REGEX Expression for Date Validation(YYYY-MM-DD)

```
^(?:(19|20)\d{2})-(0[1-9]|1[0-2])-(0[1-9]|1\d|2[08])$|
^(?:(19|20)([02468][048]|[13579][26])|2000)-02-29$|
^(?:(19|20)\d{2})-(0[13-9]|1[0-2])-(29|30)$|
^(?:(19|20)\d{2})-(0[13578]|1[02])-31$
```

The pattern is divided into four major sections:

General date validation for most months (1-28 days).

Leap year validation for February 29.

Validation for months with 29 or 30 days.

Validation for months with 31 days.

Step-by-Step Explanation:

Part 1:

General Date Validation for Months with 28 Days

```
^((19|20)\d{2})-(0[1-9]|1[0-2])-(0[1-9]|1\d|2[0-8])$
```

This part of the pattern handles the general case for all months where the day is between 1 and 28. It matches dates like YYYY-MM-DD where:

Year (YYYY): The year is between 1900 and 2099.

(19|20)\d{2}: This checks if the year starts with 19 or 20, followed by any two digits.

Month (MM): The month is valid (from 01 to 12).

(0[1-9]|1[0-2]): This matches months like 01-09 or 10-12.

Day (DD): The day is valid for months with up to 28 days.

(0[1-9]|1\d|2[0-8]): This matches days from 01 to 28.

The ^ and $ denote the start and end of the string, ensuring that the entire date string must conform to this pattern.

Part 2:

Leap Year Validation for February 29:

^((19|20)([02468][048]|[13579][26])|2000)-02-29$

This part handles February 29 for leap years. It matches dates like YYYY-02-29 where:

Year (YYYY): The year must be a leap year.

(19|20)([02468][048]|[13579][26]): This ensures that the year is divisible by 4 but not by 100, except if divisible by 400 (e.g., 2000 is valid). Specifically, the pattern checks:

19xx or 20xx: The year must start with 19 or 20.

Leap year rule: The last two digits of the year must satisfy the leap year rule, e.g., 04, 08, 12, 16, 20, 24....

|2000: This special case ensures the year 2000 is allowed as a leap year (even though it's divisible by 100, it's also divisible by 400, making it a leap year).

Month and Day: The pattern specifically matches February 29.

-02-29: This ensures that the month is February (02), and the day is the 29th.

Part 3:

Validation for Months with 29 or 30 Days

^((19|20)\d{2})-(0[13-9]|1[0-2])-(29|30)$

This part of the pattern handles months that have either 29 or 30 days. It matches dates like YYYY-MM-29 or YYYY-MM-30 where:

Year (YYYY): The year is between 1900 and 2099.

(19|20)\d{2}: This checks if the year starts with 19 or 20, followed by any two digits.

Month (MM): The month must allow 29 or 30 days.

(0[13-9]|1[0-2]): This matches months like March (03), April (04), May (05), June (06), September (09), October (10), November (11), and December (12).

Day (DD): The day must be either 29 or 30.

(29|30): This ensures that the day is either 29th or 30th.

Part 4:

Validation for Months with 31 Days:

^((19|20)\d{2})-(0[13578]|1[02])-31$

This part of the pattern handles months that have 31 days. It matches dates like YYYY-MM-31 where:

Year (YYYY): The year is between 1900 and 2099.

(19|20)\d{2}: This checks if the year starts with 19 or 20, followed by any two digits.

Month (MM): The month must have 31 days.

(0[13578]|1[02]): This matches months that have 31 days, such as January (01), March (03), May (05), July (07), August (08), October (10), and December (12).

Day (DD): The day must be 31.

-31: This ensures that the day is the 31st.

Summary of Date Validation Logic:

Non-leap year February: Only allows days up to 28.

Leap year February 29: Allows the date February 29 only in valid leap years.

Months with 29 or 30 days: March, April, May, June, September, October, November, and December allow the 29th or 30th day.

Months with 31 days: January, March, May, July, August, October, and December allow the 31st day.

Testing Examples:

Valid dates:

2024-02-29(leap year), 2023-12-31 (non-leap year)

2020-02-29 (leap year), 1996-04-30 (non-leap year)

Invalid dates:

2021-02-29(non-leap year)

2019-04-31 (April has 30 days)

1900-02-29 (not a leap year)