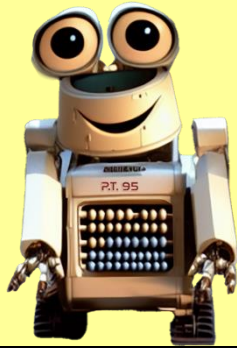


# FUNKTIONEN MIT RÜCKGABEWERT



## PT95 erklärt: Was sind Rückgabewerte?

Bisher haben wir Funktionen kennen gelernt, die etwas für uns tun, aber nichts zurückgeben. Manchmal ist es aber sinnvoll, dass eine Funktion uns ein Ergebnis liefert. Mit diesem Ergebnis können wir dann weiterarbeiten. So etwas nennt man einen *Rückgabewert*.

Du kannst dir das so vorstellen: Eine Funktion ist ein kleines Programm, dem du eine Frage stellen kannst. Und die Funktion beantwortet dir diese Frage. Das heißt, sie liefert dir einen Rückgabewert!

## Funktionen mit Rückgabewert: Wie geht das?

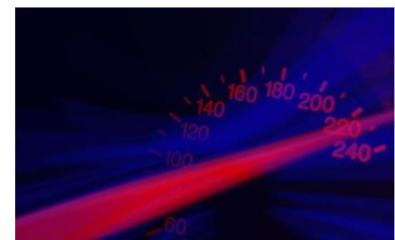
Wir schauen uns das mal in einem Beispiel an: Die folgende Funktion soll berechnen, wie lang der Bremsweg eines Autos ist. Die Formel hierfür lautet:

$$\text{Bremsweg} = (\text{Geschwindigkeit} / 10)^2$$

Weil man aber auch noch den *Reaktionsweg* berechnen muss (also die Strecke, die ein Mensch im Auto zurücklegt, bis er reagiert), wollen wir nur das Ergebnis wissen und noch nicht ausgeben!

Das kann man so mit einer Funktion mit Rückgabewert lösen:

## Berechnung des Anhaltewegs



Geschwindigkeit:

```
<?php
```

```
$geschwindigkeit = $_POST["txtGeschwindigkeit"];
```

```
function bremsweg($geschwindigkeit)
```

```
{
```

```
$bremsweg = ($geschwindigkeit / 10) * ($geschwindigkeit / 10);
```

```
return $bremsweg;
```

```
}
```

```
echo bremsweg($geschwindigkeit);
```

```
?>
```

Hier rufen wir die Funktion *bremsweg* auf!



1. Probieren Sie den obigen Code aus! Erstellen Sie dafür die PHP-Seite *anhalteweg-auswertung.php* und rufen Sie diese Seite über einen Browser auf.

Die HTML-Seite mit dem Formular wurde bereits erstellt: Verwenden Sie hier die Datei *anhalteweg.html*.

Erweitern Sie die Ausgabe dann so, dass diese so aussieht:

**Bremsweg: 25 Meter**

*Im Beispiel wurde eine Geschwindigkeit von 50 km/h eingegeben.*

2. Erweitern Sie die PHP-Seite so:

- Zusätzlich soll der **Reaktionsweg** ausgegeben werden. Das ist die Strecke, die eine durchschnittlicher Fahrer zurücklegt, bis er reagieren kann:

$$\text{Reaktionsweg} = \text{geschwindigkeit} / 10 * 3$$

- Erstellen Sie hierzu auf der PHP-Seite eine neue **Funktion mit Rückgabeparameter** - Name der Funktion: *reaktionsweg()*

Bremsweg: 25 Meter  
 Reaktionsweg: 15 Meter

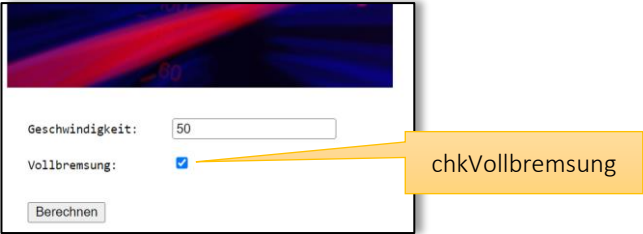
3. Der gesamte Anhalteweg berechnet sich so:

$$\text{Anhalteweg} = \text{Reaktionsweg} + \text{Bremsweg}$$

Erweitern Sie die PHP-Seite, indem Sie mit den bereits erstellten Funktionen eine solche Ausgabe realisieren:

Bremsweg: 25 Meter  
 Reaktionsweg: 15 Meter  
  
 Anhalteweg: 40 Meter

4. Bei einer **Vollbremsung** halbiert sich der Bremsweg. Ändern Sie unser Programm so, dass dieses berücksichtigt wird. Ein verändertes HTML-Formular wurde bereits erstellt (anhalteweg2.html):



Gehen Sie so vor:

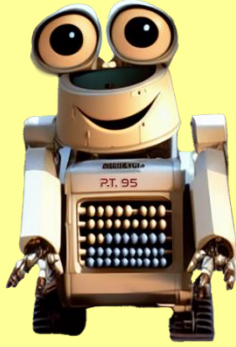
- Erstellen Sie eine Funktion *bremswegVollbremsung()* mit Rückgabewert!
- Ob die Checkbox angeklickt ist, können Sie so auswerten:

```

if(isset($_POST["chkVollbremsung"]))
{
    // Mach was!
}
else
{
    // Mach was anderes!
}
  
```

Ausgabe:

Bremsweg: 12.5 Meter  
 Reaktionsweg: 15 Meter  
  
 Der gesamte Anhalteweg beträgt: 27.5 Meter



## PT95 erklärt: Bringt uns das was?

Funktionen mit Rückgabewert sind wie kleine Helfer, die dir etwas Nützliches zurückgeben, nachdem sie eine Aufgabe erledigt haben. Hier sind einige Gründe, warum man sie verwendet:

- Wiederverwendbarkeit: Funktionen erledigen eine bestimmte Aufgabe, und du kannst sie immer wieder verwenden, ohne den gleichen Code wiederholt schreiben zu müssen.
- Strukturierung des Codes: Funktionen helfen, den Programmcode ordentlich und leicht verständlich zu gestalten, indem sie komplexe Aufgaben in kleinere, leichter verdauliche Teile aufteilen.

## Aufgaben



5. Erstellen Sie eine **Funktion mit Rückgabewert** (Name: *kreisUmfang*), die den Umfang eines Kreises berechnet. Als Eingangsparameter soll die Funktion den Radius des Kreises erhalten.
- Das Formular zur Eingabe des Radius wurde bereits erstellt. Sie finden dieses in der Datei *radius.html*.
- Kreisumfang**

Radius:
- Geben Sie das Ergebnis der Berechnung aus, indem Sie die Funktion *kreisUmfang* aufrufen!  
(Hinweis: Auf die Zahl *pi* können Sie in PHP mit der Funktion *pi()* zugreifen!)
6. Entwerfen Sie eine Funktion, die aus einem gegebenen Verkaufspreis den Nettopreis berechnet (also den Preis ohne MwSt). Verwenden Sie als Namen der Funktion *nettopreisBerechnen()*. Sowohl der Bruttopreis als auch der Steuersatz sollen der Funktion als Parameter übergeben werden.
- Die Funktion soll den berechneten Preis als Parameter zurückgeben!
- Nettopreis berechnen**

Bruttopreis:

Steuersatz:
7. Ihre Aufgabe ist es, eine Funktion zu erstellen (Name: *heizkostenBerechnen*), die berechnet, wie hoch die Heizkosten sind, wenn die Raumtemperatur abgesenkt wird. Pro Grad Temperaturabsenkung können etwa 6% Heizkosten gespart werden.
- Eingabeparameter der Funktion:  
Aktuelle Heizkosten, Prozentuale Temperaturabsenkung
- Ausgabeparamater:  
Berechnete Heizkosten
- Heizkosten: Temperaturabsenkung**

Bisherige Heizkosten:

Temperaturabsenkung: