

Data Types

Text Type	str; str()	'I'm a string'
Numeric Types	int; int()	10
	float; float()	10.3
Boolean Type	bool	True/False
Sequence Types	list; list()	[1, 2, 'a', 'b']
	tuple; tuple()	(1, 2, 3)
	range	range(4)
Set Type	set; set()	{1, 2, 3}
Mapping Type	dict; dict()	{1:'a', 2:'b', 3:'c'}

Booleans

Booleans as Numbers

True==1 False==0

Comparison Operators

<code>a==b</code>	is <i>a</i> equal to <i>b</i> ?	<code>a!=b</code>	is <i>a</i> different than <i>b</i> ?
<code>a<b</code>	is <i>a</i> less than <i>b</i> ?	<code>a<=b</code>	is <i>a</i> less than or equal to <i>b</i> ?
<code>a>b</code>	is <i>a</i> greater than <i>b</i> ?	<code>a>=b</code>	is <i>a</i> greater than or equal to <i>b</i> ?

Membership and Identity Operators

a in b	is <i>a</i> in <i>b</i> ?	a is b	are <i>a</i> and <i>b</i> the same object?
a not in b	is <i>a</i> not in <i>b</i> ?	a is not b	are <i>a</i> and <i>b</i> different objects?

Boolean Operators

not	returns <i>False</i> if operand is <i>True</i> , <i>True</i> otherwise
and	returns <i>True</i> if Both operands are <i>True</i> , <i>False</i> otherwise
or	returns <i>False</i> if both operands are <i>False</i> , <i>True</i> otherwise
^ (xor)	returns <i>False</i> if both or neither operands are <i>False</i> , <i>True</i> otherwise

Operator Precedence

1.	()	Parentheses are evaluated first
2.	**	Exponent
3.	+, -	unary + and - signs (e.g., -x)
4.	*, /, //, %	multiplication, division, floor division, and modulo
5.	+, -	Addition, subtraction
6.	^	Bitwise XOR
7.	in, not in, is, is not, <, <=, >, >=, !=, ==	Comparison, identity, and membership operators
8.	not	logical NOT
9.	and	Logical AND
10.	or	logical OR

Print Function

```
print('a', 'b', sep='*')      a*b
```

Decision Structure

```
if n == 0:
    print( "n is zero")
elif n > 0:
    print( "n is strictly positive")
else: # n < 0
    print( "n is strictly negative")
```

Repetition Structures

```
n = 0
while n < 4:
    print(n)
    n += 1
print( "n =", n)
# output is: 0 | 1 | 2 | 3 | n = 4
for i in range(4):
    print(i)
print( "i =", i)
# output is: 0 | 1 | 2 | 3 | i = 3
```

Exceptions

try:	Built-in Exceptions
# run this code	FileNotFoundError
except NameOf Err orT ype1:	IndexError
# handle error type 2	KeyError
except NameOf Err orT ype2:	ModuleNotFoundError
# handle error type 2	NameError
except:	SyntaxError
# handle any other error	TypeError
else:	ValueError
# run this code if no error	ZeroDivisionError
finally:	
# always run this code	

NumPy

```
import numpy as np
```

Creating arrays

np.array([1,2,3])	Convert python list to NumPy array
np.arange(1,5)	Return sequence from start (incl.) to end (excl.)
np.arange(1,5,2)	Return stepped sequence from start (incl.) to end (excl.)
np.repeat([1,3,-6],3)	Repeat values n times: [1,1,1,3,...]
np.tile([1,3,6],3)	Repeat values n times: [1,3,6,1,...]

Math functions and methods

All functions take an array as the input



By TeunR
cheatography.com/teunr/

Not published yet.
Last updated 3rd June, 2025.
Page 2 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>