Data.Time.Calendar

Safe Haskell	Safe
Language	Haskell2010

Days

`newtype` **Day**   # Source

The Modified Julian Day is a standard count of days, with zero being the day 1858-11-17.

Constructors

ModifiedJulianDay

  toModifiedJulianDay :: `Integer`

Instances

`Enum Day`	# Source
`Eq Day`	# Source
`Data Day`	# Source
`Ord Day`	# Source
`Ix Day`	# Source
`NFData Day`	# Source
`ParseTime Day`	# Source
`FormatTime Day`	# Source

addDays :: `Integer` `->` `Day` `->` `Day`   # Source

diffDays :: `Day` `->` `Day` `->` `Integer`   # Source

Gregorian calendar

toGregorian :: `Day` `->` (`Integer`, `Int`, `Int`)   # Source

convert to proleptic Gregorian calendar. First element of result is year, second month number (1-12), third day (1-31).

fromGregorian :: `Integer` `->` `Int` `->` `Int` `->` `Day`   # Source

convert from proleptic Gregorian calendar. First argument is year, second month number (1-12), third day (1-31). Invalid values will be clipped to the correct range, month first, then day.

fromGregorianValid :: `Integer` `->` `Int` `->` `Int` `->` `Maybe Day`   # Source

convert from proleptic Gregorian calendar. First argument is year, second month number (1-12), third day (1-31). Invalid values will return Nothing

showGregorian :: `Day` `->` `String`   # Source

show in ISO 8601 format (yyyy-mm-dd)

---

gregorianMonthLength :: `Integer -> Int -> Int`   `# Source`

The number of days in a given month according to the proleptic Gregorian calendar. First argument is year, second is month.

---

addGregorianMonthsClip :: `Integer -> Day -> Day`   `# Source`

Add months, with days past the last day of the month clipped to the last day. For instance, 2005-01-30 + 1 month = 2005-02-28.

---

addGregorianMonthsRollOver :: `Integer -> Day -> Day`   `# Source`

Add months, with days past the last day of the month rolling over to the next month. For instance, 2005-01-30 + 1 month = 2005-03-02.

---

addGregorianYearsClip :: `Integer -> Day -> Day`   `# Source`

Add years, matching month and day, with Feb 29th clipped to Feb 28th if necessary. For instance, 2004-02-29 + 2 years = 2006-02-28.

---

addGregorianYearsRollOver :: `Integer -> Day -> Day`   `# Source`

Add years, matching month and day, with Feb 29th rolled over to Mar 1st if necessary. For instance, 2004-02-29 + 2 years = 2006-03-01.

---

isLeapYear :: `Integer -> Bool`   `# Source`

Is this year a leap year according to the proleptic Gregorian calendar?