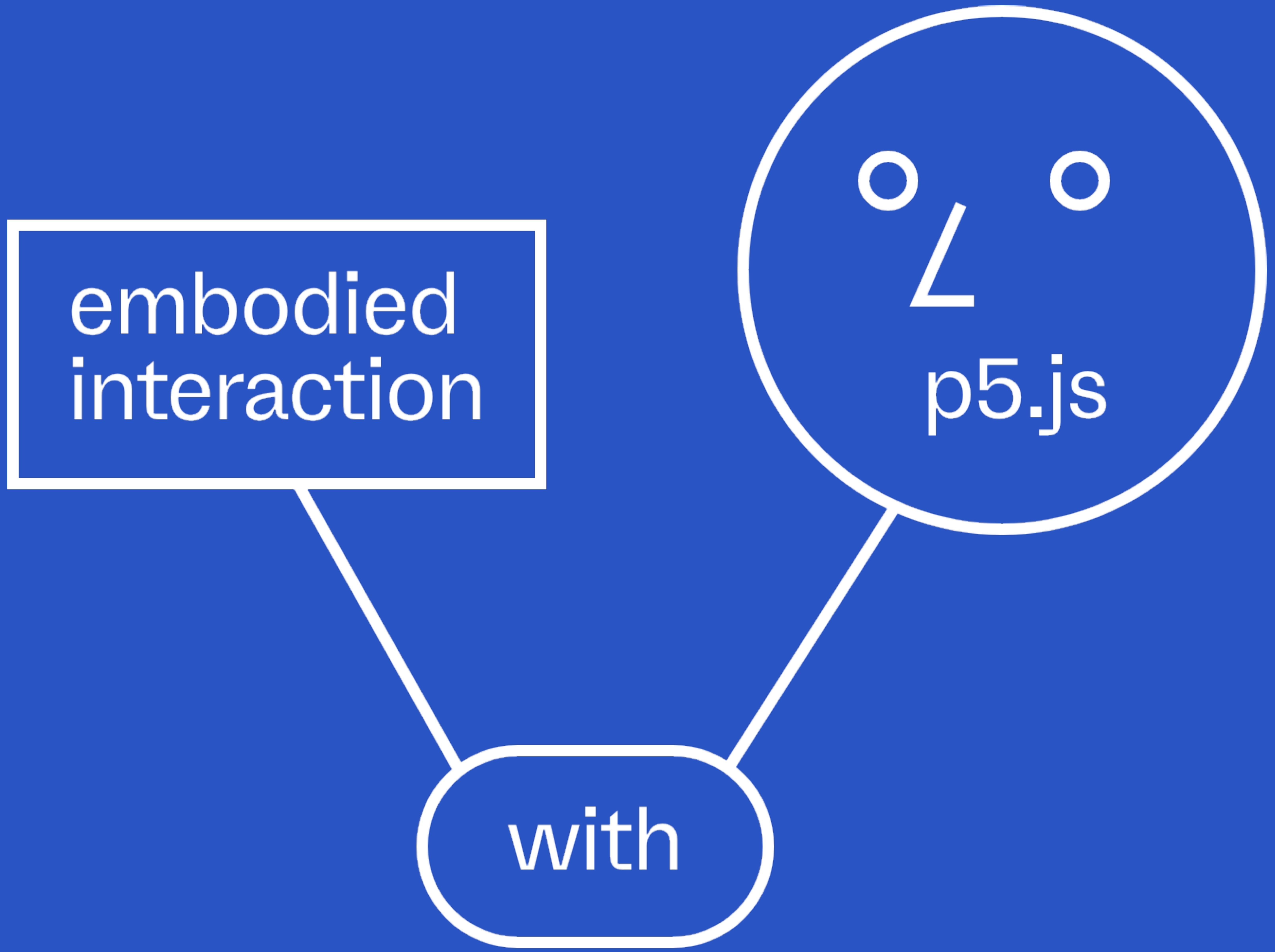
NG——VK



embodied interaction

p5.js

with

Body-Tracking mit p5.js und mediapipe

■

FHV
24.10.2025

© 2025

NG——VK

1 —
p5.js basics

© 2025

☞ conditions

```
if (a) {
    // do something
} else {
    // do something else
}
```

- if
- else
- else if
- && (and)
- || (or)
- ! (negation)

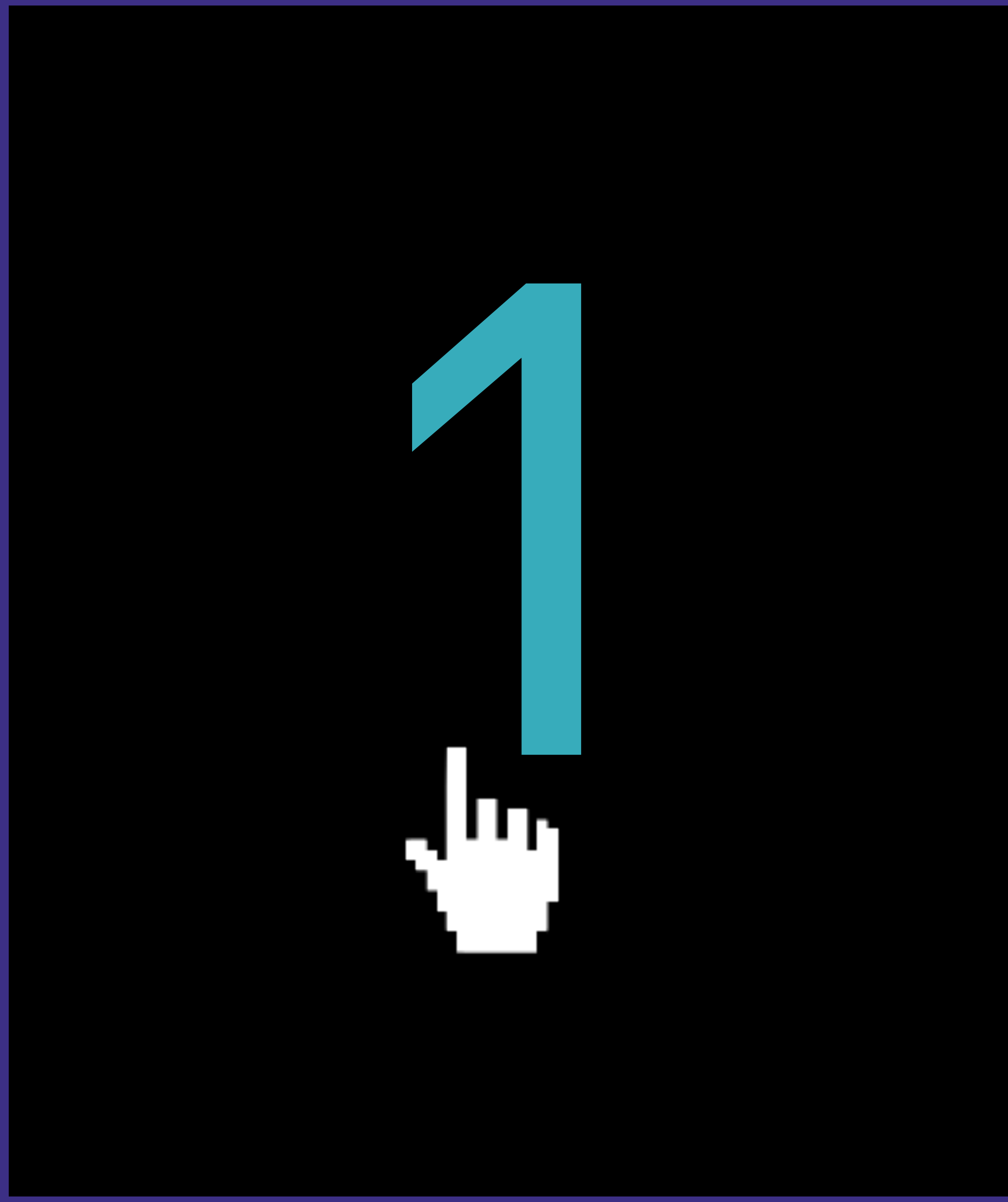
☞ arrays

```
let numbers = [1, 20, 3.5];

let cities = ["Paris", "Berlin", "Rome"];

let colors = ["blue", "#000", "#fff"];
```

© 2025

☞ using arrays

```
let numbers = [1, 20, 3.5];
text(numbers[0], mouseX, mouseY);
```



© 2025

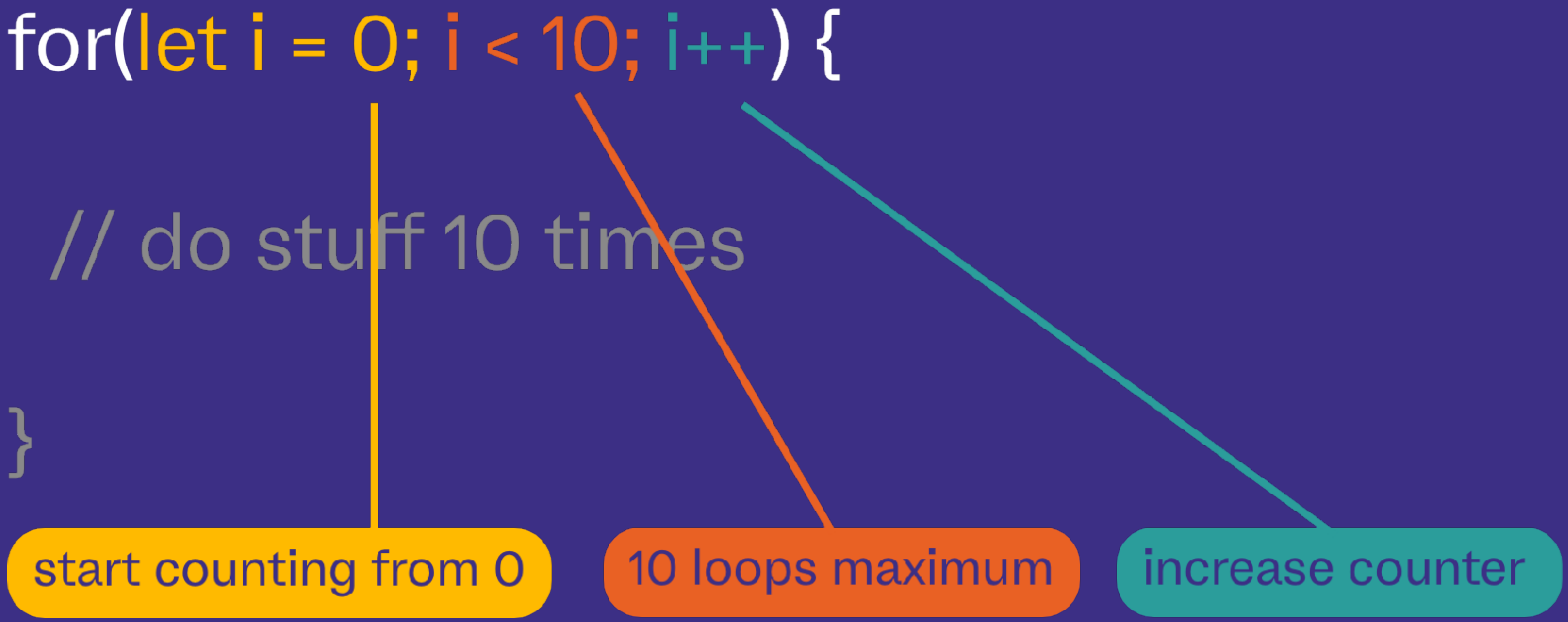
NG——VK

☞ loops (Schleifen)

```
for(let i = 0; i < 10; i++) {
  // do stuff 10 times
}
```

© 2025

☞ loops



```
for(let i = 0; i < 10; i++) {

  // do stuff 10 times

}
```

start counting from 0

10 loops maximum

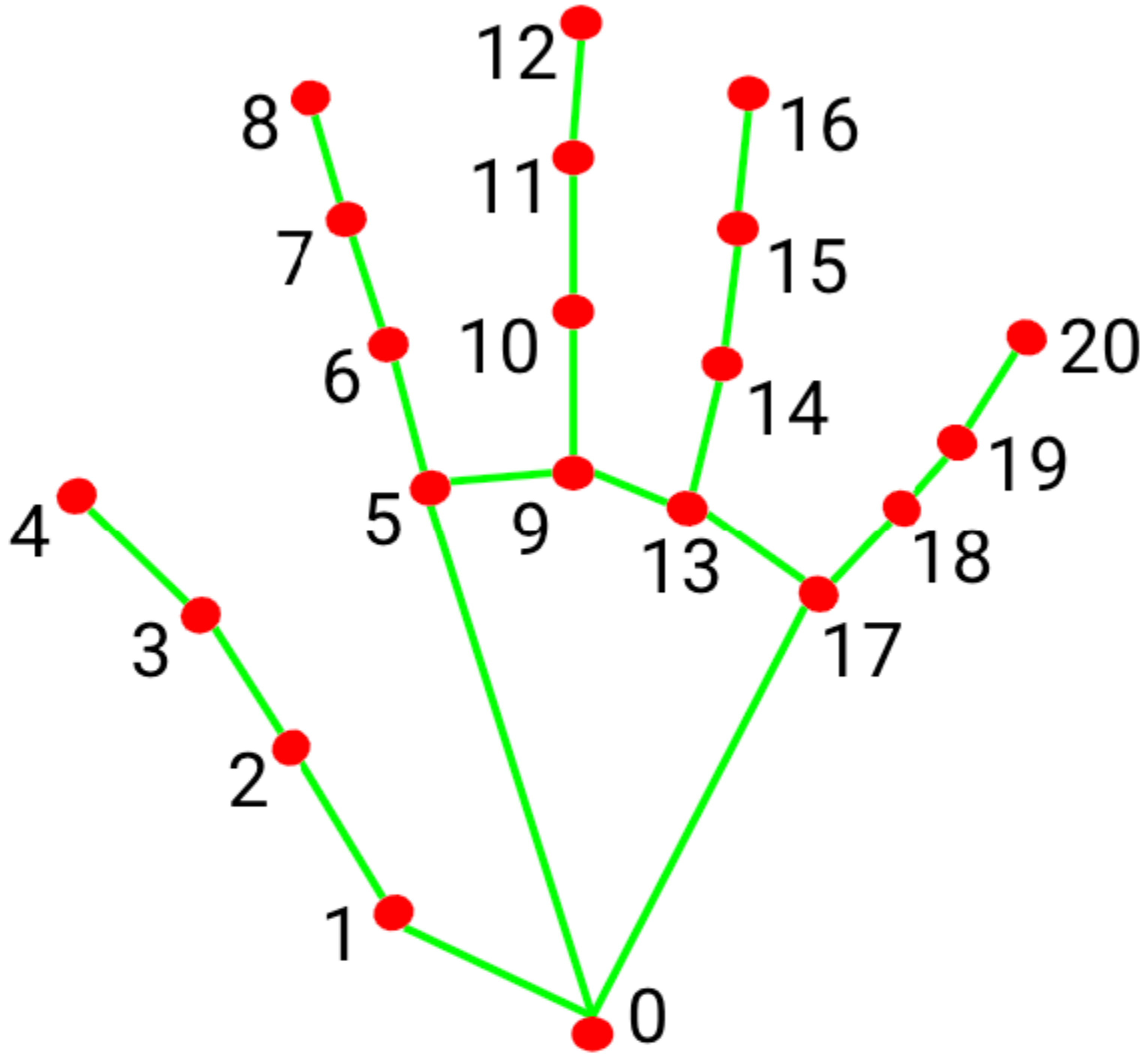
increase counter

© 2025

2 —
tracking basics

© 2025

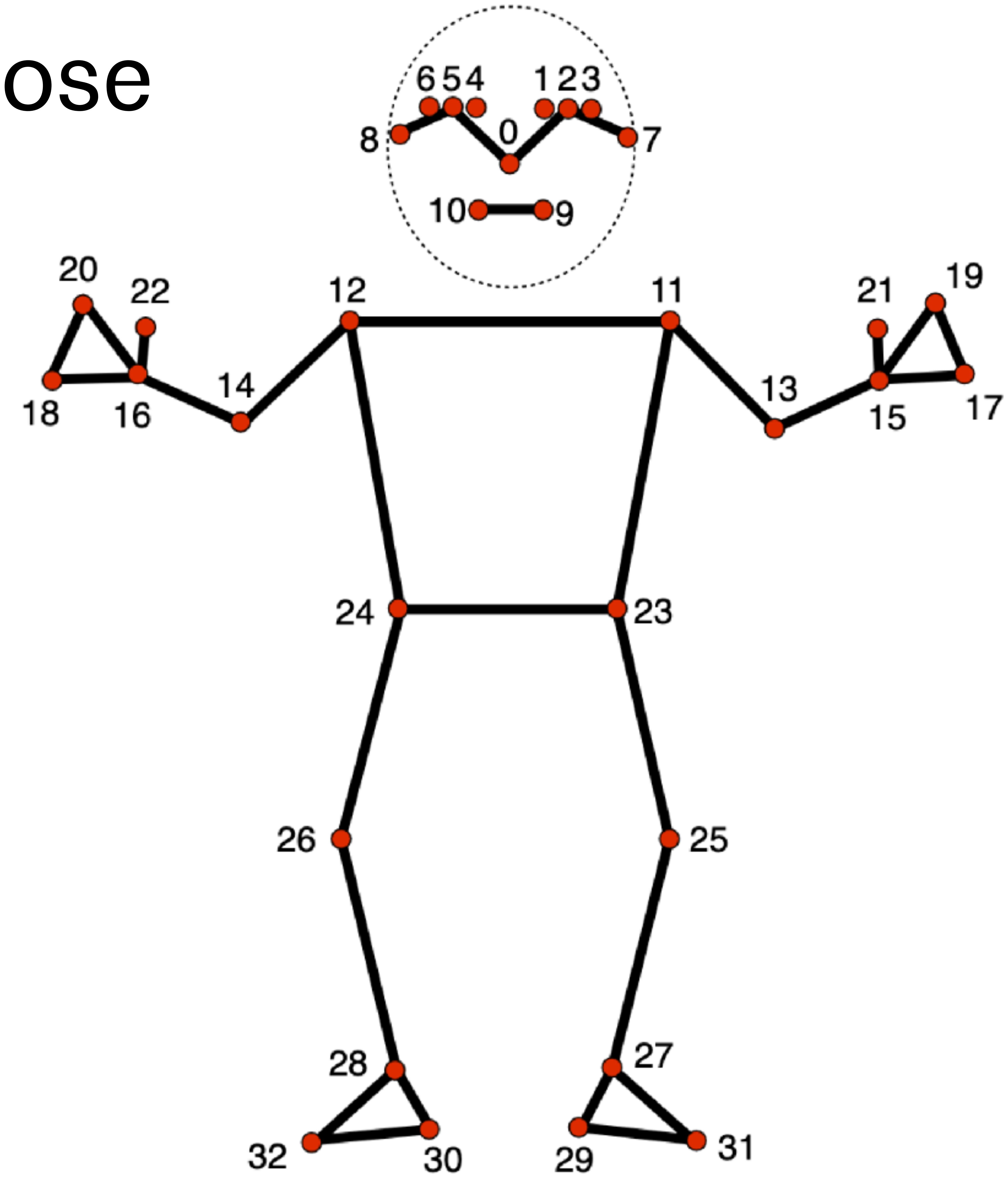
✌ hands



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

© 2025

✌ pose



0 = nose

12 = right shoulder

11 = left shoulder

26 = right knee

25 = left knee

32 = right foot

31 = left foot

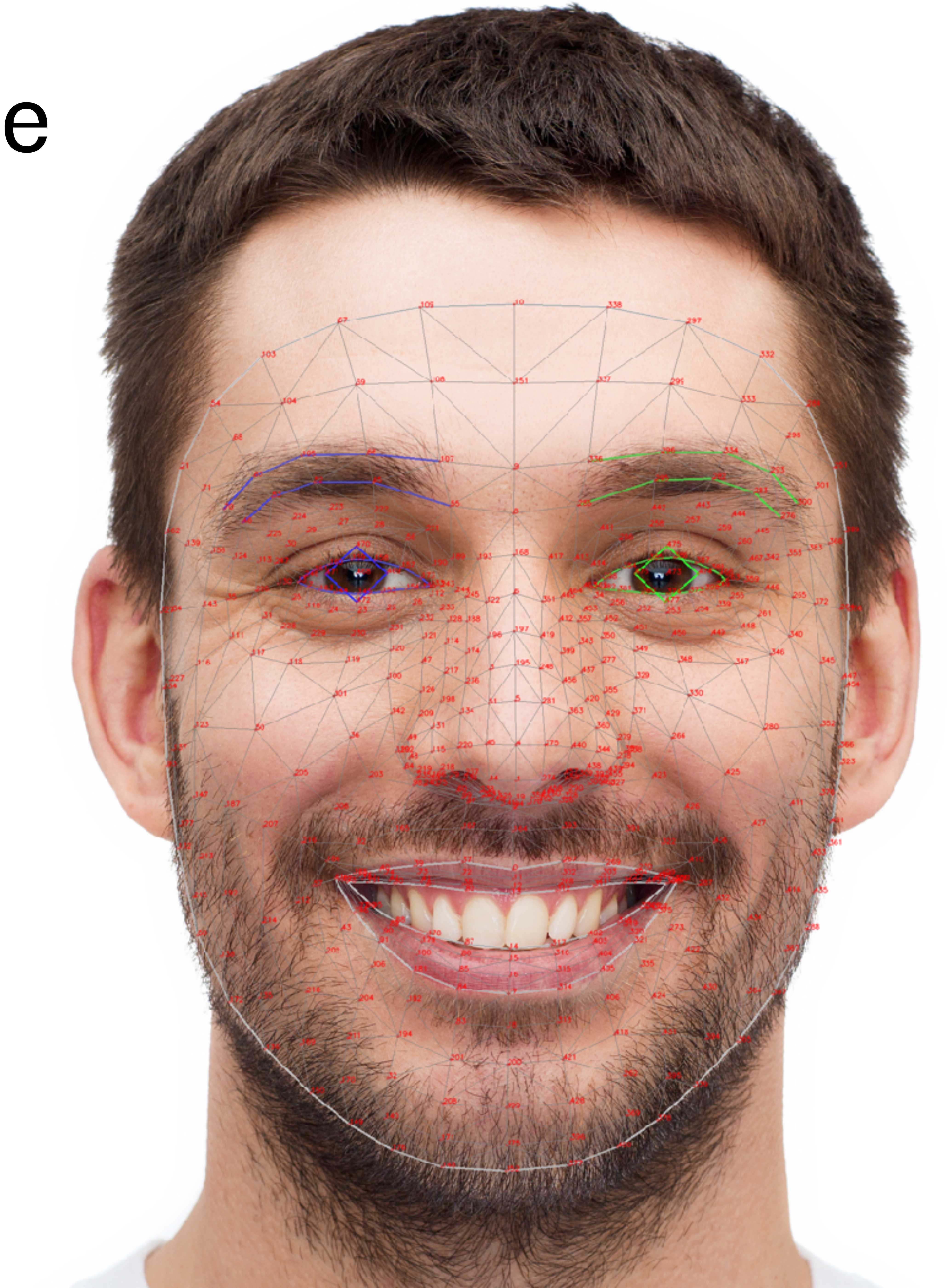
20 = right hand

19 = left hand

https://developers.google.com/static/mediapipe/images/solutions/pose_landmarks_index.png

© 2025

☻ face



4 = nose tip

13 = upper lip

14 = lower lip

310 = mouth left corner

78 = mouth right corner

473 = left eye

468 = right eye

https://storage.googleapis.com/mediapipe-assets/documentation/mediapipe_face_landmark_fullsize.png

© 2025

✎ Task (20 mins)

☞ copy example 02_follow_finger

☞ add new tracking points to fingers

☞ attach images / shapes / letters to fingers

😉 play

© 2025

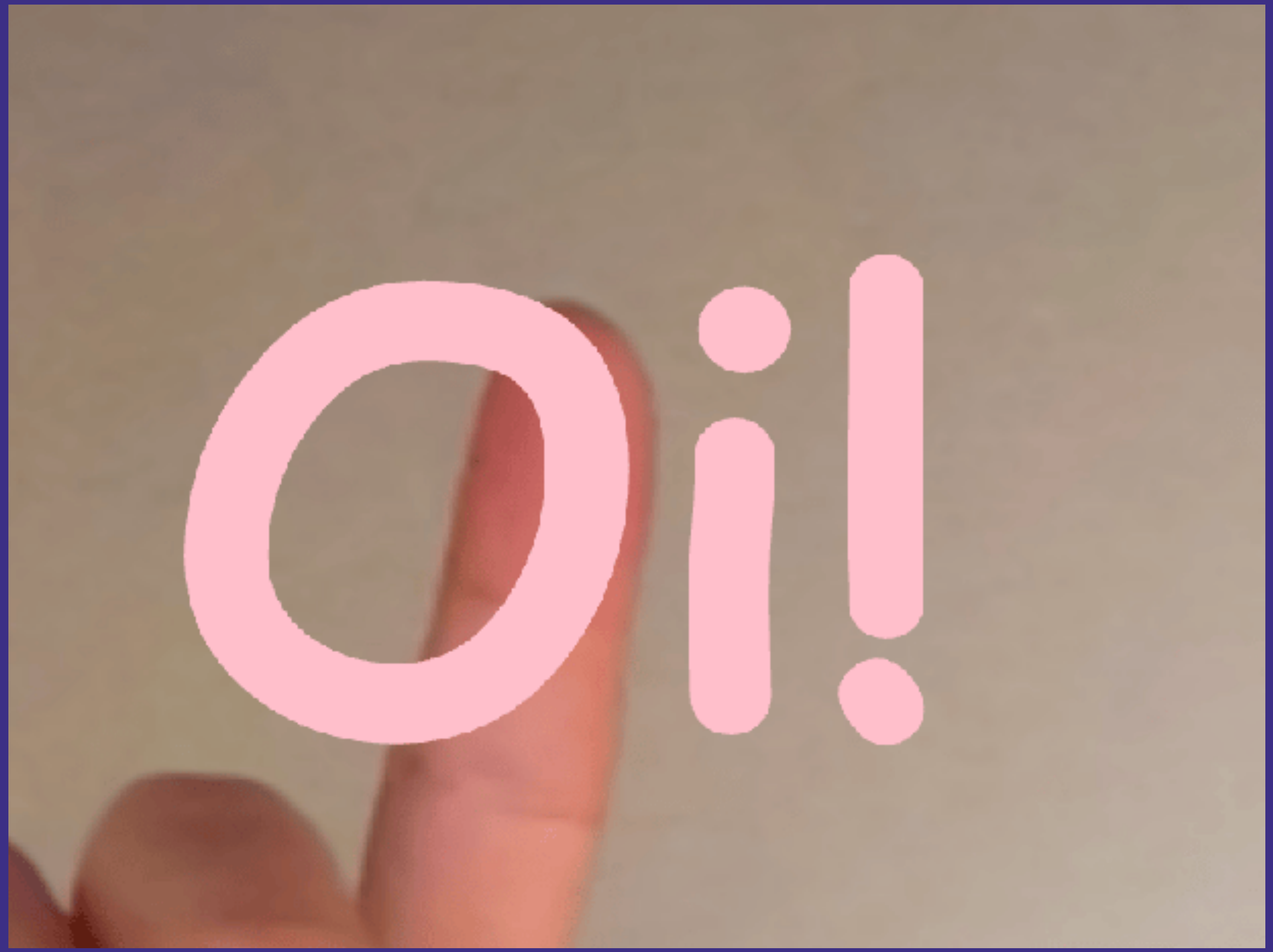
NG——VK

✌ input — introducing type

→ text function

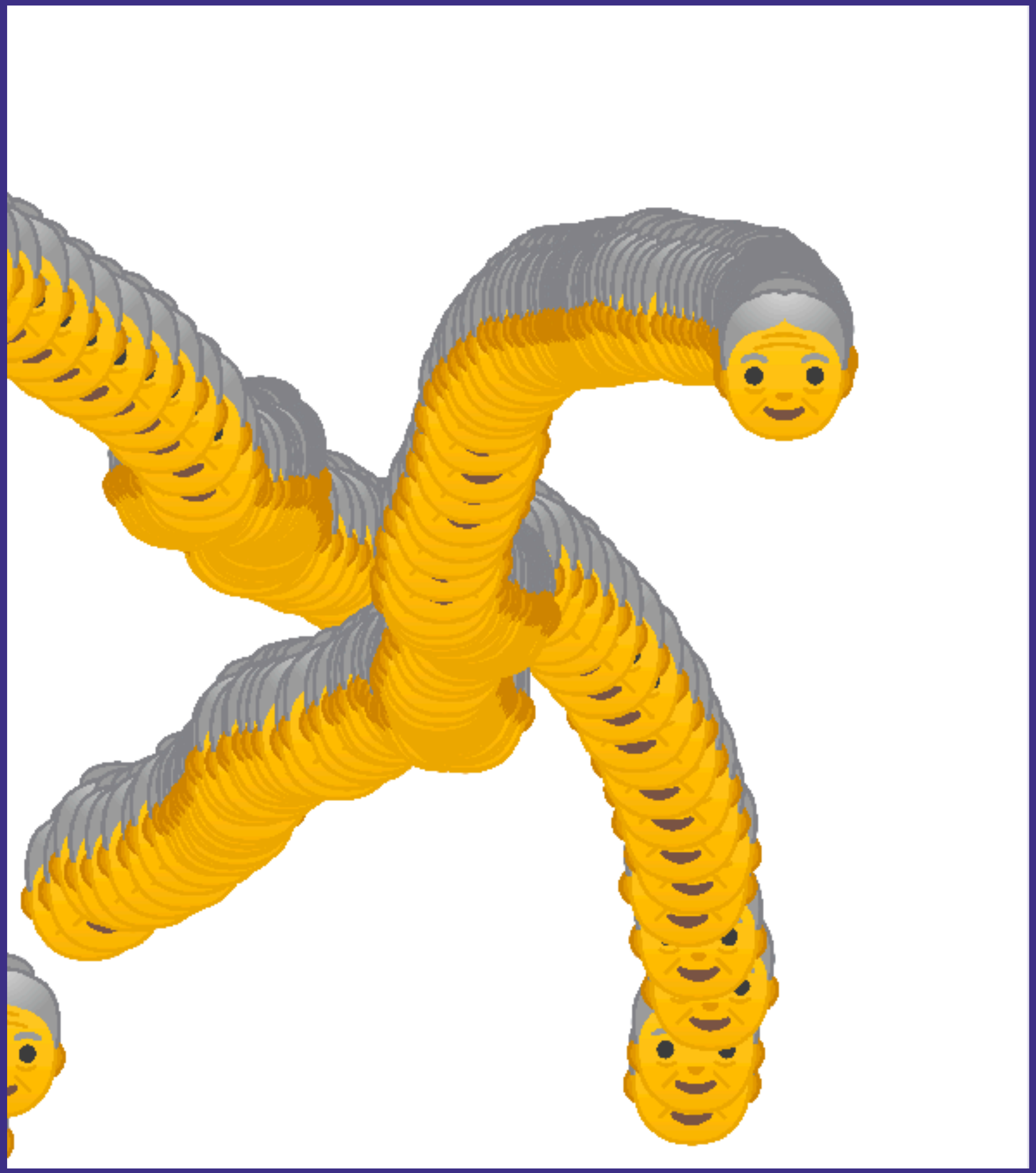
→ styling options

- font family
- size
- fill & stroke color



© 2025

✌️ interaction: emojie drawings

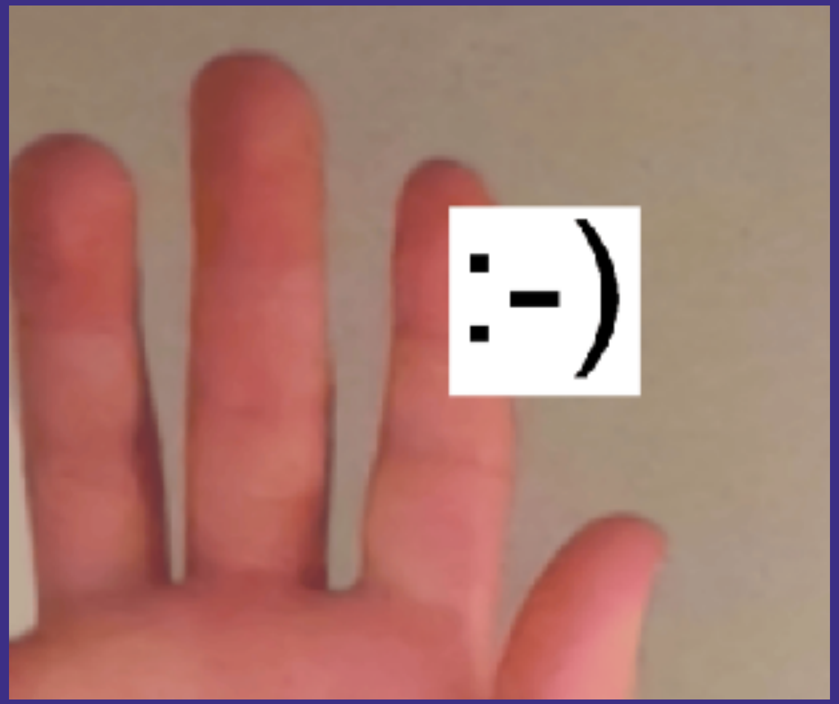


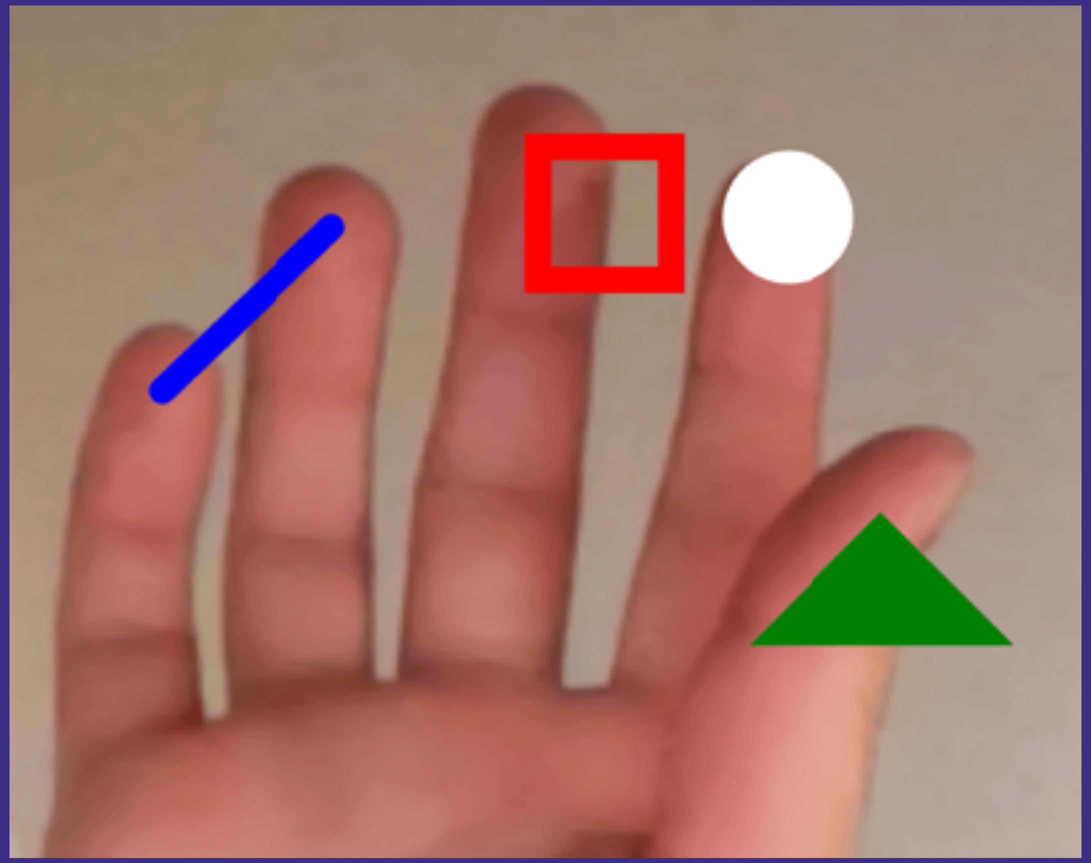
→ if we draw images without using background and webcam, we get a fun drawing app

© 2025

NG——VK

✌ input — expanding our toolbox





→ p5.js reference

→ our basic shapes

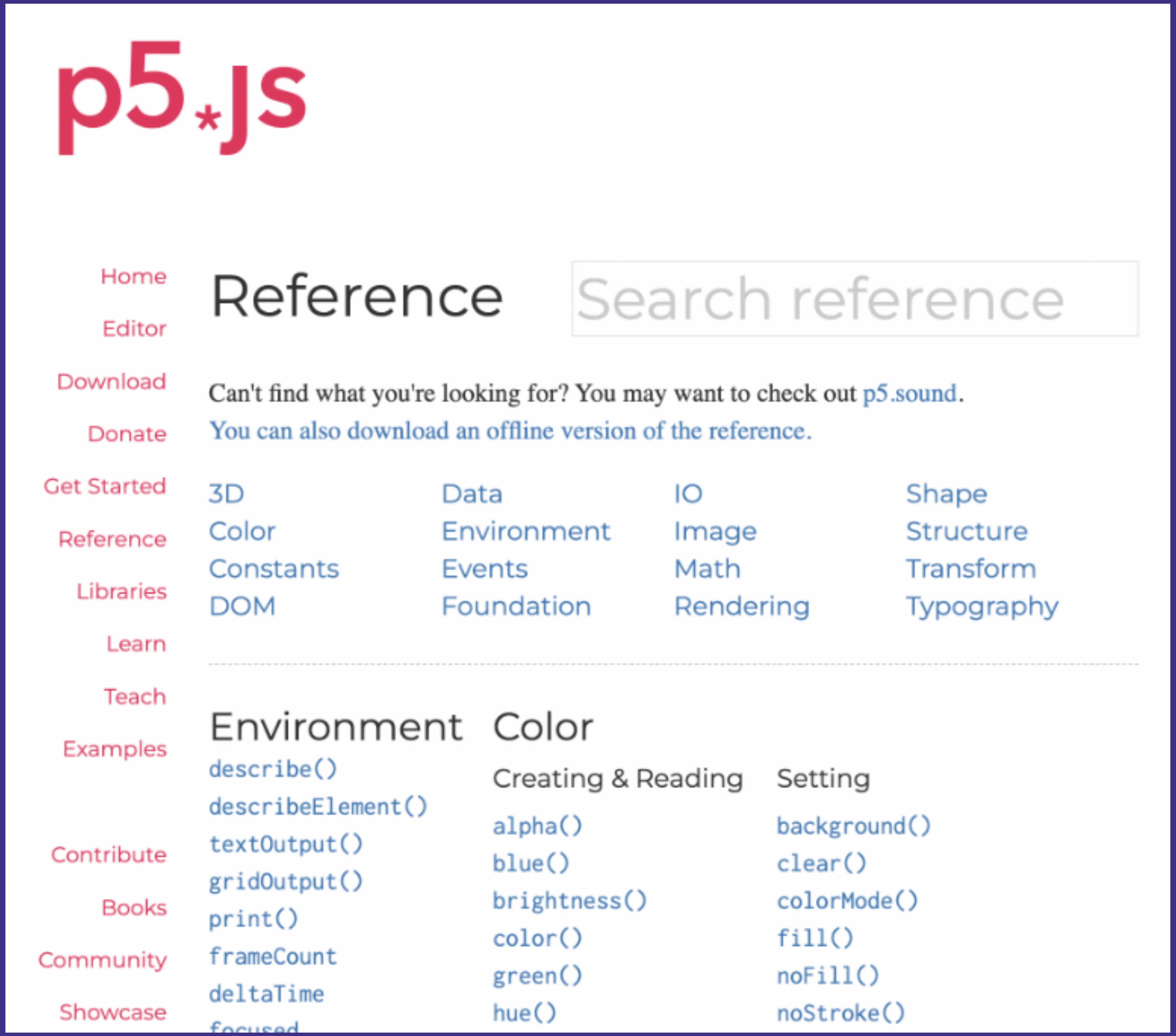
- ellipse
- rectangle
- triangle
- line
- text
- image

→ how to style them

- size
- fill & stroke color

© 2025

✌ p5.js reference



p5.js

- Home
- Editor
- Download
- Donate
- Get Started
- Reference
- Libraries
- Learn
- Teach
- Examples
- Contribute
- Books
- Community
- Showcase

Reference

Search reference

Can't find what you're looking for? You may want to check out p5.sound.
You can also download an offline version of the reference.

3D
Color
Constants
DOM
Data
Environment
Events
Foundation
IO
Image
Math
Rendering
Shape
Structure
Transform
Typography

Environment

describe()
describeElement()
textOutput()
gridOutput()
print()
frameCount
deltaTime
focused

Color

Creating & Reading

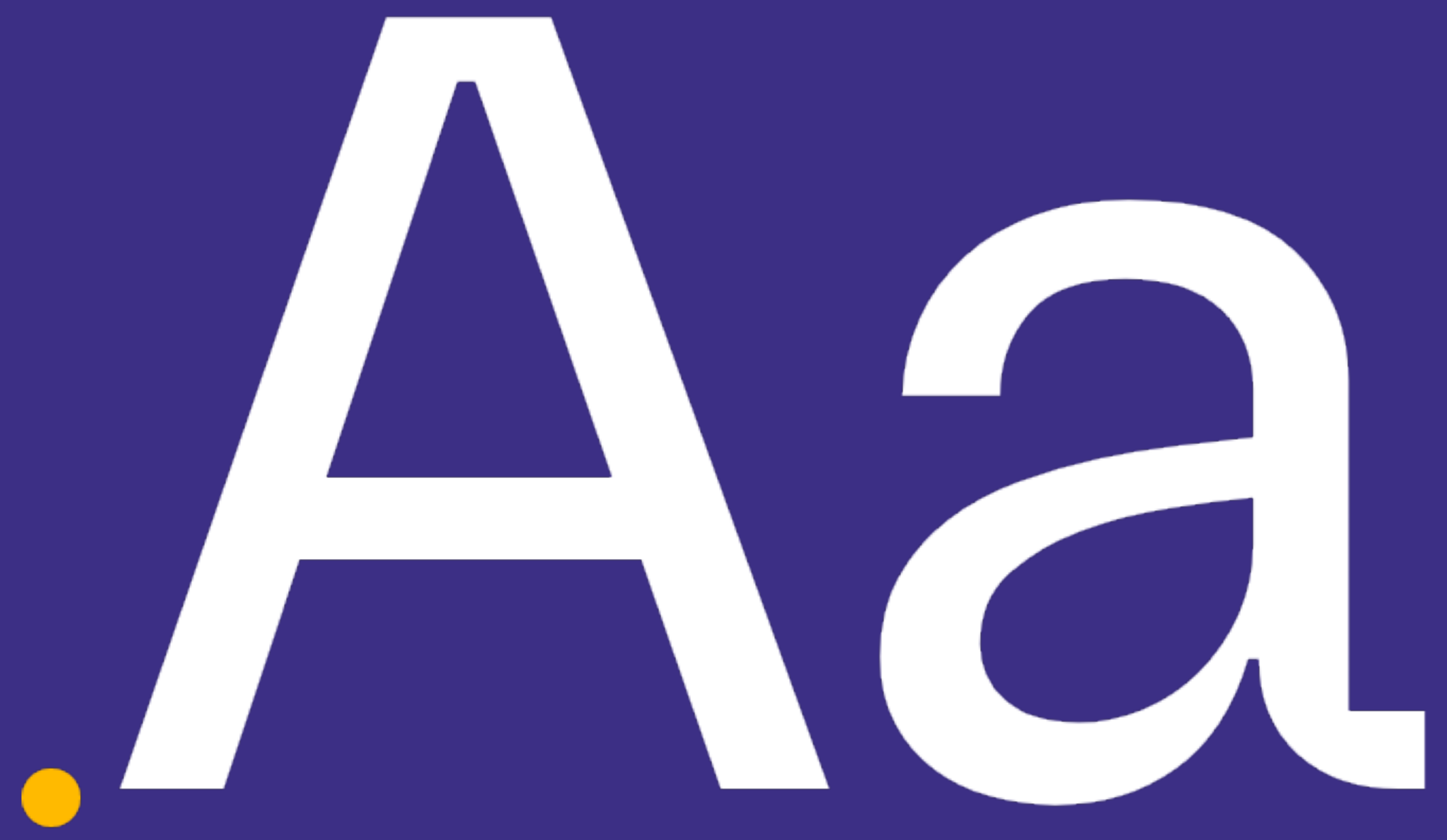
alpha()
blue()
brightness()
color()
green()
hue()

Setting

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

https://p5js.org/reference/

✌ text function



```
text("Aa", x, y);
```

© 2025

✌ text styling

→ size

```
textSize(20);
```

→ alignment (x, y)

```
textAlign(CENTER, CENTER);
textAlign(LEFT, TOP);
textAlign(RIGHT, BOTTOM);
```

→ fill

```
noFill();
fill("#FFBA00");
```

→ stroke

```
noStroke();
noFill();
stroke("#FFBA00");
strokeWeight(20);
```

© 2025

font family

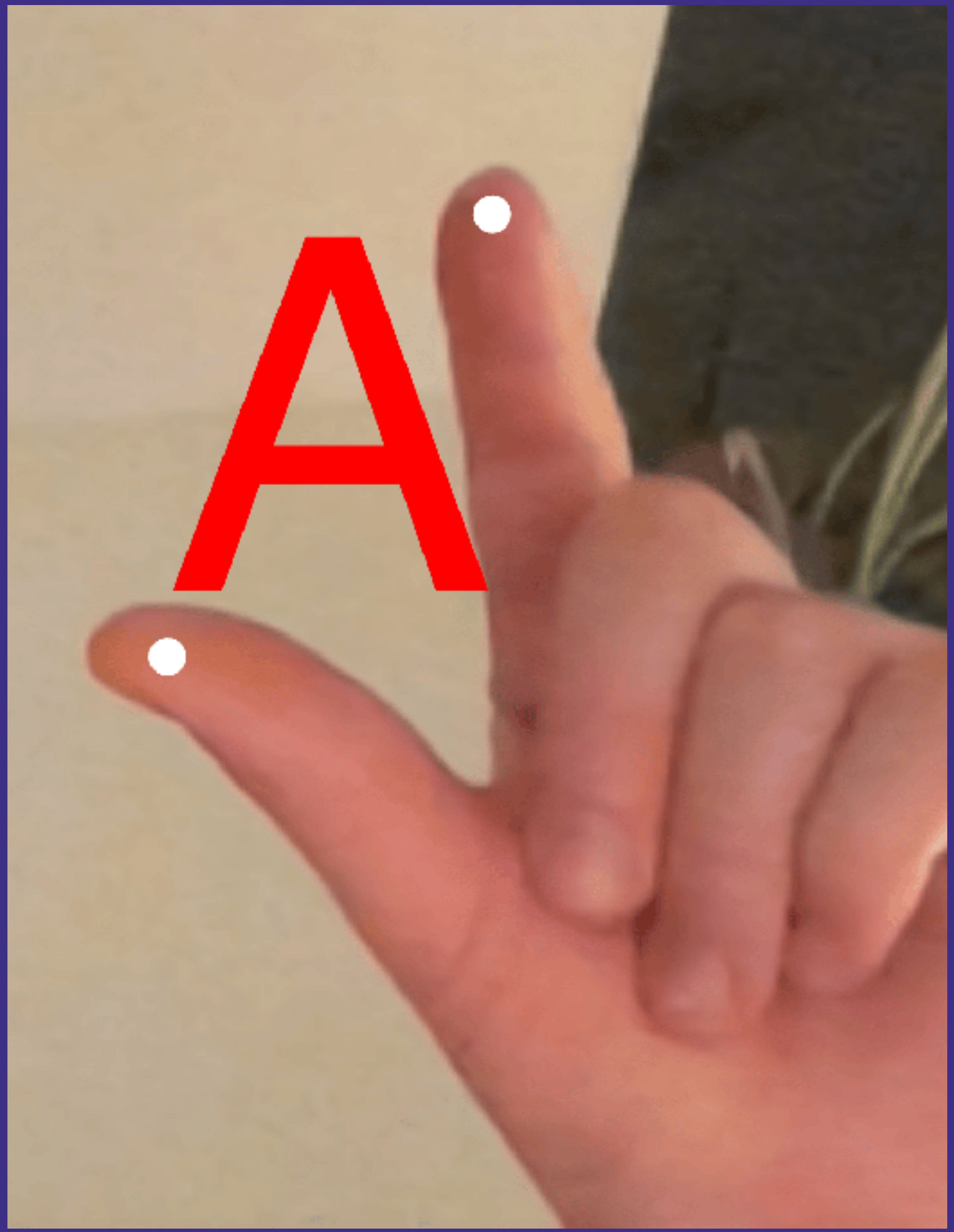
- → save your font to your sketch folder
- → create a variable to hold the font
- → preload() load the font from your folder
- → textFont() set the font before drawing text

```
let myFont;

function preload(){
  myFont = loadFont("fonts/my-nice-font.ttf");
}

function setup() {
  createCanvas(900, 900);
  textFont(myFont);
}

function draw() {
  background(255);

  text("Neat text in nice font", 100, 0, 900, 900);
}
```

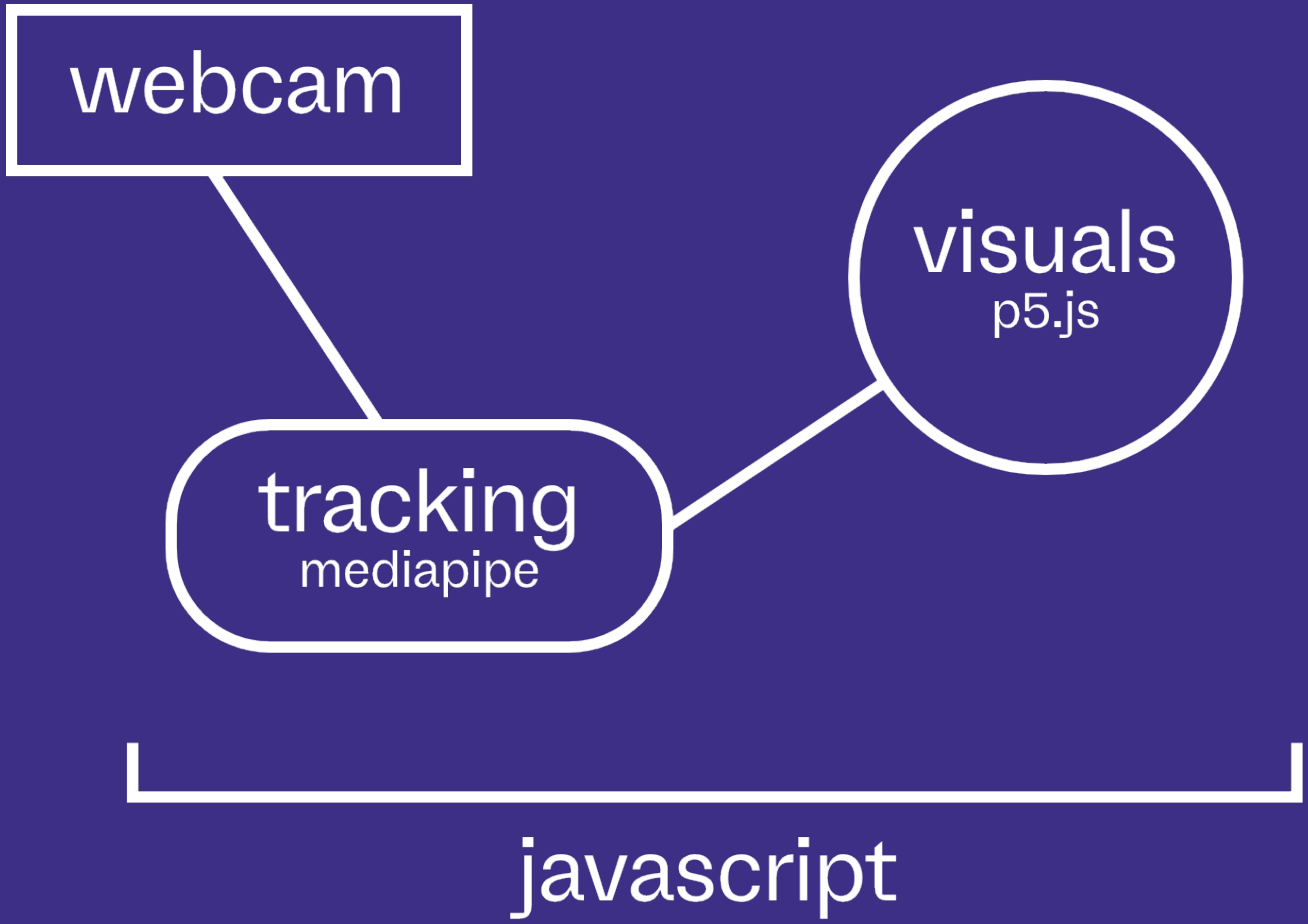
© 2025

✌ input — interaction

→ we can measure the distance between two points to scale things



© 2025



webcam

tracking
mediapipe

visuals
p5.js

javascript

© 2025