

Programmcode

Variante "short"

```
#include <desklab.h>
```

1

```
double OptischeDichte;
```

2

```
void setup() {
```

3

```
    StarteDisplay();
```

4

```
}
```

```
void loop() {
```

5

```
    OptischeDichte = PhotometerMessung(A0);
```

6

```
    PhotometerAusgabe(OptischeDichte);
```

7

```
    delay(100);
```

8

```
}
```

Hintergrundinfo: Ein `double` speichert sog. Flieskommazahlen, also Zahlen die mehrere Nachkommastellen haben können.

Aufgabe: Welche Datentypen kennst du noch?

Aufgabe: Formuliere nun in eigenen Worten, was dieses Programm macht

 **Unterrichtsreihe**

 **Version S-1**

Programmcode

Variante "short"

```
#include <desklab.h>
```

1 Einbinden einer Bibliothek mit Funktionen für das Photometer von desklab (www.desk-lab.de)

```
double OptischeDichte;
```

2 Initialisierung einer Variable vom Typ **double** zum Speichern der optischen Dichte

```
void setup() {
```

3 Beginn der **setup**-Funktion (wird einmal beim Start ausgeführt)

```
  StarteDisplay();
```

4 Starte das Display (Anzeige "Starte Messung!")

```
}
```

```
void loop() {
```

5 Beginn der **loop**-Funktion (wird nach der **setup**-Funktion in einer Dauerschleife ausgeführt)

```
  OptischeDichte = PhotometerMessung(A0);
```

6 Liest den Sensor an A0 aus, berechnet aus dem Sensorwert die optische Dichte und speichert ihn in **OptischeDichte**


```
  PhotometerAusgabe(OptischeDichte);
```


7 Ausgabe des aktuellen Werts der Variable **OptischeDichte** auf dem Display


```
  delay(100);
```


8 Warte für 100 Millisekunden

```
}
```

 **Mehr Informationen:**
Siehe Photometer-Bibliothek

 **Hintergrundinfo:** Ein **double** speichert sog. Flieskommazahlen, also Zahlen die mehrere Nachkommastellen haben können.

 **Aufgabe:** Welche Datentypen kennst du noch?

 **Aufgabe:** Formuliere nun in eigenen Worten, was dieses Programm macht

 **Unterrichtsreihe**

 **Version L-1**