

Aufbau

je Di 16:30 — 18:00 (18:30)

am Ende des Semesters (Dezember/Januar) Sprechstunde für individuelle Projektbetreuung

11.10. Einführung

18.10. Algorithmen & Parameter

25.10. Visuelle Musik

Die zweite Hälfte des Semesters sollen individuelle Projekte entwickelt und visualisiert werden. Jedes Projekt soll am Ende browserfähig aufbereitet werden und auf der Werkschau auch am Bildschirm präsentiert werden können

Feiertag

Blockwoche

Workshop Touch Designer mit Alexander Vaccaro

22.11. Text

29.11. Bewegt(bild)

6.12. Daten

1 – Einführung

Einführung
p5.js und Umgebung
Begriffe und Nachschlagewerk
Koordinaten, Formen, Farben

1 – Einführung

<https://www.werkschau.digital/kurse-ba/regelbasiertes-gestalten>

p5.js als Werkzeug für die Wochenaufgaben und Endabgabe nutzen können.

Was kann das sein?

<http://www.generative-gestaltung.de/2/>

<https://thedotisblack.com/>

1 – Einführung

Mitte 1990er:
Java für interaktives Fernsehen
Java applets

Netscape bzw. Brendan Eich
Mocha = LiveScript = Javascript

Java ≠ Javascript
Grape ≠ Grapefruit

Processing (2001) baut auf Java auf



Qianqian Ye



Lauren Lee McCarthy

1 – Einführung

heute:

HTML = Content

CSS = Style

Javascript = User Action

p5.js

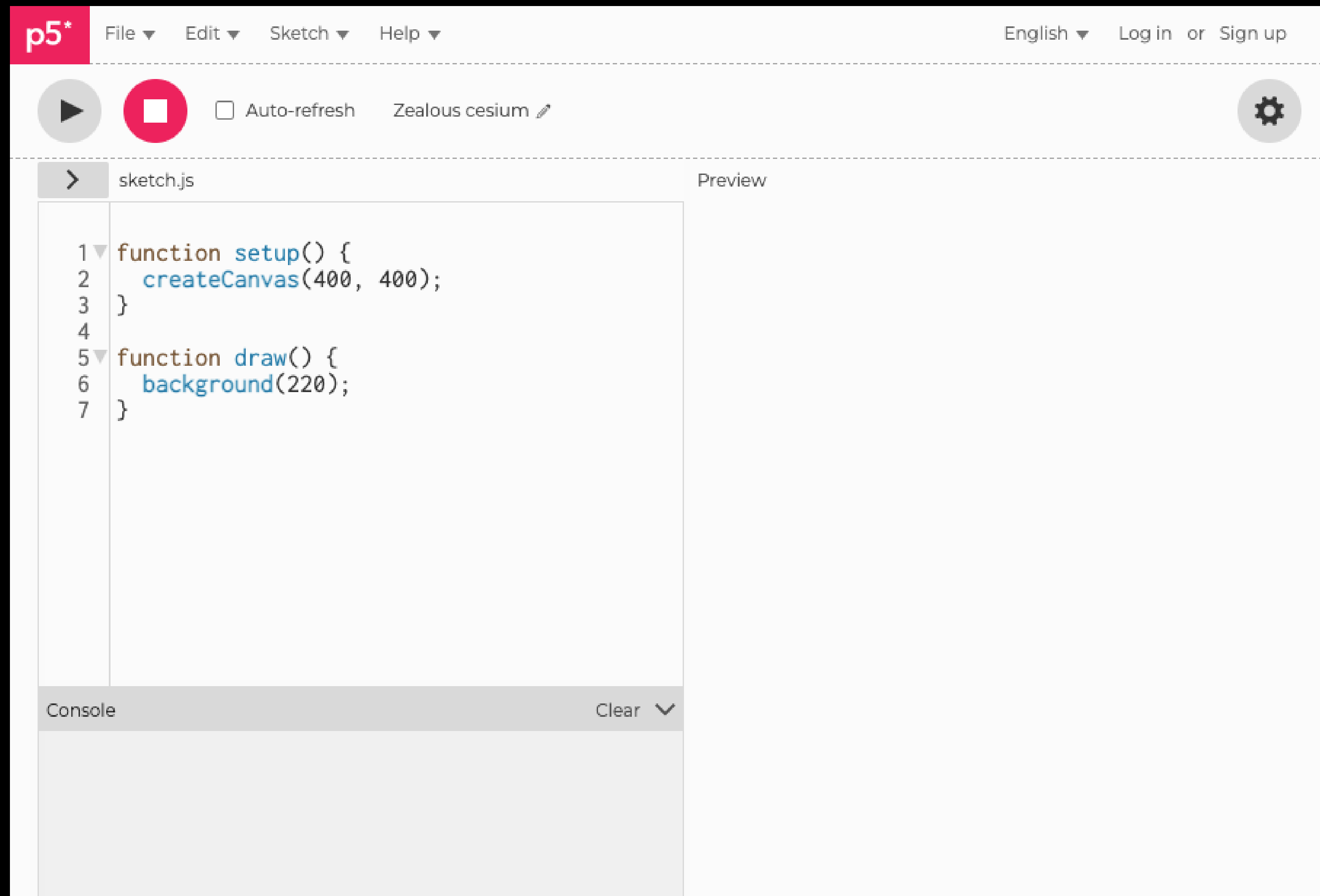
1) Library von Funktionen

2) Editor

1 – Einführung

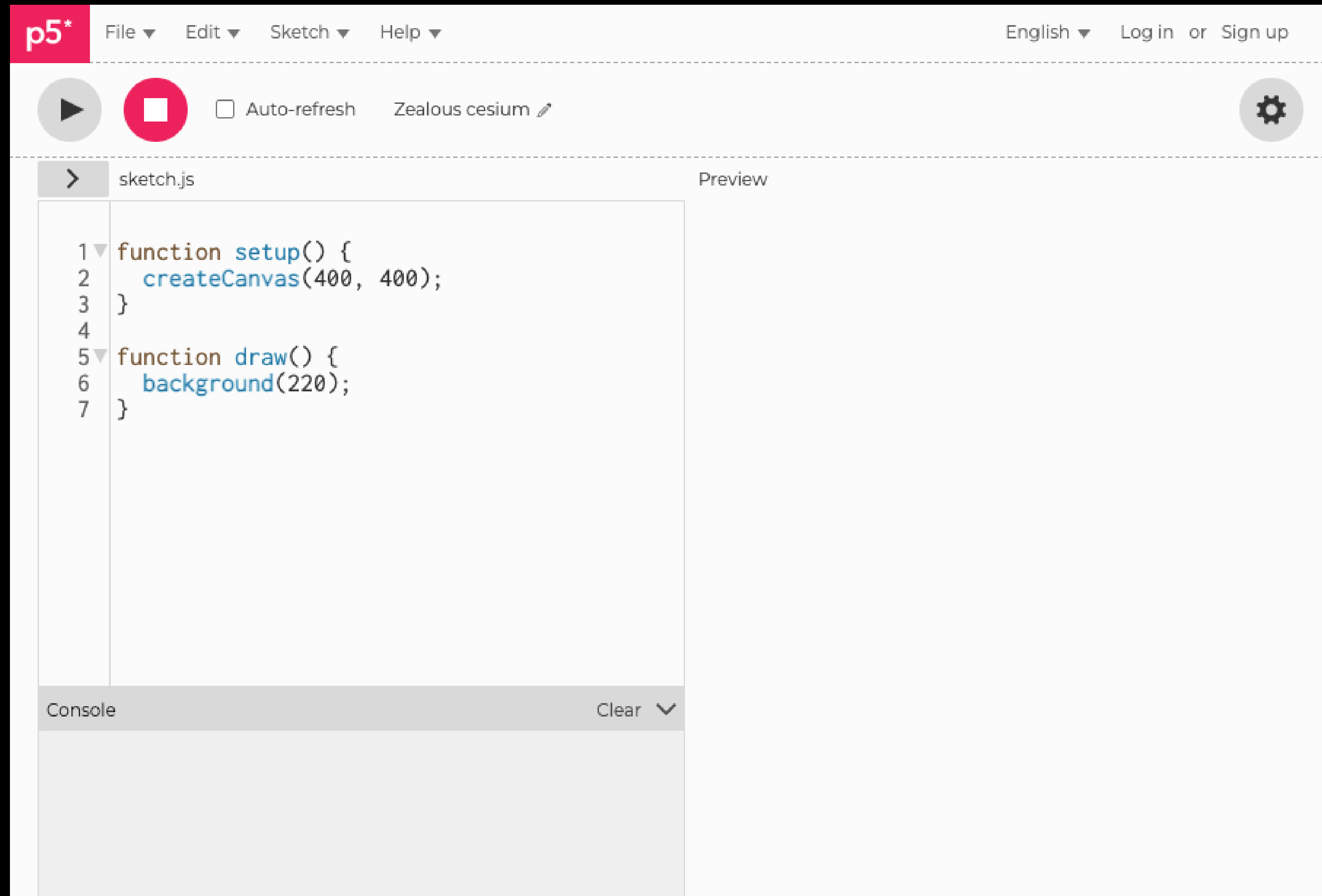
<https://editor.p5js.org/>

Sign up



1 – Einführung

Oberfläche
File speichern
Share
Download
Examples
MySketches
Collections
...



Gundlegendes

jede Befehlszeile muss abgeschlossen werden mit einem Semikolon ;

auf Klammertypen achten [] {} ()

Code kann oder muss verschachtelt sein (speziell bei Abfragen, Funktionen...)

Umgang mit Variablen beachten – es gibt eigene und auch Systemvariablen

Gundlegendes

p5.js arbeitet (zunächst) jede Zeile nacheinander ab

bestimmte Einstellungen bleiben so lange gültig, bis sie wieder durch neue ersetzt werden

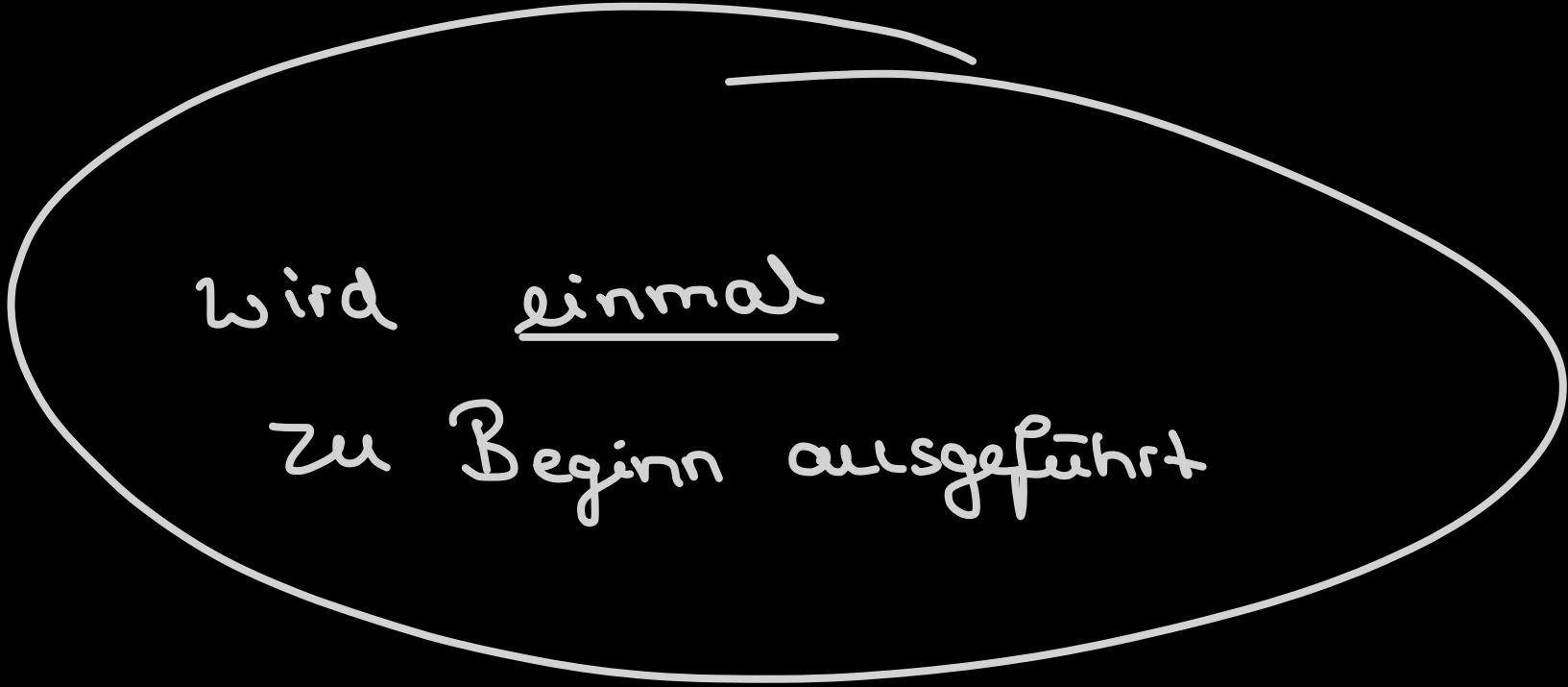
function setup

Umgebungseigenschaften wie Bildschirmgröße und Hintergrundfarbe definieren

Medien wie Bilder und Schriftarten beim Programmstart zu laden

Für jedes Programm kann es nur eine setup()-Funktion geben, und sie sollte nach der ersten Ausführung nicht erneut aufgerufen werden.

Hinweis: Die in setup() deklarierten Variablen sind in anderen Funktionen, einschließlich draw(), nicht zugänglich.



wird einmal
zu Beginn ausgeführt

Canvas

Erzeugt ein Canvas-Element im Dokument und legt dessen Abmessungen in Pixeln fest

wird nur einmal zu Beginn von `setup()` aufgerufen

function draw

die Funktion draw() führt kontinuierlich die in ihrem Block enthaltenen Codezeilen aus, bis das Programm angehalten oder noLoop() aufgerufen wird,

Wie oft draw() in jeder Sekunde ausgeführt wird, kann mit dem **frameRate()** Befehl gesteuert werden.

Es kann nur eine draw()-Funktion für jeden Sketch geben, und draw() muss vorhanden sein, wenn der Code kontinuierlich ausgeführt werden soll.



wird kontinuierlich
ausgeführt



Auskommentieren //

Unter auskommentieren versteht man, Teile des Programmcodes in einen Kommentar zu verpacken, sodass er zwar im Quelltext noch lesbar, aber unwirksam ist.

Variablen = veränderliche Größe

wie ein Umschlag oder Behälter, in dem Informationen aufbewahrt werden. Der Name auf dem Umschlag verrät welche Information darin ist.

```
let Kaffee = 12;  
let Preis = 0.5;
```

```
let Ausgaben = Kaffee*Preis;
```

Variablen, die mit `let` deklariert werden, gelten nur innerhalb ihres Blocks, während der Gültigkeitsbereich von Variablen mit `var` auf Funktionen ausgedehnt ist.

Console

spuckt eine Nachricht auf der Webkonsole des Browsers aus

man kann `print()` und `console.log()` austauschbar verwenden

Numbers

12

-12

1.23

-1.234

Vorsicht:

Dezimalziffern mit Komma in Europa
für p5.js mit Punkt arbeiten

Random

Gibt eine zufällige Kommazahl zurück.

Wenn kein Argument angegeben wird, wird eine Zufallszahl von 0 bis (aber nicht einschließlich) 1 zurückgegeben.

Wenn ein Argument angegeben wird und es sich um eine Zahl handelt, wird eine Zufallszahl von 0 bis zu (aber nicht einschließlich) der Zahl zurückgegeben.

Wenn ein Argument angegeben wird und es sich um ein Array handelt, wird ein zufälliges Element aus diesem Array zurückgegeben.

Wenn zwei Argumente angegeben werden, wird eine Zufallszahl vom ersten bis zum zweiten Argument (aber nicht einschließlich) zurückgegeben.

Eckige Klammer öffnen ([): Option + 5.

Eckige Klammer schließen (]): Option + 6.

Framerate

Gibt die Anzahl der Bilder an, die pro Sekunde angezeigt werden sollen. Der Funktionsaufruf `frameRate(30)` versucht beispielsweise, 30 Mal pro Sekunde zu aktualisieren. Wenn der Prozessor nicht schnell genug ist, um die angegebene Rate beizubehalten, wird die Bildrate nicht erreicht. Es wird empfohlen, die **Bildrate in `setup()`** festzulegen. Die voreingestellte Framerate basiert auf der Framerate des Bildschirms, die bei den meisten Computern auf 60 Bilder pro Sekunde eingestellt ist.

Arrays (or lists)

Datentyp, der nicht nur wie eine Variable genau einen Wert aufnehmen kann, sondern der mehrere Werte verwahrt, die über einen Schlüssel separat angesprochen werden können, technisch realisiert durch einen zusammenhängenden und deshalb linear adressierbaren Bereich von Speicherzellen

hochtrabendes Wort für Datentyp

ist ein Ort, um verwandte Elemente aufzubewahren – ähnlich wie man in einem Kühlschrank Lebensmittel aufbewahrt. Ein Array kann allerdings nur Strings oder Zahlen aufbewahren.

```
let Kühlschrankinhalt = ["Joghurt", "Mais", "Pudding", "Senf"];
```

Array besagt, dass die Variable Kühlschrankinhalt die Datentypen Joghurt, Mais, Pudding und Senf enthält. Jetzt kann man auch eine gezielte Abfrage machen.

```
console.log(Kühlschrankinhalt[0]);
```

Die Antwort der Console ist dann Joghurt.

creative

~~creative~~
Something

Strings

“Username”
“password”

Strings braucht man, weil ein Program Sätze oder Texte nicht versteht. Es nimmt bspw. keine Rücksicht auf Leerzeichen, die aber für die Lesbarkeit von Texten wichtig sind.

Ein Text, der als Text und **nicht als Befehl** verstanden werden soll, muss darum markiert werden. Das macht man mit dem ‘-Zeichen.

Anders sieht es bei Zahlen aus: Die kann man ohne Rücksicht auf Anführungszeichen eingeben.

System Variablen

mouseX

mouseY

windowWidth

windowHeight

Syntax

= Aufbau und der Gliederung von Sätzen und Satzstrukturen

```
rect(argument, argument, argument, argument)
```

Syntax

= Aufbau und der Gliederung von Sätzen und Satzstrukturen

```
rect(X, Y, width, height)
```

2D Primitives

`rect(x, y, width, height);`

`ellipse(x, y, width, height);`

`circle(x, y, d);`

`line(x1, y1, x2, y2);`

`point(x,y);`

`triangle(x1, y1, x2, y2, x3, y3);`

2D Primitives

```
rect(x, y, width, height);  
ellipse( x, y, width, height);  
circle(x, y, d);  
line(x1, y1, x2, y2);  
point(x,y);  
triangle(x1, y1, x2, y2, x3, y3);
```

Attribute

```
rectMode(CORNER  
CENTER );
```

```
ellipseMode(CENTER  
CORNER );
```

```
strokeWeight(the weight of the stroke in pixels);
```

```
strokeCap(ROUND  
SQUARE  
PROJECT );
```

push & pop

Die Funktion `push()` speichert die aktuellen Zeichenstil-Einstellungen und Transformationen, während `pop()` diese Einstellungen wiederherstellt. Beachten Sie, dass diese Funktionen immer zusammen verwendet werden. Sie ermöglichen es Ihnen, die Stil- und Transformationseinstellungen zu ändern und später zu dem Zustand zurückzukehren, den Sie hatten. Wenn ein neuer Zustand mit `push()` gestartet wird, baut er auf den aktuellen Stil- und Transformationsinformationen auf. Die Funktionen `push()` und `pop()` können eingebettet werden, um mehr Kontrolle zu ermöglichen.

if ... else ...

Die if-else-Anweisung dient der Steuerung des Codeflusses.

Eine Bedingung wird zwischen die Klammern nach “if” gesetzt. Wenn diese Bedingung als wahr ausgewertet wird, wird der Code zwischen den folgenden geschweiften Klammern ausgeführt. Wenn die Bedingung als falsch bewertet wird, wird stattdessen der Code zwischen den geschweiften Klammern des else-Blocks ausgeführt. Das Schreiben eines else-Blocks ist optional.

Begriffe

if ... else ...

Der Code wird Zeile für Zeile gelesen und die Bedingungen nach einander geprüft. Ist weder die Bedingung von 'if' noch aller anderen Bedingungen der 'else if' erfüllt, greift 'else'.

Beachte: else benötigt keine runden klammern, da die einzige bedingung ist, dass alle statements vorher 'false' waren

'else if' und 'else' sind optional. man kann auch nur das 'if' alleine verwenden

```
function draw() {  
    if (mouseX < width/2) {  
        tue dies und das  
    } else if (Bedingung) {  
        tue dies und das  
    } else {  
        tue das  
    }  
}
```


Begriffe

Konkrete Problemlöse-Techniken, sobald man weiß,
was man tun will:

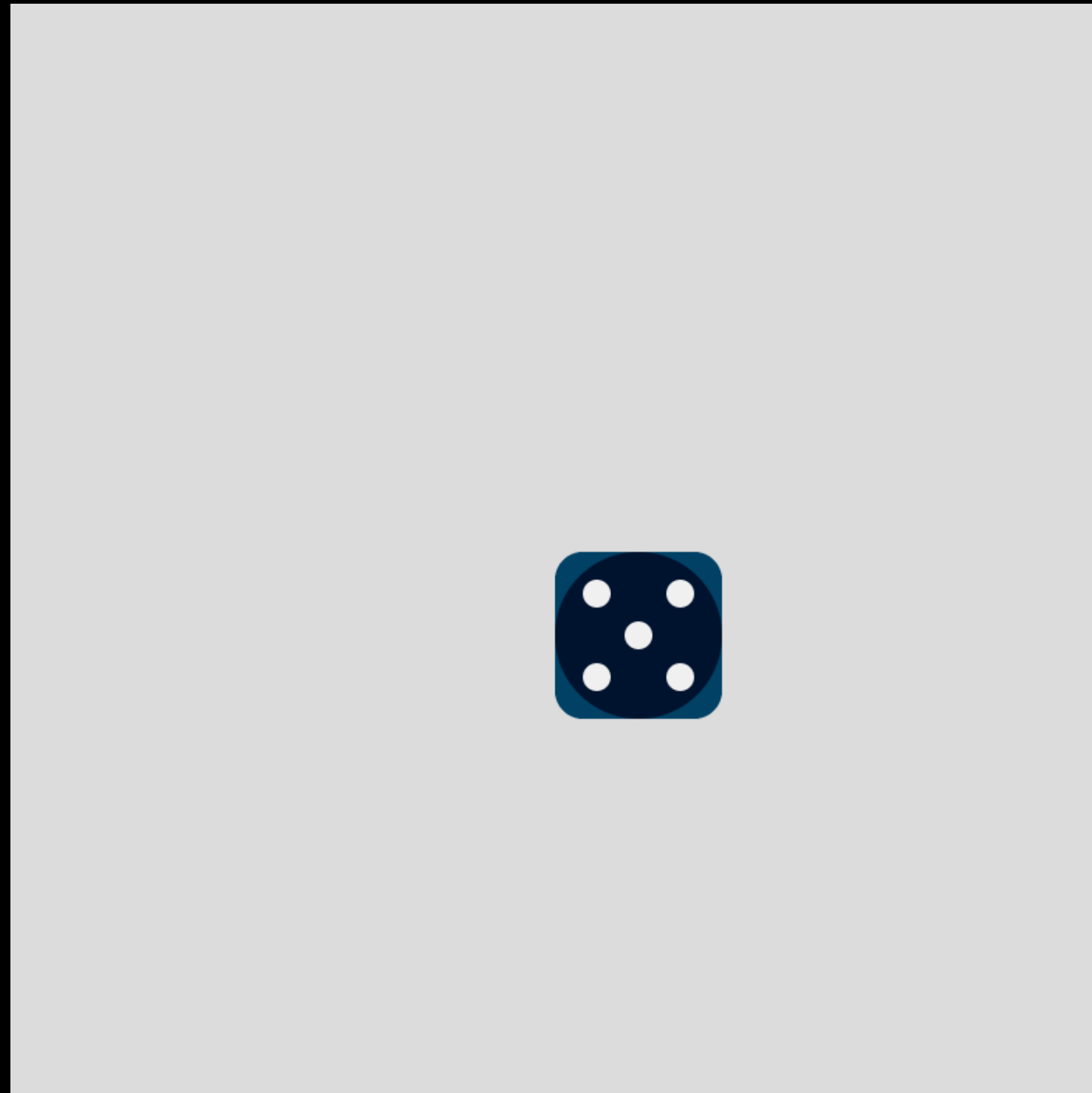
p5.js Reference

Modifikation (bestehenden Codes)

Vermehrung, Erweiterung (bestehenden Codes)

Collage (cut & paste)

from scratch



<https://editor.p5js.org/julia.vm/sketches/98j-8v6EK>