

Wiederholung

Definition:

```
void machWas() {  
    // Anweisungen  
    // ...  
}
```

Aufruf:

```
machWas();
```

Definition:

```
void machMehr(int p1, double p2) {  
    // Anweisungen  
    // ...  
}
```

Aufruf:

```
machMehr(5, 23);
```

Schleifen Teil 1

- Schreibe ein Programm, das 1000 Kreise zeichnet

Warum ist diese Aufgabe aufwändig?

Weil wir “von Hand” 1000 Mal die Methode `circle()` aufrufen müssten.
Das geht besser!

- Schleifen dienen dazu, bestimmte Anweisungen wiederholen zu lassen
- Die Anzahl der Wiederholungen kann festgelegt werden
 - Bestimmte (feste) Anzahl (z. B. 1000 Mal)
 - Abhängig von einer Bedingung (z. B. solange der Benutzer eine Taste drückt)

Die for-Schleife (Zählschleife)

Beispiel:

```
for ( int x = 0; x < 100; x = x+1 ) {  
    // Anweisungen  
    // ...  
}
```

for Schlüsselwort, das den Beginn einer for-Schleife anzeigt.

bzw. } Beginn und Ende des sogenannten Schleifenrumpfs, also der Anweisungen, die wiederholt werden sollen.

int x = 0 Initialisierung, wird **vor** Ausführung der Schleife **ein Mal** ausgeführt. Erzeugt eine Zählvariable mit dem Namen x und dem Anfangswert 0.

x < 100 Bedingung. Legt fest, dass der Schleifenrumpf so lange wiederholt werden soll, wie diese Bedingung gültig ist. Wird **vor** jedem Durchlauf des Schleifenrumpfs geprüft.

x = x+1 "Fortsetzung", wird **nach jeder** Wiederholung des Schleifenrumpfs ausgeführt. Erhöht den Wert von x jeweils um 1.

- Name, Anfangs- und Endwert der Zählvariable können frei gewählt werden.
- Die Schrittweite muss nicht zwingend +1 sein.
- Statt `x = x+1` kann man auch `x++` oder `++x` schreiben.

Aufgabe 1

Welche Werte werden jeweils ausgegeben?

// Beispiel 1

```
for ( int x = 0; x < 5; x++ ) {  
    System.out.println(x);  
}
```

// Beispiel 2

```
for ( int x = 0; x <= 6; x = x+2 ) {  
    System.out.println(x);  
}
```

// Beispiel 3

```
for ( int x = 100; x > 0; x-- ) {  
    System.out.println(x);  
}
```

// Beispiel 4

```
for ( int x = 0; x > 10; x = x+1 ) {  
    System.out.println(x);  
}
```

Bemerkungen:

- `System.out.println()` gibt Text auf der Konsole aus.
- Die Zählvariable kann innerhalb des Schleifenrumpfs verwendet werden (z. B., um sie auszugeben).

- Schreibe ein Programm, das mittels einer for-Schleife 20 überlappende Kreise in einer Reihe zeichnet
- Erweitere das Programm, dass mehrere Reihen aus je 20 Kreisen untereinander gezeichnet werden

Variablen

- Die Zählvariable der for-Schleife ist nur ein Beispiel für den Einsatz von Variablen
- Allgemein versteht man unter einer Variable einen Speicherplatz für einen Wert von einem bestimmten Datentyp (ganze Zahl, Kommazahl usw.)
- Der Wert einer Variable kann sich im Laufe der Programmausführung ändern

int Ganze Zahlen

float Kommazahlen, einfache Genauigkeit

double Kommazahlen, doppelte Genauigkeit

boolean Wahrheitswerte (true oder false)

char Einzelne Zeichen (Buchstaben usw.)

String Zeichenketten

```
int x;           // Deklaration ohne Initialisierung
x = -5;          // Wertzuweisung
int y = 10;      // Deklaration und Initialisierung
y = y * 2;       // Wertzuweisung
int z = y;       // Deklaration und Initialisierung
z++;            // Wertzuweisung
```

```
float a = 1.23f; // Deklaration und Initialisierung (1,23f => float)
double b = a / 0.5; // Deklaration und Initialisierung
```

```
boolean w = true; // Deklaration und Initialisierung
boolean f = false; // Deklaration und Initialisierung
w = false;        // Wertzuweisung
```

```
char firstLetter = 'A'; // Deklaration und Initialisierung
char lastLetter = 'Z';  // Deklaration und Initialisierung
```

Deklaration Festlegung/“Bekanntmachung” von Name und Datentyp der Variable

Initialisierung Erste Wertzuweisung an die Variable. Kann, muss aber nicht zusammen mit der Deklaration erfolgen.

Wertzuweisung Zuweisung eines neuen Werts an eine schon vorhandene Variable.

Gültigkeitsbereich “Lebensdauer” einer Variable (dazu später mehr)

Aufgabe 3

- Öffne das Programm “Variablen” mit Processing
- Ergänze innerhalb von `setup()` einige weitere Berechnungen und lasse Dir die Ergebnisse mit `System.out.println()` ausgeben
 - Verwende die Rechenoperationen `+`, `-`, `*`, `/`
 - Verwende Variablen mit unterschiedlichen Datentypen
 - Mische in den Rechnungen Variablen vom Typ `double` und `int`
- Versuche, Dir den Wert der Variable `zaehl` nach der for-Schleife ausgeben zu lassen. Was stellst Du fest?

- Eine Variable existiert nur innerhalb des Blocks, in dem sie deklariert wurde sowie in den darin eingeschlossenen Blöcken

```
void setup() {  
  int x = 0;  
  for ( ; ; ) {  
    x = 25; // geht  
  }  
  x = 30; // geht  
}
```

```
void setup() {  
  for ( ; ; ) {  
    int x = 0;  
    x = 25; // geht  
  }  
  x = 30; // geht nicht  
}
```

```
int x = 0;  
void setup() {  
  for ( ; ; ) {  
    x = 25; // geht  
  }  
  x = 30; // geht  
}
```

```
void bla() {  
  x = 10; // geht nicht  
}
```

```
void bla() {  
  x = 10; // geht nicht  
}
```

```
void bla() {  
  x = 10; // geht  
}
```