

LU09.A07 - Bremsweg

Aufgabe

Berechnen Sie die Anhaltestrecke eines Autos bei gegebener Geschwindigkeit [km/h] für trockene bzw. nasse Straße. In eckigen Klammern stehen die Maßeinheiten, die jedoch für die Programmierung nicht von Bedeutung sind.

Sie brauchen folgende Informationen um diese Aufgabe zu lösen:

- Die Geschwindigkeit [km/h] kann vom Anwender eingegeben werden.
- Geschwindigkeit [m/s] = Geschwindigkeit [km/h] / 3.6 (z. B. 72 [km/h] = 20 [m/s])
- Die Bremsbeschleunigung (auch Bremsverzögerung genannt) wird in Metern pro Quadratsekunden gemessen. Trockene Straße: Bremsbeschleunigung = 7 [m/s²]. Nasse Straße: Bremsbeschleunigung = 4 [m/s²]. Wer das Programm benutzt, kann dabei «t» für die trockene bzw. «n» für die nasse Straße eingeben.
- Reaktionszeit [s] = 1.44
- Reaktionsweg [m] = Geschwindigkeit [m/s] * Reaktionszeit [s]
- Bremsweg [m] = Geschwindigkeit²[(m/s)²] / (2 * Bremsbeschleunigung [m/s²])
- Anhaltestrecke [m] = Reaktionsweg [m] + Bremsweg [m]

Programmieren Sie möglichst viele der benötigten Größen als Funktionen mit sinnvollen Eingabeparametern. Beispiele:

braking_distance() reaction_distance() stopping_distance() in_meters_per_second()

Zusatzaufgabe (Auffahrunfall):

Wie viel Abstand (in Metern und in Sekunden) zum vor ihm fahrenden Wagen muss ein Fahrer einhalten, um einen Auffahrunfall zu vermeiden, falls der vor ihm Fahrende eine Vollbremsung einleitet?

Dazu ist zusätzlich die Bremszeit zu berechnen:

- Bremszeit [s] = Geschwindigkeit [m/s] / Bremsbeschleunigung [m/s²]

Vorlage

```
#TODO create multiple functions to solve the  
braking distance calculation  
  
def main():  
    #TODO: Call the function to calculate the  
    braking distance for 50, 80 and 120 km/h
```

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

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

Permanent link:
<https://wiki.bzz.ch/modul/archiv/m319python/learningunits/lu09/aufgaben/bremsweg>

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

