```
# Carbon Footprint Calculator
def transportation_emission(car_km=0, car_fuel_efficiency=12, flights_per_year=0)
    # Car CO2 (kg per year)
    # Assumes gasoline emits 2.31 kg CO2 per liter
    car_liters = car_km / car_fuel_efficiency
    car_emission = car_liters * 2.31

    # Flight CO2 (kg per flight)
    # Short-haul: 250 kg, Long-haul: 1100 kg (average)
    flight_emission = flights_per_year * 250

    return car_emission + flight_emission

def electricity_emission(kwh_per_month=0, renewable=False):
    # Average grid emission factor: 0.475 kg CO2/kWh
    if renewable:
        factor = 0.05
    else:
        factor = 0.475
    return kwh_per_month * 12 * factor

def diet_emission(meat_type='medium'):
    # Meat consumption CO2 per year (kg)
    if meat_type == 'high':
        return 3000
    elif meat_type == 'medium':
        return 1500
    else:
        return 800

def waste_emission(waste_kg_per_week=0, recycle=True):
    # Waste emissions per year
    if recycle:
        factor = 0.1
    else:
        factor = 0.25
    return waste_kg_per_week * 52 * factor

def total_carbon_footprint(car_km=0, car_fuel_efficiency=12, flights_per_year=0,
                           kwh_per_month=0, renewable=False, meat_type='medium',
                           waste_kg_per_week=0, recycle=True):
    total = (transportation_emission(car_km, car_fuel_efficiency, flights_per_yea
             electricity_emission(kwh_per_month, renewable) +
             diet_emission(meat_type) +
             waste_emission(waste_kg_per_week, recycle))
    return total

# Example usage
footprint = total_carbon_footprint(car_km=12000, car_fuel_efficiency=15, flights_
                                   kwh_per_month=350, renewable=False, meat_type=
                                   waste_kg_per_week=5, recycle=True)
print(f"Your estimated annual carbon footprint is {footprint:.2f} kg CO₂")
```

```
Your estimated annual carbon footprint is 5869.00 kg CO₂
```

```
car_km = float(input("Enter km d ✦ n per year by car: "))
flights_per_year = int(input("Enter number of short-haul flights per year: "))
```

```
kwh_per_month = float(input("Enter monthly electricity usage (kWh): "))
meat_type = input("Enter meat consumption (high/medium/low): ").lower()
waste_kg_per_week = float(input("Enter waste produced per week (kg): "))

footprint = total_carbon_footprint(car_km=car_km, flights_per_year=flights_per_ye
                                   kwh_per_month=kwh_per_month, meat_type=meat_ty
                                   waste_kg_per_week=waste_kg_per_week)
print(f"Your annual carbon footprint is {footprint:.2f} kg CO₂")
```

```
Enter km driven per year by car: 100
Enter number of short-haul flights per year: 90
Enter monthly electricity usage (kWh): 75
Enter meat consumption (high/medium/low): 90
Enter waste produced per week (kg): 1000000
Your annual carbon footprint is 5223746.75 kg CO₂
```

```
# Carbon Footprint Calculator - Interactive Version
# Run this in Google Colab

# Install ipywidgets if not already installed
!pip install ipywidgets --quiet

import ipywidgets as widgets
from IPython.display import display, Markdown
import matplotlib.pyplot as plt

# ----------------------------
# Carbon Footprint Calculation Functions
# ----------------------------
def transportation_emission(car_km=0, car_fuel_efficiency=12, flights_per_year=0)
    # Car CO2 (kg per year)
    car_liters = car_km / car_fuel_efficiency
    car_emission = car_liters * 2.31
    # Flights CO2 (kg per flight, short-haul)
    flight_emission = flights_per_year * 250
    return car_emission + flight_emission

def electricity_emission(kwh_per_month=0, renewable=False):
    factor = 0.05 if renewable else 0.475
    return kwh_per_month * 12 * factor

def diet_emission(meat_type='medium'):
    if meat_type == 'high':
        return 3000
    elif meat_type == 'medium':
        return 1500
    else:
        return 800

def waste_emission(waste_kg_per_week=0, recycle=True):
    factor = 0.1 if recycle else 0.25
    return waste_kg_per_week * 52 * factor

def total_carbon_footprint(car_km=0, car_fuel_efficiency=12, flights_per_year=0,
                           kwh_per_month=0, renewable=False, meat_type='medium',
                           waste_kg_per_week=0, recycle=True):
    transport = transportation_emission(car_km, car_fuel_efficiency, flights_per_
    electricity = electricity_emission(kwh_per_month, renewable)
```

```
    diet = diet_emission(meat_type)
    waste = waste_emission(waste_kg_per_week, recycle)
    total = transport + electricity + diet + waste
    return total, {'Transportation': transport, 'Electricity': electricity, 'Diet

# ----------------------------
# Interactive Widgets
# ----------------------------
car_km_slider = widgets.IntSlider(value=12000, min=0, max=50000, step=500, descri
fuel_eff_slider = widgets.IntSlider(value=15, min=5, max=25, step=1, description=
flights_slider = widgets.IntSlider(value=2, min=0, max=20, step=1, description='F
kwh_slider = widgets.IntSlider(value=350, min=0, max=1000, step=10, description='
renewable_checkbox = widgets.Checkbox(value=False, description='Renewable energy?
meat_dropdown = widgets.Dropdown(options=['low', 'medium', 'high'], value='medium
waste_slider = widgets.IntSlider(value=5, min=0, max=20, step=1, description='Was
recycle_checkbox = widgets.Checkbox(value=True, description='Recycle waste?')

button = widgets.Button(description="Calculate Carbon Footprint")

output = widgets.Output()

# ----------------------------
# Function to display result
# ----------------------------
def calculate_footprint(b):
    total, details = total_carbon_footprint(
        car_km=car_km_slider.value,
        car_fuel_efficiency=fuel_eff_slider.value,
        flights_per_year=flights_slider.value,
        kwh_per_month=kwh_slider.value,
        renewable=renewable_checkbox.value,
        meat_type=meat_dropdown.value,
        waste_kg_per_week=waste_slider.value,
        recycle=recycle_checkbox.value
    )

    with output:
        output.clear_output()
        display(Markdown(f"### Your Annual Carbon Footprint: **{total:.2f} kg CO₂*
        display(Markdown("#### Breakdown by category:"))
        for key, value in details.items():
            display(Markdown(f"- **{key}**: {value:.2f} kg CO₂"))

        # Plot a bar chart
        plt.figure(figsize=(6,4))
        plt.bar(details.keys(), details.values(), color=['#FF6F61','#6B5B95','#88
        plt.ylabel("kg CO₂/year")
        plt.title("Carbon Footprint Breakdown")
        plt.show()

button.on_click(calculate_footprint)

# ----------------------------
# Display widgets
# ----------------------------
display(Markdown("## 🌱 Carbon Footprint Calculator"))
display(Markdown("Adjust your lifestyle parameters below to estimate your annual
display(car_km_slider, fuel_eff_slider, flights_slider, kwh_slider, renewable_che
        meat_dropdown, waste_slider, recycle_checkbox, button, output)
```

🌱 Carbon Footprint Calculator

Adjust your lifestyle parameters below to estimate your annual CO_2 emissions:

Car km/year: 12000

Fuel eff km/l: 15

Flights/year: 2

Electricity k... 350

Renewable energy?

Meat intake: medium

Waste kg/w... 5

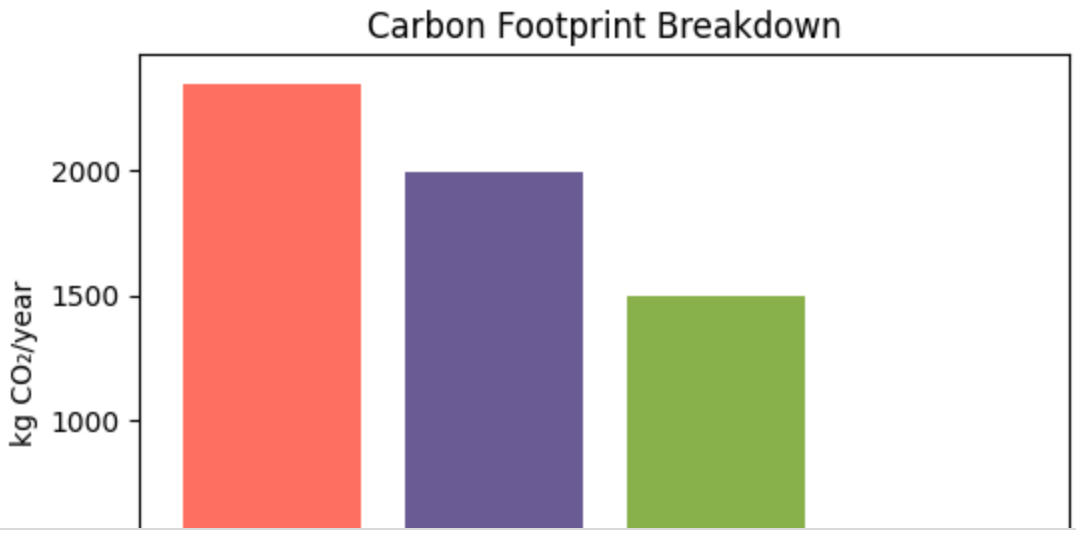
Recycle waste?

Calculate Carbon Fo...

Your Annual Carbon Footprint: 5869.00 kg CO_2

Breakdown by category:

- **Transportation**: 2348.00 kg CO_2
- **Electricity**: 1995.00 kg CO_2
- **Diet**: 1500.00 kg CO_2
- **Waste**: 26.00 kg CO_2




```
from google.colab import drive
drive.mount('/content/drive')
```

```
import shutil
import os
# Install ipynbname if not already installed
try:
    import ipynbname
except ImportError:
```

```
    !pip install ipynbname --quiet
    import ipynbname

# Replace with your folder name
folder_name = 'carbon_footprint_calculator'
zip_name = 'carbon_footprint_calculator.zip'


# Create the folder if it doesn't exist
if not os.path.exists(folder_name):
    os.makedirs(folder_name)
    print(f"Created folder: {folder_name}")

# Get the current notebook name
try:
    notebook_path = ipynbname.path()
    notebook_name = notebook_path.name
    # Move the notebook into the folder
    shutil.move(notebook_path, os.path.join(folder_name, notebook_name))
    print(f"Moved notebook '{notebook_name}' into '{folder_name}'")
except FileNotFoundError:
    print("Could not find the notebook file. Please save your notebook and try ag
    notebook_name = None # Set notebook_name to None if file not found

# Create a ZIP file if the notebook was moved successfully
if notebook_name:
    shutil.make_archive(zip_name.replace('.zip',''), 'zip', folder_name)
    print(f"Created {zip_name} successfully!")
else:
    print("Skipping zip creation as notebook file was not found.")
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 27.9 MB/s eta
Created folder: carbon_footprint_calculator
Could not find the notebook file. Please save your notebook and try again.
Skipping zip creation as notebook file was not found.
```