

# Build your Robot

Handout 6.1 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022



<https://p5js.org/>

## function preload

Dieser Code dient dazu, ein Bild auf der Canvas zu platzieren.

```
let img;
function preload() {
  img = loadImage("Datei_Name.png");
}

function setup() {
  createCanvas(420, 660);
}

function draw() {
  background(150);
  image(img, 0, 0);
  img.resize(420, 660);
  if (mouseIsPressed) {
    print(mouseX, mouseY);
  }
}
```

# Build your Robot

Handout 6.2 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022



<https://p5js.org/>

## function preload

Erklärungen zum Code

```
let img;
function preload() {
  img = loadImage("Datei_Name.png");
}

function setup() {
  createCanvas(420, 660);
}

function draw() {
  background(150);
  image(img, 0, 0);
  img.resize(420, 660);
  if (mouseIsPressed) {
    print(mouseX, mouseY);
  }
}
```

if (mouseIsPressed) ("mouse is pressed") prüft, ob die Maustaste gedrückt wird.  
mouseX, mouseY sind die horizontalen und senkrechten Koordinaten der Maus.  
Der Code hat die Funktion, dass beim Drücken der Mausetaste die x-y-Koordinaten des Cursors in der console angezeigt werden.

let img;

let erstellt und benennt eine neue Variable. Eine Variable ist ein Container für einen Wert.

function preload()

preload platziert das Bild auf der Canvas.

img = loadImage ("Datei\_Name");

Der Code img. loadImage gibt den Befehl, das Bild zu laden. ("Datei\_Name") in diese Klammer kommt der Dateiname des Bildes.

> Für "Datei\_Name" wird der jeweilige Dateiname der Bild-Datei inklusive Endung (.jpg oder .png) eingegeben.

image (img, 0, 0);

In der function draw() wird dieser Code benötigt um den Inhalt der Variablen img auf der Canvas zu platzieren. Die Zahlen (img, 0, 0) stehen für die Koordinaten der linken oberen Bildecke.

img.resize (420, 660);

resize() skaliert das Bild auf die Anzahl Pixel, in diesem Beispiel auf 420 x 660 Pixel.

# Build your Robot

Handout 6.3 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022



<https://p5js.org/>

## function moving eyes

Dieser Code dient dazu, die Augen mittels der Position des Mauszeigers zu bewegen.

```
function setup() {
  createCanvas(420, 660);
}
function draw() {
  background(255);

  // Large eye right
  fill(255);
  ellipse(218, 160, 30, 30);

  //large eye left
  ellipse(180, 160, 30, 30);

  // moving Pupil right
  fill(0, 0, 255);
  let p1 = map(mouseX, 0, width, 210, 220);
  let p2 = map(mouseY, 0, height, 150, 162);
  ellipse(p1, p2, 10, 10);

  // moving Pupil left
  fill(0, 0, 255);
  let p3 = map(mouseX, 0, width, 172, 182);
  let p4 = map(mouseY, 0, height, 150, 162);
  ellipse(p3, p4, 10, 10);
}
```

# Build your Robot

Handout 6.4 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022

p5.js

<https://p5js.org/>

## function moving eyes

Erklärungen zum Code

```
function setup() {
  createCanvas(420, 660);
}
function draw() {
  background(255);

  // Large eye right
  fill(255);
  ellipse(218, 160, 30, 30);

  //large eye left
  ellipse(180, 160, 30, 30);

  // moving Pupil right
  fill(0, 0, 255);
  let p1 = map(mouseX, 0, width, 210, 220);
  let p2 = map(mouseY, 0, height, 150,
162);
  ellipse(p1, p2, 10, 10);

  // moving Pupil left
  fill(0, 0, 255);
  let p3 = map(mouseX, 0, width, 172, 182);
  let p4 = map(mouseY, 0, height, 150,
162);
  ellipse(p3, p4, 10, 10);
}
```

```
// Large eye right
fill(255);
ellipse(218, 160, 30, 30);
//large eye left
ellipse(180, 160, 30, 30);
```

`fill` definiert die Farbe der Augen.

› Der Code für die beiden Pupillen wird wiederholt, lediglich die Variablen müssen umbenannt werden: `let p1` wird zu `p3`, `p2` zu `p4`.

```
// moving Pupil right
fill(0,0,255);
```

// `moving Pupil right` ist ein erklärender Kommentar zum Code.

`fill(0, 0, 255)` bestimmt die Farbe der bewegten Pupille.

```
let p1 = map
(mouseX, 0, width, 210, 220);
```

`mouseX` beinhaltet die horizontale Koordinate der Maus, `mouseY` die vertikale Koordinate.

`map` weist Zahlen eines bestimmten Wertebereichs einem anderen Wertebereich zu.

`width` entspricht der Breite der Canvas. Bei diesem Bsp. stehen die Zahlen (210, 220) für die Pixel-Werte, in denen sich die Pupille (`ellipse`) bewegen wird.

```
let p2 = map
(mouseY, 0, height, 150, 162);
```

`height` entspricht der Höhe der Canvas.

```
ellipse(p1, p2, 10, 10);
```

Die Werte `p1`, `p2` bezeichnen die Position (x- und y- Koordinaten) der Ellipse.

# Build your Robot

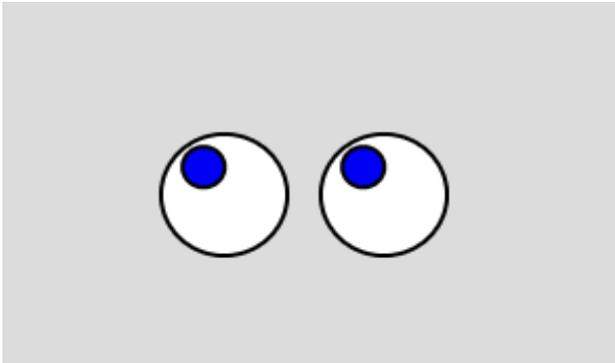
Handout 6.5 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022

p5.js

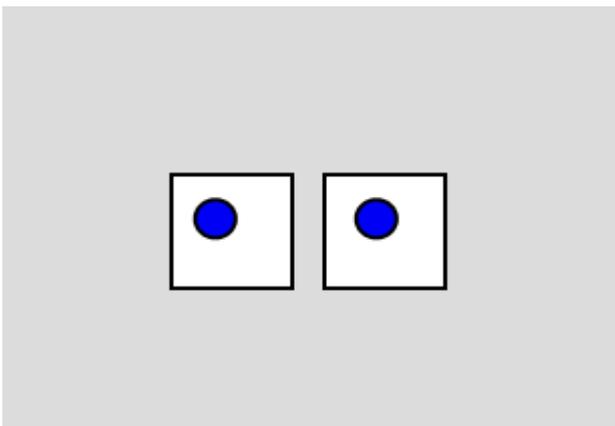
<https://p5js.org/>

## function moving eyes

Dieser Code dient dazu, die Augen mittels der Position des Mauszeigers zu bewegen.



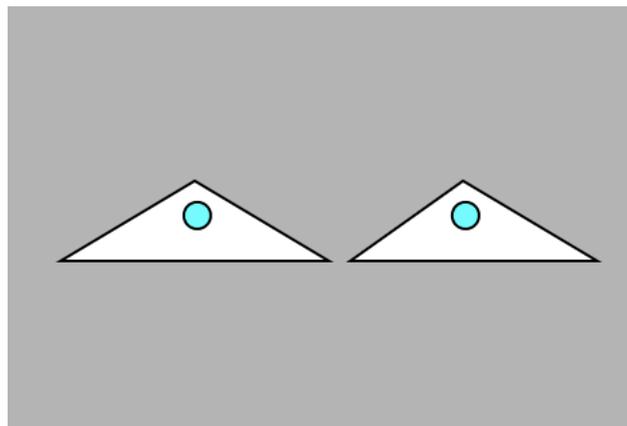
Bsp. des Codes auf dem Handout



Bsp. mit rechteckigen Augen:

```
function draw() {  
  background (220);  
  
  // Large eye right  
  fill (255);  
  rect (218, 160, 30, 30);  
  //large eye left  
  rect (180,160,30,30);
```

Anstatt `ellipse` kann `rect` eingefügt werden damit die Augen eine andere Wirkung bekommen.



Bsp. mit dreieckigen Augen

```
function draw() {  
  background (180);  
  
  // Large eye right  
  fill (255);  
  triangle(218, 190,260,160,310,190);  
  //large eye left  
  triangle(110,190, 160,160,210,190);
```

Es ist möglich die `ellipse` mit `triangle` zu ersetzen um einen z.B. stechenderen Blick der Augen zu generieren.

# Build your Robot

Handout 6.6 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022



<https://p5js.org/>

## **function mic, micLevel, AudioIn**

Dieser Code aktiviert das Computer-Mikrofon und misst die Lautstärke mit `getLevel()`.

```
let mic, micLevel, c;
function setup() {
  createCanvas(420, 660);
  mic = new p5.AudioIn();
  mic.start();
}

function draw() {
  background(255);
  micLevel = mic.getLevel();
  c = map(micLevel, 0, 0.5, 0, 30);
  fill(0);
  rect(190, 185, 40, 10 + c);
  print(micLevel);
}
```

# Build your Robot

Handout 6.7 - Grundlagen II. Olivia Schwarz und Mirco Leali, Mai 2022



<https://p5js.org/>

## function mic, micLevel, AudioIn

Erklärungen zum Code

```
let mic, micLevel, c;
function setup() {
  createCanvas(420, 660);
  mic = new p5.AudioIn();
  mic.start();
}

function draw() {
  background(255);
  micLevel = mic.getLevel();
  c = map(micLevel, 0, 0.5, 0, 30);
  fill(0);
  rect(190, 185, 40, 10 + c);
  print(micLevel);
}
```

### let mic, micLevel, c;

let erstellt neue Variablen (**mic**, **micLevel**, **c**). Eine Variable ist ein Container für einen Wert. **mic** bezeichnet das Mikrofon des Computers. **micLevel** entspricht dem Pegel der Lautstärke.

### mic = new p5.AudioIn();

Der Wert des Audio-Inputs (AudioIn) wird in die Variable **mic** abgefüllt .

### mic.start();

Wird in der Funktion **setup** benötigt um das Mikrofon zu aktivieren.

### micLevel = mic.getLevel ();

**micLevel** beinhaltet den Wert der der Lautstärke, die vom Mikrofon gemessen wird.

### c = map (micLevel, 0, 0.5, 0, 30);

**map** ordnet eine Zahl von einem Wertebereich einem neuen Wertebereich zu. Der Wert von **micLevel** wird in diesem Beispiel in eine Zahl zwischen 0 und 30 umgewandelt.

### rect (190, 185, 40, 10 + c);

Die Basic-Form **rect** steht hier für den Mund.  
Die ersten beiden Zahlen bezeichnen die Koordinaten Rechtecks. Die Zahl 40 ist Breite des Rechtecks. 10+c definiert die Höhe. Die Variable **c** ist mit der Lautstärke gekoppelt, je lauter desto grösser wird **c** und desto höher wird das Rechteck.

### print (micLevel);

**print (micLevel)** zeigt den Mikrofonpegel im Feld **console** an. So wird ersichtlich, wie laut der gemessene Ton ist.