

Random module

random.random()	random float between 0.0 and 1.0
random.uniform(a, b)	random float between a and b
random.randint(a, b)	random integer between a and b
random.randrange(0, 10, 2)	random number from [0, 2, 4, 6, 8, 10]
random.choice(list)	random element from a list
random.choices(list, weights=None, k=2)	k no. of random elements from a list with replacement, weights is a list that specifies the probability of choosing a specific element
random.sample(list, k=2)	k no. of unique elements(no replacement)
random.shuffle(list)	shuffles a list
random.seed(a=None)	use this to get the same result every time

Types of errors

NameError	Doesn't recognize the name you are using
TypeError	When you try to combine or manipulate data in a way python doesn't allow
IndexError	The index doesn't exist
KeyError	When you try to access a value in a dictionary using a key that doesn't exist
ZeroDivisionError	When you divide a number by 0
ValueError	Function receives a correct type but invalid value
AttributeError	Invalid attribute or method for an object
ImportError / ModuleNotFoundError	Failed to import a module

Types of errors (cont)

FileNotFoundError	File does not exist when trying to open it
-------------------	--

Pandas module

df = pd.DataFrame(dictionary)	To convert a dictionary into a pandas dataframe
df = pd.read_csv('file.csv')	To convert a csv file into a dataframe
df = pd.read_excel('file.xlsx')	To convert an excel file into a dataframe
df = pd.read_json('file.json')	To convert a json file into a dataframe
df.to_csv('output.csv', index=False)	Convert a dataframe into a csv file
df.to_excel('output.excel')	Convert a dataframe into an excel file
df.head(k)	First k rows, leave empty for five
df.tail(k)	Last k rows, leave empty for five
df.info()	Data types and non-null values
df.describe()	Summary statistics
df.shape	No. of rows and columns
df.columns	Column names
df.dtypes	Data types
df['col']	A specified column
df.iloc[k, l]	A specified cell by index, leave l empty for an entire row
df.loc[k, 'col']	A specified cell by index, 'col' is column name
df[0:5]	Slicing rows
df[df['col'] > 25]	Filter data by condition
df[(df['col'] > 25 & (df['Age'] < 40))]	Filter data by multiple conditions



By kush9220

cheatography.com/kush9220/

Not published yet.

Last updated 16th July, 2025.

Page 1 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Pandas module (cont)		Matplotlib module	
<code>df[df['Name'].isin(['Alice'])]</code>	Filter by values	<code>plt.plot(x, y, color='red', linestyle='--', marker='o', label='line 1')</code>	Line plot with color red, dashed lines, o marker labeled as 'line 1'
<code>df.rename(columns={'old': 'new'})</code>	Renaming a column	<code>plt.title("Title")</code>	Set title of the chart
<code>df.drop(columns=['Col1', 'Col2'])</code>	Dropping columns	<code>plt.xlabel("x-axis")</code>	Label of x-axis
<code>df.drop(index=[0, 1])</code>	Dropping rows	<code>plt.ylabel("y-axis")</code>	Label of y-axis
<code>df[col].sum()</code>	Sum of values in col	<code>plt.legend()</code>	Show legend
<code>df[col].mean()</code>	Mean of values in col	<code>plt.grid(True)</code>	Show grid
<code>df[col].value_counts()</code>	Number of values in col	<code>plt.show()</code>	Display the chart
<code>df.groupby(col).mean()</code>	Grouped stats	<code>plt.figure(figsize=(6, 4))</code>	Set figure size
<code>df.isnull()</code>	Returns null values of boolean dataframes	<code>plt.subplot(2, 1, 1)</code>	2 rows, 1 column, 1st plot
<code>df.isnull().sum()</code>	No. of null values	<code>plt.tight_layout()</code>	Avoid overlap
<code>df.dropna()</code>	Drop the row with null values	<code>plt.scatter(x, y)</code>	Scatter plot
<code>df.fillna(k)</code>	Fill the missing values with value k	<code>plt.bar(x, y)</code>	Bar plot
<code>df['col'] = df['col'].str.strip()</code>	Remove whitespace	<code>plt.barh(x, y)</code>	Horizontal bar plot
<code>df['col'] = df['col'].str.lower()</code>	Present data in lowercase	<code>plt.hist(list, bins=5)</code>	Histogram plot
<code>df['col'] = pd.to_datetime(df['col'])</code>	Convert to datetime	<code>plt.pie(data_list, labels=label_list, autopct='%1.1f%%')</code>	Pie chart plot
<code>df.sort_values('Age')</code>	Sort data by age	<code>plt.style.use('ggplot')</code>	Set global chart style
<code>df.sort_values(['Age', 'Name'])</code>	Sort data by multiple values	<code>plt.style.available</code>	Show all chart styles
<code>df.reset_index(drop=True)</code>	Reset index	<code>plt.savefig('plot.pdf', dpi=300)</code>	Save chart as pdf with resolution
<code>pd.concat([df1, df2])</code>	Appending rows	<code>plt.savefig('plot.png')</code>	Save chart as png
<code>pd.merge(df1, df2, on='ID')</code>	Joining data by column value	<code>plt.text(2, 20, "Sample Text")</code>	Add sample text to x=2, y=20
<code>pd.merge(df1, df2, how='left', on='ID')</code>	Left joining data by column value	<code>plt.annotate("Important", xy=(2, 20), xytext=(3, 25), arrowprops=dict(facecolor='black'))</code>	For annotating
<code>df.pivot_table(index='Gender', values='Age', aggfunc='mean')</code>	Create a pivot table with mean of the values categorized by index	<code>plt.xscale('log')</code>	Logarithmic x-axis
		<code>plt.yscale('log')</code>	Logarithmic y-axis
		<code>plt.xlim(0, 5)</code>	X-axis limits
		<code>plt.ylim(0, 5)</code>	Y-axis limits
		<code>plt.xticks([1, 2, 3])</code>	Custom ticks in x-axis
		<code>plt.yticks([1, 2, 3])</code>	Custom ticks in y-axis

Plotly module

```
import plotly.graph_objects as go
```

```
import plotly.express as px
```

```
df = px.data.gapminder()
```

Returning a Gapminder dataset as a pandas dataframe

```
px.line(df[df['country'] == 'India'],
x='year', y='gdpPercap', title='GDP
over time')
```

Line plot country dataframe, x=year, y=gdp-percap and title is GDP over time

```
px.bar(x=['A', 'B'], y=[10, 20], title='Bar
Plot')
```

Bar plot

```
px.scatter(df, x='gdpPercap', y='lifeExp',
color='continent', title='GDP vs
Life Expectancy')
```

Scatter plot

```
px.scatter(df, x='gdpPercap', y='lifeExp',
size='pop', color='continent',
hover_name='country', log_x=True)
```

Bubble sort

```
px.choropleth(df[df['year']==2007],
locations="iso_alpha", color="lifeExp",
hover_name="country")
```

Map plot (Choropleth)

```
fig.update_layout(title='New Title',
xaxis_title='X Axis', yaxis_title='Y
Axis', template='plotly_dark')
```

To customize layout

```
fig.add_trace(go.Scatter(x=[1, 2, 3], y=
[4, 5, 6], mode='lines+markers',
name='Line'))
```

Line plot

```
fig = go.Figure(go.Bar(x=['A', 'B'], y=
[10, 15]))
```

Bar plot

```
go.Figure(go.Pie(labels=['A', 'B'],
values=[30, 70]))
```

Pie plot

```
fig.write_html("plot.html")
```

Save as html file

```
fig.write_image("plot.png")
```

Save as image file

Plotly module (cont)

```
fig.update_layout(hovermode='x unified')
```

Tooltip follows x

```
fig.update_traces(marker=dict(size=10))
```

Change marker size

```
fig.update_layout(dragmode='zoom')
```

Default zoom tool

```
fig.update_layout(template='plotly_dark')
```

Update the style of theme

```
px.scatter_geo(px.data.gapminder()).query("year==2-
007"), locations="iso_alpha", color="continent",
size="pop")
```

Map visualizations

```
from plotly.subplots import make_subplots
```

```
fig = make_subplots(rows=1, cols=2)
```

To set subplots

```
fig.add_trace(go.Scatter(x=[1, 2], y=[3, 4]), row=1,
col=1)
```

add trace in a subplot

Glob module

```
glob.glob("*.txt")
```

All .txt files in current directory

```
glob.glob("*/txt", recursive=True)
```

Match files in subdirectories

```
glob.glob("*.txt", recursive=False, includ-
e_hidden=False)
```

Sort matched files

Glob module works best when worked with Regex expressions

Types of data structures

Lists Indexing, Slicing, Extending and Mutability, syntax: `my_list = [1, 1.21, "hello", True]`

Tuples Indexing, Slicing and Immutable, syntax: `my_tuple = (1, 10, "hello")`

Sets Unordered nature, Key operations are `add()`, `remove()`, `union()`, `intersection()`, `difference()`, syntax: `my_set = {1, 2, 3, 3}`



By **kush9220**

cheatography.com/kush9220/

Not published yet.

Last updated 16th July, 2025.

Page 3 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Types of data structures (cont)

Dictionary Accessing values by key, Mutability and flexibility, common operations are `get()`, `items()`, `keys()`, `values()`, `update()`, syntax: `my_dict = {"name": "Alice", "age": 30, "city": "New York"}`

Pytest module

<code>assert result == k</code>	checks if the result variable is the same as the variable assigned as k
<code>@pytest.fixture</code>	to define a fixture to use as a reusable piece of code to use before or after a test
<code>@pytest.mark.parametrize("a, b, result", [(1, 2, 3), (4, 5, 9)])</code>	checks the result variable with a and b by performing numerous tests based on the data we give
<code>@pytest.mark.skip(reason="Not implemented yet")</code>	skip a particular test
<code>@pytest.mark.skipif(condition, reason="...")</code>	skip the test given the condition
<code>@pytest.mark.xfail</code>	If you are expecting a test to fail
<code>pytest.raises()</code>	to raise a specific type of error

Numpy module

<code>np.array([1, 2, 3], [4, 5, 6])</code>	Creating a 2D array
<code>np.zeros((3, 3))</code>	3x3 array of zeros
<code>np.ones((3, 3))</code>	3x3 array of ones
<code>np.full((2, 2), 7)</code>	2x2 array of sevens
<code>np.eye(3)</code>	Identity matrix 3x3
<code>np.arange(0, 10, 2)</code>	An array of this: [0, 2, 4, 6, 8]
<code>np.linspace(0, 1, 5)</code>	5 values from 0 to 1
<code>arr.shape</code>	Dimensions of the array

Numpy module (cont)

<code>arr.ndim</code>	No. of dimensions
<code>arr.size</code>	Total no. of elements
<code>arr.dtype</code>	Data type
<code>arr.reshape((2, 3))</code>	Reshape an array to 2x3
<code>arr.ravel()</code>	Compress an array to 1D
<code>arr.T</code>	Transpose the array
<code>np.add(a, b)</code>	$a + b$
<code>np.subtract(a, b)</code>	$a - b$
<code>np.multiply(a, b)</code>	$a * b$
<code>np.divide(a, b)</code>	a / b
<code>np.power(a, 2)</code>	a to the power of 2
<code>np.sqrt(a)</code>	Square root of a
<code>np.exp(a)</code>	Exponential value of a
<code>np.log(a)</code>	Natural log of a
<code>np.mean(list)</code>	Mean of the list
<code>np.median(list)</code>	Median of the list
<code>np.std(list)</code>	Standard deviation of the list
<code>np.sum(list)</code>	Sum of the list
<code>np.max(list)</code>	Maximum value in a list
<code>np.min(list)</code>	Minimum value in a list
<code>np.argmax(list)</code>	Index of maximum value
<code>np.argmin(list)</code>	Index of minimum value
<code>np.concatenate([a, b])</code>	Join arrays
<code>np.vstack([a, b])</code>	Stack vertically
<code>np.hstack([a, b])</code>	Stack horizontally
<code>np.split(a, 3)</code>	Split the array into 3 parts
<code>np.unique(a)</code>	Unique elements of the array
<code>np.random.rand(2, 2)</code>	a 2x2 array of random elements from 0 to 1
<code>np.random.randn(2, 2)</code>	a 2x2 array of random elements, this will be a normal distribution
<code>np.random.randint(0, 10, size=5)</code>	a 1D array of 5 random integers from 0 to 10
<code>np.isnan(a)</code>	Check for NaN values
<code>np.isinf(a)</code>	Check for Inf values

Numpy module (cont)

<code>np.nan_to_num(a)</code>	Convert NaN to 0
<code>np.clip(a, 0, 1)</code>	Limit values between 0 to 1
<code>np.where(a > 0, 1, 0)</code>	Conditional values
<code>np.cumsum(a)</code>	Cumulative sum
<code>np.cumprod(a)</code>	Cumulative product

Bokeh module

<code>from bokeh.plotting import figure, show</code>	
<code>from bokeh.io import output_file, output_notebook</code>	
<code>from bokeh.layouts import column, row</code>	
<code>output_file("plot.html")</code>	Output to html file
<code>output_notebook()</code>	Output to Jupyter notebook
<code>p = figure(title="Simple Line", x_axis_label='x', y_axis_label='y')</code>	Label the figure
<code>p.line([1, 2, 3], [4, 6, 2])</code>	Line plot
<code>show(p)</code>	Show the chart
<code>p.circle(x, y, size=10)</code>	Scatter plot
<code>p.vbar(x=x, top=y, width=0.5)</code>	Vertical bar plot
<code>p.hbar(x=x, top=y, width=0.5)</code>	Horizontal bar plot
<code>p.triangle(x, y, size=12, color="-green")</code>	Shape plot, other glyphs available ex: square, diamond etc.
<code>p.title.text = "Custom Title"</code>	Set title
<code>p.xaxis.axis_label = "X Axis"</code>	Label x-axis
<code>p.yaxis.axis_label = "Y Axis"</code>	Label y-axis
<code>p.background_fill_color = "lightgray"</code>	Set background color
<code>p.border_fill_color = "whitesmoke"</code>	Set border color
<code>p.outline_line_color = "black"</code>	Set outline line color

Bokeh module (cont)

<code>p.line(x, y, legend_label="My Line", line_width=2)</code>	define legend_label for legend
<code>p.legend.location = "top_left"</code>	Set interactive legend
<code>p.legend.click_policy = "hide"</code>	
<code>layout = row(p1, p2)</code>	To set layout of a row
<code>layout = column(p1, p2)</code>	To set layout of a column
<code>show(layout)</code>	Show layout
<code>from bokeh.models import ColumnDataSource</code>	
<code>source = ColumnDataSource(data={'x': [1, 2, 3], 'y': [4, 6, 5]})</code>	Set a data source
<code>p.circle(x='x', y='y', source=source, size=10)</code>	Plot a circle chart from data source
<code>from bokeh.io.export import export_png</code>	
<code>export_png(p, filename="plot.png")</code>	Export chart to png file
<code>p1.x_range = p2.x_range</code>	Link x-axis
<code>p1.y_range = p2.y_range</code>	Link y-axis
<code>from bokeh.embed import components</code>	
<code>script, div = components(p)</code>	Use in html templates

Os module

<code>os.getcwd()</code>	Returns the current working directory
<code>os.chdir('path/to/directory')</code>	Changes current working directory
<code>os.listdir('path')</code>	Lists files and folders in the specified path
<code>os.mkdir('dirname')</code>	Creates a single directory
<code>os.makedirs('dir/subdir')</code>	Creates intermediate directories as needed
<code>os.rmdir('dirname')</code>	Removes an empty directory
<code>os.removedirs('dir/subdir')</code>	Removes nested empty directories
<code>os.remove('filename')</code>	Removes a file
<code>os.path.exists('path')</code>	Returns true if path exists



By **kush9220**

cheatography.com/kush9220/

Not published yet.

Last updated 16th July, 2025.

Page 5 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Os module (cont)

<code>os.path.isfile('path')</code>	True if it's a file
<code>os.path.isdir('path')</code>	True if it's a directory
<code>os.path.join('folder', 'file.txt')</code>	Combines path using the right separator
<code>os.path.basename('path/to/file.txt')</code>	Returns file.txt
<code>os.path.dirname('path/to/file.txt')</code>	Returns 'path/to'
<code>os.path.split('path/to/file.txt')</code>	Returns ('path/to', 'file.txt')
<code>os.path.abspath('file.txt')</code>	Returns absolute path to the file
<code>os.environ.get('HOME')</code>	Retrieves the path to the current user's home directory
<code>os.environ['MY_VAR'] = 'value'</code>	Sets or creates an environment variable within the current environment process
<code>os.system('ls')</code>	Executes a shell command
<code>os.getpid()</code>	Current process ID
<code>os.getppid()</code>	Current parent process ID
<code>os.rename('old.txt', 'new.txt')</code>	Renaming a file or a directory

Shutil module

<code>shutil.copy(src, destination)</code>	Copies file to destination
<code>shutil.copy2(src, dst)</code>	It's like copying but preserves metadata
<code>shutil.copymode(src, dst)</code>	Copies file permissions only
<code>shutil.copystat(src, dst)</code>	Copies file's metadata only
<code>shutil.copytree(src, dst)</code>	Copies entire directory tree
<code>shutil.move(src, dst)</code>	Moves or renames a file
<code>shutil.rmtree('dir')</code>	Deletes directory and everything in it
<code>shutil.make_archive(base_name, format, root_dir)</code>	Creates archive in any format

Shutil module (cont)

<code>shutil.unpack_archive(filename, extract_dir)</code>	Unpacks the archive
<code>shutil.disk_usage(path)</code>	Gets disk usage stats

Re module

<code>re.search(pattern, string)</code>	Searches for first match everywhere
<code>re.match(pattern, string)</code>	Checks for a match only at the beginning
<code>re.fullmatch(pattern, str)</code>	Matches entire string to pattern
<code>re.findall(pattern, string)</code>	Returns all non-overlapping matches
<code>re.finditer(pattern, string)</code>	Returns iterators yielding match objects
<code>re.sub(pattern, repl, string)</code>	Replace matches with repl
<code>re.split(pattern, string)</code>	Split string by the matches
<code>re.compile(pattern)</code>	Precompile a pattern for reuse

Regex Expressions used in Python:

<code>.</code>	Matches any character except newline, use like this: a.b.c to match a1b7c, asbfc, a9bkc etc.
<code>?</code>	Use after a character to define it occurs 0 or 1 times
<code>\</code>	To define a Regex pattern / Escape character
<code>*</code>	Use after a pattern to define 0 or more repetitions
<code>+</code>	Use after a pattern to define 1 or more repetitions
<code>^</code>	Use before a pattern to define start of a string
<code>\$</code>	Use after a pattern to define end of string

Re module (cont)

{k}	Use to define k number of repetitions for a pattern
{k, l}	Use to define between k and l repetitions
[]	define a list of characters and use if you match from one of them
\d	Specifies digits [0-9]
\D	Anything that's not a digit
\w	Any word character [a-zA-Z0-9_]
\W	Anything that is not a word character
\s	Whitespace \ Spacing between two words
\S	Non-whitespace
\b	Word boundary, used to match whole words only like: \bcat\b to match 'cat', 'little cat' and not 'tomocat' or 'catatine'
\B	Non-word boundary, best used to match a word which has that letter like: \bun\b matches 'unmalicious', 'unnasty' and not 'un' or 'we un'



By **kush9220**

cheatography.com/kush9220/

Not published yet.

Last updated 16th July, 2025.

Page 7 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>