**Azure Data Factory Data Flow Function Cheat Sheet**

| Data Flow Expression Function | | |
|---|---|---|
| **Expression** | **Short Description** | **Example** |
| abs | Returns the absolute value of the input. | `abs(-10)` returns `10`. |
| acos | Returns the arc cosine of the input. | `acos(0.5)` returns `1.0472`. |
| add | Adds two or more numeric inputs. | `add(5, 10)` returns `15`. |
| and | Returns true if all of the input expressions evaluate to true. | `and(true, true, false)` returns `false`. |
| asin | Returns the arc sine of the input. | `asin(0.5)` returns `0.5236`. |
| assertErrorMessages | Throws an error if there are any error messages in the input. | `assertErrorMessages()` throws an error if there are any error messages. |
| atan | Returns the arc tangent of the input. | `atan(1)` returns `0.7854`. |
| atan2 | Returns the arc tangent of y/x, where y and x are the inputs. | `atan2(1, 1)` returns `0.7854`. |
| between | Returns true if the input is between the lower and upper bounds (inclusive). | `between(5, 1, 10)` returns `true`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| bitwiseAnd | Returns the bitwise AND of the input expressions. | `bitwiseAnd(3, 6)` returns 2. |
| bitwiseOr | Returns the bitwise OR of the input expressions. | `bitwiseOr(3, 6)` returns 7. |
| bitwiseXor | Returns the bitwise XOR of the input expressions. | `bitwiseXor(3, 6)` returns 5. |
| cbrt | Returns the cube root of the input. | `cbrt(27)` returns 3. |
| ceil | Returns the smallest integer that is greater than or equal to the input. | `ceil(3.14)` returns 4. |
| coalesce | Returns the first non-null input. | `coalesce(null, "hello", null, "world")` returns `"hello"`. |
| columnNames | Returns a list of column names from the input table. | `columnNames(table)` returns `["column1", "column2", "column3"]`. |
| compare | Compares two inputs and returns -1 if the first input is less than the second, 0 if they are equal, or 1 if the first input is greater than the second. | `compare(1, 2)` returns -1. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| concat | Concatenates two or more input strings. | `concat("hello", "world")` returns `"helloworld"`. |
| concatWS | Concatenates two or more input strings with a separator. | `concatWS(",", "apple", "banana", "cherry")` returns `"apple,banana,cherry"`. |
| cos | Returns the cosine of the input (in radians). | `cos(0)` returns `1`. |
| cosh | Returns the hyperbolic cosine of the input. | `cosh(1)` returns `1.5431`. |
| crc32 | Returns the CRC-32 checksum of the input. | `crc32("Hello, world!")` returns `222957957`. |
| degrees | Converts the input from radians to degrees. | `degrees(3.14159)` returns `180`. |
| divide | Divides two input values. | `divide(10, 2)` returns `5`. |
| dropLeft | Removes a specified number of characters from the beginning of the input. | `dropLeft("hello", 2)` returns `"llo"`. |
| dropRight | Removes a specified number of characters from the end of the input. | `dropRight("hello", 2)` returns `"hel"`. |
| endsWith | Returns true if the input ends with | `endsWith("hello", "lo")` returns `true`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| | the specified substring. | |
| equals | Returns true if the two input values are equal. | `equals(1, 2)` returns `false`. |
| equalsIgnoreCase | Returns true if the two input strings are equal (ignoring case). | `equalsIgnoreCase("hello", "HELLO")` returns `true`. |
| escape | Escapes special characters in the input string. | `escape("It's a beautiful day!")` returns `"It¥'s a beautiful day!"`. |
| expr | Evaluates a string expression. | `expr("1 + 2")` returns `3`. |
| factorial | Returns the factorial of the input. | `factorial(5)` returns `120`. |
| FALSE | Returns the boolean value false. | `FALSE` returns `false`. |
| floor | Returns the largest integer that is less than or equal to the input. | `floor(3.14)` returns `3`. |
| fromBase64 | Decodes a base64-encoded input string. | `fromBase64("SGVsbG8sIHdvcmxkIQ==")` returns `"Hello, world!"`. |
| greater | Returns true if the first input is greater than the second. | `greater(2, 1)` returns `true`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| greaterOrEqual | Returns true if the first input is greater than or equal to the second. | `greaterOrEqual(2, 2)` returns `true`. |
| greatest | Returns the greatest value among the input expressions. | `greatest(1, 2, 3, 4)` returns `4`. |
| hasColumn | Returns true if the input table contains the specified column. | `hasColumn(table, "column1")` returns `true`. |
| hasError | Returns true if the input contains an error message. | `hasError(input)` returns `true` if `input` contains an error message. |
| iif | Returns the second input if the first input is true, otherwise returns the third input. | `iif(1 > 2, "One is greater than two", "One is not greater than two")` returns `` `"One is not greater than`` |
| iifNull | Returns the second input if the first input is not null, otherwise returns the third input. | `iifNull(null, "Input is null", "Input is not null")` returns `"Input is null"`. |
| initCap | Capitalizes the first letter of each word in the input string. | `initCap("hello world")` returns `"Hello World"`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| instr | Returns the position of the first occurrence of a substring in the input string. | `instr("hello world", "world")` returns `7`. |
| isDelete | Returns true if the input is a delete operation. | `isDelete(input)` returns `true` if `input` is a delete operation. |
| isError | Returns true if the input contains an error message. | `isError(input)` returns `true` if `input` contains an error message. |
| isIgnore | Returns true if the input is an ignore operation. | `isIgnore(input)` returns `true` if `input` is an ignore operation. |
| isInsert | Returns true if the input is an insert operation. | `isInsert(input)` returns `true` if `input` is an insert operation. |
| isMatch | Returns true if the input matches the specified regular expression. | `isMatch("hello", "hel*o")` returns `true`. |
| isNull | Returns true if the input is null. | `isNull(null)` returns `true`. |
| isUpdate | Returns true if the input is an update operation. | `isUpdate(input)` returns `true` if `input` is an update operation. |
| isUpsert | Returns true if the input is an upsert operation. | `isUpsert(input)` returns `true` if `input` is an upsert operation. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| jaroWinkler | Returns the Jaro-Winkler distance between two input strings. | `jaroWinkler("hello", "world")` returns `0`. |
| least | Returns the least value among the input expressions. | `least(1, 2, 3, 4)` returns `1`. |
| left | Returns the specified number of characters from the beginning of the input. | `left("hello", 2)` returns `"he"`. |
| length | Returns the length of the input string. | `length("hello")` returns `5`. |
| lesser | Returns true if the first input is less than the second. | `lesser(1, 2)` returns `true`. |
| lesserOrEqual | Returns true if the first input is less than or equal to the second. | `lesserOrEqual(2, 2)` returns `true`. |
| levenshtein | Returns the Levenshtein distance between two input strings. | `levenshtein("hello", "world")` returns `5`. |
| like | Returns true if the input matches the specified pattern. | `like("hello world", "%world")` returns `true`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| locate | Returns the position of the first occurrence of a substring in the input string. | `locate("world", "hello world")` returns **6**. |
| log | Returns the natural logarithm of the input. | `log(2.718)` returns **1**. |
| log10 | Returns the base-10 logarithm of the input. | `log10(100)` returns **2**. |
| lower | Converts the input string to lowercase. | `lower("HELLO")` returns `"hello"`. |
| lpad | Pads the input string with the specified character until it is the specified length. | `lpad("hello", 7, "*")` returns `"**hello"`. |
| ltrim | Removes leading whitespace from the input string. | `` `ltrim `` |
| ltrim | Removes leading whitespace from the input string. | `ltrim(" hello")` returns `"hello"`. |
| md5 | Returns the MD5 hash of the input. | `md5("Hello, world!")` returns `"ed076287532e86365e841e92bfc50d8c"`. |
| minus | Subtracts the second | `minus(10, 2)` returns **8**. |

**Azure Data Factory Data Flow Function Cheat Sheet**

|  |  |  |
|---|---|---|
|  | input from the first. |  |
| mod | Returns the remainder when the first input is divided by the second. | `mod(10, 3)` returns `1`. |
| multiply | Multiplies two or more input values. | `multiply(2, 3, 4)` returns `24`. |
| negate | Negates the input value. | `negate(5)` returns `-5`. |
| nextSequence | Returns the next value in a sequence. | `nextSequence("mySequence")` returns the next value in the "mySequence" sequence. |
| normalize | Normalizes the input string to Unicode NFC normalization form. | `normalize("NFD", "Å")` returns `"A¥u030A"`. |
| not | Returns the negation of the input boolean. | `not(true)` returns `false`. |
| notEquals | Returns true if the two input values are not equal. | `notEquals(1, 2)` returns `true`. |
| null | Returns null. | `null` returns `null`. |
| or | Returns true if any of the input expressions evaluate to true. | `or(false, true, false)` returns `true`. |
| pMod | Returns the positive remainder when the | `pMod(-10, 3)` returns `2`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| | first input is divided by the second. | |
| partitionId | Returns the partition ID for the input. | `partitionId(input)` returns the partition ID for `input`. |
| power | Returns the first input raised to the power of the second. | `power(2, 3)` returns `8`. |
| radians | Converts the input from degrees to radians. | `radians(180)` returns `3.14159`. |
| random | Returns a random value between 0 and 1. | `random()` returns a random value. |
| regexExtract | Returns the first substring that matches the specified regular expression. | `regexExtract("hello world", "w.*")` returns `"world"`. |
| regexMatch | Returns true if the input matches the specified regular expression. | `regexMatch("hello", "hel*o")` returns `true`. |
| regexReplace | Replaces all occurrences of a substring that matches the specified regular expression | `regexReplace("hello world", "¥¥w+", "123")` returns `"123 123"`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

|  |  |  |
|---|---|---|
|  | with another string. |  |
| regexSplit | Splits the input string using the specified regular expression as the delimiter. | `regexSplit("hello,world", ",")` returns `["hello", "world"]`. |
| replace | Replaces all occurrences of a substring with another string. | `replace("hello world", "o", "0")` returns `"hell0 w0rld"`. |
| reverse | Reverses the characters in the input string. | `reverse("hello")` returns `"olleh"`. |
| right | Returns the specified number of characters from the end of the input. | `right("hello", 2)` returns `"lo"`. |
| rlike | Returns true if the input matches the specified regular expression (case-sensitive). | `rlike("hello", "hel*o")` returns `true`. |
| round | Rounds the input to the nearest integer. | `round(3.14)` returns `3`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| rpad | Pads the input string with the specified character until it is the | |
| rpad | Pads the input string with the specified character until it is the specified length. | `rpad("hello", 7, "*")` returns `"hello**"`. |
| rtrim | Removes trailing whitespace from the input string. | `rtrim("hello ")` returns `"hello"`. |
| sha1 | Returns the SHA-1 hash of the input. | `sha1("Hello, world!")` returns `"0a4d55a8d778e5022fab701977c5d840bbc486d0"`. |
| sha2 | Returns the SHA-2 hash of the input with the specified bit length (256 or 512). | `sha2("Hello, world!", 256)` returns `"a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146"`. |
| sin | Returns the sine of the input (in radians). | `sin(0)` returns `0`. |
| sinh | Returns the hyperbolic sine of the input. | `sinh(1)` returns `1.1752`. |
| soundex | Returns the Soundex code for the input string. | `soundex("hello")` returns `"H400"`. |
| split | Splits the input string using the | `split("hello,world", ",")` returns `["hello", "world"]`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| | specified delimiter. | |
| sqrt | Returns the square root of the input. | `sqrt(16)` returns `4`. |
| startsWith | Returns true if the input starts with the specified substring. | `startsWith("hello", "he")` returns `true`. |
| substring | Returns a substring of the input starting at the specified position and with the specified length. | `substring("hello", 1, 3)` returns `"ell"`. |
| substringIndex | Returns the position of the first occurrence of the specified delimiter in the input string, starting from the beginning or end of the string. | `substringIndex("hello,world", ",", 1)` returns `6`. |
| tan | Returns the tangent of the input (in radians). | `tan(0)` returns `0`. |
| tanh | Returns the hyperbolic tangent of the input. | `tanh(1)` returns `0.7616`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| translate | Replaces each character in the input string that matches a character in the specified from string with the corresponding character in the to string. | `translate("hello world", "aeiou", "12345")` returns `"h2ll4 w4rld"`. |
| trim | Removes leading and trailing whitespace from the input string. | `trim(" hello ")` returns `"hello"`. |
| TRUE | Returns the boolean value true. | `TRUE` returns `true`. |
| typeMatch | Returns true if the input matches the specified type. | `typeMatch(input, "type")` returns `true` if `input` matches the specified type. |
| unescape | Unescapes special characters in the input string. | `unescape("It¥¥'s a beautiful day¥¥!")` returns `"It's a beautiful day!"`. |
| upper | Converts the input string to uppercase. | `upper("hello")` returns `"HELLO"`. |
| uuid | Returns a randomly generated UUID (Universally Unique Identifier). | `uuid()` returns a UUID. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | Returns true if exactly one of the input expressions evaluates to true. | |
|---|---|---|
| xor | Returns true if exactly one of the input expressions evaluates to true. | `xor(true, false)` returns `true`. |

| **Data Flow Aggregate Function** | | |
|---|---|---|
| **Function** | **Short Description** | **Example** |
| approxDistinctCount | Returns the approximate number of distinct values in the input. | `approxDistinctCount(column)` returns the approximate number of distinct values in the `column`. |
| avg | Returns the average of the input values. | `avg(column)` returns the average of the values in the `column`. |
| avgIf | Returns the average of the input values that match a specified condition. | `avgIf(column, condition)` returns the average of the values in the `column` that match the `condition`. |
| collect | Returns an array of the input values. | `collect(column)` returns an array of the values in the `column`. |
| collectUnique | Returns an array of the unique input values. | `collectUnique(column)` returns an array of the unique values in the `column`. |
| count | Returns the number of rows in the input. | `count(*)` returns the number of rows in the input. |
| countAll | Returns the number of non-null values in the input. | `countAll(column)` returns the number of non-null values in the `column`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| countDistinct | Returns the number of distinct values in the input. | `countDistinct(column)` returns the number of distinct values in the `column`. |
| countAllDistinct | Returns the number of non-null distinct values in the input. | `countAllDistinct(column)` returns the number of non-null distinct values in the `column`. |
| countIf | Returns the number of rows in the input that match a specified condition. | `countIf(condition)` returns the number of rows in the input that match the `condition`. |
| covariancePopulation | Returns the population covariance of two input columns. | `covariancePopulation(column1, column2)` returns the population covariance of the values in `column1` and `column2`. |
| covariancePopulationIf | Returns the population covariance of two input columns that match a specified condition. | `covariancePopulationIf(column1, column2, condition)` returns the population covariance of the values in `column1` and `column2` that match the `condition`. |
| covarianceSample | Returns the sample covariance of two input columns. | `covarianceSample(column1, column2)` returns the sample covariance of the values in `column1` and `column2`. |
| covarianceSampleIf | Returns the sample covariance of two input columns that match a specified condition. | `covarianceSampleIf(column1, column2, condition)` returns the sample covariance of the values in `column1` and `column2` that match the `condition`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| first | Returns the first value in the input. | `first(column)` returns the first value in the `column`. |
|---|---|---|
| isDistinct | Returns true if the input values are distinct. | `isDistinct(column)` returns `true` if the values in the `column` are distinct. |
| kurtosis | Returns the kurtosis of the input values. | `kurtosis(column)` returns the kurtosis of the values in the `column`. |
| kurtosisIf | Returns the kurtosis of the input values that match a specified condition. | `kurtosisIf(column, condition)` returns the kurtosis of the values in the `column` that match the `condition`. |
| last | Returns the last value in the input. | `last(column)` returns the last value in the `column`. |
| max | Returns the maximum value in the input. | `max(column)` returns the maximum value in the `column`. |
| maxIf | Returns the maximum value in the input that matches a specified condition. | `maxIf(column, condition)` returns the maximum value in the `column` that matches the `condition`. |
| mean | Returns the mean of the input values. | `mean |
| mean | Returns the mean of the input values. | `mean(column)` returns the mean of the values in the `column`. |
| meanIf | Returns the mean of the input values that match a specified condition. | `meanIf(column, condition)` returns the mean of the values in the `column` that match the `condition`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| min | Returns the minimum value in the input. | `min(column)` returns the minimum value in the `column`. |
| minIf | Returns the minimum value in the input that matches a specified condition. | `minIf(column, condition)` returns the minimum value in the `column` that matches the `condition`. |
| skewness | Returns the skewness of the input values. | `skewness(column)` returns the skewness of the values in the `column`. |
| skewnessIf | Returns the skewness of the input values that match a specified condition. | `skewnessIf(column, condition)` returns the skewness of the values in the `column` that match the `condition`. |
| stddev | Returns the standard deviation of the input values. | `stddev(column)` returns the standard deviation of the values in the `column`. |
| stddevIf | Returns the standard deviation of the input values that match a specified condition. | `stddevIf(column, condition)` returns the standard deviation of the values in the `column` that match the `condition`. |
| stddevPopulation | Returns the population standard deviation of the input values. | `stddevPopulation(column)` returns the population standard deviation of the values in the `column`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| stddevPopulationIf | Returns the population standard deviation of the input values that match a specified condition. | `stddevPopulationIf(column, condition)` returns the population standard deviation of the values in the `column` that match the `condition`. |
| stddevSample | Returns the sample standard deviation of the input values. | `stddevSample(column)` returns the sample standard deviation of the values in the `column`. |
| stddevSampleIf | Returns the sample standard deviation of the input values that match a specified condition. | `stddevSampleIf(column, condition)` returns the sample standard deviation of the values in the `column` that match the `condition`. |
| sum | Returns the sum of the input values. | `sum(column)` returns the sum of the values in the `column`. |
| sumDistinct | Returns the sum of the distinct input values. | `sumDistinct(column)` returns the sum of the distinct values in the `column`. |
| sumDistinctIf | Returns the sum of the distinct input values that match a specified condition. | `sumDistinctIf(column, condition)` returns the sum of the distinct values in the `column` that match the `condition`. |
| sumIf | Returns the sum of the input values that match a specified condition. | `sumIf(column, condition)` returns the sum of the values in the `column` that match the `condition`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| topN | Returns the top N values in the input. | `topN(column, N)` returns the top `N` values in the `column`. |
| variance | Returns the variance of the input values. | `variance(column)` returns the variance of the values in the `column`. |
| varianceIf | Returns the variance of the input values that match a specified condition. | `varianceIf(column, condition)` returns the variance of the values in the `column` that match the `condition`. |
| variancePopulation | Returns the population variance of the input values. | `variancePopulation(column)` returns the population variance of the values in the `column`. |
| variancePopulation | Returns the population variance of the input values. | `variancePopulation(column)` returns the population variance of the values in the `column`. |
| variancePopulationIf | Returns the population variance of the input values that match a specified condition. | `variancePopulationIf(column, condition)` returns the population variance of the values in the `column` that match the `condition`. |
| varianceSample | Returns the sample variance of the input values. | `varianceSample(column)` returns the sample variance of the values in the `column`. |
| varianceSampleIf | Returns the sample variance of the input values that match a | `varianceSampleIf(column, condition)` returns the sample variance of the values in the `column` that match the `condition` |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | specified condition. | |
|---|---|---|
| | | |

| Array Functions in Data Flow | | |
|---|---|---|
| **Function** | **Short Description** | **Example** |
| array | Creates an array from the input values. | `array("a", "b", "c")` returns `["a", "b", "c"]`. |
| at | Returns the value at the specified index in the input array. | `at(array, index)` returns the value at the `index` in the `array`. |
| contains | Returns true if the input array contains the specified value. | `contains(array, value)` returns `true` if the `array` contains the `value`. |
| distinct | Returns an array of the distinct values in the input array. | `distinct(array)` returns an array of the distinct values in the `array`. |
| except | Returns an array containing the elements in the first input array that are not present in any of the subsequent input arrays. | `except(array1, array2)` returns an array containing the elements in `array1` that are not present in `array2`. |
| filter | Returns an array containing the elements in the input | `filter(array, condition)` returns an array containing the elements in the `array` that match the `condition`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| | array that match a specified condition. | |
| find | Returns the index of the first occurrence of the specified value in the input array, or -1 if the value is not found. | `find(array, value)` returns the index of the first occurrence of the **value** in the **array**, or -1 if the **value** is not found. |
| flatten | Returns an array containing all the elements of the input arrays flattened into a single array. | `flatten(array1, array2)` returns an array containing all the elements of **array1** and **array2**. |
| in | Returns true if the input value is found in the input array. | `in(value, array)` returns **true** if the **value** is found in the **array**. |
| intersect | Returns an array containing the elements that are present in all of the input arrays. | `intersect(array1, array2)` returns an array containing the elements that are present in both **array1** and **array2**. |
| map | Returns an array containing the result of applying a function to | `map(array, function)` returns an array containing the result of applying the **function** to each element of the **array**. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| | each element of the input array. | |
| mapIf | Returns an array containing the result of applying a function to each element of the input array that matches a specified condition. | `mapIf(array, condition, function)` returns an array containing the result of applying the `function` to each element of the `array` that matches the `condition`. |
| mapIndex | Returns an array containing the result of applying a function to each element and index of the input array. | `mapIndex(array, function)` returns an array containing the result of applying the `function` to each element and index of the `array`. |
| mapLoop | Returns an array containing the result of applying a function to each element of the input array and the previous and next elements. | `mapLoop(array, function)` returns an array containing the result of applying the `function` to each element of the `array` and the previous and next elements. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| reduce | Returns the result of applying a function to each element of the input array to produce a single output value. | `reduce(array, function)` returns the result of applying the `function` to each element of the `array` to produce a single output value. |
| size | Returns the number of elements in the input array. | `size(array)` returns the number of elements in the `array`. |
| slice | Returns a subarray of the input array starting at the specified index and with the specified length. | `slice(array, start, length)` returns a subarray of the `array` starting at the `start` index and with the specified `length`. |
| sort | Returns an array containing the input array sorted in ascending order. | `sort(array)` returns an array containing the `array` sorted in ascending order. |
| unfold | Returns an array containing the input array with each element expanded into a | `unfold(array)` returns an array containing the `array` with each element expanded into a separate row. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | separate row. | |
|---|---|---|
| union | Returns an array containing the elements that are present in any of the input arrays. | `union(array1, array2)` returns an array containing the elements that are present in **array1** or **array2**. |
| **Date Time Function in DataFlow** | | |
| **Function** | **Short Description** | **Example** |
| add | Adds two numbers. | `add(2, 3)` returns **5**. |
| addDays | Adds a specified number of days to the input date. | `addDays(date, days)` returns a new date that is **days** days after the **date**. |
| addMonths | Adds a specified number of months to the input date. | `addMonths(date, months)` returns a new date that is **months** months after the **date**. |
| between | Returns true if the input value is between two specified values. | `between(value, lower, upper)` returns **true** if the **value** is between **lower** and **upper**. |
| currentDate | Returns the current date. | `currentDate()` returns the current date. |
| currentTimestamp | Returns the current timestamp. | `currentTimestamp()` returns the current timestamp. |
| currentUTC | Returns the current UTC datetime. | `currentUTC()` returns the current UTC datetime. |

### Azure Data Factory Data Flow Function Cheat Sheet

| | | |
|---|---|---|
| dayOfMonth | Returns the day of the month of the input date. | `dayOfMonth(date)` returns the day of the month of the **date**. |
| dayOfWeek | Returns the day of the week of the input date. | `dayOfWeek(date)` returns the day of the week of the **date**. |
| dayOfYear | Returns the day of the year of the input date. | `dayOfYear(date)` returns the day of the year of the **date**. |
| days | Returns the number of days between two dates. | `days(date1, date2)` returns the number of days between **date1** and **date2**. |
| fromUTC | Converts the input UTC datetime to a local datetime in the specified timezone. | `fromUTC(utcDatetime, timezone)` returns the local datetime for the **utcDatetime** in the specified **timezone**. |
| hour | Returns the hour component of the input datetime. | `hour(datetime)` returns the hour component of the **datetime**. |
| hours | Returns the number of hours between two datetimes. | `hours(datetime1, datetime2)` returns the number of hours between **datetime1** and **datetime2**. |
| isDate | Returns true if the input value is a valid date. | `isDate(value)` returns **true** if the **value** is a valid date. |
| isTimestamp | Returns true if the input value is a valid timestamp. | `isTimestamp(value)` returns **true** if the **value** is a valid timestamp. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| lastDayOfMonth | Returns the last day of the month of the input date. | `lastDayOfMonth(date)` returns the last day of the month of the `date`. |
| millisecond | Returns the millisecond component of the input datetime. | `millisecond(datetime)` returns the millisecond component of the `datetime`. |
| milliseconds | Returns the number of milliseconds between two datetimes. | `milliseconds(datetime1, datetime2)` returns the number of milliseconds between `datetime1` and `datetime2`. |
| minus | Subtracts two numbers. | `minus(5, 3)` returns `2`. |
| minute | Returns the minute component of the input datetime. | `minute(datetime)` returns the minute component of the `datetime`. |
| minutes | Returns the number of minutes between two datetimes. | `minutes(datetime1, datetime2)` returns the number of minutes between `datetime1` and `datetime2`. |
| month | Returns the month component of the input date. | `month(date)` returns the month component of the `date`. |
| monthsBetween | Returns the number of months between two dates. | `monthsBetween(date1, date2)` returns the number of months between `date1` and `date2`. |
| second | Returns the second component of the input datetime. | `second(datetime)` returns the second component of the `datetime`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | | |
|---|---|---|
| seconds | Returns the number of seconds between two datetimes. | `seconds(datetime1, datetime2)` returns the number of seconds between `datetime1` and `datetime2`. |
| subDays | Subtracts a specified number of days from the input date. | `subDays(date, days)` returns a new date that is `days` days before the `date`. |
| subMonths | Subtracts a specified number of months from the input date. | `subMonths(date, months)` returns a new date that is `months` months before the `date`. |
| toDate | Converts a string to a date. | `toDate(string, format)` returns the date represented by the `string` in the specified `format`. |
| toTimestamp | Converts a string to a timestamp. | `toTimestamp(string, format)` returns the timestamp represented by the `string` in the specified `format`. |
| toUTC | Converts the input datetime to a UTC datetime. | `toUTC(datetime)` returns the UTC datetime for the `datetime`. |
| weekOfYear | Returns the ISO week of the year of the input date. | `weekOfYear(date)` returns the ISO week of the year of the `date`. |
| weeks | Returns the number of weeks between two dates. | `weeks(date1, date2)` returns the number of weeks between `date1` and `date2`. |
| year | Returns the year component | `year(date)` returns the year component of the `date`. |

**Azure Data Factory Data Flow Function Cheat Sheet**

| | of the input date. | |
|---|---|---|
| | | |