

# Конвертер единиц измерения

## Описание проекта

Проект реализует конвертер единиц измерения на языке программирования Python. С его помощью можно преобразовывать значения между различными единицами длины, веса и температуры. Конвертер предлагает простой интерфейс для выбора типа преобразования и вывода результата.

## Теория по переводу единиц измерения

### 1. Перевод единиц длины

Длина – это физическая величина, характеризующая протяженность объектов или расстояние между ними. Основной единицей длины в международной системе единиц (СИ) является **метр (м)**. Преобразование длины осуществляется по следующим формулам (прямые и обратные преобразования):

**Метр → километры**

$$\text{kilometers} = \text{meters} \times 0.001$$

**Километры → метры**

$$\text{meters} = \text{kilometers} \div 0.001$$

**Метр → мили**

$$\text{miles} = \text{meters} \times 0.000621371$$

**Мили → метры**

$$\text{meters} = \text{miles} \div 0.000621371$$

### 2. Перевод единиц массы (веса)

Масса – это физическая величина, измеряющая количество вещества в теле. Основной единицей массы в СИ является **килограмм (кг)**. Перевод значений массы осуществляется по следующим формулам:

**Килограммы → граммы**

$$\text{grams} = \text{kilograms} \times 1000$$

**Граммы → килограммы**

$$\text{kilograms} = \text{grams} \div 1000$$

**Килограммы → фунты**

$$\text{pounds} = \text{kilograms} \times 2.20462$$

**Фунты → килограммы**

$$\text{kilograms} = \text{pounds} \div 2.20462$$

### 3. Перевод единиц температуры

Температура – это физическая величина, измеряющая степень нагретости или охлажденности объекта. Основные шкалы температуры: **Цельсий (°C)**, **Кельвин (K)** и **Фаренгейт (°F)**. Преобразование температуры осуществляется по следующим формулам:

**Цельсий → Фаренгейт**

$$\text{Fahrenheit} = \text{Celsius} \times \frac{9}{5} + 32$$

**Фаренгейт → Цельсий**

$$\text{Celsius} = (\text{Fahrenheit} - 32) \times \frac{5}{9}$$

**Цельсий → Кельвин**

$$\text{Kelvin} = \text{Celsius} + 273.15$$

**Кельвин → Цельсий**

$$\text{Celsius} = \text{Kelvin} - 273.15$$

Фаренгейт → Кельвин

$$\text{Kelvin} = (\text{Fahrenheit} - 32) \times \frac{5}{9} + 273.15$$

Кельвин → Фаренгейт

$$\text{Fahrenheit} = (\text{Kelvin} - 273.15) \times \frac{9}{5} + 32$$

## Структура проекта

- `converter.py` — основной файл программы, который предоставляет интерфейс пользователя для выбора типа конверсии и ввода значений.
- `conversion_utils.py` — модуль, содержит функции для преобразования различных единицами измерения: длины, массы и температуры.

## Основные функции преобразования

### Функция `convert_length()`

Функция принимает значение `value` и преобразует его из одной единицы измерения длины (`from_unit`) в другую (`to_unit`). Внутри функции используется словарь с коэффициентами преобразования относительно метра.

Мы переводим любые введенные единицы в метры (далее в килограммы) для упрощения преобразований, т.к. все единицы сводятся к **базовой (метрам или килограммам)**. Добавление новых единиц упрощается — нужен только **коэффициент к базовой единице**.

```
def convert_length(value: float, from_unit: str, to_unit: str) -> float:
    """
    Преобразует значение длины из одной единицы в другую.

    Параметры:
    value (float): значение для преобразования.
    from_unit (str): исходная единица (например, "meters", "kilometers", "miles").
    to_unit (str): целевая единица (например, "meters", "kilometers", "miles").

    Возвращает:
    float: значение, преобразованное в целевую единицу.
    """

    # Словарь с коэффициентами преобразования относительно метра
    length_units = {
        "meters": 1,
        "kilometers": 0.001,
        "miles": 0.000621371
    }

    if from_unit not in length_units or to_unit not in length_units:
        raise ValueError("Invalid units for length conversion")

    # Преобразование из исходной единицы в метры
    value_in_meters = value / length_units[from_unit]
    # Преобразование из метров в целевую единицу
    converted_value = value_in_meters * length_units[to_unit]
    return converted_value
```

### Функция `convert_weight()`

Функция преобразует значение веса из одной единицы (`from_unit`) в другую (`to_unit`). Используется словарь с коэффициентами преобразования относительно килограмма.

```
def convert_weight(value: float, from_unit: str, to_unit: str) -> float:
    # Словарь с коэффициентами преобразования относительно килограмма
    weight_units = {
        "grams": 1000,
        "kilograms": 1,
        "pounds": 2.20462
    }

    if from_unit not in weight_units or to_unit not in weight_units:
        raise ValueError("Invalid units for weight conversion")

    # Преобразование из исходной единицы в килограммы
    value_in_kilograms = value / weight_units[from_unit]
    # Преобразование из килограммов в целевую единицу
    converted_value = value_in_kilograms * weight_units[to_unit]
    return converted_value
```

## Функция convert\_temperature()

Функция преобразует значение температуры из одной шкалы (from\_unit) в другую (to\_unit). Для преобразования используются соответствующие формулы перехода между шкалами Цельсия, Фаренгейта и Кельвина.

```
def convert_temperature(value: float, from_unit: str, to_unit: str) -> float:
    """
    Возвращает:
    float: значение температуры, преобразованное в целевую единицу.
    """
    if from_unit == "celsius" and to_unit == "fahrenheit":
        return (value * 9/5) + 32
    elif from_unit == "celsius" and to_unit == "kelvin":
        return value + 273.15
    elif from_unit == "fahrenheit" and to_unit == "celsius":
        return (value - 32) * 5/9
    elif from_unit == "fahrenheit" and to_unit == "kelvin":
        return (value - 32) * 5/9 + 273.15
    elif from_unit == "kelvin" and to_unit == "celsius":
        return value - 273.15
    elif from_unit == "kelvin" and to_unit == "fahrenheit":
        return (value - 273.15) * 9/5 + 32
    else:
        raise ValueError("Invalid units for temperature conversion")
```

## Основной файл программы

**Описание:** converter.py — это основной файл программы, который предоставляет пользователю интерфейс для выбора типа преобразования, ввода значения и выполнения конверсии. Он взаимодействует с функциями из conversion\_utils.py для выполнения самих преобразований.

```
from conversion_utils import convert_length, convert_weight, convert_temperature
```

## Функция main()

**Описание:** Функция main() управляет основным взаимодействием с пользователем. Она выводит доступные опции для преобразования, принимает выбор пользователя и вызывает обработку преобразования.

```
def main():
    # Словарь с параметрами для каждой конверсии: тип, метка и функция преобразования
    conversion_options = {
        "1": ("length", "Длина", convert_length),
        "2": ("weight", "Вес", convert_weight),
        "3": ("temperature", "Температура", convert_temperature)
    }
```

**Пояснение:** Словарь conversion\_options содержит информацию о возможных типах преобразования. Ключи (1, 2, 3) соответствуют доступным опциям, а значения кортежей включают тип преобразования (например, "length"), текстовую метку ("Длина") и функцию преобразования (например, convert\_length).

```
while True:
    print("\nДобро пожаловать в конвертер единиц измерения!")
    # Печать доступных опций для пользователя
    for key, (conv_type, label, _) in conversion_options.items():
        print(f"{key}. {label}")
    print("4. Выход")
```

**Пояснение:** Программа приветствует пользователя и выводит доступные опции для преобразования (например, длина, вес, температура). Пользователь делает выбор, вводя число от 1 до 4.

```
choice = input("Выберите тип конвертации (1-4): ")

if choice in conversion_options:
    # Получение параметров для выбранной опции
    conv_type, _, conversion_func = conversion_options[choice]
    # Вызов обработчика конверсии
    handle_conversion(conv_type, conversion_func)
elif choice == "4":
    print("Выход из программы.")
    break
else:
    print("Неверный выбор, попробуйте снова.")
```

**Пояснение:** - Если пользователь выбирает значение 1, 2 или 3, извлекаются соответствующие параметры (тип конверсии и функция) и вызывается функция `handle_conversion()`. - Если пользователь выбирает "4" программа завершает работу. - В случае некорректного выбора выводится сообщение об ошибке.

## Функция `handle_conversion()`

**Описание:** Функция `handle_conversion()` выполняет преобразование значений в соответствии с выбранным пользователем типом преобразования. Она принимает тип конверсии (`conversion_type`) и функцию преобразования (`conversion_func`), которую нужно вызвать.

```
def handle_conversion(conversion_type, conversion_func):
    # Словарь с доступными единицами для каждого типа конверсии
    units_dict = {
        "length": {
            "1": "meters",
            "2": "kilometers",
            "3": "miles"
        },
        "weight": {
            "1": "grams",
            "2": "kilograms",
            "3": "pounds"
        },
        "temperature": {
            "1": "celsius",
            "2": "fahrenheit",
            "3": "kelvin"
        }
    }
```

**Пояснение:** Словарь `units_dict` содержит информацию о доступных единицах измерения для каждого типа конверсии. Например, для длины доступны метры, километры и мили.

```
# Получение словаря единиц для выбранного типа конверсии
units = units_dict[conversion_type]
# Ввод значения для преобразования
value = float(input("\nВведите значение: "))
print("Доступные единицы:")
# Печать доступных единиц
for key, unit in units.items():
    print(f"{key}. {unit}")
```

**Пояснение:** - Программа получает доступные единицы для выбранного типа преобразования. - Пользователь вводит значение для преобразования. - Далее программа отображает доступные единицы, чтобы пользователь мог выбрать исходную и целевую.

```
# Ввод исходной и целевой единиц с проверкой по ключам
from_unit_key = input("Введите номер исходной единицы: ").strip()
to_unit_key = input("Введите номер целевой единицы: ").strip()

if from_unit_key in units and to_unit_key in units:
    from_unit = units[from_unit_key]
    to_unit = units[to_unit_key]
    try:
        # Вызов функции преобразования
        result = conversion_func(value, from_unit, to_unit)
        print(f"{value} {from_unit} равно {result} {to_unit}.")
    except ValueError as e:
        # Обработка ошибки, если выбранные единицы недействительны
        print(e)
else:
    # Вывод сообщения об ошибке при некорректном выборе
    print("Неверный выбор единицы.")
```

**Пояснение:** - Пользователь выбирает исходную и целевую единицы. - Если выбор корректен, вызывается переданная функция преобразования `conversion_func` и выполняется преобразование. - Если произошла ошибка, например, недопустимые единицы, программа выводит сообщение об ошибке.

## Пример работы программы

Пример работы программы показывает, как пользователь взаимодействует с программой для выполнения преобразования значений. Рассмотрим пример, где пользователь выбирает преобразование длины:

1. Программа приветствует пользователя и предлагает выбрать тип преобразования:

```
Добро пожаловать в конвертер единиц измерения!  
1. Длина  
2. Вес  
3. Температура  
4. Выход  
Выберите тип конвертации (1-4):
```

2. Пользователь вводит 1, чтобы выбрать преобразование длины.
3. Программа запрашивает значение для преобразования и отображает доступные единицы:

```
Введите значение: 100  
Доступные единицы:  
1. meters  
2. kilometers  
3. miles  
Введите номер исходной единицы:
```

4. Пользователь вводит 1 для выбора метров и затем 2 для преобразования в километры.
5. Программа выполняет преобразование и выводит результат:

```
100 meters равно 0.1 kilometers.
```

Аналогично, пользователь может выбрать преобразование массы или температуры, следуя подсказкам программы.