Home (/) About (https://sparkbyexamples.com/about-sparkbyexamples/) | { One stop for all Spark Examples }

Spark by {Examples} (https://sparkbyexamples.com/)

Spark (https://sparkbyexamples.com/)

PySpark (https://sparkbyexamples.com/pyspark-tutorial/)

Hive (https://sparkbyexamples.com/apache-hive-tutorial/)

HBase (https://sparkbyexamples.com/apache-hbase-tutorial/)

Kafka (https://sparkbyexamples.com/apache-kafka-tutorials-with-examples/)

FAQ's (https://sparkbyexamples.com/spark-questions/)

More (https://sparkbyexamples.com/)

PySpark Tutorial

PySpark Tutorial For Beginners (https://sparkbyexamples.com/pyspark-tutorial/)

PySpark – Features (https://sparkbyexamples.com/pyspark-tutorial/#features)

PySpark – Advantages (https://sparkbyexamples.com/pyspark-tutorial/#advantages)

PySpark – Modules & Packages (https://sparkbyexamples.com/pyspark-tutorial/#modules-packages)

PySpark – Cluster Managers (https://sparkbyexamples.com/pyspark-tutorial/#cluster-manager)

PySpark – Install on Windows (https://sparkbyexamples.com/pyspark-tutorial/#pyspark-installation)

PySpark – Web/Application UI (https://sparkbyexamples.com/spark/spark-web-ui-understanding/)

PySpark – SparkSession (https://sparkbyexamples.com/pyspark/pyspark-what-is-sparksession/)

PySpark – RDD (https://sparkbyexamples.com/pyspark-rdd)

PySpark – Parallelize (https://sparkbyexamples.com/pyspark/pyspark-parallelize-create-rdd/)

PySpark – repartition() vs coalesce() (https://sparkbyexamples.com/pyspark/pyspark-repartition-vs-coalesce/)

PySpark – Broadcast Variables (https://sparkbyexamples.com/pyspark/pyspark-broadcast-variables/)

PySpark SQL Date and Timestamp Functions

NNK (https://sparkbyexamples.com/author/admin/) - PySpark (https://sparkbyexamples.com/category/pyspark/)

PySpark Date and Timestamp Functions are supported on DataFrame and SQL queries and they work similarly to traditional SQL, Date and Time are very important if you are using PySpark for ETL. Most of all these functions accept input as, Date type, Timestamp type, or String. If a String used, it should be in a default format that can be cast to date.

- `DateType` default format is `yyyy-MM-dd`

PySpark – Accumulator (https://sparkbyexamples.com/pyspark/pyspark-accumulator-with-example/)

PySpark DataFrame

PySpark – Create a DataFrame (https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/)

PySpark – Create an empty DataFrame (https://sparkbyexamples.com/pyspark/pyspark-create-an-empty-dataframe/)

PySpark – Convert RDD to DataFrame (https://sparkbyexamples.com/pyspark/convert-pyspark-rdd-to-dataframe/)

PySpark – Convert DataFrame to Pandas (https://sparkbyexamples.com/pyspark/convert-pyspark-dataframe-to-pandas/)

PySpark – show() (https://sparkbyexamples.com/pyspark/pyspark-show-display-dataframe-contents-in-table/)

PySpark – StructType & StructField (https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/)

PySpark – Row Class (https://sparkbyexamples.com/pyspark/pyspark-row-using-rdd-dataframe/)

PySpark – Column Class (https://sparkbyexamples.com/pyspark/pyspark-column-functions/)

PySpark – select() (https://sparkbyexamples.com/pyspark/select-columns-from-pyspark-dataframe/)

PySpark – collect() (https://sparkbyexamples.com/pyspark/pyspark-collect/)

PySpark – withColumn() (https://sparkbyexamples.com/pyspark/pyspark-withcolumn/)

- `TimestampType` default format is `yyyy-MM-dd HH:mm:ss.SSSS`
- Returns `null` if the input is a string that can not be cast to Date or Timestamp.

PySpark SQL provides several Date & Timestamp functions hence keep an eye on and understand these. Always you should choose these functions instead of writing your own functions (UDF) as these functions are compile-time safe, handles null, and perform better when compared to PySpark UDF (https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/). If your PySpark application is critical on performance (https://sparkbyexamples.com/spark/spark-performance-tuning/) try to avoid using custom UDF at all costs as these are not guarantee performance.

For readable purposes, I've grouped these functions into the following groups.

- Date Functions
- Timestamp Functions
- Date and Timestamp Window Functions

Before you use any examples below, make sure you Create PySpark Sparksession (https://sparkbyexamples.com/pyspark/pyspark-what-is-sparksession/) and import SQL functions.

```
from pyspark.sql.functions impo
```

PySpark SQL Date Functions

Below are some of the PySpark SQL Date functions, these functions operate on the just Date.

The default format of the PySpark Date is `yyyy-MM-dd`.

PySpark – withColumnRenamed() (https://sparkbyexamples.com/pyspark/pyspark-rename-dataframe-column/)

PySpark – where() & filter() (https://sparkbyexamples.com/pyspark/pyspark-where-filter/)

PySpark – drop() & dropDuplicates() (https://sparkbyexamples.com/pyspark/pyspark-distinct-to-drop-duplicates/)

PySpark – orderBy() and sort() (https://sparkbyexamples.com/pyspark/pyspark-orderby-and-sort-explained/)

PySpark – groupBy() (https://sparkbyexamples.com/pyspark/pyspark-groupby-explained-with-example/)

PySpark – join() (https://sparkbyexamples.com/pyspark/pyspark-join-explained-with-examples/)

PySpark – union() & unionAll() (https://sparkbyexamples.com/pyspark/pyspark-union-and-unionall/)

PySpark – unionByName() (https://sparkbyexamples.com/spark/spark-merge-two-dataframes-with-different-columns/)

PySpark – UDF (User Defined Function) (https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/)

PySpark – map() (https://sparkbyexamples.com/pyspark/pyspark-map-transformation/)

PySpark – flatMap() (https://sparkbyexamples.com/pyspark/pyspark-flatmap-transformation/)

pyspark – foreach() (https://sparkbyexamples.com/pyspark/pyspark-loop-iterate-through-rows-in-dataframe/#use-foreach-loop-through-dataframe)

PySpark – sample() vs sampleBy()

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
current_date()	Returns the current date as a date column.
date_format(dateExpr,format)	Converts a date/timestamp/string to a value of string in the format specified by the date format given by the second argument.
to_date()	Converts the column into `DateType` by casting rules to `DateType`.
to_date(column, fmt)	Converts the column into a `DateType` with a specified format
add_months(Column, numMonths)	Returns the date that is `numMonths` after `startDate`.

date_add(column, days)
date_sub(column, days)
	Returns the date that is `days` days after `start`

datediff(end, start)	Returns the number of days from `start` to `end`.
months_between(end, start)	Returns number of months between dates `start` and `end`. A whole number is returned if both inputs have the same day of month or both are the last day of their respective months. Otherwise, the difference is calculated assuming 31 days per month.

(https://sparkbyexamples.com/pyspark/pyspark-sampling-example/)

PySpark – fillna() & fill() (https://sparkbyexamples.com/pyspark/pyspark-fillna-fill-replace-null-values/)

PySpark – pivot() (Row to Column) (https://sparkbyexamples.com/pyspark/pyspark-pivot-and-unpivot-dataframe/)

PySpark – partitionBy() (https://sparkbyexamples.com/pyspark/pyspark-partitionby-example/)

PySpark – ArrayType Column (Array) (https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/)

PySpark – MapType (Map/Dict) (https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/)

PySpark SQL Functions

PySpark – Aggregate Functions (https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/)

PySpark – Window Functions (https://sparkbyexamples.com/pyspark/pyspark-window-functions/)

PySpark – Date and Timestamp Functions (https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/)

PySpark – JSON Functions (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/)

PySpark Datasources

PySpark – Read & Write CSV File (https://sparkbyexamples.com/pyspark/pyspark-read-csv-file-into-dataframe/)

PySpark – Read & Write Parquet File (https://sparkbyexamples.com/

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
months_between(end, start, roundOff)	Returns number of months between dates \`end\` and \`start\`. If \`roundOff\` is set to true, the result is rounded off to 8 digits; it is not rounded otherwise.
next_day(column, dayOfWeek)	Returns the first date which is later than the value of the \`date\` column that is on the specified day of the week. For example, \`next_day('2015-07-27', "Sunday")\` returns 2015-08-02 because that is the first Sunday after 2015-07-27.
trunc(column, format)	Returns date truncated to the unit specified by the format. For example, \`trunc("2018-11-19 12:01:19", "year")\` returns 2018-01-01 format: 'year', 'yyyy', 'yy' to truncate by year, 'month', 'mon', 'mm' to truncate by month

pyspark/pyspark-read-and-write-parquet-file/)

PySpark – Read & Write JSON file (https://sparkbyexamples.com/pyspark/pyspark-read-json-file-into-dataframe/)

PySpark Built-In Functions

PySpark – when() (https://sparkbyexamples.com/pyspark/pyspark-when-otherwise/)

PySpark – expr() (https://sparkbyexamples.com/pyspark/pyspark-sql-expr-expression-function/)

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
date_trunc(format, timestamp)	Returns timestamp truncated to the unit specified by the format. For example, `date_trunc("year", "2018-11-19 12:01:19")` returns 2018-01-01 00:00:00 format: 'year', 'yyyy', 'yy' to truncate by year, 'month', 'mon', 'mm' to truncate by month, 'day', 'dd' to truncate by day, Other options are: 'second', 'minute', 'hour', 'week', 'month', 'quarter'
ar(column)	Extracts the year as an integer from a given date/timestamp/string
arter(column)	Extracts the quarter as an integer from a given date/timestamp/string.
nth(column)	Extracts the month as an integer from a given date/timestamp/string
yofweek(column)	Extracts the day of the week as an integer from a given date/timestamp/string. Ranges from 1 for a Sunday through to 7 for a Saturday
dayofmonth(column)	Extracts the day of the month as an integer from a given date/timestamp/string.
dayofyear(column)	Extracts the day of the year as an integer from a given date/timestamp/

- PySpark – lit() (https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/)
- PySpark – split() (https://sparkbyexamples.com/pyspark/pyspark-convert-string-to-array-column/)
- PySpark – concat_ws() (https://sparkbyexamples.com/pyspark/pyspark-convert-array-column-to-string-column/)
- Pyspark – substring() (https://sparkbyexamples.com/pyspark/pyspark-substring-from-a-column/)
- PySpark – translate() (https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#translate-replace-character-by-character)
- PySpark – regexp_replace() (https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#regexp_replace-replace-string-columns)
- PySpark – overlay() (https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#overlay-function)
- PySpark – to_timestamp() (https://sparkbyexamples.com/spark/pyspark-to_timestamp-convert-string-to-timestamp-type/)
- PySpark – to_date() (https://sparkbyexamples.com/pyspark/pyspark-to_date-convert-timestamp-to-date/)
- PySpark – date_format() (https://sparkbyexamples.com/pyspark/pyspark-date_format-convert-date-to-string-format/)
- PySpark – datediff() (https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#datediff)
- PySpark – months_between() (https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#months_between())

PYSPARK DATE FUNCTION	DATE FUNCTIC DESCRIPTION
weekofyear(column)	Extracts the wee number as an in from a given date/timestamp/ A week is consic to start on a Mo and week 1 is th week with more days, as defined ISO 8601
last_day(column)	Returns the last the month which given date belor For example, inp "2015-07-27" re "2015-07-31" sir July 31 is the la of the month in 2015.
from_unixtime(column)	Converts the number of seconds from unix epoch (1970-01-01 00:00:00 UTC) to a string representing the timestamp of that moment in the current system time zone in the yyyy-MM-dd HH:mm:ss format.
from_unixtime(column, f)	Converts the number of seconds from unix epoch (1970-01-01 00:00:00 UTC) to a string representing the timestamp of that moment in the current system time zone in the given format.
unix_timestamp()	Returns the current Unix timestamp (in seconds) as a long

PySpark – explode() (https://sparkbyexamples.com/pyspark/pyspark-explode-nested-array-into-rows/)

PySpark – array_contains() (https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array_contains)

PySpark – array() (https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array)

PySpark – collect_list() (https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-list)

PySpark – collect_set() (https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-set)

PySpark – create_map() (https://sparkbyexamples.com/pyspark/pyspark-convert-dataframe-columns-to-maptype-dict/)

PySpark – map_keys() (https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_keys)

PySpark – map_values() (https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_values)

PySpark – struct() (https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/#update-struct-function)

PySpark – countDistinct() (https://sparkbyexamples.com/pyspark/pyspark-count-distinct-from-dataframe/)

PySpark – sum(), avg() (https://sparkbyexamples.com/pyspark/pyspark-dataframe-groupby-and-sort-by-descending-order/)

PySpark – row_number() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#row_number)

PySpark – rank() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#rank)

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
unix_timestamp(column)	Converts time string in format yyyy-MM-dd HH:mm:ss to Unix timestamp (in seconds), using the default timezone and the default locale.
unix_timestamp(column, p)	Converts time string with given pattern to Unix timestamp (in seconds).

PySpark SQL Timestamp Functions

Below are some of the PySpark SQL Timestamp functions, these functions operate on both date and timestamp values.

The default format of the Spark Timestamp is `yyyy-MM-dd HH:mm:ss.SSSS`

PYSPARK TIMESTAMP FUNCTION SIGNATURE	TIMESTAMP FUNCTION DESCRIPTION
current_timestamp ()	Returns the current timestamp as a timestamp column
hour(column)	Extracts the hours as an integer from a given date/timestamp/string.
minute(column)	Extracts the minutes as an integer from a given date/timestamp/string.
second(column)	Extracts the seconds as an integer from a given date/timestamp/string.

PySpark – dense_rank() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#dense_rank)

PySpark – percent_rank() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#percent_rank)

PySpark – typedLit() (https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/#typedlit)

PySpark – from_json() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#from_json)

PySpark – to_json() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#to_json)

PySpark – json_tuple() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#json_tuple)

PySpark – get_json_object() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#get_json_object)

PySpark – schema_of_json() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#schema_of_json)

PYSPARK TIMESTAMP FUNCTION SIGNATURE	TIMESTAMP FUNCTION DESCRIPTION
to_timestamp(column)	Converts to a timestamp by casting rules to `TimestampType`.
to_timestamp(column, fmt)	Converts time string with the given pattern to timestamp.

Date and Timestamp Window Functions

Below are PySpark Data and Timestamp window functions.

Search:

DATE & TIME WINDOW FUNCTION SYNTAX	DATE & TIME WINDOW FUNCTION DESCRIPTION
window(timeColumn: Column, windowDuration: String, slideDuration: String, startTime: String): Column	Bucketize rows into one or more time windows given a timestamp specifying column. Window starts are inclusive but the window ends are exclusive, e.g. 12:05 will be in the window [12:05,12:10) but not in [12:00,12:05). Windows can support microsecond precision. Windows in the order of months are not supported.

DATE & TIME WINDOW FUNCTION SYNTAX	DATE & TIME WINDOW FUNCTION DESCRIPTION
window(timeColumn: Column, windowDuration: String, slideDuration: String): Column	Bucketize rows into one or more time windows given a timestamp specifying column. Window starts are inclusive but the window ends are exclusive, e.g. 12:05 will be in the window [12:05,12:10) but not in [12:00,12:05). Windows can support microsecond precision. Windows in the order of months are not supported. The windows start beginning at 1970-01-01 00:00:00 UTC

DATE & TIME WINDOW FUNCTION SYNTAX	DATE & TIME WINDOW FUNCTION DESCRIPTION
window(timeColumn: Column, windowDuration: String): Column	Generates tumbling time windows given a timestamp specifying column. Window starts are inclusive but the window ends are exclusive, e.g. 12:05 will be in the window [12:05,12:10) but not in [12:00,12:05). Windows can support microsecond precision. Windows in the order of months are not supported. The windows start beginning at 1970-01-01 00:00:00 UTC.

PySpark SQL Date and Timestamp Functions Examples

Following are the most used **PySpark SQL Date and Timestamp Functions** with examples, you can use these on DataFrame and SQL expressions.

```
from pyspark.sql import SparkSe
from pyspark.sql.functions impo

# Create SparkSession
spark = SparkSession.builder \
            .appName('SparkByEx
            .getOrCreate()
data=[["1","2020-02-01"],["2","
df=spark.createDataFrame(data,[
df.show()

#Result
+---+----------+
| id|input     |
+---+----------+
|  1|2020-02-01|
|  2|2019-03-01|
|  3|2021-03-01|
+---+----------+
```

current_date()

Use current_date() to get the current system date. By default, the data will be returned in `yyyy-dd-mm` format.

```
#current_date()
df.select(current_date().alias(
  ).show(1)

#Result
+------------+
|current_date|
+------------+
|  2021-02-22|
+------------+
```

date_format()

The below example uses `date_format()` to parses the date and converts from `yyyy-dd-mm` to `MM-dd-yyyy` format.

```
#date_format()
df.select(col("input"),
    date_format(col("input"), "
  ).show()

#Result
+----------+-----------+
|input     |date_format|
+----------+-----------+
|2020-02-01| 02-01-2020|
|2019-03-01| 03-01-2019|
|2021-03-01| 03-01-2021|
+----------+-----------+
```

to_date()

Below example converts string in date format `yyyy-MM-dd` to a DateType `yyyy-MM-dd` using `to_date()`. You can also use this to convert into any specific format. PySpark supports all patterns supports on Java DateTimeFormatter (https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/time/format/DateTimeFormatter.html).

```
#to_date()
df.select(col("input"),
    to_date(col("input"), "yyy-
  ).show()

#Result
+----------+----------+
|     input|   to_date|
+----------+----------+
|2020-02-01|2020-02-01|
|2019-03-01|2019-03-01|
|2021-03-01|2021-03-01|
+----------+----------+
```

datediff()

The below example returns the difference between two dates using `datediff()`.

```
#datediff()
df.select(col("input"),
    datediff(current_date(),col
  ).show()

#Result
+----------+--------+
|     input|datediff|
+----------+--------+
|2020-02-01|     387|
|2019-03-01|     724|
|2021-03-01|      -7|
+----------+--------+
```

months_between()

The below example returns the months between two dates using `months_between()`.

```
#months_between()
df.select(col("input"),
    months_between(current_date
  ).show()

#Result
+----------+--------------+
|     input|months_between|
+----------+--------------+
|2020-02-01|   12.67741935|
|2019-03-01|   23.67741935|
|2021-03-01|   -0.32258065|
+----------+--------------+
```

trunc()

The below example truncates the date at a specified unit using `trunc()`.

```
#trunc()
df.select(col("input"),
    trunc(col("input"),"Month")
    trunc(col("input"),"Year").
    trunc(col("input"),"Month")
   ).show()

#Result
+----------+----------+-------
|     input|Month_Trunc|Month_Y
+----------+----------+-------
|2020-02-01| 2020-02-01|2020-01
|2019-03-01| 2019-03-01|2019-01
|2021-03-01| 2021-03-01|2021-01
+----------+----------+-------
```

add_months() , date_add(), date_sub()

Here we are adding and subtracting date and month from a given input.

```
#add_months() , date_add(), dat
df.select(col("input"),
    add_months(col("input"),3).
    add_months(col("input"),-3)
    date_add(col("input"),4).al
    date_sub(col("input"),4).al
  ).show()

#Result
+----------+----------+--------
|     input|add_months|sub_mont
+----------+----------+--------
|2020-02-01|2020-05-01|2019-11-
|2019-03-01|2019-06-01|2018-12-
|2021-03-01|2021-06-01|2020-12-
+----------+----------+--------
```

year(), month(), month(),next_day(), weekofyear()

```
df.select(col("input"),
     year(col("input")).alias("
     month(col("input")).alias(
     next_day(col("input"),"Sun
     weekofyear(col("input")).a
  ).show()

#Result
+----------+----+-----+--------
|     input|year|month|   next_d
+----------+----+-----+--------
|2020-02-01|2020|    2|2020-02-
|2019-03-01|2019|    3|2019-03-
|2021-03-01|2021|    3|2021-03-
+----------+----+-----+--------
```

dayofweek(), dayofmonth(), dayofyear()

```
df.select(col("input"),
     dayofweek(col("input")).al
     dayofmonth(col("input")).a
     dayofyear(col("input")).al
  ).show()

#Result
+----------+---------+---------
|     input|dayofweek|dayofmont
+----------+---------+---------
|2020-02-01|        7|
|2019-03-01|        6|
|2021-03-01|        2|
+----------+---------+---------
```

current_timestamp()

Following are the Timestamp Functions that you can use on SQL and on DataFrame. Let’s learn these with examples.

Let’s create a test data.

```
data=[["1","02-01-2020 11 01 19
df2=spark.createDataFrame(data,
df2.show(truncate=False)

#Result
+---+-----------------------+
|id |input                  |
+---+-----------------------+
|1  |02-01-2020 11 01 19 06 |
|2  |03-01-2019 12 01 19 406|
|3  |03-01-2021 12 01 19 406|
+---+-----------------------+
```

Below example returns the current timestamp in spark default format **yyyy-MM-dd HH:mm:ss**

```
#current_timestamp()
df2.select(current_timestamp().
  ).show(1,truncate=False)

#Result
+-----------------------+
|current_timestamp      |
+-----------------------+
|2021-02-22 20:13:29.673|
+-----------------------+
```

to_timestamp()

Converts string timestamp to Timestamp type format.

```
#to_timestamp()
df2.select(col("input"),
    to_timestamp(col("input"),
  ).show(truncate=False)

#Result
+-----------------------+------
|input                  |to_tim
+-----------------------+------
|02-01-2020 11 01 19 06 |2020-0
|03-01-2019 12 01 19 406|2019-0
|03-01-2021 12 01 19 406|2021-0
+-----------------------+------
```

hour(), Minute() and second()

```
#hour, minute,second
data=[["1","2020-02-01 11:01:19
df3=spark.createDataFrame(data,

df3.select(col("input"),
    hour(col("input")).alias("h
    minute(col("input")).alias(
    second(col("input")).alias(
  ).show(truncate=False)

#Result
+-----------------------+----+-
|input                  |hour|m
+-----------------------+----+-
|2020-02-01 11:01:19.06 |11  |1
|2019-03-01 12:01:19.406|12  |1
|2021-03-01 12:01:19.406|12  |1
+-----------------------+----+-
```

Conclusion:

In this post, I've consolidated the complete list of Date and Timestamp Functions with a description and example of some commonly used. You can find the complete list on the following blog (https://databricks.com/blog/2015/09/16/apache-spark-1-5-dataframe-api-highlights.html).

Happy Learning !!

Related

- PySpark Aggregate Functions With Examples (https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/)
- PySpark Window Functions With Examples (https://sparkbyexamples.com/pyspark/pyspark-window-functions/)

Share this:

(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=facebook&nb=1)

(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=reddit&nb=1)

(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=pinterest&nb=1)

1

(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=tumblr&nb=1)

(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=pocket&nb=1)

(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=linkedin&nb=1)

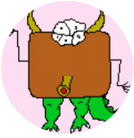
(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/?share=twitter&nb=1)

(https://sparkbyexamples.com/author/admin/)

NNK (Https://Sparkbyexamples.Com/Author/Admin/)

SparkByExamples.com is a Big Data and Spark examples community page, all examples are simple and easy to understand and well tested in our development environment Read more .. (https://sparkbyexamples.com/about-sparkbyexamples/)

THIS POST HAS 2 COMMENTS



Anonymous

28 FEB 2021 REPLY

Very helpful

NNK 1 MAR 2021 REPLY

Thanks.

Leave a Reply

Categories

Apache Hadoop (https://sparkbyexamples.com/category/hadoop/)

Apache Spark (https://sparkbyexamples.com/category/spark/)

Apache Spark Streaming (https://sparkbyexamples.com/category/spark/apache-spark-streaming/)

Apache Kafka (https://sparkbyexamples.com/category/kafka/)

Apache HBase (https://sparkbyexamples.com/category/hbase/)

Apache Cassandra (https://sparkbyexamples.com/category/cassandra/)

Snowflake Database (https://sparkbyexamples.com/catego

Recent Posts

Spark regexp_replace() – Replace String Value (https://sparkbyexamples.com/spark/spark-regexp_replace-replace-string-value/)

How to Run a PySpark Script from Python? (https://sparkbyexamples.com/pyspark/run-pyspark-script-from-python-subprocess/)

Spark SQL like() Using Wildcard Example (https://sparkbyexamples.com/spark/spark-sql-like-using-wildcard-example/)

Spark isin() & IS NOT IN Operator Example (https://sparkbyexamples.com/spark/spark-isin-is-not-in-operator-example/)

Spark – Get Size/Length of Array & Map Column (https://sparkbyexamples.com/spark/spark-get-size-length-of-array-map-column/)

About SparkByExamples.Com

SparkByExamples.com is a Big Data and Spark examples community page, all examples are simple and easy to understand, and well tested in our development environment Read more .. (https://sparkbyexamples.com/about-sparkbyexamples/)

Follow Us

ry/snowflake/)

H2O Sparkling Water (https://sparkbyexamples.com/category/h2o-sparkling-water/)

PySpark (https://sparkbyexamples.com/category/pyspark/)

Spark Using Length/Size Of a DataFrame Column (https://sparkbyexamples.com/spark/spark-using-length-size-of-a-dataframe-column/)

Spark rlike() Working with Regex Matching Examples (https://sparkbyexamples.com/spark/spark-rlike-regex-matching-examples/)

Spark Check String Column Has Numeric Values (https://sparkbyexamples.com/spark/spark-check-string-column-has-numeric-values/)

Spark Check Column Data Type is Integer or String (https://sparkbyexamples.com/spark/spark-check-column-data-type-is-integer-or-string/)

(https://twitter.com/sparkbyexamples)

(https://www.facebook.com/sparkbyexamples/)

(https://www.linkedin.com/in/n-nk-b860a8193/)

(https://github.com/spark-examples/)