

# LU09.L03 - Erweiterte Aufgaben

## Mehrere Return-Werte

```
import math

def div(dividend, divisor):
    """
    Calculates the result and the rest of an
    integer division
    :param dividend: Dividend of the division
    :param divisor: Divisor of the division
    :return: Result of the division as whole
    number and the rest of the division
    """

    result = math.floor(dividend / divisor)
    rest = dividend % divisor

    return result, rest

def main():
    result, rest = div(34, 6)
    print(f'Result: {result} and Rest:
    {rest}')

if __name__ == '__main__':
    main()
```

## 2. Längenumrechner

### 1.Code anpassen

```
def convert(length, from_unit = 1, to_unit =
2):
    """
    Converts lengths. The following lengths
    can be converted: meters, miles, nautical
    miles, yards, inches
```

```
    :length: length to convert
    :from_unit: 1=Meter, 2=Meilen,
3=Seemeilen, 4=Yard, 5=Inches
    :to_unit: 1=Meter, 2=Meilen, 3=Seemeilen,
4=Yard, 5=Inches
    :return: converted length
    """
    if (from_unit == 1):
        result = length
    elif (from_unit == 2):
        result = length * 1609.34
    elif (from_unit == 3):
        result = length * 1852.0
    elif (from_unit == 4):
        result = length * 0.9144
    elif (from_unit == 5):
        result = length * 0.0254

    if (to_unit == 1):
        return result
    if (to_unit == 2):
        result = result / 1609.34
        return result
    if (to_unit == 3):
        result = result / 1852.0
        return result
    if (to_unit == 4):
        result = result / 0.9144
        return result
    if (to_unit == 5):
        result = result / 0.0254
        return result

def main():
    print("1=Meter, 2=Meilen, 3=Seemeilen,
4=Yard, 5=Inches")
    print(convert(1000))

if __name__ == '__main__':
    main()
```

## 2. Erweiterung

```
def convert(length, from_unit = 1, to_unit =
```

```
2):
    """
    Converts lengths. The following lengths
    can be converted: meters, miles, nautical
    miles, yards, inches
    :length: length to convert
    :from_unit: 1=Meter, 2=Meilen,
    3=Seemeilen, 4=Yard, 5=Inches
    :to_unit: 1=Meter, 2=Meilen, 3=Seemeilen,
    4=Yard, 5=Inches
    :return: converted length
    """

    if (from_unit == 1 or
str(from_unit).casefold() ==
'Meter'.casefold()):
        result = length
    elif (from_unit == 2 or
str(from_unit).casefold() ==
'Meilen'.casefold()):
        result = length * 1609.34
    elif (from_unit == 3 or
str(from_unit).casefold() ==
'Seemeilen'.casefold()):
        result = length * 1852.0
    elif (from_unit == 4 or
str(from_unit).casefold() ==
'Yard'.casefold()):
        result = length * 0.9144
    elif (from_unit == 5 or
str(from_unit).casefold() ==
'Inches'.casefold()):
        result = length * 0.0254

    if (to_unit == 1 or
str(to_unit).casefold() ==
'Meter'.casefold()):
        return result
    if (to_unit == 2 or
str(to_unit).casefold() ==
'Meilen'.casefold()):
        result = result / 1609.34
        return result
    if (to_unit == 3 or
str(to_unit).casefold() ==
'Seemeilen'.casefold()):
        result = result / 1852.0
        return result
    if (to_unit == 4 or
str(to_unit).casefold() ==
'Yard'.casefold()):
        result = result / 0.9144
```

```
        return result
    if (to_unit == 5 or
str(to_unit).casefold() ==
'Inches'.casefold()):
        result = result / 0.0254
    return result

def main():
    print('1=Meter, 2=Meilen, 3=Seemeilen,
4=Yard, 5=Inches')
    print(convert(1000))
    print(convert(1000, from_unit='Meilen',
to_unit=5))

if __name__ == '__main__':
    main()
```

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/archiv/m319python/learningunits/lu09/loesungen/erweitert>

Last update: **2024/03/28 14:07**

