

```
// einzeliger Kommentar, /* mehrzeiliger Kommentar */
let x; const x = 10; // Deklaration, Zuweisung (blockscope)
var x; var x = 10; // functionscope
```

```
# Allerlei
** Exponentiation, i++ möglich, Kurzformen möglich,
== // equal to
=== // equal value and equal type
a = (b > 8) ? 1 : 0; // bedingte Zuweisung
&&, ||, !, // boolesche Operatoren
&, |, ~, ^ // bitweise
true, false
```

```
# setup und draw
function setup() {
  createCanvas(400, 400);
  // createCanvas(windowWidth, windowHeight);
}
function draw() {
  background(220);
}
```

```
# Systemvariablen
width, height # wie in createCanvas gesetzt
mouseX, mouseY, pmouseX, pmouseY
frameCount # Anzahl draw-Durchgänge
keyCode # Code des zuletzt gedrückten keys
mouseIsPressed, keyIsPressed (irgendeiner)
keyIsDown(UP_ARROW)
```

```
# Sonstige Funktionen
dist(x1,y1,x2,y2) # Abstand
# Zeichnen
point(x,y);
line(x1,y1,x2,y2);
rect(x,y,breite,höhe);
rectMode(CORNER/CENTER/CORNERS);
ellipse(x,y,xdurch,ydurch);
ellipseMode(CENTER/CORNER/CORNERS);
triangle(x1,y1,x2,y2,x3,y3);
quad(x1,y1,...);
```

```
# Farben
background(0);
stroke(100); strokeWeight(4);
noStroke();
```

```
fill(grauwert); fill(r,g,b); fill(r,g,b,alpha);
noFill();
colorMode(HSB,360,100,100);
colorMode(RGB,255,255,255);
var farbe = '#D7F052';
background(farbe), stroke(farbe), fill(farbe);
```

```
# Zufall
randomSeed(99);
x = random() # Dezimalzahl x in [0,1)
x = random(4) # x in [0,4)
x = random(4,9) # x in [4,9)
x = random(a) # a array zur Auswahl
```

```
# Text
textAlign(LEFT); textSize(16);
text('Bitte eine Zahl eingeben', 20, 40);
```

```
# Events
function mousePressed() { ... }
function mouseReleased() {...}
function mouseClicked(){...}
function keyPressed() { ... }
function keyReleased() {...}
```

```
# Keys
keyCode // Variable mit Code des letzten gedrückten keys
BACKSPACE, DELETE, ENTER, RETURN, TAB, ESCAPE, SHIFT,
CONTROL, OPTION, ALT, UP_ARROW, DOWN_ARROW,
LEFT_ARROW, RIGHT_ARROW.
```

```
# Functions
```

```
# Arrays
var a = [];
a.push("Hi"); a=[-1,2,4];
a[4] = 3;
a.toString(), a.join(" * ");
x = a.pop(); # letztes Element
k = a.push(y); # y dranhängen, k neue Länge
x = a.shift(); # 1.Element wird gelöscht und zurückgegeben
k = a.unshift(x); # fügt x vorne ein, k neue Länge
a[a.length] = 7; # etwas dranhängen
a[10] = 7; # ggf. mit Leerstellen was dranhängen
delete a[0]; # lässt vorne undefined Leerstelle
```

```
a.splice(i,n,x1,x2,...); # i Einfügeindex, n weg, x1,x2 ... rein
a.splice(0, 1);          # löscht erstes Element ohne undefined Loch
c = a.concat(a1,a2,...)  # konkatenieren
b = a.slice(2)           # Teilarray ab Index 2
b = a.slice(2,5)         # Teilarray mit Index [2,5)
matrix=[ [0,1,2], [10,11,12] ];
```

```
# Sortieren von Arrays
a.sort() # die Elemente werden als Strings sortiert
a.reverse()
a.sort(function(x,y) {return abs(x) - abs(y)}) # für zahlen
```

```
# P5 Array-Funktionen
max(a), min(a)
```

```
# Ein Array durchlaufen
for (let i = 0; i < a.length; i++) {print(a[i]);}
a.forEach(m); function m(x) {print(x);}
```

```
# ein zweidimensionales Array 20x20:
let tmp = []; let size = 20;
while(size --) {tmp.push([]);}
```

```
# Dictionaries
spielkarte={farbe:"Pik", wert:8};
```

```
# Klassen
class Auto {
  constructor(farbe, kW) {
    this.farbe=farbe;
    this.kW=kW;
  }
  zeigeFarbe() {
    print("Die Farbe ist " + this.farbe);
  }
}
class BestandsAuto extends Auto {
  constructor(farbe, kW, preis) {
    super(farbe, kW);
    this.preis=preis;
  }
  rabattiere(prozent) {
    this.preis=this.preis*(100-prozent)/100;
  }
}
```