Data in Motion: Data Cleaning Challenge

Let's take a look at the data...

In [ ]:
```
import pandas as pd, re
def get_data():
 ''' This function reads the data from the csv file
 and returns a pandas dataframe '''

 data = pd.read_csv('historical_events_data.csv')
 data.fillna('null', inplace=True)

 return data

get_data()
```

Out[ ]:

	Event	Date
0	Moon Landing	07-20-1969
1	End of WWII	1945
2	Chernobyl Disaster	26th April 1986
3	Fall of Berlin Wall	null
4	Invention of Internet	1960s

Challenge Tasks:

Date Parsing: Standardize the 'Date' column into a consistent YYYY-MM-DD format. For dates with only the year mentioned, use YYYY-01-01 as the standard format.

Time Period Extraction: Create separate columns for 'Year', 'Month', and 'Day'. If a specific detail is missing, leave it as 'Unknown'.

Date Imputation: For the 'Fall of Berlin Wall' event, the date is missing. Impute it with '1989-11-09'.

Chronological Ordering: Sort the events in ascending order of their occurrence.

Building Our Data Cleaning Program

Now that we have our steps, let's build a program that will clean the data.

In [ ]:
```
def convert_date(Date):
 ''' This function converts the date
 into a pandas datetime format '''

# regex pattern for our text date
 pattern = r'(\d{1,2})\w{2} (\w+) (\d{4})'
 match = re.match(pattern, Date)
 month_dict = {'April': 4}

 if len(str(Date)) <= 5:   # find years and convert to date
    Date = re.sub(r'[^0-9]','',Date)
    Date = pd.to_datetime(Date, format='%Y-%m-%d')

 elif Date.startswith('0'): # find date format 04/01/2020 & convert to date
    Date = pd.to_datetime(Date, format='%m-%d-%Y')

 elif match:      # match text date like 1st April 2020 and convert to date
    day = int(match.group(1))
    month = match.group(2)
    year = int(match.group(3))
    month_value = month_dict.get(month)
    Date = pd.to_datetime(f'{year}-{month_value}-{day}', format='%Y-%m-%d')

 return Date
```

In [ ]:
```
def clean_Data(df):
 ''' This function will clean the data '''
 print('Commencing data cleaning...')
 cleaned_df = (
 df
 .assign(Date = lambda x: x['Date'].apply(convert_date))
 .assign(Date = lambda x: x['Date'].replace(pd.NaT, pd.to_datetime('1989-11-09')))
 .assign(Year = lambda x: x['Date'].dt.year)
 .assign(Month = lambda x: x['Date'].dt.month)
 .assign(Day = lambda x: x['Date'].dt.day)
 .sort_values(by='Date', ascending=True)
 )
 print('Data successfully cleaned.')
 return cleaned_df
```

In [ ]:
```
def main():
 ''' This is the main function '''
 historical_magazine_data = get_data()
 cleaned_df = clean_Data(historical_magazine_data)

 return cleaned_df
```

In [ ]:
```
main()
```

```
Commencing data cleaning...
Data successfully cleaned.
```

Out[ ]:

	Event	Date	Year	Month	Day
1	End of WWII	1945-01-01	1945	1	1
4	Invention of Internet	1960-01-01	1960	1	1
0	Moon Landing	1969-07-20	1969	7	20
2	Chernobyl Disaster	1986-04-26	1986	4	26
3	Fall of Berlin Wall	1989-11-09	1989	11	9