pandas.tseries.offsets.SemiMonthBegin.freqstr

SemiMonthBegin.**freqstr**

pandas.tseries.offsets.SemiMonthBegin.kwds

property SemiMonthBegin.**kwds**

pandas.tseries.offsets.SemiMonthBegin.name

property SemiMonthBegin.**name**

pandas.tseries.offsets.SemiMonthBegin.nanos

property SemiMonthBegin.**nanos**

pandas.tseries.offsets.SemiMonthBegin.normalize

SemiMonthBegin.**normalize = False**

pandas.tseries.offsets.SemiMonthBegin.rule_code

property SemiMonthBegin.**rule_code**

Methods

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

pandas.tseries.offsets.SemiMonthBegin.apply

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

pandas.tseries.offsets.SemiMonthBegin.apply_index

SemiMonthBegin.**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.SemiMonthBegin.copy

SemiMonthBegin.**copy**(*self*)

pandas.tseries.offsets.SemiMonthBegin.isAnchored

SemiMonthBegin.**isAnchored**(*self*)

pandas.tseries.offsets.SemiMonthBegin.onOffset

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

pandas.tseries.offsets.SemiMonthBegin.is_anchored

SemiMonthBegin.**is_anchored**(*self*)

pandas.tseries.offsets.SemiMonthBegin.is_on_offset

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

pandas.tseries.offsets.SemiMonthBegin.__call__

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

Call self as a function.

3.8.16 Week

Week([n, normalize, weekday])	Weekly offset.

pandas.tseries.offsets.Week

class pandas.tseries.offsets.**Week**(*n=1*, *normalize=False*, *weekday=None*)

Weekly offset.

Parameters

weekday [int, default None] Always generate specific day of week. 0 for Monday.

Attributes

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

pandas.tseries.offsets.Week.base

property Week.**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.Week.rollback

Week.**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.Week.rollforward

Week.**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

Week.freqstr
Week.kwds
Week.name
Week.nanos
Week.normalize
Week.rule_code

pandas.tseries.offsets.Week.freqstr

Week.**freqstr**

pandas.tseries.offsets.Week.kwds

property Week.**kwds**

pandas.tseries.offsets.Week.name

property Week.**name**

pandas.tseries.offsets.Week.nanos

property Week.**nanos**

pandas.tseries.offsets.Week.normalize

Week.**normalize = False**

pandas.tseries.offsets.Week.rule_code

property Week.**rule_code**

Methods

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

pandas.tseries.offsets.Week.apply

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

pandas.tseries.offsets.Week.apply_index

Week.**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.Week.copy

Week.**copy**(*self*)

pandas.tseries.offsets.Week.isAnchored

Week.**isAnchored**(*self*)

pandas.tseries.offsets.Week.onOffset

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

pandas.tseries.offsets.Week.is_anchored

Week.**is_anchored**(*self*)

pandas.tseries.offsets.Week.is_on_offset

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

pandas.tseries.offsets.Week.__call__

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

3.8.17 WeekOfMonth

WeekOfMonth([n, normalize, week, weekday])	Describes monthly dates like "the Tuesday of the 2nd week of each month".

pandas.tseries.offsets.WeekOfMonth

class pandas.tseries.offsets.**WeekOfMonth**(*n=1*, *normalize=False*, *week=0*, *weekday=0*)
Describes monthly dates like "the Tuesday of the 2nd week of each month".

Parameters

n [int]

week [int {0, 1, 2, 3, ...}, default 0] A specific integer for the week of the month. e.g. 0 is 1st week of month, 1 is the 2nd week, etc.

weekday [int {0, 1, ..., 6}, default 0] A specific integer for the day of the week.

- 0 is Monday
- 1 is Tuesday
- 2 is Wednesday

- 3 is Thursday
- 4 is Friday
- 5 is Saturday
- 6 is Sunday.

Attributes

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

pandas.tseries.offsets.WeekOfMonth.base

property WeekOfMonth.**base**

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

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.WeekOfMonth.apply_index

WeekOfMonth.**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.WeekOfMonth.rollback

WeekOfMonth.**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.WeekOfMonth.rollforward

WeekOfMonth.**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

WeekOfMonth.freqstr
WeekOfMonth.kwds
WeekOfMonth.name
WeekOfMonth.nanos
WeekOfMonth.normalize
WeekOfMonth.rule_code

pandas.tseries.offsets.WeekOfMonth.freqstr

WeekOfMonth.**freqstr**

pandas.tseries.offsets.WeekOfMonth.kwds

property WeekOfMonth.**kwds**

pandas.tseries.offsets.WeekOfMonth.name

property WeekOfMonth.**name**

pandas.tseries.offsets.WeekOfMonth.nanos

property WeekOfMonth.**nanos**

pandas.tseries.offsets.WeekOfMonth.normalize

WeekOfMonth.**normalize = False**

pandas.tseries.offsets.WeekOfMonth.rule_code

property WeekOfMonth.**rule_code**

Methods

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

pandas.tseries.offsets.WeekOfMonth.apply

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

pandas.tseries.offsets.WeekOfMonth.copy

WeekOfMonth.**copy**(*self*)

pandas.tseries.offsets.WeekOfMonth.isAnchored

WeekOfMonth.**isAnchored**(*self*)

pandas.tseries.offsets.WeekOfMonth.onOffset

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

pandas.tseries.offsets.WeekOfMonth.is_anchored

WeekOfMonth.**is_anchored**(*self*)

pandas.tseries.offsets.WeekOfMonth.is_on_offset

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

pandas.tseries.offsets.WeekOfMonth.__call__

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

3.8.18 LastWeekOfMonth

LastWeekOfMonth([n, normalize, weekday])	Describes monthly dates in last week of month like "the last Tuesday of each month".

pandas.tseries.offsets.LastWeekOfMonth

class pandas.tseries.offsets.**LastWeekOfMonth**(*n=1*, *normalize=False*, *weekday=0*)
Describes monthly dates in last week of month like "the last Tuesday of each month".

Parameters

n [int, default 1]

weekday [int {0, 1, …, 6}, default 0] A specific integer for the day of the week.

- 0 is Monday
- 1 is Tuesday
- 2 is Wednesday
- 3 is Thursday
- 4 is Friday

- 5 is Saturday
- 6 is Sunday.

Attributes

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

pandas.tseries.offsets.LastWeekOfMonth.base

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

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.LastWeekOfMonth.apply_index

LastWeekOfMonth.**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.LastWeekOfMonth.rollback

LastWeekOfMonth.**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.LastWeekOfMonth.rollforward

LastWeekOfMonth.**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

LastWeekOfMonth.freqstr
LastWeekOfMonth.kwds
LastWeekOfMonth.name
LastWeekOfMonth.nanos
LastWeekOfMonth.normalize
LastWeekOfMonth.rule_code

pandas.tseries.offsets.LastWeekOfMonth.freqstr

LastWeekOfMonth.**freqstr**

pandas.tseries.offsets.LastWeekOfMonth.kwds

property LastWeekOfMonth.**kwds**

pandas.tseries.offsets.LastWeekOfMonth.name

property LastWeekOfMonth.**name**

pandas.tseries.offsets.LastWeekOfMonth.nanos

property LastWeekOfMonth.**nanos**

pandas.tseries.offsets.LastWeekOfMonth.normalize

LastWeekOfMonth.**normalize = False**

pandas.tseries.offsets.LastWeekOfMonth.rule_code

property LastWeekOfMonth.**rule_code**

Methods

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

pandas.tseries.offsets.LastWeekOfMonth.apply

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

pandas.tseries.offsets.LastWeekOfMonth.copy

LastWeekOfMonth.**copy**(*self*)

pandas.tseries.offsets.LastWeekOfMonth.isAnchored

LastWeekOfMonth.**isAnchored**(*self*)

pandas.tseries.offsets.LastWeekOfMonth.onOffset

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

pandas.tseries.offsets.LastWeekOfMonth.is_anchored

LastWeekOfMonth.**is_anchored**(*self*)

pandas.tseries.offsets.LastWeekOfMonth.is_on_offset

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

pandas.tseries.offsets.LastWeekOfMonth.__call__

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

3.8.19 QuarterOffset

QuarterOffset([n, normalize, startingMonth])	Quarter representation - doesn't call super.

pandas.tseries.offsets.QuarterOffset

class pandas.tseries.offsets.**QuarterOffset**(*n=1*, *normalize=False*, *startingMonth=None*)
Quarter representation - doesn't call super.

Attributes

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

pandas.tseries.offsets.QuarterOffset.base

property QuarterOffset.**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.QuarterOffset.rollback

QuarterOffset.**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.QuarterOffset.rollforward

QuarterOffset.**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

QuarterOffset.freqstr
QuarterOffset.kwds
QuarterOffset.name
QuarterOffset.nanos
QuarterOffset.normalize
QuarterOffset.rule_code

pandas.tseries.offsets.QuarterOffset.freqstr

QuarterOffset.**freqstr**

pandas.tseries.offsets.QuarterOffset.kwds

property QuarterOffset.**kwds**

pandas.tseries.offsets.QuarterOffset.name

property QuarterOffset.**name**

pandas.tseries.offsets.QuarterOffset.nanos

property QuarterOffset.**nanos**

pandas.tseries.offsets.QuarterOffset.normalize

QuarterOffset.**normalize = False**

pandas.tseries.offsets.QuarterOffset.rule_code

property QuarterOffset.**rule_code**

Methods

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

continues on next page

Table 274 – continued from previous page

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

pandas.tseries.offsets.QuarterOffset.apply

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

pandas.tseries.offsets.QuarterOffset.apply_index

QuarterOffset.**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.QuarterOffset.copy

QuarterOffset.**copy**(*self*)

pandas.tseries.offsets.QuarterOffset.isAnchored

QuarterOffset.**isAnchored**(*self*)

pandas.tseries.offsets.QuarterOffset.onOffset

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

pandas.tseries.offsets.QuarterOffset.is_anchored

QuarterOffset.**is_anchored**(*self*)

pandas.tseries.offsets.QuarterOffset.is_on_offset

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

pandas.tseries.offsets.QuarterOffset.__call__

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

3.8.20 BQuarterEnd

BQuarterEnd([n, normalize, startingMonth])	DateOffset increments between business Quarter dates.

pandas.tseries.offsets.BQuarterEnd

class pandas.tseries.offsets.**BQuarterEnd**(*n=1*, *normalize=False*, *startingMonth=None*)
DateOffset increments between business Quarter dates.

startingMonth = 1 corresponds to dates like 1/31/2007, 4/30/2007, ... startingMonth = 2 corresponds to dates like 2/28/2007, 5/31/2007, ... startingMonth = 3 corresponds to dates like 3/30/2007, 6/29/2007, ...

Attributes

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

pandas.tseries.offsets.BQuarterEnd.base

property BQuarterEnd.**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.BQuarterEnd.rollback

BQuarterEnd.**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.BQuarterEnd.rollforward

BQuarterEnd.**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

BQuarterEnd.freqstr
BQuarterEnd.kwds
BQuarterEnd.name
BQuarterEnd.nanos
BQuarterEnd.normalize
BQuarterEnd.rule_code

pandas.tseries.offsets.BQuarterEnd.freqstr

BQuarterEnd.**freqstr**

pandas.tseries.offsets.BQuarterEnd.kwds

property BQuarterEnd.**kwds**

pandas.tseries.offsets.BQuarterEnd.name

property BQuarterEnd.**name**

pandas.tseries.offsets.BQuarterEnd.nanos

property BQuarterEnd.**nanos**

pandas.tseries.offsets.BQuarterEnd.normalize

BQuarterEnd.**normalize = False**

pandas.tseries.offsets.BQuarterEnd.rule_code

property BQuarterEnd.**rule_code**

Methods

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

pandas.tseries.offsets.BQuarterEnd.apply

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

pandas.tseries.offsets.BQuarterEnd.apply_index

BQuarterEnd.**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.BQuarterEnd.copy

BQuarterEnd.**copy**(*self*)

pandas.tseries.offsets.BQuarterEnd.isAnchored

BQuarterEnd.**isAnchored**(*self*)

pandas.tseries.offsets.BQuarterEnd.onOffset

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

pandas.tseries.offsets.BQuarterEnd.is_anchored

BQuarterEnd.**is_anchored**(*self*)

pandas.tseries.offsets.BQuarterEnd.is_on_offset

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

pandas.tseries.offsets.BQuarterEnd.__call__

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

Call self as a function.

3.8.21 BQuarterBegin

BQuarterBegin([n, normalize, startingMonth])

Attributes

pandas.tseries.offsets.BQuarterBegin

class pandas.tseries.offsets.**BQuarterBegin**(*n=1*, *normalize=False*, *startingMonth=None*)

Attributes

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

pandas.tseries.offsets.BQuarterBegin.base

property BQuarterBegin.**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.BQuarterBegin.rollback

BQuarterBegin.**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.BQuarterBegin.rollforward

BQuarterBegin.**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

BQuarterBegin.freqstr
BQuarterBegin.kwds
BQuarterBegin.name
BQuarterBegin.nanos
BQuarterBegin.normalize
BQuarterBegin.rule_code

pandas.tseries.offsets.BQuarterBegin.freqstr

BQuarterBegin.**freqstr**

pandas.tseries.offsets.BQuarterBegin.kwds

property BQuarterBegin.**kwds**

pandas.tseries.offsets.BQuarterBegin.name

property BQuarterBegin.**name**

pandas.tseries.offsets.BQuarterBegin.nanos

property BQuarterBegin.**nanos**

pandas.tseries.offsets.BQuarterBegin.normalize

BQuarterBegin.**normalize = False**

pandas.tseries.offsets.BQuarterBegin.rule_code

property BQuarterBegin.**rule_code**

Methods

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

pandas.tseries.offsets.BQuarterBegin.apply

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

pandas.tseries.offsets.BQuarterBegin.apply_index

BQuarterBegin.**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.BQuarterBegin.copy

BQuarterBegin.**copy**(*self*)

pandas.tseries.offsets.BQuarterBegin.isAnchored

BQuarterBegin.**isAnchored**(*self*)

pandas.tseries.offsets.BQuarterBegin.onOffset

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

pandas.tseries.offsets.BQuarterBegin.is_anchored

BQuarterBegin.**is_anchored**(*self*)

pandas.tseries.offsets.BQuarterBegin.is_on_offset

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

pandas.tseries.offsets.BQuarterBegin.__call__

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

3.8.22 QuarterEnd

QuarterEnd([n, normalize, startingMonth])	DateOffset increments between business Quarter dates.

pandas.tseries.offsets.QuarterEnd

class pandas.tseries.offsets.**QuarterEnd**(*n=1*, *normalize=False*, *startingMonth=None*)
DateOffset increments between business Quarter dates.

startingMonth = 1 corresponds to dates like 1/31/2007, 4/30/2007, ... startingMonth = 2 corresponds to dates like 2/28/2007, 5/31/2007, ... startingMonth = 3 corresponds to dates like 3/31/2007, 6/30/2007, ...

Attributes

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

pandas.tseries.offsets.QuarterEnd.base

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

freqstr	
kwds	
name	
nanos	
rule_code	