

Jetzt weißt du schon, wie man einfache Zeichnungen erstellt und mit Farben arbeitet. Ziel dieses Arbeitsblattes ist es, dass du mit Variablen arbeiten kannst.

Gut zu wissen

Im Internet findet man viele Lösungen zu den Aufgaben. Aber ein Programm einfach nur zu kopieren, ohne zu verstehen, wie das Programm die Aufgabe löst, ist nicht sehr sinnvoll. Versuche deswegen die Aufgaben selbst zu lösen.

Aufgabe 1. p5js-Cheat sheet.

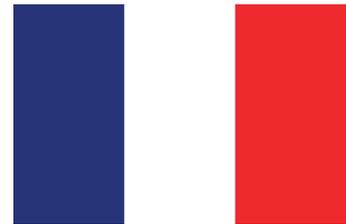
Beim Lernen einer Programmiersprache ist es hilfreich, ein Cheat sheet selbst zu erstellen, sodass man nicht immer in die Referenz sehen muss. Ein einfaches Cheat sheet mit einigen Eintragungen findest du hier: <http://l.phina.be/p5cheat>

Lade es herunter und erweitere es um die Befehle aus dem Arbeitsblatt 1. Führe es immer weiter.

Aufgabe 2. Vive la France!

Die französische Flagge besteht aus drei Rechtecken mit den Farben:

Farbe	Code
blau	(39, 53, 120)
weiß	(255)
rot	(238,42,46)



Schreibe ein Programm, das diese Flagge zeichnet. Die Rechtecke sind doppelt so hoch wie breit. Damit die Rechtecke keine Linien haben, kannst du den Befehl `strokeWeight()` benutzen (siehe Code 2.1.1, dieser Code zeichnet einen roten Kreis ohne Umrandung).

Code 2.1.1

```
function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(255);
  strokeWeight(0);
  fill(255,0,0);
  circle(200, 200, 100); //
}
```

Kopiere deinen Code und füge ihn in ein neues Text-Dokument im Programm Notepad++ ein. Speichere ihn unter dem Namen `p5js-AB2-2.js` in deinem Laufwerk ab.

Aufgabe 3. Die ganze Zeichenfläche bemalen.

Die gesamte Zeichenfläche soll nun mit der Flagge „befüllt“ werden. Vielleicht will man ja auch noch, dass man eingeben kann, wie groß die Zeichenfläche sein soll. Dazu kann man Variablen verwenden.

Gut zu wissen

Variablen

In Variablen kann man bestimmte Werte speichern. Jede Variable hat einen **Namen**, einen dazugehörigen **Wert** und einen **Variablentyp**. Anders als in der Mathematik können die Variablennamen auch länger sein, deswegen sollte man sprechende Namen benutzen, d.h., aus dem Namen sollte man auch ablesen können, welchen Wert man in dieser Variablen speichert. Bevor man Variablen nutzen kann, muss man sie deklarieren. Das macht man mit dem Schlüsselwort **var** (siehe Code 2.3.1).

Code 2.3.1

```
1 var breite = 100;
2 var hoehe = 1.5*breite;
3
4 function setup() {
5   createCanvas(3*breite, hoehe);
6 }
7
8 function draw() {
9   background(220);
10  circle(breite/2, hoehe/2, breite);
11  circle(3/2*breite, hoehe/2, breite);
12  circle(5/2*breite, hoehe/2, breite);
13 }
```

Code 2.3.1 arbeitet mit den Variablen `breite` und `hoehe`:

1. In den Zeilen 1 und 2 werden diese Variablen definiert. Man sagt also dem Computer, dass `breite` und `hoehe` Variablen sind. Beiden Variablen wird auch gleich ein Wert zugeordnet: `breite` wird auf 100 gesetzt. `hoehe` wird aus `breite` berechnet. D.h., der Computer sieht nach, welchen Wert wir in `breite` gespeichert haben und multipliziert diesen mit 1,5. Achte bitte darauf, dass man statt des Kommas einen Punkt setzen muss.
 2. In Zeile 5 wird Größe der Zeichenfläche definiert als das dreifache von `breite` x `hoehe`.
 3. In den Zeilen 10 bis 12 werden drei Kreise gezeichnet. Der erste Kreis hat den Mittelpunkt $(0,5 \cdot \text{breite}, \text{hoehe}/2)$ und den Durchmesser `breite`. Der zweite und dritte Kreis haben den gleichen Durchmesser, aber unterschiedliche x-Koordinaten für die Kreise, nämlich $1,5 \cdot \text{breite}$ und $2,5 \cdot \text{breite}$.
- a) Gib an, wie das Bild aussieht, das durch Code 2.3.1 gezeichnet wird. Übernimm den Code und teste deine Vermutung.
 - b) Ändere den Code 2.3.1 so ab, dass das gleiche Bild wie in Aufgabe 2 entsteht. `breite` soll dabei angeben, wie breit z.B. die blaue Fläche ist. Die Fahne soll den ganzen Zeichenbereich ausfüllen.

Kopiere deinen Code und füge ihn in ein neues Text-Dokument im Programm Notepad++ ein. Speichere ihn unter dem Namen `p5js-AB2-3.js` in deinem Laufwerk ab.

Aufgabe 4. Schland!

Ändere deine Lösung aus Aufgabe 3 so ab, dass die gesamte Zeichenfläche mit der Deutschlandflagge ausgefüllt ist.

Zur Festlegung der Farben kannst du einen Color-Picker verwenden. Die gibt es online. Du musst nur danach suchen.

Kopiere deinen Code und füge ihn in ein neues Text-Dokument im Programm Notepad++ ein. Speichere ihn unter dem Namen `p5js-AB2-4.js` in deinem Laufwerk ab.

Aufgabe 5. Zufall!

Gut zu wissen

Zufall

Möchte man Spiele programmieren, benötigt man oft Zufallswerte. Der Befehl, um sich einen Zufallswert generieren zu lassen, ist `random`:

`random()` generiert eine Zahl zwischen 0 und 1 (z.B. 0.542).

`random(20)` generiert eine ganze Zahl zwischen 0 und 20 (z.B. 4, 18, 3).

`random(20,100)` generiert eine ganze Zahl zwischen 20 und 100 (z.B. 50, 87, 23).

In `p5js` muss der Befehl `random()` in einer Funktion verwendet werden. Deswegen wird in Code 2.5.1 dem Computer in Zeile 1 nur angegeben, dass es zwei Variablen mit den Namen `breite` und `hoehe` gibt. Diesen Variablen wurde noch kein Wert zugeordnet. In der Funktion `setup()` in den Zeile 3-7 wird in Zeile 4 der Variable `breite` ein Zufallswert von 20 bis 100 zugewiesen. In Zeile 5 wird der Wert für `hoehe` aus dem Wert von `breite` berechnet.

Code 2.5.1

```
1 var breite, hoehe;
2
3 function setup() {
4   breite = random(20,100);
5   hoehe = 1.5*breite;
6   createCanvas(3*breite, hoehe);
7 }
8
9 function draw() {
10  background(220);
11  circle(breite/2, hoehe/2, breite);
12  circle(3/2*breite, hoehe/2, breite);
13  circle(5/2*breite, hoehe/2, breite);
14 }
```

Ändern Sie den Ihre Lösung für Aufgabe 4 ab, sodass Flaggen bei jedem Start des Programms die gezeichnete Flagge unterschiedlich groß ist.

Übernimm den Code 2.5.1. Ändere ihn so, dass die gezeichneten Bilder zwischen 150 und 270 Pixel breit sind.

Kopiere deinen Code und füge ihn in ein neues Text-Dokument im Programm Notepad++ ein. Speichere ihn unter dem Namen `p5js-AB2-5.js` in deinem Laufwerk ab.

Aufgabe 6. Phantasieflagge.

Gut zu wissen

p5js und Animationen

Ohne es zu wissen, hast du schon die ganze Zeit Animationen gezeichnet. Dir ist sicher schon aufgefallen, dass es in p5js-Programmen zwei unterschiedliche Teile gibt. In der Funktion `setup()` werden alle Anweisungen geschrieben, die nur einmal am Anfang ausgeführt werden sollen. Alle Anweisungen, die in der Funktion `draw()` stehen, werden immer wieder ausgeführt.

Mit Animationen wollen wir uns (noch) nicht beschäftigen. Schreibe deswegen dein Programm komplett in der Funktion `setup()`, deine Anweisungen sollten unter dem `createCanvas()`-Befehl stehen.

Ändere dein Programm aus Aufgabe 4 so ab, dass alle deine Anweisungen in der `setup()`-Funktion stehen (siehe „Gut zu wissen – p5js und Animationen“-Box). Das Programm soll jetzt die Flagge nicht mehr mit vorgegebenen Farben zeichnen, sondern für jede der zu zeichnenden Flächen einen Farbton zufällig zuordnen.

Kopiere deinen Code und füge ihn in ein neues Text-Dokument im Programm Notepad++ ein. Speichere ihn unter dem Namen `p5js-AB2-6.js` in deinem Laufwerk ab.

Aufgabe 7. PhantasieflaggeN.

Gut zu wissen

Framerate

Wie schnell die Animation verläuft, kannst du in der `setup()`-Funktion mit dem Befehl `frameRate()` steuern. Nähere Informationen findest du in der Referenz von p5js.

Kopiere deine Lösung aus Aufgabe 6 in die `draw()`-Funktion. Setze die Framerate (siehe „Gut zu wissen – Framerate“-Box) auf ein sinnvolles Maß.

Kopiere deinen Code und füge ihn in ein neues Text-Dokument im Programm Notepad++ ein. Speichere ihn unter dem Namen `p5js-AB2-6.js` in deinem Laufwerk ab.