
NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# PROCESSING

## NEW MEDIA


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# INTRODUCTIE

INTRODUCTIE

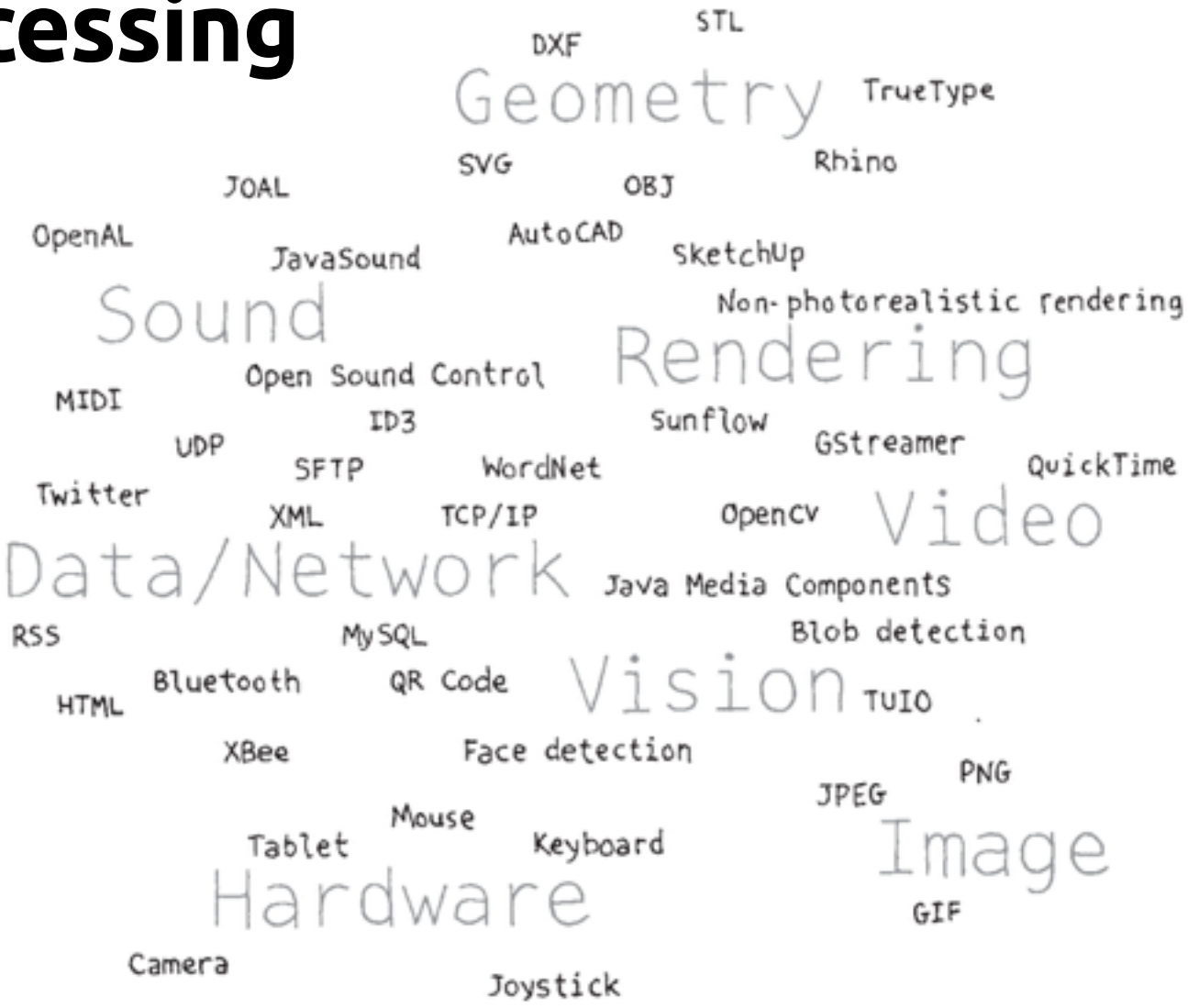
# Processing

- Casey Reas & Ben Fry
- Open source programmeertaal en IDE
- Bouwt verder op Java
- Uitbreidbare build mode: JavaScript, Android…

- Veel open source libraries beschikbaar
- Ideaal om gemakkelijk te communiceren met hardware
- Werkt vlot en platform onafhankelijk
- Ideale omgeving om te experimenteren

INTRODUCTIE

# Processing



INTRODUCTIE

# Processing

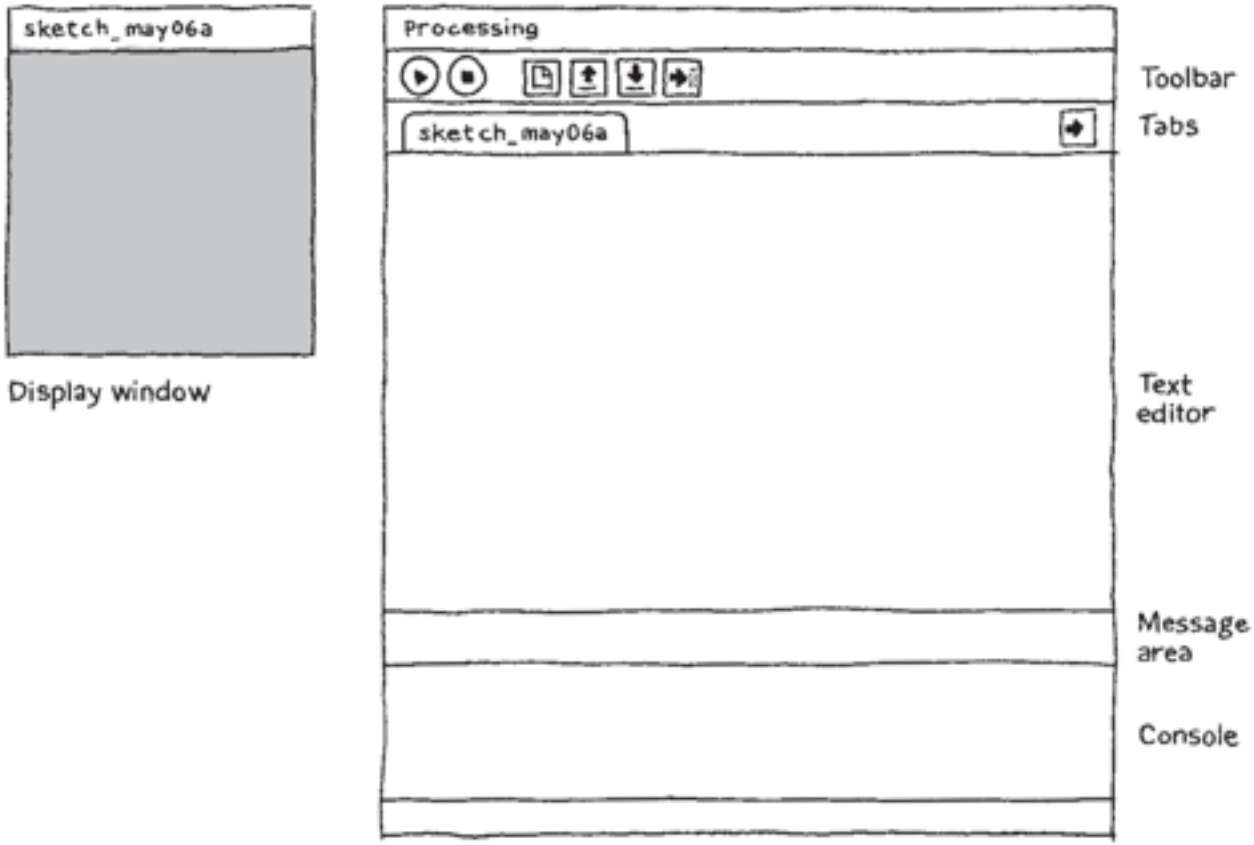
- Gebruik Processing IDE is geen intelliSense
- Gebruik Sublime Text, wel autocomplete en code snippets. Uitleg en koppeling in slides "**Sublime Text IDE**".

- Download Processing

INTRODUCTIE

# Eerste programma

- SAVE: PDE-file moet in map zitten met dezelfde naam



INTRODUCTIE

# Eerste programma

## Opbouw

```
void setup() {
        size(800, 600); //sketch
        background(#000000); //achtergrondkleur
        //voorbereiding, instellen
}

void draw() {
        //tekenen, oneindige loop
}
```

INTRODUCTIE

# Open Source Community

Online reference:

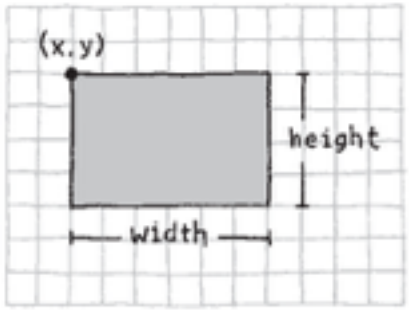
Processing reference


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

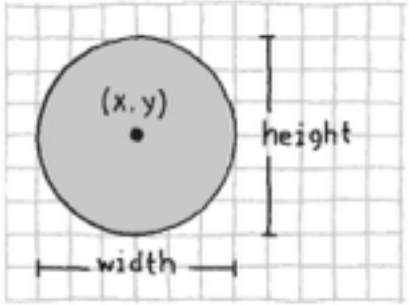
# DRAWING

DRAWING

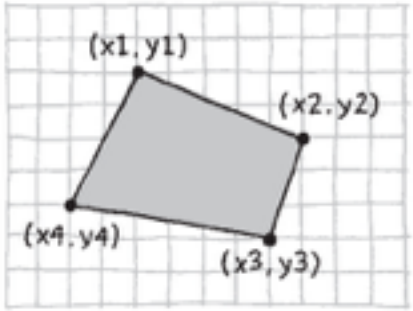
# Basic shapes




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



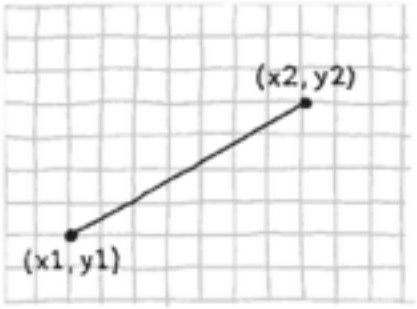

```
ellipse(x, y, width, height)
```



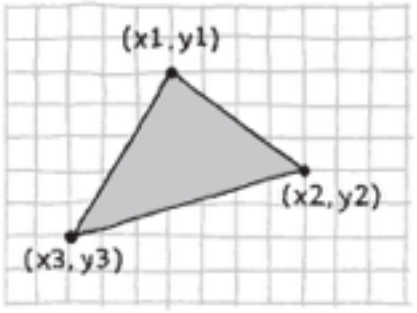

```
quad(x1, y1, x2, y2, x3, y3, x4, y4)
```

DRAWING

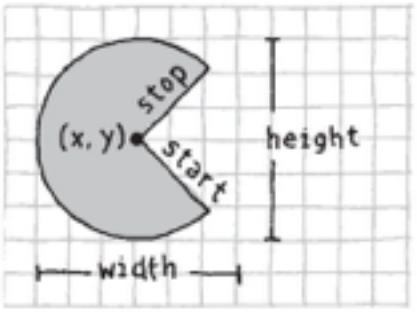
# Basic shapes




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




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




```
arc(x, y, width, height, start, stop)
```

DRAWING

# Drawing order

De volgorde van de code is ook de volgorde van tekenen.

```
rect();
ellipse();
triangle();
```

DRAWING

# Styling

```
strokeWeight(5);
background(255, 255, 255);  //of background(#FFFFFF)
fill(#FF0000);
stroke(#0000FF);
noFill();
noStroke();
```

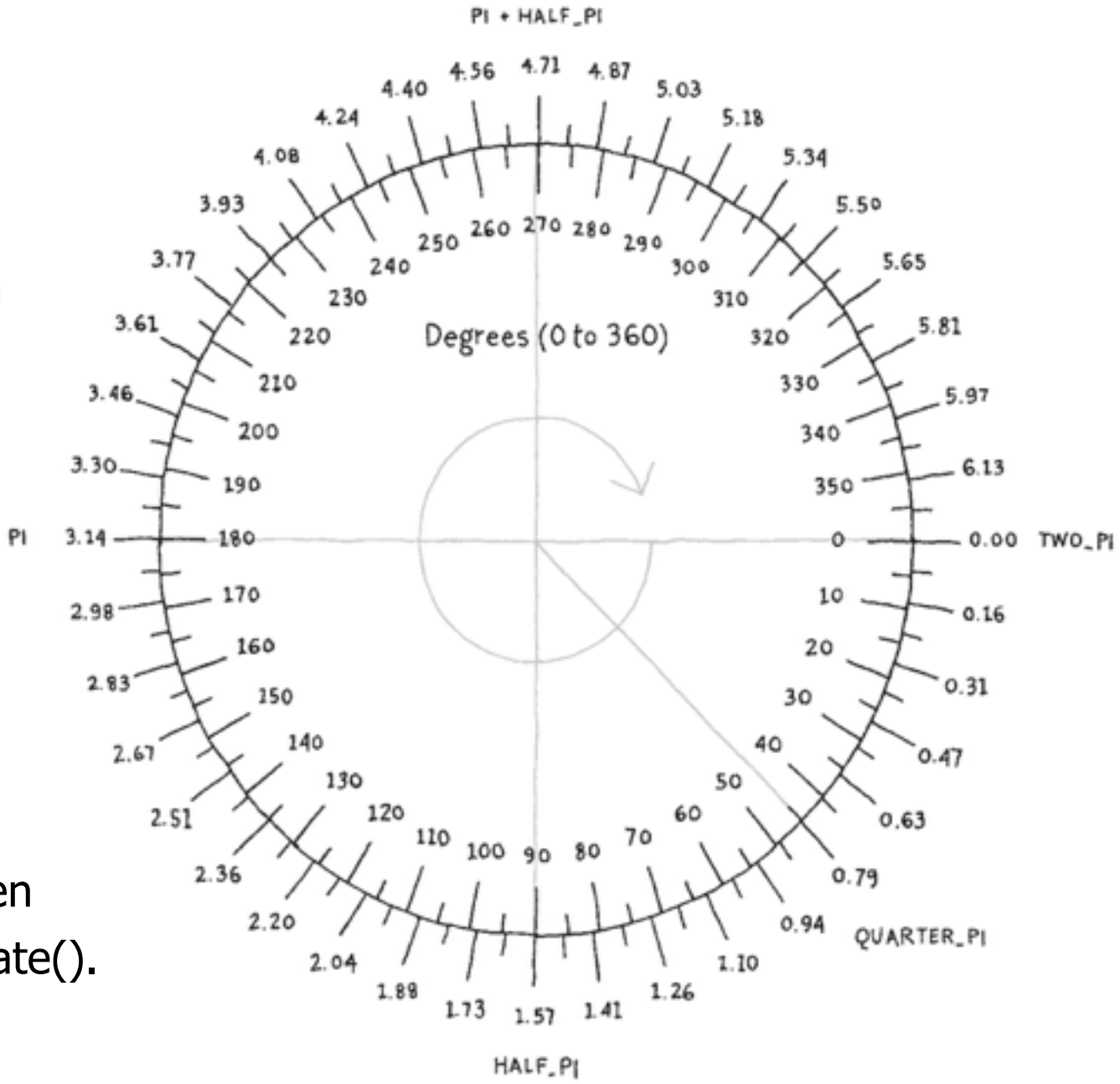
Transparency via 4de parameter bij kleuren:

```
fill(32, 50, 120, 100);
```

DRAWING

# Radians

Hoeken moeten in radialen.



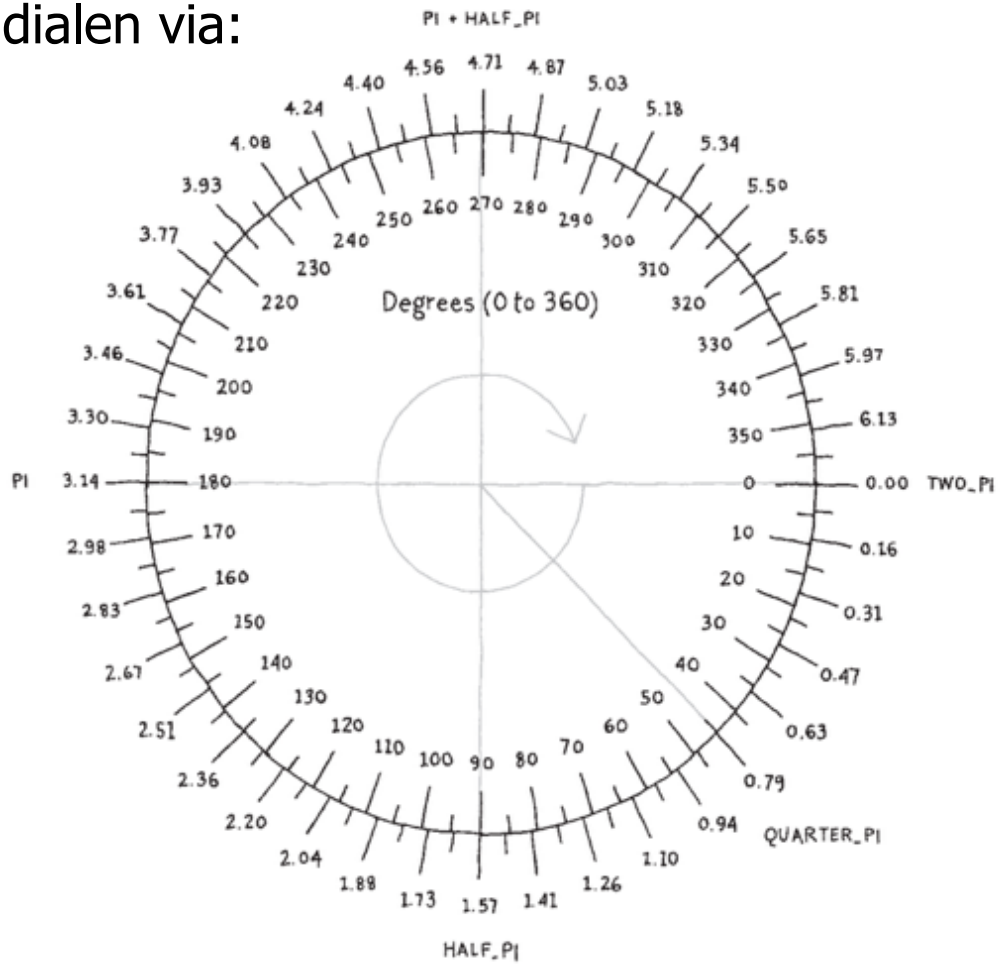

Bijvoorbeeld bij een arc() shape of rotate().

DRAWING

# Radians

Je kan ook converteren naar radialen via:

radians(180)



DRAWING

# Smooth

smooth()

noSmooth()

Vlotte rendering van randen.

DRAWING

# Custom Shapes

```
beginShape();
vertex(180, 82);
vertex(207, 36);
vertex(214, 63);
vertex(407, 11);
vertex(412, 30);
vertex(219, 82);
vertex(226, 109);
endShape(CLOSE);
```


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# Vars en structure

VARS

# Variabelen en notatie

```
int xPos;
xPos = 200;
```

Operators:

| | |
|---|---|
| + | Optellen |
| - | Aftrekken |
| * | Vermenigvuldigen |
| / | Delen |
| = | Toekennen |

VARS

# Types

boolean, byte, char, color, double

float, int, long

int[] numbers = { 90, 80, 70 }

String tekst = „new media”

…

STRUCTURE

# Controlestructuren

```
if (test) {
  statements
}
```

STRUCTURE

# Controlestructuren

```
if (expression) {
  statements
} else if (expression) {
  statements
} else {
  statements
}
```

STRUCTURE

# Controlestructuren

```
switch(letter) {
  case 'A':
    println("Alpha");
    break;
  case 'B':
    println("Bravo");
    break;
  default:
    println("Zulu");
    break;
}
```

STRUCTURE

# Controlestructuren

```
for (int i = 0; i < 100; i += 10) {
}
```

STRUCTURE

# Controlestructuren

```
while (expression) {
  statements
}
```


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# Input

INPUT

# Mouse input

mouseX = huidige x-positie cursor

mouseY = huidige y-positie cursor

pmouseX = vorige x-positie cursor

pmouseY = vorige y-positie cursor

INPUT

# Key input

```
void keyPressed() {
  if (key == CODED) {
    if (keyCode == UP) {
      ...
    } else if (keyCode == DOWN) {
      ...
    }
  }
}
```

INPUT

# Key input

```
if (keyPressed) {
    if (key == 'b' || key == 'B') {
      ...
    }
  }
```

INPUT

# Binnenkort meer input

Leap Motion

MaKey Makey

…


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# Math, Map, Random, Timer…

MATH

# Math functies

min()

max()

dist()

...

Check altijd de online reference voor meer!

MAP

# Remap waarden

**Syntax**

map(value, start1, stop1, start2, stop2);

map (knop, 0, 128, 0, 255);

**Parameters**

| | |
|---|---|
| value | float: waarde |
| start1 | float: laagste waarde |
| stop1 | float: hoogste waarde |
| start2 | float: laagste doelwaarde |
| stop2 | float: hoogste doelwaarde |

Bijvoorbeeld: inkomende draaiknop waarde is min 0 en max 128, maar we willen naar min 0 en max 255.

RANDOM

# Random functie

random(high)

random(low, high)

TIMER

# Timer

**Via millis()**

```
int timer;
if (millis() > timer + 1000) {
   //actie uitvoeren
   timer = millis();
}
```

# ...
# Constrain

Een waarde beperken tot een maximum en minimum

**Syntax**

constrain(value, low, high)

**Parameters**

value int, float: value die moet worden beperkt

low int, float: minimum

high int, float: maximum


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# Media

MEDIA

# Images

```
PImage photo;

void setup() {
  size(400, 400);
  photo = loadImage("picture.jpg");
}

void draw() {
  image(photo, 0, 0);
}
```

MEDIA

# Varia

Ook fonts, geluid, vectoren, …


NEW MEDIA &
COMMUNICATION
TECHNOLOGY

# JAVA

JAVA

# Java stuff

Je kan Java libraries importeren en gebruiken.

(Later in de labo's.)