

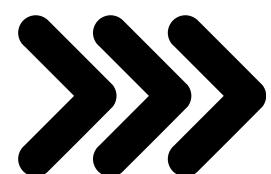


Tajamul Khan

Python for Data Science Cheat Sheet



@Tajamulkhan



Array Creation

- **np.array():** Creates an array from a list or tuple.
- **np.zeros():** Generates an array filled with zeros.
- **np.ones():** Generates an array filled with ones.
- **np.arange(start, stop, step):** Creates an array with evenly spaced values within a range.
- **np.linspace(start, stop, num):** Generates an array with evenly spaced numbers between two values.
- **np.eye(n):** Creates an identity matrix.
- **np.full(shape, fill_value):** Creates an array filled with a specified value.
- **np.empty():** Allocates an array without initializing its values.



@Tajamulkhan



Array Operations

- **np.add(a, b)**: Element-wise addition of arrays.
- **np.subtract(a, b)**: Element-wise subtraction of arrays.
- **np.multiply(a, b)**: Element-wise multiplication.
- **np.divide(a, b)**: Element-wise division.
- np.sqrt(a): Computes the square root of each element.
- **np.power(a, b)**: Raises each element of a to the power of b.
- **np.mod(a, b)**: Element-wise modulus operation.
- **np.clip(a, min, max)**: Limits values in an array to a specified range.



@Tajamulkhan



Statistical Functions

- **np.mean():** Computes the mean of the array.
- **np.median():** Computes the median of the array.
- **np.std():** Computes the standard deviation.
- **np.var():** Computes the variance.
- **np.min():** Returns the minimum value in the array.
- **np.max():** Returns the maximum value in the array.
- **np.percentile(a, q):** Computes the qth percentile.
- **np.average():** Computes the weighted average of an array.



@Tajamulkhan



Aggregate Functions

- **np.sum():** Computes the sum of elements.
- **np.prod():** Computes the product of elements.
- **np.cumsum():** Cumulative sum of array elements.
- **np.cumprod():** Cumulative product of array elements.
- **np.argmax():** Index of the maximum value in an array.
- **np.argmin():** Index of the minimum value in an array.
- **np.all():** Checks if all elements evaluate to True.
- **np.any():** Checks if any element evaluates to True.



@Tajamulkhan



Linear Algebra

- **np.dot(a, b)**: Computes the dot product of two arrays.
- **np.matmul(a, b)**: Performs matrix multiplication.
- **np.linalg.inv(a)**: Computes the inverse of a matrix.
- **np.linalg.det(a)**: Computes the determinant of a matrix.
- **np.linalg.eig(a)**: Computes the eigenvalues and eigenvectors.
- **np.linalg.norm(a)**: Computes the norm of a vector or matrix.
- **np.linalg.qr(a)**: Computes the QR decomposition.
- **np.linalg.svd(a)**: Computes the Singular Value Decomposition (SVD).



@Tajamulkhan



Random Operations

- **np.random.rand()**: Generates random numbers in [0, 1).
- **np.random.randint(low, high)**: Generates random integers within a range.
- **np.random.normal(mean, std)**: Generates random samples from a normal distribution.
- **np.random.choice()**: Chooses random elements from an array.
- **np.random.shuffle()**: Shuffles an array in place.
- **np.random.seed()**: Sets the seed for random number generation.
- **np.random.uniform()**: Generates samples from a uniform distribution.
- **np.random.poisson()**: Generates samples from a Poisson distribution.



@Tajamulkhan



Shape Operations

- **np.reshape(a, new_shape)**: Reshapes an array without changing data.
- **np.flatten()**: Flattens a multi-dimensional array.
- **np.transpose()**: Permutes the dimensions of an array.
- **np.resize()**: Changes the shape and size of an array.
- **np.expand_dims()**: Expands an array by adding a dimension.
- **np.squeeze()**: Removes axes of length one.
- **np.ravel()**: Returns a flattened array.
- **np.hstack(), np.vstack()**: Stacks arrays horizontally or vertically.



@Tajamulkhan



Indexing Slicing

- **np.where(condition):** Returns indices where a condition is True.
- **np.take(a, indices):** Selects elements from an array.
- **np.unique(a):** Finds unique elements in an array.
- **np.nonzero():** Returns indices of non-zero elements.
- **np.argsort():** Returns indices to sort an array.
- **np.split(a, indices):** Splits an array into sub-arrays.
- **np.flip():** Reverses the order of elements.
- **np.roll():** Rolls elements along an axis.



@Tajamulkhan



Create Dataframe

- **pd.DataFrame(data)**: Creates a DataFrame from data.
- **pd.Series(data)**: Creates a Series.
- **pd.read_csv(filepath)**: Reads a CSV file into a DataFrame.
- **pd.read_excel(filepath)**: Reads an Excel file into a DataFrame.
- **pd.DataFrame.from_dict(data)**: Creates a DataFrame from a dictionary.
- **pd.read_sql()**: Reads data from a SQL database.
- **pd.read_json()**: Reads a JSON file.
- **pd.read_html()**: Parses HTML tables into DataFrames.



@Tajamulkhan



Data Operations

- **df.head():** Displays the first 5 rows.
- **df.tail():** Displays the last 5 rows.
- **df.info():** Displays a summary of the DataFrame.
- **df.describe():** Provides summary statistics.
- **df.shape:** Returns the dimensions of the DataFrame.
- **df.columns:** Lists column names.
- **df.dtypes:** Returns the data types of columns.
- **df.memory_usage():** Returns memory usage.



@Tajamulkhan



Data Selection

- **df.iloc[row, col]**: Selects data by integer-based indexing.
- **df.loc[row, col]**: Selects data by label-based indexing.
- **df.at[row, col]**: Accesses a single value by label.
- **df.iat[row, col]**: Accesses a single value by integer index.
- **df.query(expr)**: Queries the DataFrame with a string expression.
- **df.filter()**: Filters rows or columns.
- **df.sample()**: Randomly samples rows.
- **df.select_dtypes()**: Selects columns of a specific data type.



@Tajamulkhan



Column Operations

- **df['col']:** Accesses a column.
- **df.drop(labels):** Drops rows or columns.
- **df.rename():** Renames rows or columns.
- **df.set_index(col):** Sets a column as the index.
- **df.reset_index():** Resets the index to default.
- **df.insert():** Inserts a new column.
- **df.assign():** Creates new columns.
- **df.pop():** Removes a column.



@Tajamulkhan



**Found
Helpful?**

Repost



Follow for more!