



```
line(15,10, 85,90);  
line(25,10, 75,90);  
line(35,10, 65,90);  
line(45,10, 55,90);
```



```
line(10,10, 90,90);
```



```
strokeCrandom(255);  
line(60, 60, random(100), random(100));  
}
```



# Einführung in die Programmierung

::digipool

# Code Syntax

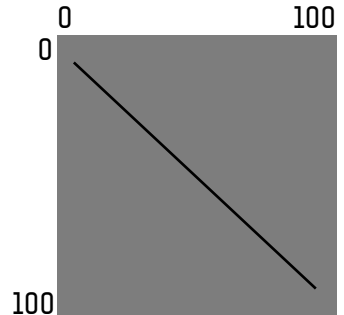
**Der Befehl:**

```
befehl();
```

```
line(10,10, 90,90);
```

**Der Scope:**

```
scope{  
    befehl();  
    ...  
}
```



**Die Kondition:**

```
if(x < y){  
    befehl();  
}
```

**Die Funktion:**

*Definition einer Funktion*

```
void meineFunktion(){  
    befehl();  
}
```

*Aufrufen einer Funktion*

```
meineFunktion();
```

# Typische Struktur

**Beispiel:**

```
int MeineGlobaleVariable;
```

```
void setup() {  
    befehl();  
    ...  
}
```

```
void draw() {  
    befehl();  
    ...  
}
```

**Beispiel:**

```
int MeineGlobaleVariable = 10;
```

```
void setup() {  
    size(MeineGlobaleVariable, MeineGlobaleVariable);  
}
```

```
void draw() {  
    print(„Hello World!“);  
}
```

# reference 1

## Structure

() (parentheses)  
, (comma)  
. (dot)  
/\* \*/ (multiline comment)  
/\*\* \*/ (doc comment)  
// (comment)  
;(semicolon)  
= (assign)  
[] (array access)  
{ } (curly braces)  
catch

## Environment

cursor()  
displayHeight  
displayWidth  
focused  
frameCount  
frameRate()  
frameRate  
height  
noCursor()  
size()  
width

# reference 2

## Data

### *Primitive:*

boolean  
byte  
char  
color  
double  
float  
int  
long

### *Composite:*

Array  
ArrayList  
HashMap  
Object  
String  
Table  
XML

### *Conversion:*

binary()  
boolean()  
byte()  
char()  
float()  
hex()  
int()  
str()  
unbinary()  
unhex()

### *String Functions:*

join()  
match()  
matchAll()  
nf()  
nfc()  
nfp()  
nfs()  
split()  
splitTokens()  
trim()

### *Array Functions:*

append()  
arrayCopy()  
concat()  
expand()  
reverse()  
shorten()  
sort()  
splice()  
subset()

# reference 3

## Control

### *Relational Operators:*

!= (inequality)

< (less than)

<= (less than or equal to)

== (equality)

> (greater than)

>= (greater than or equal to)

### *Iteration:*

for

while

### *Conditionals:*

?: (conditional)

break

case

continue

default

else

if

switch

### *Logical Operators:*

! (logical NOT)

&& (logical AND)

|| (logical OR)

# reference 4

## Shape

createShape()  
loadShape()  
PShape

### *2D Primitives:*

arc()  
ellipse()  
line()  
point()  
quad()  
rect()  
triangle()

### *Curves:*

bezier()  
bezierDetail()  
bezierPoint()  
bezierTangent()  
curve()  
curveDetail()  
curvePoint()  
curveTangent()  
curveTightness()

### *3D Primitives:*

box()  
sphere()  
sphereDetail()

### *Attributes:*

ellipseMode()  
noSmooth()  
rectMode()  
smooth()  
strokeCap()  
strokeJoin()  
strokeWeight()

### *Vertex:*

beginContour()  
beginShape()  
bezierVertex()  
curveVertex()  
endContour()  
endShape()  
quadraticVertex()  
vertex()

### *Loading & Displaying:*

shape()  
shapeMode()

# reference 5

## Input

### *Mouse:*

mouseButton  
mouseClicked()  
mouseDragged()  
mouseMoved()  
mousePressed()  
mousePressed  
mouseReleased()  
mouseX  
mouseY  
pmouseX  
pmouseY

### *Keyboard:*

key  
keyCode  
keyPressed()  
keyPressed  
keyReleased()  
keyTyped()

### *Files:*

BufferedReader  
createInput()  
createReader()  
createTable()  
createXML()  
loadBytes()  
loadStrings()  
loadTable()  
loadXML()  
open()  
parseXML()  
saveTable()  
selectFolder()  
selectInput()

## *Time & Date:*

day()  
hour()  
millis()  
minute()  
month()  
second()  
year()



# reference 6

## Output

### *Text Area:*

print()  
println()

### *Image:*

save()  
saveFrame()

### *Files:*

beginRaw()  
beginRecord()  
createOutput()  
createWriter()  
endRaw()  
endRecord()  
PrintWriter  
saveBytes()  
saveStream()  
saveStrings()  
saveXML()  
selectOutput()

## Transform

applyMatrix()  
popMatrix()  
printMatrix()  
pushMatrix()  
resetMatrix()  
rotate()  
rotateX()  
rotateY()  
rotateZ()  
scale()  
shearX()  
shearY()  
translate()

# reference 7

## Lights, Camera

### *Lights:*

ambientLight()  
directionalLight()  
lightFalloff()  
lights()  
lightSpecular()  
noLights()  
normal()  
pointLight()  
spotLight()

### *Camera:*

beginCamera()  
camera()  
endCamera()  
frustum()  
ortho()  
perspective()  
printCamera()  
printProjection()

### *Coordinates:*

modelX()  
modelY()  
modelZ()  
screenX()  
screenY()  
screenZ()

### *Material Properties:*

ambient()  
emissive()  
shininess()  
specular()

## Color

### *Setting:*

background()  
clear()  
colorMode()  
fill()  
noFill()  
noStroke()  
stroke()

### *Creating & Reading:*

alpha()  
blue()  
brightness()  
color()  
green()  
hue()  
lerpColor()  
red()  
saturation()

# reference 8

## Image

createImage()  
Plmage

### *Loading & Displaying:*

image()  
imageMode()  
loadImage()  
noTint()  
requestImage()  
tint()

### *Textures:*

texture()  
textureMode()  
textureWrap()

### *Pixels:*

blend()  
copy()  
filter()  
get()  
loadPixels()  
pixels[]  
set()  
updatePixels()

## Rendering

blendMode()  
createGraphics()  
PGraphics

### *Shaders:*

loadShader()  
PShader  
resetShader()  
shader()

## Typography

PFont

### *Loading & Displaying:*

createFont()  
loadFont()  
text()  
textFont()

### *Attributes:*

textAlign()  
textLeading()  
textMode()  
textSize()  
textWidth()

### *Metrics:*

textAscent()  
textDescent()

# reference 9

## Math

### *Operators:*

% (modulo)  
\* (multiply)  
\*= (multiply assign)  
+ (addition)  
++ (increment)  
+= (add assign)  
- (minus)  
-- (decrement)  
-= (subtract assign)  
/ (divide)  
/= (divide assign)

### *Bitwise Operators:*

& (bitwise AND)  
<< (left shift)  
>> (right shift)  
| (bitwise OR)

### *Calculation:*

abs()  
ceil()  
constrain()  
dist()  
exp()  
floor()  
lerp()  
log()  
mag()  
map()  
max()  
min()  
norm()  
pow()  
round()  
sq()  
sqrt()

### *Trigonometry:*

acos()  
asin()  
atan()  
atan2()  
cos()  
degrees()  
radians()  
sin()  
tan()

### *Random:*

noise()  
noiseDetail()  
noiseSeed()  
random()  
randomSeed()

### *Constants:*

HALF\_PI  
PI  
QUARTER\_PI  
TWO\_PI