

Variable Assignment

temporarily store data	<code>x = 5</code>
	<code>x = 'one'</code>

Calculations :

sum	<code>y = x+2</code>
Subtraction	<code>y = x-2</code>
Multiplication	<code>y = x * 2</code>
Exponentiation	<code>y = x ** 2</code>
Remainder	<code>y = x % 2</code>
Division	<code>y = x/2</code>

Comparison Operators

<code>x < y</code>	Less
<code>x <= y</code>	Less or eq
<code>x > y</code>	Greater
<code>x >= y</code>	Greater or eq
<code>x == y</code>	Equal
<code>x != y</code>	Not equal

Type Conversion

<code>str()</code>	'5', '3.45', 'True'
<code>int()</code>	5, 3, 1
<code>float()</code>	5.0, 1.0
<code>bool()</code>	True, True, False

string methods

<code>my_string.upper()</code>	String to uppercase
<code>my_string.lower()</code>	String to lowercase
<code>my_string.count('w')</code>	Count String elements
<code>my_string.replace('e', 'i')</code>	Replace String elements
<code>my_string.strip()</code>	Strip whitespaces

functions

```
def user(name):
    print("Hi {name}")
user("\n")
```

string

```
my_string = 'thisString'
my_string * 2      'thisStringthis-String'
>>> my_string + 'isnew'      'thisStringisnew'
't' in my_string    True
```

lists

```
Lists numbers = [1, 2, 3, 4, 5]
numbers[0]      returns the first item
```

lists

```
numbers.a-      # adds 6 to the end
append(6)

numbers.r-      # removes 6
remove(6)

numbers.i-      # adds 6 at index position of
insert(0, 6)    0

number-        # removes the last item
s.pop()

numbers.c-      # removes all the items
lear()

numbers.i-      # returns the index of first
index(8)        occurrence of 8

numbers.s-      # sorts the list
ort()

numbers.c-      # returns a copy of the list
opy()

numbers.r-      # reverses the list
erse()
```

Statements

```
If Statement
if expres sion:
    sta tements
elif expres sion:
    sta tements
else:
    sta tements
While Loop
while expres sion:
    sta tements
For Loop
for var in collec tion:
    sta tements
Counting For Loop
for i in range( start, end [,
step]):
    sta tements
```

Exception Handling

```
try:
    sta tements
except [ exception type [ as var
] ]:
    sta tements
finally:
    sta tements
```

C

By **romina**
cheatography.com/romina/

Not published yet.
 Last updated 29th August, 2022.
 Page 1 of 1.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>