# Spark by {Examples}(https://sparkbyexamples.com/) Spark by {Examples}(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 SQL expr() (Expression ) Function

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

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

PySpark `expr()` is a SQL function to execute SQL-like expressions and to use an existing DataFrame column value as an expression argument to Pyspark built-in functions. Most of the commonly used SQL functions are either part of the PySpark Column class (https://sparkbyexamples.com/pyspark/pyspark-column-functions/) or built-in `pyspark.sql.functions` API, besides these PySpark also supports many other SQL functions, so in order to use these, you have to use `expr()` function.

Below are 2 use cases of **PySpark expr() funcion**.

- First, allowing to use of SQL-like functions that are not present in

PySpark Column type & `pyspark.sql.functions` API. for example CASE WHEN, regr_count().

- Second, it extends the PySpark SQL Functions by allowing to use DataFrame columns in functions for expression. for example, if you wanted to add a month value from a column to a Date column. I will explain this in the example below.

## 1. PySpark expr() Syntax

Following is syntax of the expr() function.

```
expr(str)
```

`expr()` function takes SQL expression as a string argument, executes the expression, and returns a PySpark Column type. Expressions provided with this function are not a compile-time safety like DataFrame operations.

## 2. PySpark SQL expr() Function Examples

Below are some of the examples of using expr() SQL function.
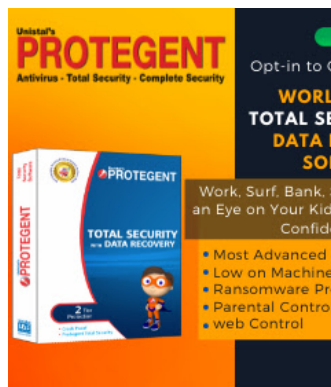
## 2.1 Concatenate Columns using || (similar to SQL)

If you have SQL background, you pretty much familiar using || to concatenate values from two string columns, you can use expr() expression to do exactly same.

```
#Concatenate columns using || (sql
data=[("James","Bond"),("Scott","Va
df=spark.createDataFrame(data).toDF
df.withColumn("Name",expr(" col1 ||


+-----+-----+-----------+
| col1| col2|       Name|
+-----+-----+-----------+
|James| Bond| James,Bond|
|Scott|Varsa|Scott,Varsa|
+-----+-----+-----------+
```
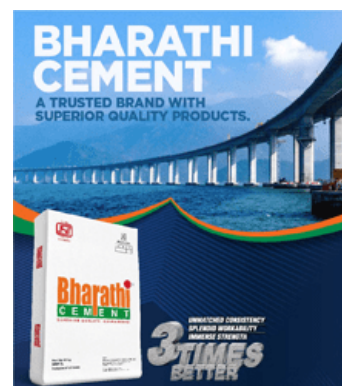
## 2.2 Using SQL CASE WHEN with expr()

PySpark doesn't have SQL Like CASE WHEN (https://sparkbyexamples.com/pyspark/pyspark-when-otherwise/) so in order to use this on PySpark DataFrame withColumn() (https://sparkbyexamples.com/pyspark/pyspark-withcolumn/) or select(), you should use `expr()` function with expression as shown below.

Here, I have used CASE WHEN expression on withColumn() by using expr(), this example updates an existing column gender with the derived values, **M for male, F for Female,** and unknown **for others**

```
from pyspark.sql.functions import e
data = [("James","M"),("Michael","F
columns = ["name","gender"]
df = spark.createDataFrame(data = c

#Using CASE WHEN similar to SQL.
from pyspark.sql.functions import e
df2=df.withColumn("gender", expr("C
          "WHEN gender = 'F' THEN
df2.show()
+-------+-------+
|   name| gender|
+-------+-------+
|  James|   Male|
|Michael| Female|
|    Jen|unknown|
+-------+-------+
```

If you have any errors in the expression you will get the run time error but not during the compile time.

## 2.3 Using an Existing Column Value for Expression

Most of the PySpark function takes constant literal values but sometimes we need to use a value from an existing column instead of a constant and this is not possible without expr() expression. The below example adds a number of months from an existing column instead of a Python constant.

```
from pyspark.sql.functions import e
data=[("2019-01-23",1),("2019-06-24
df=spark.createDataFrame(data).toDF

#Add Month value from another colum
df.select(df.date,df.increment,
    expr("add_months(date,incremen
  .alias("inc_date")).show()

+----------+---------+----------+
|      date|increment|  inc_date|
+----------+---------+----------+
|2019-01-23|        1|2019-02-23|
|2019-06-24|        2|2019-08-24|
|2019-09-20|        3|2019-12-20|
+----------+---------+----------+
```

ote that Importing SQL functions are not equired when using them with expr(). You ee above add_months() is used without nporting.

## 2.4 Giving Column Alias along with expr()

You can also use SQL like syntax to provide the alias name to the column (https://sparkbyexamples.com/pyspark/pysp ark-column-alias-after-groupby/) expression.

```
# Providing alias using 'as'
from pyspark.sql.functions import 
df.select(df.date,df.increment,
    expr("""add_months(date,increm
).show()
# This yields same output as above
```

## 2.5 Case Function with expr()

below example converts long data type to
string type.

```
# Using Cast() Function
df.select("increment",expr("cast(in
  .printSchema()

root
 |-- increment: long (nullable = tr
 |-- str_increment: string (nullabl
```

## 2.7 Arithmetic operations

expr() is also used to provide arithmetic
operations, below examples add value 5 to
`increment` and creates a new column
`new_increment`

```
# Arthemetic operations
df.select(df.date,df.increment,
    expr("increment + 5 as new_in
).show()

+----------+---------+-------------
|      date|increment|new_increment
+----------+---------+-------------
|2019-01-23|        1|            6
|2019-06-24|        2|            7
|2019-09-20|        3|            8
+----------+---------+-------------
```

## 2.8 Using Filter with expr()

Filter the DataFrame rows
(https://sparkbyexamples.com/pyspark/pysp
ark-where-filter/) can done using expr()
expression.

```
#Use expr()  to filter the rows
from pyspark.sql.functions import e
data=[(100,2),(200,3000),(500,500)]
df=spark.createDataFrame(data).toDF
df.filter(expr("col1 == col2")).sho


+----+----+
|col1|col2|
+----+----+
| 500| 500|
+----+----+
```

## 3. Complete Example of PySpark SQL expr() Function

This example is also available at GitHub PySpark Examples (https://github.com/spark-examples/pyspark-examples/blob/master/pyspark-expr.py) Project.

```python
from pyspark.sql import SparkSessi
spark = SparkSession.builder.appNam

from pyspark.sql.functions import e
#Concatenate columns
data=[("James","Bond"),("Scott","V
df=spark.createDataFrame(data).toDF
df.withColumn("Name",expr(" col1 ||

#Using CASE WHEN sql expression
data = [("James","M"),("Michael","F
columns = ["name","gender"]
df = spark.createDataFrame(data = d
df2 = df.withColumn("gender", expr(
            "WHEN gender = 'F' THEN
df2.show()

#Add months from a value of another
data=[("2019-01-23",1),("2019-06-24
df=spark.createDataFrame(data).toDF
df.select(df.date,df.increment,
      expr("add_months(date,incremer
   .alias("inc_date")).show()

# Providing alias using 'as'
df.select(df.date,df.increment,
      expr("""add_months(date,increm
   ).show()

# Add
df.select(df.date,df.increment,
      expr("increment + 5 as new_inc
   ).show()

# Using cast to convert data types
df.select("increment",expr("cast(ir
   .printSchema()

#Use expr()  to filter the rows
data=[(100,2),(200,3000),(500,500)]
df=spark.createDataFrame(data).toDF
df.filter(expr("col1 == col2")).sho
```

## Conclusion

PySpark expr() function provides a way to
run SQL like expression with DataFrames,
here you have learned how to use
expression with select(), withColumn() and
to filter the DataFrame rows.

Happy Learning !!

## References

- https://spark.apache.org/docs/2.3.1/api/python/_modules/pyspark/sql/functions.html (https://spark.apache.org/docs/2.3.1/api/python/_modules/pyspark/sql/functions.html)

---

**Share this:**

---

(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/)

---

## Leave a Reply

**Categories**

Apache Hadoop
(https://sparkbyexamples.com/category/h
adoop/)

Apache Spark
(https://sparkbyexamples.com/category/s
park/)

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

Apache Kafka
(https://sparkbyexamples.com/category/k
afka/)

Apache HBase
(https://sparkbyexamples.com/category/h
base/)

Apache Cassandra
(https://sparkbyexamples.com/category/c
assandra/)

Snowflake Database
(https://sparkbyexamples.com/category/s
nowflake/)

H2O Sparkling Water
(https://sparkbyexamples.com/category/h
2o-sparkling-water/)

**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/)

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/)

**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**

| (https: | (https: |
| --- | --- |
| //www. | //www. |

| (https: | facebo | linkedi | (https: |
| --- | --- | --- | --- |
| //twitte | ok.co | n.com/ | //githu |
| r.com/ | m/spar | in/n- | b.com/ |
| sparkb | kbyex | nk- | spark- |
| yexam | ample | b860a | examp |
| ples) | s/) | 8193/) | les/) |

PySpark
(https://sparkbyexamples.com/category/p
yspark/)

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/)