

Files

```
file = open("data.txt") # open file
contents = file.read() # whole file
as string
lines = file.readlines() # all lines
as list
line = file.readline() # one line
for line in file: # loop lines
line = line.strip() # remove \n and
spaces
parts = line.split(",") # split by
comma
file.close() # close file
```

exceptions

except ZeroDivisionError:

except ValueError:

except IndexError:

except TypeError:

list slicing

```
nums = [10, 20, 30, 40, 50, 60]
nums[1:4] [20, 30, 40]
nums[:3]. [10, 20, 30]
nums[-2:] last 2 items
nums[:-1] all but last
nums[::-2] every 2nd item
nums[::-1]. reverse
```

If statment

```
if num % 2 == 0: #even
if num % 2 != 0: #odd
6 / 2 #3.0 7 // 2 #3
7 % 2 #1 3 ** 2 #9
```

matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
# ===== BASIC LINE PLOT
=====
x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y, label='sin(x)') # Line
plot
plt.title("Sine Curve") # Title
plt.xlabel("x-axis") # X label
plt.ylabel("y-axis") # Y label
plt.grid(True) # Show grid
plt.legend() # Show legend
plt.xlim(0, 10) # Limit x-axis
plt.ylim(-1, 1) # Limit y-axis
plt.xticks([0, 5, 10]) # Custom x
ticks
plt.yticks([-1, 0, 1]) # Custom y
ticks
plt.savefig("sine_plot.png") #
Save to file
```

matplotlib (cont)

```
plt.show() # Show the plot
# ===== SCATTER PLOT
=====
x = np.random.rand(50)
y = np.random.rand(50)
plt.scatter(x, y)
plt.title("Random Scatter")
plt.show()
# ===== BAR CHART
=====
categories = ['A', 'B', 'C']
values = [10, 20, 15]
plt.bar(categories, values)
plt.title("Bar Chart")
plt.show()
# ===== HISTOGRAM
=====
data = np.random.randn(1000)
plt.hist(data, bins=30)
plt.title("Histogram")
plt.show()
# ===== CLEARING /
CLOSING =====
plt.clf() # Clear current figure (if
needed)
plt.close() # Close the plot
window
```

For Loops

```
for i in range(5): print(i) # 0 1 2 3
4
for i in range(2,6): print(i) # 2 3 4
5
for i in range(1,10,3): print(i) # 1
4 7
lst=['a','b','c']
for i in range(len(lst)): print(i,l-
st[i]) # 0 a 1 b 2 c
for item in lst: print(item) # a b c
for i,item in enumerate(lst):
print(i,item) # 0 a 1 b 2 c
```

switching while loop to for loop

```
# For loops
for op1 in range(0, value1):
for op2 in range(value1, value2,
-1):
print(f"{op1:3} x {op2:3} =
{op1*op2:3}")
# Equivalent while loops
op1 = 0
while op1 < value1:
op2 = value1
while op2 > value2:
print(f"{op1:3} x {op2:3} =
{op1*op2:3}")
op2 -= 1
op1 += 1
```



By **luckytime**

cheatography.com/luckytime/

Not published yet.

Last updated 17th June, 2025.

Page 1 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Sets and Dictionaries

```
# Sets from list and operations
lst=[1,2,2,3,4,4]
s=set(lst) # {1,2,3,4}
s.add(5) # {1,2,3,4,5}
s.remove(2) # {1,3,4,5}
s.discard(6) # {1,3,4,5} no error
print(3 in s) # True
s2={3,4,5,6}
print(s.union(s2)) # {1,3,4,5,6}
print(s.intersection(s2)) # {3,4,5}
print(s.difference(s2)) # {1}
print(s.issubset(s2)) # False
for x in s: print(x) # 1 3 4 5 (any order)

# Dictionaries and methods
d={"a":1,"b":2}
print(d.keys()) # dict_keys(['a','b'])
print(d.values()) # dict_values([1,2])
print(d.items()) # dict_items([('a',1),('b',2)])
print(d.get("a")) # 1
print(d.get("z",0)) # 0 default if missing
```

Sets and Dictionaries (cont)

```
d["c"]=3
for k in d: print(k) # a b c
for v in d.values(): print(v) # 1 2 3
for k,v in d.items(): print(k,v) # a 1 b 2 c 3
```

Lists

```
fruits = ["apple", "banana", "cherry"]
fruits[0] # apple
fruits[-1] # cherry
len(fruits) # 3
fruits.append("kiwi")
fruits.insert(1, "pear")
fruits.remove("banana")
fruits.pop() # removes last
fruits.sort()
sorted(fruits)
```



By **luckytime**

cheatography.com/luckytime/

Not published yet.

Last updated 17th June, 2025.

Page 2 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>