Funktionen

`createCanvas(width, height)`	LeinwandGrösse(Breite, Höhe)	in Pixel
`colorMode(mode, max1, max2, max3)`	FarbModus(Modus, Max1, Max2, Max3) *Konkretes Beispiel für HSB:* `colorMode(HSB,360,100,100);`	Mögliche Modi: HSB oder RGB (ist Standard). Max1 bis Max3 beziehen sich auf die erwünschten Maximalwerte für **R**ot, **G**rün, **B**lau beziehungsweise **H**ue, **S**aturation, **B**rightness.
`color(value1, value2, value3)`	Farbe(Wert1, Wert2, Wert3) *Konkretes Beispiel:* `meineFarbe = color(180,100,80);`	Wert1 bis Wert3 beziehen sich auf die Stärke der Farbkomponenten **R**ot, **G**rün, **B**lau beziehungsweise **H**ue, **S**aturation, **B**rightness.
`background(h, s, b)`	Hintergrundfarbe(hue, saturation, brightness)	hue: Farbwert | saturation: Sättigung | brightness: Helligkeit
`fill(h, s, b)`	Füllfarbe(hue, saturation, brightness)	hue: Farbwert | saturation: Sättigung | brightness: Helligkeit
`noFill()`	keine Füllung	
`stroke(h, s, b)`	Strichfarbe(hue, saturation, brightness)	hue: Farbwert | saturation: Sättigung | brightness: Helligkeit
`strokeWeight(weight)`	Strichdicke(Dicke), Rahmendicke(Dicke)	Dicke in Anzahl Pixel (Kommazahl)
`strokeCap(cap)`	Strichende(cap)	Mögliche Werte für cap: SQUARE, PROJECT oder ROUND
`strokeJoin(join)`	Strichverbindung(join)	Mögliche Werte für Join: MITER, BEVEL oder ROUND
`noStroke()`	kein Strich, kein Rahmen	
`point(x, y)`	Punkt(xPosition, yPosition)	
`line(x1, y1, x2, y2)`	Linie(x1, y1, x2, y2)	Linie von Punkt1(x1, y1) zu Punkt2(x2, y2)
`square(x, y, s)`	Quadrat(xPosition, yPosition, Seite)	in Pixel
`rect(x, y, width, height)`	Rechteck(xPosition, yPosition, Breite, Höhe)	in Pixel → Quadrat bei identischer Breite und Höhe
`triangle(x1,y1,x2,y2,x3,y3)`	Dreieck	x- und y-Position dreier Punkte
`quad(x1,y1,x2,y2,x3,y3,x4,y4)`	Viereck	x- und y-Position von vier Punkten
`circle(x, y, d)`	Kreis(xPosition, yPosition, Durchmesser)	in Pixel
`ellipse(x, y, width, height)`	Ellipse(xPosition, yPosition, Breite, Höhe)	in Pixel → Kreis bei identischer Breite und Höhe
`arc(x,y,width,height,start,stop)`	Kreisbogen(xPosition, yPosition, Breite, Höhe, Startwinkel, Stopwinkel)	Winkel in Bogenmass [0..2π]. Evtl. mit `radians()` von Grad in Bogenmass umrechnen.
`random(min, max)`	Zufallszahl(untereGrenze, obereGrenze) *Konkretes Beispiel 6er Würfel:* `z = int(random(1, 7));`	Liefert eine zufällige Kommazahl ab der unteren Grenze bis exklusiv oberer Grenze. Soll eine Ganzzahl resultieren, so entfernt die Funktion `int()` die Dezimalstellen.
`int(wert)`	Ganzzahl(Wert)	als Wert: u. A. `boolean, byte, char, color, float`
`frameRate(fps)`	Bildfrequenz(Bilder pro Sekunde)	Ganzzahl, als Standard sind 60 fps eingestellt.
`console.log(what)`	schreibeZeile(was) *Konkretes Beispiel:* `console.log("aktueller Wert " + i);`	Gibt im Konsolenfenster in doppelten Anführungszeichen stehende Zeichenketten, Werte von Variablen oder eine mit + komponierte Zeichenkette.
`save("dateiname.png")`	Sketchfenster im Sketch-Ordner speichern	mögliche Erweiterungen: .png oder .jpg
`bild =loadImage("dateiname.png")`	Bild in Var laden – Achtung in function preload()	mögliche Erweiterungen: .png, .jpg oder .gif
`image(bild, x, y, w, h)`	Bild anzeigen(Bildvariable, x, y, Breite, Höhe)	Die Angabe von Breite und Höhe ist optional.
`filter(mode)`	Filter(Modus)	Werte für Modus: THRESHOLD, GRAY, OPAQUE, INVERT, POSTERIZE, BLUR, ERODE, DILATE
`img.filter(mode)`	Auf ein Bild einen Filter anwenden.	Werte für Modus s. Eintrag `filter()`.

`map(value, start1, stop1, start2, stop2)`	abbilden(wert, start1, stop1, start2, stop2)	Vom Wert wird der bisherige Wertebereich von start1 bis stop1 in einen neuen von start2 bis stop2 umgerechnet.
`translate(x, y)`	Verschiebung(xRichtung, yRichtun)	Verschiebt den Ursprung 0/0 des Koordinatensystems.
`rotate(phi)`	Rotiere(Winkel)	Rotiert die Zeichenrichtung um den Winkel in Bogenmass.
`function preload() { … }`	Die Anweisungen werden vor dem setup()-Block ausgeführt: laden von Bildern oder Schriften.	
`function setup() { … }`	Die Anweisungen innerhalb der setup() Funktion werden beim Start einmalig ausgeführt. Sie wird zur Einrichtung der Umgebung benutzt. Z.B. werden Fenstergrösse, Bildfrequenz oder Farbmodus festgelegt.	
`function draw() { … }`	Wird nach setup()-Block kontinuierlich ausgeführt, bis das Programm beendet oder `noLoop()` ausgeführt wird.	
`function keyPressed() { … }`	Die Anweisungen innerhalb der geschweiften Klammern werden beim Druck einer Taste einmal ausgeführt.	
`function mousePressed() { … }`	Die Anweisungen innerhalb der geschweiften Klammern werden beim Druck der Maus einmal ausgeführt.	
`if (test) { … }`	Die umfassten Anweisungen werden nur ausgeführt, wenn die Testbedingung WAHR ist.	
`if (test) { … } else { … }`	Die Anweisungen werden abhängig von Test ausgeführt: erster Teil bei WAHR und der else-Teil bei FALSCH.	
`for (init; test; update) { … }`	Die For-Schleife wiederholt die umfassten Anweisungen solange die im Bereich `test` formulierte Bedingung WAHR ist. Das konkrete Beispiel rechts schreibt 5 7 9 11 zeilenweise in die Konsole.	`for (let i=5; i<13; i=i+2) { console.log(i); }`
`while (test) { … }`	Die While-Schleife wiederholt die umfassten Anweisungen solange die unter `test` formulierte Bedingung WAHR ist. Das konkrete Beispiel rechts schreibt 5 4 3 2 1 zeilenweise in die Konsole.	`let num = 5; while (num > 0) { console.log(num); num = num - 1; }`

Datentypen

`let laenge = 16;`	Number
`let name = "Huber";`	String
`let on = false;`	Boolean
`let person = { vName: "Joe", nName: "Meyer" }`	Object
`let serie = []; serie.push(124); serie.push(75); serie.push(13);`	Array mit Index von 0 bis Länge minus 1. Hier Länge 3 und Index im Bereich [0..2].

Datentypen umwandeln (casting)

`parseInt( )`	`Number( )`
`parseFloat( )`	`String( )`
	`Boolean( )`

Arithmetische Operatoren

`+`	Addition
`-`	Subtraktion
`*`	Multiplikation
`/`	Division
`%`	Restdivision (Modulo)
`=`	Wertzuweisung („wird zu“)

Logische Operatoren

`==`	Gleichheit
`<`	Kleiner
`>`	Grösser
`<=`	Kleiner gleich
`>=`	Grösser gleich
`!=`	Nicht gleich
`&&`	UND, verknüpft Bedingungen
`||`	ODER, verknüpft Bedingung.
`!`	NICHT, negiert Bedingunen

Systemvariablen

`mouseX`	x-Koordinate der Maus
`mouseY`	y-Koordinate der Maus
`pmouseX`	vorherige x-Koordinate der Maus
`pmouseY`	vorherige y-Koordinate der Maus
`mouseIsPressed`	ist TRUE wenn die Maustaste gedrückt wird, sonst FALSE
`keyIsPressed`	ist TRUE wenn eine Taste gedrückt wird, sonst FALSE
`width`	Fensterbreite des Canvas in Pixel
`height`	Fensterhöhe des Canvas in Pixel
`windowWidth`	Breite des Browserfensters in px
`windowHeight`	Höhe des Browserfensters in px
`frameCount`	Nr. des Anzeigefensters seit Start
`PI`	PI (3.14159…), auch `QUARTER_PI`, `HALF_PI`, `TWO_PI`