pandas.tseries.offsets.BusinessDay.name

property BusinessDay.**name**

pandas.tseries.offsets.BusinessDay.nanos

property BusinessDay.**nanos**

pandas.tseries.offsets.BusinessDay.normalize

BusinessDay.**normalize = False**

pandas.tseries.offsets.BusinessDay.rule_code

property BusinessDay.**rule_code**

Methods

BusinessDay.apply(self, other)	
BusinessDay.apply_index(self, other)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
BusinessDay.copy(self)	
BusinessDay.isAnchored(self)	
BusinessDay.onOffset(self, dt)	
BusinessDay.is_anchored(self)	
BusinessDay.is_on_offset(self, dt)	
BusinessDay.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.BusinessDay.apply

BusinessDay.**apply**(*self*, *other*)

pandas.tseries.offsets.BusinessDay.apply_index

BusinessDay.**apply_index**(*self*, *other*)

Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.BusinessDay.copy

BusinessDay.**copy**(*self*)

pandas.tseries.offsets.BusinessDay.isAnchored

BusinessDay.**isAnchored**(*self*)

pandas.tseries.offsets.BusinessDay.onOffset

BusinessDay.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.BusinessDay.is_anchored

BusinessDay.**is_anchored**(*self*)

pandas.tseries.offsets.BusinessDay.is_on_offset

BusinessDay.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.BusinessDay.__call__

BusinessDay.**__call__**(*self*, *other*)
Call self as a function.

3.8.3 BusinessHour

BusinessHour([n, normalize, start, end, offset])	DateOffset subclass representing possibly n business hours.

pandas.tseries.offsets.BusinessHour

class pandas.tseries.offsets.**BusinessHour**(*n=1*, *normalize=False*, *start='09:00'*, *end='17:00'*, *offset=datetime.timedelta(0)*)
DateOffset subclass representing possibly n business hours.

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.
next_bday	Used for moving to next business day.
offset	Alias for self._offset.

pandas.tseries.offsets.BusinessHour.base

property BusinessHour.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.BusinessHour.next_bday

BusinessHour.**next_bday**
Used for moving to next business day.

pandas.tseries.offsets.BusinessHour.offset

property BusinessHour.**offset**
Alias for self._offset.

freqstr	
kwds	
name	
nanos	
rule_code	

Methods

apply_index(self, other)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
rollback(self, dt)	Roll provided date backward to next offset only if not on offset.
rollforward(self, dt)	Roll provided date forward to next offset only if not on offset.

pandas.tseries.offsets.BusinessHour.apply_index

BusinessHour.**apply_index**(*self*, *other*)

Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.BusinessHour.rollback

BusinessHour.**rollback**(*self*, *dt*)

Roll provided date backward to next offset only if not on offset.

pandas.tseries.offsets.BusinessHour.rollforward

BusinessHour.**rollforward**(*self*, *dt*)

Roll provided date forward to next offset only if not on offset.

__call__	
apply	
copy	
isAnchored	
is_anchored	
is_on_offset	
onOffset	

Properties

BusinessHour.freqstr
BusinessHour.kwds
BusinessHour.name
BusinessHour.nanos
BusinessHour.normalize
BusinessHour.rule_code

pandas.tseries.offsets.BusinessHour.freqstr

BusinessHour.**freqstr**

pandas.tseries.offsets.BusinessHour.kwds

property BusinessHour.**kwds**

pandas.tseries.offsets.BusinessHour.name

property BusinessHour.**name**

pandas.tseries.offsets.BusinessHour.nanos

property BusinessHour.**nanos**

pandas.tseries.offsets.BusinessHour.normalize

BusinessHour.**normalize = False**

pandas.tseries.offsets.BusinessHour.rule_code

property BusinessHour.**rule_code**

Methods

BusinessHour.apply(self, other)	
BusinessHour.copy(self)	
BusinessHour.isAnchored(self)	
BusinessHour.onOffset(self, dt)	
BusinessHour.is_anchored(self)	
BusinessHour.is_on_offset(self, dt)	
BusinessHour.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.BusinessHour.apply

BusinessHour.**apply**(*self*, *other*)

pandas.tseries.offsets.BusinessHour.copy

BusinessHour.**copy**(*self*)

pandas.tseries.offsets.BusinessHour.isAnchored

BusinessHour.**isAnchored**(*self*)

pandas.tseries.offsets.BusinessHour.onOffset

BusinessHour.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.BusinessHour.is_anchored

BusinessHour.**is_anchored**(*self*)

pandas.tseries.offsets.BusinessHour.is_on_offset

BusinessHour.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.BusinessHour.__call__

BusinessHour.**__call__**(*self*, *other*)
Call self as a function.

3.8.4 CustomBusinessDay

CustomBusinessDay([n, normalize, weekmask, ...])	DateOffset subclass representing possibly n custom business days, excluding holidays.

pandas.tseries.offsets.CustomBusinessDay

class pandas.tseries.offsets.**CustomBusinessDay**(*n=1*, *normalize=False*, *weekmask='Mon Tue Wed Thu Fri'*, *holidays=None*, *calendar=None*, *offset=datetime.timedelta(0)*)

DateOffset subclass representing possibly n custom business days, excluding holidays.

Parameters

n [int, default 1]

normalize [bool, default False] Normalize start/end dates to midnight before generating date range.

weekmask [str, Default 'Mon Tue Wed Thu Fri'] Weekmask of valid business days, passed to `numpy.busdaycalendar`.

holidays [list] List/array of dates to exclude from the set of valid business days, passed to

numpy.busdaycalendar.

calendar [pd.HolidayCalendar or np.busdaycalendar]

offset [timedelta, default timedelta(0)]

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.
offset	Alias for self._offset.

pandas.tseries.offsets.CustomBusinessDay.base

property CustomBusinessDay.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.CustomBusinessDay.offset

property CustomBusinessDay.**offset**
Alias for self._offset.

freqstr	
kwds	
name	
nanos	
rule_code	

Methods

apply_index(self, i)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
rollback(self, dt)	Roll provided date backward to next offset only if not on offset.
rollforward(self, dt)	Roll provided date forward to next offset only if not on offset.

pandas.tseries.offsets.CustomBusinessDay.apply_index

CustomBusinessDay.**apply_index**(*self*, *i*)
Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.CustomBusinessDay.rollback

CustomBusinessDay.**rollback**(*self*, *dt*)

Roll provided date backward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

pandas.tseries.offsets.CustomBusinessDay.rollforward

CustomBusinessDay.**rollforward**(*self*, *dt*)

Roll provided date forward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

__call__	
apply	
copy	
isAnchored	
is_anchored	
is_on_offset	
onOffset	

Properties

CustomBusinessDay.freqstr
CustomBusinessDay.kwds
CustomBusinessDay.name
CustomBusinessDay.nanos
CustomBusinessDay.normalize
CustomBusinessDay.rule_code

pandas.tseries.offsets.CustomBusinessDay.freqstr

CustomBusinessDay.**freqstr**

pandas.tseries.offsets.CustomBusinessDay.kwds

property CustomBusinessDay.**kwds**

pandas.tseries.offsets.CustomBusinessDay.name

property CustomBusinessDay.**name**

pandas.tseries.offsets.CustomBusinessDay.nanos

property CustomBusinessDay.**nanos**

pandas.tseries.offsets.CustomBusinessDay.normalize

CustomBusinessDay.**normalize = False**

pandas.tseries.offsets.CustomBusinessDay.rule_code

property CustomBusinessDay.**rule_code**

Methods

CustomBusinessDay.apply(self, other)	
CustomBusinessDay.copy(self)	
CustomBusinessDay.isAnchored(self)	
CustomBusinessDay.onOffset(self, dt)	
CustomBusinessDay.is_anchored(self)	
CustomBusinessDay.is_on_offset(self, dt)	
CustomBusinessDay.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.CustomBusinessDay.apply

CustomBusinessDay.**apply**(*self*, *other*)

pandas.tseries.offsets.CustomBusinessDay.copy

CustomBusinessDay.**copy**(*self*)

pandas.tseries.offsets.CustomBusinessDay.isAnchored

CustomBusinessDay.**isAnchored**(*self*)

pandas.tseries.offsets.CustomBusinessDay.onOffset

CustomBusinessDay.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.CustomBusinessDay.is_anchored

CustomBusinessDay.**is_anchored**(*self*)

pandas.tseries.offsets.CustomBusinessDay.is_on_offset

CustomBusinessDay.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.CustomBusinessDay.__call__

CustomBusinessDay.**__call__**(*self*, *other*)
Call self as a function.

3.8.5 CustomBusinessHour

CustomBusinessHour([n, normalize, weekmask, ...])	DateOffset subclass representing possibly n custom business days.

pandas.tseries.offsets.CustomBusinessHour

class pandas.tseries.offsets.**CustomBusinessHour**(*n=1*, *normalize=False*, *weekmask='Mon Tue Wed Thu Fri'*, *holidays=None*, *calendar=None*, *start='09:00'*, *end='17:00'*, *offset=datetime.timedelta(0)*)
DateOffset subclass representing possibly n custom business days.

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.
next_bday	Used for moving to next business day.
offset	Alias for self._offset.

pandas.tseries.offsets.CustomBusinessHour.base

property CustomBusinessHour.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.CustomBusinessHour.next_bday

CustomBusinessHour.**next_bday**
Used for moving to next business day.

pandas.tseries.offsets.CustomBusinessHour.offset

property CustomBusinessHour.**offset**
Alias for self._offset.

freqstr	
kwds	
name	
nanos	
rule_code	

Methods

apply_index(self, other)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
rollback(self, dt)	Roll provided date backward to next offset only if not on offset.
rollforward(self, dt)	Roll provided date forward to next offset only if not on offset.

pandas.tseries.offsets.CustomBusinessHour.apply_index

CustomBusinessHour.**apply_index**(*self*, *other*)

Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.CustomBusinessHour.rollback

CustomBusinessHour.**rollback**(*self*, *dt*)

Roll provided date backward to next offset only if not on offset.

pandas.tseries.offsets.CustomBusinessHour.rollforward

CustomBusinessHour.**rollforward**(*self*, *dt*)

Roll provided date forward to next offset only if not on offset.

__call__	
apply	
copy	
isAnchored	
is_anchored	
is_on_offset	
onOffset	

Properties

CustomBusinessHour.freqstr
CustomBusinessHour.kwds
CustomBusinessHour.name
CustomBusinessHour.nanos
CustomBusinessHour.normalize
CustomBusinessHour.rule_code

pandas.tseries.offsets.CustomBusinessHour.freqstr

CustomBusinessHour.**freqstr**

pandas.tseries.offsets.CustomBusinessHour.kwds

property CustomBusinessHour.**kwds**

pandas.tseries.offsets.CustomBusinessHour.name

property CustomBusinessHour.**name**

pandas.tseries.offsets.CustomBusinessHour.nanos

property CustomBusinessHour.**nanos**

pandas.tseries.offsets.CustomBusinessHour.normalize

CustomBusinessHour.**normalize = False**

pandas.tseries.offsets.CustomBusinessHour.rule_code

property CustomBusinessHour.**rule_code**

Methods

CustomBusinessHour.apply(self, other)	
CustomBusinessHour.copy(self)	
CustomBusinessHour.isAnchored(self)	
CustomBusinessHour.onOffset(self, dt)	
CustomBusinessHour.is_anchored(self)	
CustomBusinessHour.is_on_offset(self, dt)	
CustomBusinessHour.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.CustomBusinessHour.apply

CustomBusinessHour.**apply**(*self*, *other*)

pandas.tseries.offsets.CustomBusinessHour.copy

CustomBusinessHour.**copy**(*self*)

pandas.tseries.offsets.CustomBusinessHour.isAnchored

CustomBusinessHour.**isAnchored**(*self*)

pandas.tseries.offsets.CustomBusinessHour.onOffset

CustomBusinessHour.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.CustomBusinessHour.is_anchored

CustomBusinessHour.**is_anchored**(*self*)

pandas.tseries.offsets.CustomBusinessHour.is_on_offset

CustomBusinessHour.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.CustomBusinessHour.__call__

CustomBusinessHour.**__call__**(*self*, *other*)
Call self as a function.

3.8.6 MonthOffset

MonthOffset([n, normalize])

Attributes

pandas.tseries.offsets.MonthOffset

class pandas.tseries.offsets.**MonthOffset**(*n=1*, *normalize=False*)

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.MonthOffset.base

property MonthOffset.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

freqstr	
kwds	
name	
nanos	
rule_code	

Methods

rollback(self, dt)	Roll provided date backward to next offset only if not on offset.
rollforward(self, dt)	Roll provided date forward to next offset only if not on offset.

pandas.tseries.offsets.MonthOffset.rollback

MonthOffset.**rollback**(*self*, *dt*)
Roll provided date backward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

pandas.tseries.offsets.MonthOffset.rollforward

MonthOffset.**rollforward**(*self*, *dt*)
Roll provided date forward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

__call__	
apply	
apply_index	
copy	
isAnchored	
is_anchored	
is_on_offset	
onOffset	

Properties

MonthOffset.freqstr
MonthOffset.kwds
MonthOffset.name
MonthOffset.nanos
MonthOffset.normalize
MonthOffset.rule_code

pandas.tseries.offsets.MonthOffset.freqstr

MonthOffset.**freqstr**

pandas.tseries.offsets.MonthOffset.kwds

property MonthOffset.**kwds**

pandas.tseries.offsets.MonthOffset.name

property MonthOffset.**name**

pandas.tseries.offsets.MonthOffset.nanos

property MonthOffset.**nanos**

pandas.tseries.offsets.MonthOffset.normalize

MonthOffset.**normalize = False**

pandas.tseries.offsets.MonthOffset.rule_code

property MonthOffset.**rule_code**

Methods

MonthOffset.apply(self, other)	
MonthOffset.apply_index(self, other)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
MonthOffset.copy(self)	
MonthOffset.isAnchored(self)	
MonthOffset.onOffset(self, dt)	
MonthOffset.is_anchored(self)	
MonthOffset.is_on_offset(self, dt)	

continues on next page

Table 209 – continued from previous page

MonthOffset.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.MonthOffset.apply

MonthOffset.**apply**(*self*, *other*)

pandas.tseries.offsets.MonthOffset.apply_index

MonthOffset.**apply_index**(*self*, *other*)

Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.MonthOffset.copy

MonthOffset.**copy**(*self*)

pandas.tseries.offsets.MonthOffset.isAnchored

MonthOffset.**isAnchored**(*self*)

pandas.tseries.offsets.MonthOffset.onOffset

MonthOffset.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.MonthOffset.is_anchored

MonthOffset.**is_anchored**(*self*)

pandas.tseries.offsets.MonthOffset.is_on_offset

MonthOffset.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.MonthOffset.__call__

MonthOffset.**__call__**(*self*, *other*)
Call self as a function.

3.8.7 MonthEnd

MonthEnd([n, normalize])	DateOffset of one month end.

pandas.tseries.offsets.MonthEnd

class pandas.tseries.offsets.**MonthEnd**(*n=1*, *normalize=False*)
DateOffset of one month end.

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.MonthEnd.base

property MonthEnd.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

freqstr	
kwds	
name	
nanos	
rule_code	

Methods

rollback(self, dt)	Roll provided date backward to next offset only if not on offset.
rollforward(self, dt)	Roll provided date forward to next offset only if not on offset.

pandas.tseries.offsets.MonthEnd.rollback

MonthEnd.**rollback**(*self*, *dt*)

Roll provided date backward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

pandas.tseries.offsets.MonthEnd.rollforward

MonthEnd.**rollforward**(*self*, *dt*)

Roll provided date forward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

__call__	
apply	
apply_index	
copy	
isAnchored	
is_anchored	
is_on_offset	
onOffset	

Properties

MonthEnd.freqstr
MonthEnd.kwds
MonthEnd.name
MonthEnd.nanos
MonthEnd.normalize
MonthEnd.rule_code

pandas.tseries.offsets.MonthEnd.freqstr

MonthEnd.**freqstr**

pandas.tseries.offsets.MonthEnd.kwds

property MonthEnd.**kwds**

pandas.tseries.offsets.MonthEnd.name

property MonthEnd.**name**

pandas.tseries.offsets.MonthEnd.nanos

property MonthEnd.**nanos**

pandas.tseries.offsets.MonthEnd.normalize

MonthEnd.**normalize = False**

pandas.tseries.offsets.MonthEnd.rule_code

property MonthEnd.**rule_code**

Methods

MonthEnd.apply(self, other)	
MonthEnd.apply_index(self, other)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
MonthEnd.copy(self)	
MonthEnd.isAnchored(self)	
MonthEnd.onOffset(self, dt)	
MonthEnd.is_anchored(self)	
MonthEnd.is_on_offset(self, dt)	
MonthEnd.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.MonthEnd.apply

MonthEnd.**apply**(*self*, *other*)

pandas.tseries.offsets.MonthEnd.apply_index

MonthEnd.**apply_index**(*self*, *other*)

Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.MonthEnd.copy

MonthEnd.**copy**(*self*)

pandas.tseries.offsets.MonthEnd.isAnchored

MonthEnd.**isAnchored**(*self*)

pandas.tseries.offsets.MonthEnd.onOffset

MonthEnd.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.MonthEnd.is_anchored

MonthEnd.**is_anchored**(*self*)

pandas.tseries.offsets.MonthEnd.is_on_offset

MonthEnd.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.MonthEnd.__call__

MonthEnd.**__call__**(*self*, *other*)

Call self as a function.

3.8.8 MonthBegin

MonthBegin([n, normalize])	DateOffset of one month at beginning.

pandas.tseries.offsets.MonthBegin

class pandas.tseries.offsets.**MonthBegin**(*n=1*, *normalize=False*)
DateOffset of one month at beginning.

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.MonthBegin.base

property MonthBegin.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

freqstr	
kwds	
name	
nanos	
rule_code	

Methods

rollback(self, dt)	Roll provided date backward to next offset only if not on offset.
rollforward(self, dt)	Roll provided date forward to next offset only if not on offset.

pandas.tseries.offsets.MonthBegin.rollback

MonthBegin.**rollback**(*self*, *dt*)
Roll provided date backward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

pandas.tseries.offsets.MonthBegin.rollforward

MonthBegin.**rollforward**(*self*, *dt*)
Roll provided date forward to next offset only if not on offset.

Returns

TimeStamp Rolled timestamp if not on offset, otherwise unchanged timestamp.

__call__	
apply	
apply_index	
copy	
isAnchored	
is_anchored	
is_on_offset	
onOffset	

Properties

MonthBegin.freqstr
MonthBegin.kwds
MonthBegin.name
MonthBegin.nanos
MonthBegin.normalize
MonthBegin.rule_code

pandas.tseries.offsets.MonthBegin.freqstr

MonthBegin.**freqstr**

pandas.tseries.offsets.MonthBegin.kwds

property MonthBegin.**kwds**

pandas.tseries.offsets.MonthBegin.name

property MonthBegin.**name**

pandas.tseries.offsets.MonthBegin.nanos

property MonthBegin.**nanos**

pandas.tseries.offsets.MonthBegin.normalize

MonthBegin.**normalize = False**

pandas.tseries.offsets.MonthBegin.rule_code

property MonthBegin.**rule_code**

Methods

MonthBegin.apply(self, other)	
MonthBegin.apply_index(self, other)	Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.
MonthBegin.copy(self)	
MonthBegin.isAnchored(self)	
MonthBegin.onOffset(self, dt)	
MonthBegin.is_anchored(self)	
MonthBegin.is_on_offset(self, dt)	
MonthBegin.__call__(self, other)	Call self as a function.

pandas.tseries.offsets.MonthBegin.apply

MonthBegin.**apply**(*self*, *other*)

pandas.tseries.offsets.MonthBegin.apply_index

MonthBegin.**apply_index**(*self*, *other*)

Vectorized apply of DateOffset to DatetimeIndex, raises NotImplentedError for offsets without a vectorized implementation.

Parameters

i [DatetimeIndex]

Returns

y [DatetimeIndex]

pandas.tseries.offsets.MonthBegin.copy

MonthBegin.**copy**(*self*)

pandas.tseries.offsets.MonthBegin.isAnchored

MonthBegin.**isAnchored**(*self*)

pandas.tseries.offsets.MonthBegin.onOffset

MonthBegin.**onOffset**(*self*, *dt*)

pandas.tseries.offsets.MonthBegin.is_anchored

MonthBegin.**is_anchored**(*self*)

pandas.tseries.offsets.MonthBegin.is_on_offset

MonthBegin.**is_on_offset**(*self*, *dt*)

pandas.tseries.offsets.MonthBegin.__call__

MonthBegin.**__call__**(*self*, *other*)
Call self as a function.

3.8.9 BusinessMonthEnd

BusinessMonthEnd([n, normalize])	DateOffset increments between business EOM dates.

pandas.tseries.offsets.BusinessMonthEnd

class pandas.tseries.offsets.**BusinessMonthEnd**(*n=1*, *normalize=False*)
DateOffset increments between business EOM dates.

Attributes

base	Returns a copy of the calling offset object with n=1 and all other attributes equal.

pandas.tseries.offsets.BusinessMonthEnd.base

property BusinessMonthEnd.**base**
Returns a copy of the calling offset object with n=1 and all other attributes equal.

freqstr	
kwds	
name	
nanos	
rule_code	