CSprojectteam11convertor

March 8, 2025

Group 11 Project

Members : 1. Shadi Sabzali 2. Kapugedara Harshani Nirmala Kumari Abesinghe 3. Rivini Chethya Yakupitiyage 4. Samudra Sandamali Bandara Herath Mudiyanselage 5. Shanki Chaminda Sumanarathna Aaryachakra Bandaranayaka Thewanga Mudiyanselage 6. Zahra Sadeghi Jalalabadi

Convertor

Create a program that can be used to converttemperature, length, weight, pressure.

Your program should have a menu displayed for the user to choose from, where are listed the convertion options:

-temperature fahrenheit to celsius celsius to fahrenheit

-length miles to km km to miles

-weight pound to kilogramms kg to poundexit

The program should allow the user to choose the desired convertion over and over again until user chooses to quit using it.

UI packages importing section:

[17]:

```
import ipywidgets as widgets
from IPython.display import display, clear_output
```

functions

[19]:

```
# Temperature conversion functions
def fahrenheit_to_celsius(f):
    return (f - 32) * 5/9

def celsius_to_fahrenheit(c):
    return (c * 9/5) + 32

# Length conversion functions
def miles_to_km(miles):
    return miles * 1.60934

def km_to_miles(km):
    return km / 1.60934
```

```
# Weight conversion functions
def pounds_to_kg(pounds):
    return pounds * 0.453592

def kg_to_pounds(kg):
    return kg / 0.453592

# Pressure conversion functions
def pascal_to_bar(pa):
    return pa / 100000

def bar_to_pascal(bar):
    return bar * 100000
```

UI

[21]:

```
# UI elements
conversion_type = widgets.Dropdown(
    options=['Choose','Temperature', 'Length', 'Weight', 'Pressure'],
    description='Conversion Type:',
)

input_value = widgets.FloatText(
    description='Input Value:',
    value=0
)

input_unit = widgets.Dropdown(
    options=[],
    description='Input Unit:',
)

output_value = widgets.FloatText(
    description='Output Value:',
    value=0,
    disabled=True
)
```

Convertor with UI interaction

[23]:

```
#check the conversion updates
def update_units(change):
    if conversion_type.value == 'Temperature':
        input_unit.options = ['Fahrenheit', 'Celsius']
    elif conversion_type.value == 'Length':
        input_unit.options = ['Miles', 'Kilometers']
    elif conversion_type.value == 'Weight':
        input_unit.options = ['Pounds', 'Kilograms']
```

```python
    elif conversion_type.value == 'Pressure':
        input_unit.options = ['Pascal', 'Bar']

#Select Conversion formula by changing units
def convert(change):
    if conversion_type.value == 'Temperature':
        if input_unit.value == 'Fahrenheit':
            output_value.value = fahrenheit_to_celsius(input_value.value)
        else:
            output_value.value = celsius_to_fahrenheit(input_value.value)
    elif conversion_type.value == 'Length':
        if input_unit.value == 'Miles':
            output_value.value = miles_to_km(input_value.value)
        else:
            output_value.value = km_to_miles(input_value.value)
    elif conversion_type.value == 'Weight':
        if input_unit.value == 'Pounds':
            output_value.value = pounds_to_kg(input_value.value)
        else:
            output_value.value = kg_to_pounds(input_value.value)
    elif conversion_type.value == 'Pressure':
        if input_unit.value == 'Pascal':
            output_value.value = pascal_to_bar(input_value.value)
        else:
            output_value.value = bar_to_pascal(input_value.value)

# Button to trigger conversion
convert_button = widgets.Button(
    description='Convert'
)

# Display the output
output_area = widgets.Output()

# Link functions to events
conversion_type.observe(update_units, names='value')
convert_button.on_click(convert)

# Display the UI components
display(conversion_type, input_unit, input_value, convert_button, output_value, output_area)
```

```
Dropdown(description='Conversion Type:', options=('Choose', 'Temperature', 'Length', 'Weight', 'Pressure'), va…

Dropdown(description='Input Unit:', options=(), value=None)

FloatText(value=0.0, description='Input Value:')
```

```
Button(description='Convert', style=ButtonStyle())

FloatText(value=0.0, description='Output Value:', disabled=True)

Output()
```

[ ]: