

## **Panda Pocka Card Template**

### **Data Frame Attributes**

<b>df.<a href="#">T</a></b>	<b>Transpose index and columns.</b>
<b>df.<a href="#">at</a></b>	<b>Access a single value for a row/column label pair.</b>
<b>df.<a href="#">axes</a></b>	<b>Return a list representing the axes of the DataFrame.</b>
<b>df.<a href="#">columns</a></b>	<b>The column labels of the DataFrame.</b>

<b>df.<a href="#">dtypes</a></b>	<b>Return the dtypes in the DataFrame.</b>
<b>df.<a href="#">empty</a></b>	<b>Indicator whether DataFrame is empty.</b>
<b>df.<a href="#">ftypes</a></b>	<b>Return the ftypes (indication of sparse/dense and dtype) in DataFrame.</b>
<b>df.<a href="#">iat</a></b>	<b>Access a single value for a row/column pair by integer position.</b>

<b>df.<a href="#">iloc</a></b>	<b>Purely integer-location based indexing for selection by position.</b>
<b>df.<a href="#">index</a></b>	<b>The index (row labels) of the DataFrame.</b>
<b>df.<a href="#">ix</a></b>	<b>A primarily label-location based indexer, with integer position fallback.</b>
<b>df.<a href="#">loc</a></b>	<b>Access a group of rows and columns by label(s) or a boolean array.</b>

<b>df.<a href="#">ndim</a></b>	<b>Return an int representing the number of axes / array dimensions.</b>
<b>df.<a href="#">shape</a></b>	<b>Return a tuple representing the dimensionality of the DataFrame.</b>
<b>df.<a href="#">size</a></b>	<b>Return an int representing the number of elements in this object.</b>
<b>df.<a href="#">style</a></b>	<b>Property returning a Styler object containing methods for building a styled HTML representation fo the DataFrame.</b>

**df.[values](#)**

**Return a Numpy  
representation of the  
DataFrame.**

<b>df.<a href="#">abs</a>( )</b>	<b>Return a Series/DataFrame with absolute numeric value of each element.</b>
<b>df.<a href="#">add</a>( other )</b>	<b>Addition of dataframe and other, element-wise (binary operator add).</b>
<b>df.<a href="#">add_prefix</a>( )</b>	<b>Prefix labels with string prefix.</b>
<b>df.<a href="#">add_suffix</a>( )</b>	<b>Suffix labels with string suffix.</b>

<b>df.<a href="#">agg</a>( )</b>	<b>Aggregate using one or more operations over the specified axis.</b>
<b>df.<a href="#">aggregate</a>( )</b>	<b>Aggregate using one or more operations over the specified axis.</b>
<b>df.<a href="#">align</a>( )</b>	<b>Align two objects on their axes with the specified join method for each axis Index</b>
<b>df.<a href="#">all</a>( )</b>	<b>Return whether all elements are True, potentially over an axis.</b>



<b>df.<a href="#">any()</a></b>	<b>Return whether any element is True over requested axis.</b>
<b>df.<a href="#">append()</a></b>	<b>Append rows of other to the end of this frame, returning a new object.</b>
<b>df.<a href="#">apply()</a></b>	<b>Apply a function along an axis of the DataFrame.</b>
<b>df.<a href="#">applymap()</a></b>	<b>Apply a function to a Dataframe elementwise.</b>

<b>df.<a href="#">asfreq</a>(freq)</b>	<b>Convert TimeSeries to specified frequency.</b>
<b>df.<a href="#">asof</a>( )</b>	<b>The last row without any NaN is taken (or the last row without NaN considering only the subset of columns in the case of a DataFrame)</b>
<b>df.<a href="#">assign</a>(**kwargs)</b>	<b>Assign new columns to a DataFrame, returning a new object (a copy) with the new columns added to the original ones.</b>
<b>df.<a href="#">astype</a>( )</b>	<b>Cast a pandas object to a specified dtype dtype.</b>

<b>df.<a href="#">at time</a>(time)</b>	<b>Select values at particular time of day (e.g.</b>
<b>df.<a href="#">between time</a>( )</b>	<b>Select values between particular times of the day (e.g., 9:00-9:30 AM).</b>
<b>df.<a href="#">bfill</a>( )</b>	<b>Synonym for <a href="#">DataFrame.fillna</a></b>
<b>df.<a href="#">bool</a>( )</b>	<b>Return the bool of a single element PandasObject.</b>

**df.[boxplot](#)( )**

**Make a box plot from  
DataFrame columns.**

**df.[clip](#)( )**

**Trim values at input  
threshold(s).**

**df.[clip\\_lower](#)( )**

**Return copy of the input with  
values below a threshold  
truncated.**

**df.[clip\\_upper](#)( )**

**Return copy of input with  
values above given value(s)  
truncated.**

<b>df.<a href="#">combine</a>( other )</b>	<b>Add two DataFrame objects and do not propagate NaN values, so if for a (column, time) one frame is missing a value, it will default to the other frame's value (which might be NaN as well)</b>
<b>df.<a href="#">combine_first</a>( )</b>	<b>Combine two DataFrame objects and default to non-null values in frame calling the method.</b>
<b>df.<a href="#">compound</a>( )</b>	<b>Return the compound percentage of the values for the requested axis</b>
<b>df.<a href="#">copy</a>( [deep] )</b>	<b>Make a copy of this object's indices and data.</b>

<b>df.<a href="#">corr</a>( [method, min_periods] )</b>	<b>Compute pairwise correlation of columns, excluding NA/null values</b>
<b>df.<a href="#">corrwith</a>( )</b>	<b>Compute pairwise correlation between rows or columns of two DataFrame objects.</b>
<b>df.<a href="#">count</a>( )</b>	<b>Count non-NA cells for each column or row.</b>
<b>df.<a href="#">cov</a>( [min_periods] )</b>	<b>Compute pairwise covariance of columns, excluding NA/null values.</b>

<b>df.<a href="#">cummax</a>( )</b>	<b>Return cumulative maximum over a DataFrame or Series axis.</b>
<b>df.<a href="#">cummin</a>( )</b>	<b>Return cumulative minimum over a DataFrame or Series axis.</b>
<b>df.<a href="#">cumprod</a>( )</b>	<b>Return cumulative product over a DataFrame or Series axis.</b>
<b>df.<a href="#">cumsum</a>( )</b>	<b>Return cumulative sum over a DataFrame or Series axis.</b>

<b>df.<a href="#">describe</a>( )</b>	<b>Generates descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding NaN values.</b>
<b>df.<a href="#">diff</a>( [periods, axis] )</b>	<b>First discrete difference of element.</b>
<b>df.<a href="#">div</a>( other )</b>	<b>Floating division of dataframe and other, element-wise (binary operator truediv).</b>
<b>df.<a href="#">divide</a>( )</b>	<b>Floating division of dataframe and other, element-wise (binary operator truediv).</b>



<b>df.<a href="#">dot</a>( other )</b>	<b>Matrix multiplication with DataFrame or Series objects.</b>
<b>df.<a href="#">drop</a>( )</b>	<b>Drop specified labels from rows or columns.</b>
<b>df.<a href="#">drop_duplicates</a>( )</b>	<b>Return DataFrame with duplicate rows removed, optionally only considering certain columns</b>
<b>df.<a href="#">dropna</a>( )</b>	<b>Remove missing values.</b>

<b>df.<a href="#">duplicated</a>( )</b>	<b>Return boolean Series denoting duplicate rows, optionally only considering certain columns</b>
<b>df.<a href="#">eq</a>( other[, axis, level] )</b>	<b>Wrapper for flexible comparison methods eq</b>
<b>df.<a href="#">equals</a>( other )</b>	<b>Determines if two NDFrame objects contain the same elements.</b>
<b>df.<a href="#">eval</a>(expr[, inplace] )</b>	<b>Evaluate a string describing operations on DataFrame columns.</b>

<b>df.<a href="#">ewm()</a></b>	<b>Provides exponential weighted functions</b>
<b>df.<a href="#">expanding()</a></b>	<b>Provides expanding transformations.</b>
<b>df.<a href="#">ffill()</a></b>	<b>Synonym for <a href="#">DataFrame.fillna(method='ffill')</a></b>
<b>df.<a href="#">fillna()</a></b>	<b>Fill NA/NaN values using the specified method</b>

<b>df.<a href="#">filter</a>( )</b>	<b>Subset rows or columns of dataframe according to labels in the specified index.</b>
<b>df.<a href="#">first</a>(offset)</b>	<b>Convenience method for subsetting initial periods of time series data based on a date offset.</b>
<b>df.<a href="#">first_valid_index</a>( )</b>	<b>Return index for first non-NA/null value.</b>
<b>df.<a href="#">floordiv</a>( other )</b>	<b>Integer division of dataframe and other, element-wise (binary operator floordiv).</b>

<b>df.<a href="#">from_dict</a>(data)</b>	<b>Construct DataFrame from dict of array-like or dicts.</b>
<b>df.<a href="#">from_records</a>(data)</b>	<b>Convert structured or record ndarray to DataFrame</b>
<b>df.<a href="#">ge</a>( )</b>	<b>Wrapper for flexible comparison methods ge</b>
<b>df.<a href="#">get</a>( key[, default] )</b>	<b>Get item from object for given key (DataFrame column, Panel slice, etc.).</b>

<b>df.<a href="#">get_dtypes_counts()</a></b>	<b>Return counts of unique dtypes in this object.</b>
<b>df.<a href="#">get_values()</a></b>	<b>Return an ndarray after converting sparse values to dense.</b>
<b>df.<a href="#">groupby()</a></b>	<b>Group series using mapper (dict or key function, apply given function to group, return result as series) or by a series of columns.</b>
<b>df.<a href="#">gt()</a></b>	<b>Wrapper for flexible comparison methods gt</b>

<code>df.<a href="#">head</a>( [n] )</code>	<b>Return the first n rows.</b>
<code>df.<a href="#">hist</a>( )</code>	<b>Make a histogram of the DataFrame's.</b>
<code>df.<a href="#">idxmax</a>( )</code>	<b>Return index of first occurrence of maximum over requested axis.</b>
<code>df.<a href="#">idxmin</a>( [axis, skipna] )</code>	<b>Return index of first occurrence of minimum over requested axis.</b>

<b>df.<a href="#">infer_objects</a>( )</b>	<b>Attempt to infer better dtypes for object columns.</b>
<b>df.<a href="#">info</a>( [verbose, ...] )</b>	<b>Print a concise summary of a DataFrame.</b>
<b>df.<a href="#">insert</a>( )</b>	<b>Insert column into DataFrame at specified location.</b>
<b>df.<a href="#">interpolate</a>( )</b>	<b>Interpolate values according to different methods.</b>



<b>df.<a href="#">isin</a>(values)</b>	<b>Return boolean DataFrame showing whether each element in the DataFrame is contained in values.</b>
<b>df.<a href="#">isna</a>( )</b>	<b>Detect missing values.</b>
<b>df.<a href="#">isnull</a>( )</b>	<b>Detect missing values.</b>
<b>df.<a href="#">items</a>( )</b>	<b>Iterator over (column name, Series) pairs.</b>

<b>df.<a href="#">iteritems</a>( )</b>	<b>Iterator over (column name, Series) pairs.</b>
<b>df.<a href="#">iterrows</a>( )</b>	<b>Iterate over DataFrame rows as (index, Series) pairs.</b>
<b>df.<a href="#">itertuples</a>( [index, name] )</b>	<b>Iterate over DataFrame rows as namedtuples, with index value as first element of the tuple.</b>
<b>df.<a href="#">join</a>( other )</b>	<b>Join columns with other DataFrame either on index or on a key column.</b>

<p><b>df.<a href="#">keys</a>( )</b></p>	<p><b>Get the 'info axis' (see Indexing for more)</b></p>
<p><b>df.<a href="#">kurt</a>( [axis, ... ] )</b></p>	<p><b>Return unbiased kurtosis over requested axis using Fisher's definition of kurtosis (kurtosis of normal == 0.0).</b></p>
<p><b>df.<a href="#">kurtosis</a>( [axis, skipna... ] )</b></p>	<p><b>Return unbiased kurtosis over requested axis using Fisher's definition of kurtosis (kurtosis of normal == 0.0).</b></p>
<p><b>df.<a href="#">last</a>( offset )</b></p>	<p><b>Convenience method for subsetting final periods of time series data based on a date offset.</b></p>

<b>df.<a href="#">last_valid_index</a>( )</b>	<b>Return index for last non-NA/null value.</b>
<b>df.<a href="#">le</a>( other )</b>	<b>Wrapper for flexible comparison methods le</b>
<b>df.<a href="#">lookup</a>(row_labels, col_labels)</b>	<b>Label-based “fancy indexing” function for DataFrame.</b>
<b>df.<a href="#">lt</a>( )</b>	<b>Wrapper for flexible comparison methods lt</b>

<b>df.<a href="#">mad</a>( )</b>	<b>Return the mean absolute deviation of the values for the requested axis</b>
<b>df.<a href="#">mask</a>( )</b>	<b>Return an object of same shape as self and whose corresponding entries are from self where cond is False and otherwise are from other.</b>
<b>df.<a href="#">max</a>( [axis, skipna, ...] )</b>	<b>This method returns the maximum of the values in the object.</b>
<b>df.<a href="#">mean</a>( )</b>	<b>Return the mean of the values for the requested axis</b>

<b>df.<a href="#">median</a>( )</b>	<b>Return the median of the values for the requested axis</b>
<b>df.<a href="#">melt</a>( )</b>	<b>“Unpivots” a DataFrame from wide format to long format, optionally leaving identifier variables set.</b>
<b>df.<a href="#">memory usage</a>( )</b>	<b>Return the memory usage of each column in bytes.</b>
<b>df.<a href="#">merge</a>(right)</b>	<b>Merge DataFrame objects by performing a database-style join operation by columns or indexes.</b>

<b>df.<a href="#">min</a>( )</b>	<b>This method returns the minimum of the values in the object.</b>
<b>df.<a href="#">mod</a>( )</b>	<b>Modulo of dataframe and other, element-wise (binary operator mod).</b>
<b>df.<a href="#">mode</a>( )</b>	<b>Gets the mode(s) of each element along the axis selected.</b>
<b>df.<a href="#">mul</a>( )</b>	<b>Multiplication of dataframe and other, element-wise (binary operator mul).</b>

<b>df.<a href="#">multiply</a>( other )</b>	<b>Multiplication of dataframe and other, element-wise (binary operator mul).</b>
<b>df.<a href="#">ne</a>( )</b>	<b>Wrapper for flexible comparison methods ne</b>
<b>df.<a href="#">nlargest</a>( )</b>	<b>Return the first n rows ordered by columns in descending order.</b>
<b>df.<a href="#">notna</a>( )</b>	<b>Detect existing (non-missing) values.</b>



<b>df.<a href="#">notnull</a>( )</b>	<b>Detect existing (non-missing) values.</b>
<b>df.<a href="#">nsmallest</a>(n, columns ... )</b>	<b>Get the rows of a DataFrame sorted by the n smallest values of columns.</b>
<b>df.<a href="#">nunique</a>( [axis, dropna] )</b>	<b>Return Series with number of distinct observations over requested axis.</b>
<b>df.<a href="#">pct_change</a>( )</b>	<b>Percentage change between the current and a prior element.</b>

<b>df.<a href="#">pipe</a>(func, *args, **kwargs)</b>	<b>Apply func(self, *args, **kwargs)</b>
<b>df.<a href="#">pivot</a>( [index, columns,... ])</b>	<b>Return reshaped DataFrame organized by given index / column values.</b>
<b>df.<a href="#">pivot table</a>( )</b>	<b>Create a spreadsheet-style pivot table as a DataFrame.</b>
<b>df.<a href="#">plot</a></b>	<b>alias of pandas.plotting._core.FrameP lotMethods</b>

<b>df.<a href="#">pop</a>(item)</b>	<b>Return item and drop from frame.</b>
<b>df.<a href="#">pow</a>( )</b>	<b>Exponential power of dataframe and other, element-wise (binary operator pow).</b>
<b>df..<a href="#">prod</a>( [axis, skipna, level, ... ])</b>	<b>Return the product of the values for the requested axis</b>

<b>df.<a href="#">product</a>( [axis, skipna, ... ] )</b>	<b>Return the product of the values for the requested axis</b>
<b>df.<a href="#">quantile</a>( [q, axis, ... ] )</b>	<b>Return values at the given quantile over requested axis, a la numpy.percentile.</b>
<b>df.<a href="#">query</a>( expr[, inplace] )</b>	<b>Query the columns of a frame with a boolean expression.</b>
<b>df.<a href="#">radd</a>( other[, axis, level,... ] )</b>	<b>Addition of dataframe and other, element-wise (binary operator radd).</b>

<b>df.<a href="#">rank</a>( [axis, method,... ] )</b>	<b>Compute numerical data ranks (1 through n) along axis.</b>
<b>df.<a href="#">rdiv</a>( other[, axis, level,... ] )</b>	<b>Floating division of dataframe and other, element-wise (binary operator rtruediv).</b>
<b>df.<a href="#">reindex</a>( [labels, index, ...] )</b>	<b>Conform DataFrame to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index.</b>
<b>df.<a href="#">reindex_axis</a>( labels ... )</b>	<b>Conform input object to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index.</b>

<b>df.<a href="#">reindex_like</a>( other... )</b>	<b>Return an object with matching indices to myself.</b>
<b>df.<a href="#">rename</a>( [mapper, index,...] )</b>	<b>Alter axes labels.</b>
<b>df.<a href="#">rename_axis</a>( mapper )</b>	<b>Alter the name of the index or columns.</b>
<b>df.<a href="#">reorder_levels</a>( order ... )</b>	<b>Rearrange index levels using input order.</b>

<b>df.<a href="#">replace</a>( )</b>	<b>Replace values given in to_replace with value.</b>
<b>df.<a href="#">resample</a>( )</b>	<b>Convenience method for frequency conversion and resampling of time series.</b>
<b>df.<a href="#">reset_index</a>( )</b>	<b>For DataFrame with multi-level index, return new DataFrame with labeling information in the columns under the index names, defaulting to 'level_0', 'level_1', etc.</b>
<b>df.<a href="#">rfloordiv</a>( )</b>	<b>Integer division of dataframe and other, element-wise (binary operator rfloordiv).</b>

<b>df.<a href="#">rmod</a>( )</b>	<b>Modulo of dataframe and other, element-wise (binary operator rmod).</b>
<b>df.<a href="#">rmul</a>( )</b>	<b>Multiplication of dataframe and other, element-wise (binary operator rmul).</b>
<b>df.<a href="#">rolling</a>(window ... )</b>	<b>Provides rolling window calculations.</b>
<b>df.<a href="#">round</a>( [decimals] )</b>	<b>Round a DataFrame to a variable number of decimal places.</b>



<b>df.<a href="#">rpow</a>( )</b>	<b>Exponential power of dataframe and other, element-wise (binary operator rpow).</b>
<b>df.<a href="#">rsub</a>( )</b>	<b>Subtraction of dataframe and other, element-wise (binary operator rsub).</b>
<b>df.<a href="#">rtruediv</a>( other[, axis,...] )</b>	<b>Floating division of dataframe and other, element-wise (binary operator rtruediv).</b>
<b>df.<a href="#">sample</a>( )</b>	<b>Return a random sample of items from an axis of object.</b>

<b>df.<a href="#">select_dtypes</a>( )</b>	<b>Return a subset of the DataFrame's columns based on the column dtypes.</b>
<b>df.<a href="#">sem</a>( )</b>	<b>Return unbiased standard error of the mean over requested axis.</b>
<b>df.<a href="#">set_axis</a>( )</b>	<b>Assign desired index to given axis.</b>
<b>df.<a href="#">set_index</a>( )</b>	<b>Set the DataFrame index (row labels) using one or more existing columns.</b>

<b>df.<a href="#">shift</a>( )</b>	<b>Shift index by desired number of periods with an optional time freq</b>
<b>df.<a href="#">skew</a>( )</b>	<b>Return unbiased skew over requested axis Normalized by N-1</b>
<b>df.<a href="#">slice_shift</a>( )</b>	<b>Equivalent to shift without copying data.</b>
<b>df.<a href="#">sort_index</a>( )</b>	<b>Sort object by labels (along an axis)</b>

<b>df.<a href="#">sort values</a>( by[, ...] )</b>	<b>Sort by the values along either axis</b>
<b>df.<a href="#">squeeze</a>( [axis] )</b>	<b>Squeeze length 1 dimensions.</b>
<b>df.<a href="#">stack</a>( [level, dropna] )</b>	<b>Stack the prescribed level(s) from columns to index.</b>
<b>df.<a href="#">std</a>( )</b>	<b>Return sample standard deviation over requested axis.</b>

<b>df.<a href="#">sub</a>( )</b>	<b>Subtraction of dataframe and other, element-wise (binary operator sub).</b>
<b>df.<a href="#">subtract</a>( )</b>	<b>Subtraction of dataframe and other, element-wise (binary operator sub).</b>
<b>df.<a href="#">sum</a>( [axis, skipna, ...] )</b>	<b>Return the sum of the values for the requested axis</b>
<b>df.<a href="#">swapaxes</a>( )</b>	<b>Interchange axes and swap values axes appropriately</b>

<b>df.<a href="#">swaplevel</a>( )</b>	<b>Swap levels i and j in a MultiIndex on a particular axis</b>
<b>df.<a href="#">tail</a>( [n] )</b>	<b>Return the last n rows.</b>
<b>df.<a href="#">take</a>( )</b>	<b>Return the elements in the given <i>positional</i> indices along an axis.</b>
<b>df.<a href="#">to_clipboard</a>( )</b>	<b>Copy object to the system clipboard.</b>

<b>df.<a href="#">to_csv</a>( )</b>	<b>Write DataFrame to a comma-separated values (csv) file</b>
<b>df.<a href="#">to_dense</a>( )</b>	<b>Return dense representation of NDFrame (as opposed to sparse)</b>
<b>df.<a href="#">to_dict</a>( )</b>	<b>Convert the DataFrame to a dictionary.</b>
<b>df.<a href="#">to_excel</a>( )</b>	<b>Write DataFrame to an excel sheet</b>

<b>df.<a href="#">to feather</a>(fname)</b>	<b>write out the binary feather-format for DataFrames</b>
<b>df.<a href="#">to_gbq</a>( )</b>	<b>Write a DataFrame to a Google BigQuery table.</b>
<b>df.<a href="#">to_hdf</a>( )</b>	<b>Write the contained data to an HDF5 file using HDFStore.</b>
<b>df.<a href="#">to_html</a>( )</b>	<b>Render a DataFrame as an HTML table.</b>



<b>df.<a href="#">to_json()</a></b>	<b>Convert the object to a JSON string.</b>
<b>df.<a href="#">to_latex()</a></b>	<b>Render an object to a tabular environment table.</b>
<b>df.<a href="#">to_msgpack()</a></b>	<b>msgpack (serialize) object to input file path</b>
<b>df.<a href="#">to_parquet()</a></b>	<b>Write a DataFrame to the binary parquet format.</b>

<b>df.<a href="#">to_period</a>( )</b>	<b>Convert DataFrame from DatetimeIndex to PeriodIndex with desired frequency (inferred from index if not passed)</b>
<b>df.<a href="#">to_pickle</a>( )</b>	<b>Pickle (serialize) object to file.</b>
<b>df.<a href="#">to_records</a>( )</b>	<b>Convert DataFrame to a NumPy record array.</b>
<b>df.<a href="#">to_sparse</a>( )</b>	<b>Convert to SparseDataFrame</b>

<b>df.<a href="#">to_sql</a>( )</b>	<b>Write records stored in a DataFrame to a SQL database.</b>
<b>df.<a href="#">to_stata</a>(fname)</b>	<b>Export Stata binary dta files.</b>
<b>df.<a href="#">to_string</a>( )</b>	<b>Render a DataFrame to a console-friendly tabular output.</b>
<b>df.<a href="#">to_timestamp</a>( )</b>	<b>Cast to DatetimeIndex of timestamps, at <i>beginning</i> of period</b>

<b>df.<a href="#">to_xarray</a>( )</b>	<b>Return an xarray object from the pandas object.</b>
<b>df.<a href="#">transform</a>( )</b>	<b>Call function producing a like-indexed NDFrame and return a NDFrame with the transformed values</b>
<b>df.<a href="#">transpose</a> ( )</b>	<b>Transpose index and columns.</b>
<b>df.<a href="#">truediv</a>( )</b>	<b>Floating division of dataframe and other, element-wise (binary operator truediv).</b>

<b>df.<a href="#">truncate</a>( ...)]</b>	<b>Truncate a Series or DataFrame before and after some index value.</b>
<b>df.<a href="#">tshift</a>( [ ] )</b>	<b>Shift the time index, using the index's frequency if available.</b>
<b>df.<a href="#">tz_convert</a>( )</b>	<b>Convert tz-aware axis to target time zone.</b>
<b>df.<a href="#">tz_localize</a>(tz)</b>	<b>Localize tz-naive TimeSeries to target time zone.</b>

<b>df.<a href="#">unstack</a>( )</b>	<b>Pivot a level of the (necessarily hierarchical) index labels, returning a DataFrame having a new level of column labels whose inner-most level consists of the pivoted index labels.</b>
<b>df.<a href="#">update</a>( other )</b>	<b>Modify in place using non-NA values from another DataFrame.</b>
<b>df.<a href="#">var</a>(,)</b>	<b>Return unbiased variance over requested axis.</b>
<b>df.<a href="#">where</a>(cond)</b>	<b>Return an object of same shape as self and whose corresponding entries are from self where cond is True and otherwise are from other.</b>

**df.[xs](#)(key[, axis, ...] )**

**Returns a cross-section  
(row(s) or column(s)) from  
the Series/DataFrame.**