my name : {name} ")
this is my name : shivank
>>> 
>>> print(f"this is my name : {fname} ")
this is my name : shivank
>>> name
'shivank'
>>> fname = 'sam'
>>>  print(f"this is my name : {name} ")

SyntaxError: unexpected indent
```

```
>>>  print(f"this is my name : {fname} ")

SyntaxError: unexpected indent
>>> print(f"this is my name : {name} ")
this is my name : shivank
>>> print(f"this is my name : {fname} ")
this is my name : sam
>>> printf("This is my name : {}".format(fname))
Traceback (most recent call last):
  File "<pyshell#101>", line 1, in <module>
    printf("This is my name : {}".format(fname))
NameError: name 'printf' is not defined
>>> print("This is my name : {}".format(fname))
This is my name : sam
>>> fname
'sam'
>>> fname = 'shivank'
>>> lname ='singh'
>>> print("Fulll name is {} {} ".format(fname,lname))
Fulll name is shivank singh
>>> print(f"full name : {fname} {lname}")
full name : shivank singh
>>> print("first name is",fname")

SyntaxError: EOL while scanning string literal
>>> print("first name is",fname)
first name is shivank
>>> print("first name is",fname + lname)
first name is shivanksingh
>>> print("first name is",fname + lname)
first name is shivanksingh
>>> print("first name is",fname , lname)
first name is shivank singh
>>> type(['c', 'c++', 'python'])
<class 'list'>
>>> lang = ['c', 'c++', 'python']
>>> print(lang)
['c', 'c++', 'python']
>>> lang2 = ['c', 100, 100.6]
>>> print(lang2)
['c', 100, 100.6]
>>> lang
['c', 'c++', 'python']
>>> lang[0]
'c'
>>> lang
['c', 'c++', 'python']
>>> lang[0]
'c'
>>> lang[0] = 'pearl'
>>> lang
['pearl', 'c++', 'python']
>>> lang[0]= 10
>>> lang
[10, 'c++', 'python']
>>> type(lang[0])
<class 'int'>
>>> lang[0]= 'c'
>>> print(lang)
['c', 'c++', 'python']
>>> dir(lang)
['__add__', '__class__', '__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']
>>> lang[0] = c
Traceback (most recent call last):
  File "<pyshell#130>", line 1, in <module>
    lang[0] = c
NameError: name 'c' is not defined
>>> lang[0] = '100'
>>> lang
['100', 'c++', 'python']
>>> type(lang[0])
<class 'str'>
>>> lang
['100', 'c++', 'python']
>>> lang.append("java")
>>> lang
['100', 'c++', 'python', 'java']
>>> lang.insert(1, 'c')
>>> print(lang)
['100', 'c', 'c++', 'python', 'java']
>>> lang.extend(['bash', 'pearl'])
>>> lang
['100', 'c', 'c++', 'python', 'java', 'bash', 'pearl']
>>> del lang
>>> lang
Traceback (most recent call last):
  File "<pyshell#142>", line 1, in <module>
    lang
NameError: name 'lang' is not defined
>>> lang = ['ruby', 'c', 'c++', 'python', 'java', 'bash', 'pearl']
>>> lang[0:3] = []
>>> lang
['python', 'java', 'bash', 'pearl']
```

```
>>> lang = ['ruby', 'c', 'c++', 'python', 'java', 'bash', 'pearl']
>>> lang.sort()
>>> print(lang)
['bash', 'c', 'c++', 'java', 'pearl', 'python', 'ruby']
>>> lang.sort(reverse = True)
>>> lang
['ruby', 'python', 'pearl', 'java', 'c++', 'c', 'bash']
>>>
```