DIGITAL SKILLS & CODING

CODING FOR NON-CODERS

Annemieke Frank

Dr. Joachim Krois

29.05.2019 & 13.06.2019

YOUR TRAINERS

Fokus: Coding, Predictive Modelling, Spatial Data Analytics, Artificial Intelligence and Computer Vision

Referenzen: Freie Universität Berlin, Charité – Universitätsmedizin Berlin, Berliner Institut für Gesundheitsforschung

Fokus: Digital Competencies, Design Thinking, App Prototyping, Gamification

Referenzen: Volkswagen, Bentley, Deutsche Bahn, Facebook, eduvation, Technologie Stiftung

HELLO WORLD

- Tell us about yourself.
- Which software is most relevant in your day to day life?
- Have you ever coded?
- What are your expectations for today?

1min

LET'S CONNECT

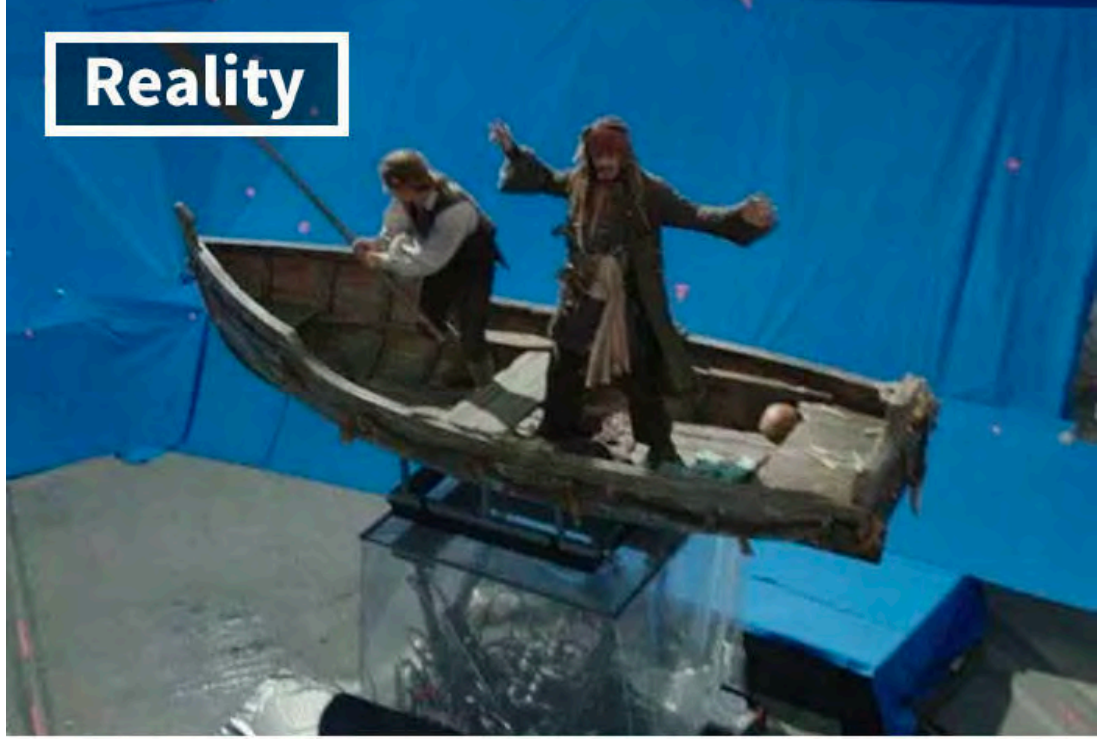
https://etherpad.hello-world.academy/p/beiersdorf

// AGENDA – 2019/05/29

1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it


JURASSIC PARK
™


Reality


Reality

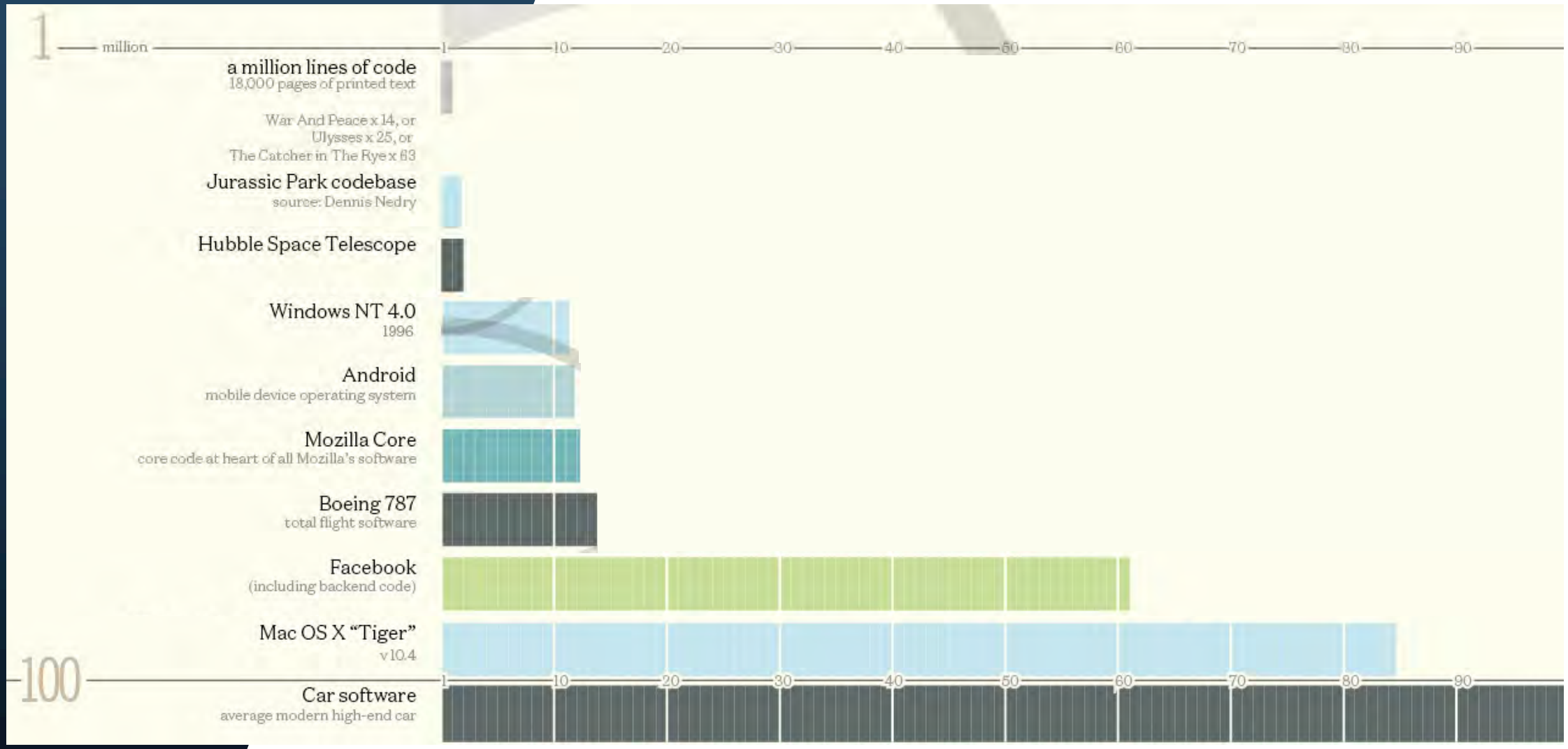

Movie


Movie


Hamburg
Donnerstag
Vereinzelte Regenschauer
17 °C | °F
Niederschlag: 30%
Luftfeuchte: 66%
Wind: 24 km/h
Temperatur
Niederschlag
Wind
9 8 9 13 16 17 17 15 13
01:00 04:00 07:00 10:00 13:00 16:00 19:00 22:00
Mo. Di. Mi. Do. Fr. Sa. So. Mo.
18° 8° 16° 4° 16° 8° 17° 9° 18° 9° 22° 11° 25° 13° 24° 12°
weather.com
Feedback geben


60
5
55
10
50
21:16
15
45
20
12
40
25
35
6

INTRODUCTION



// AGENDA

1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it

COMPUTATIONAL THINKING

"Software is eating the world"

Marc Andreessen
Developer & Investor

COMPUTATIONAL THINKING

HELLO WORLD

"Everybody should learn to program a computer, because it teaches you how to think"

Steve Jobs
former CEO Apple

COMPUTATIONAL THINKING

"Programming languages should be part of the curriculum. They are at least as important as multiplying, reading and foreign languages."

Timothy Höttges
CEO Deutsche Telekom

COMPUTATIONAL THINKING

Part 1

1. Give the robot commands so that 3 same-sized pieces of baguette are cut.
2. Write down each command on a post-it.
3. Sort post-its in the right order.

COMPUTATIONAL THINKING

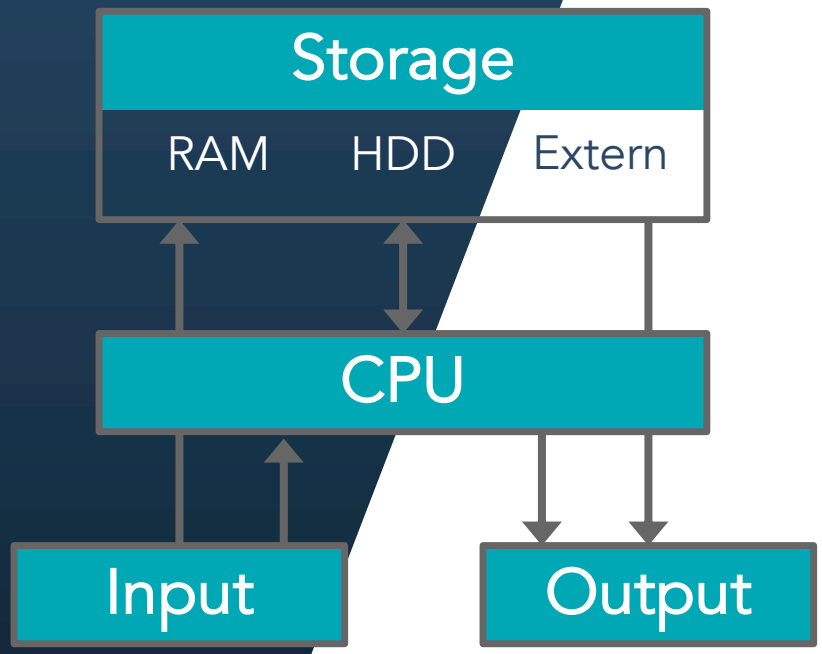
Part 2

1. Give the robot commands so that 3 same-sized pieces of baguette are cut.
2. Write down each command on a post-it.
3. Sort post-its in the right order.
4. Create a mockup with (a) the type of bread, (b) the number of pieces & (c) the thickness of the slices.

// AGENDA

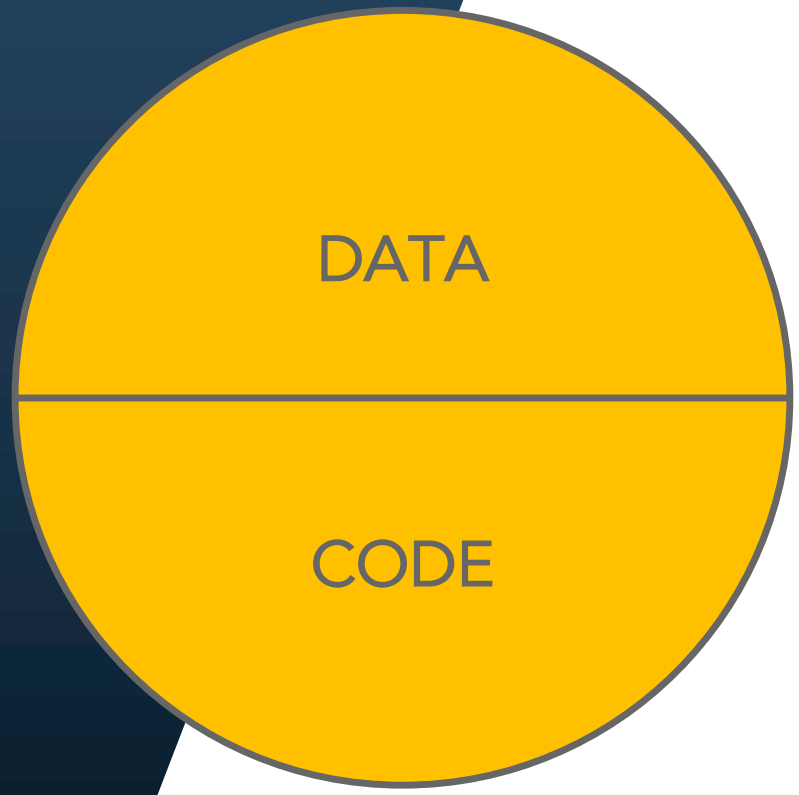
1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it

HARDWARE



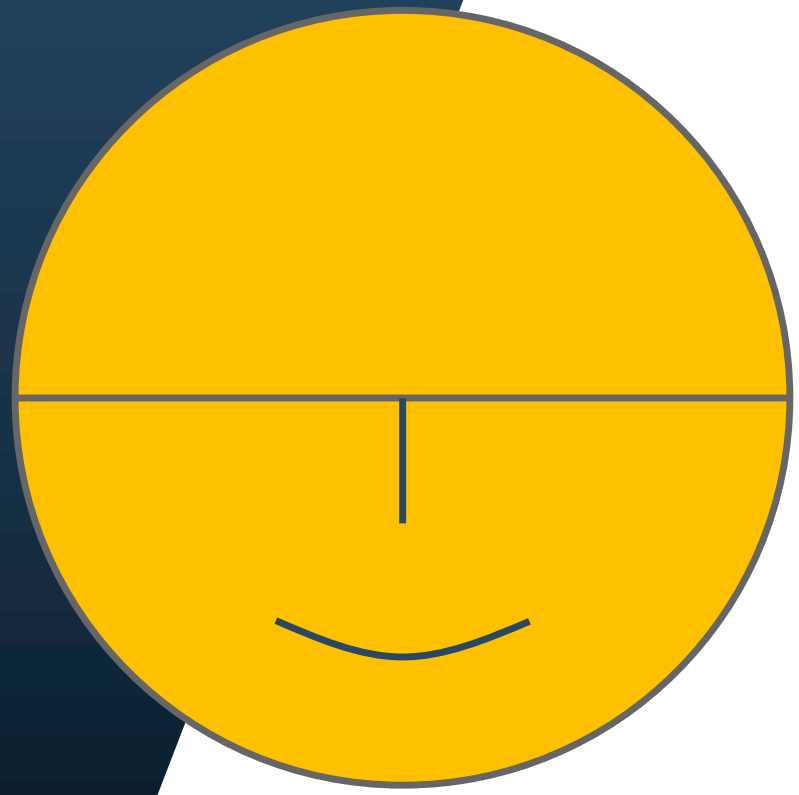
- Hardware = physical components of a computer
- The CPU (Central Processing Unit, prozessor) executes software
- Storage saves data
- Input: Mouse, keyboard, touchscreen …
- Output: Monitor, sound, vibration …
- *Analogy: Hardware = **body***

SOFTWARE



- Software executes commands on hardware
- Data is stored information
- Code is text-based commands

FUNCTIONS

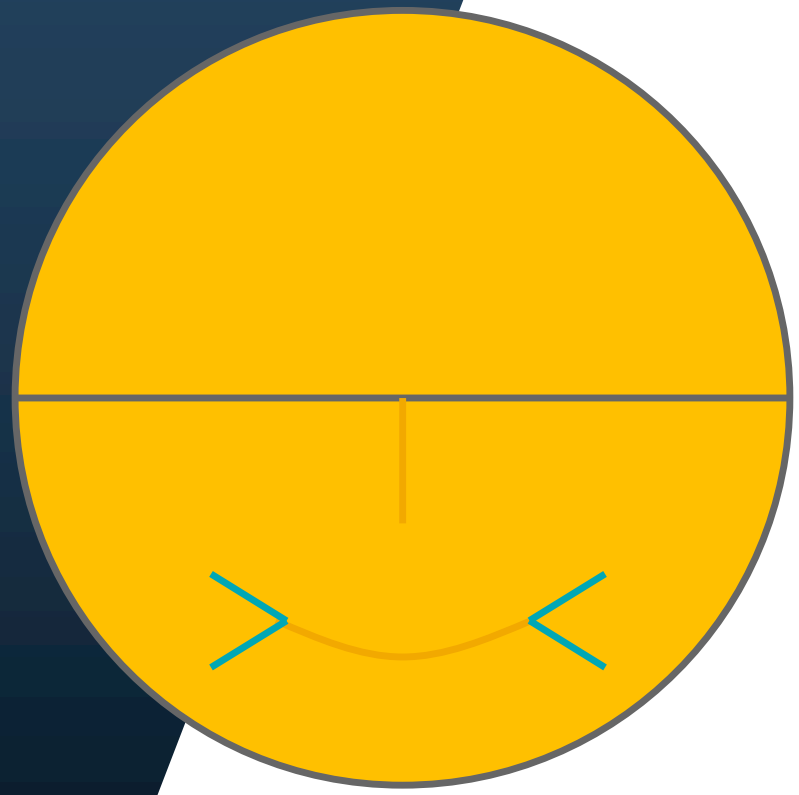


Function

- Reusable collection of commands

```
function roll (time, speed, dir) {
   setDirection(dir);
   setSpeed(speed);
}

function order(type, client) {
   var pizza = bake(type);
   deliver(pizza, client);

   …
}
```

FLOW CONTROL

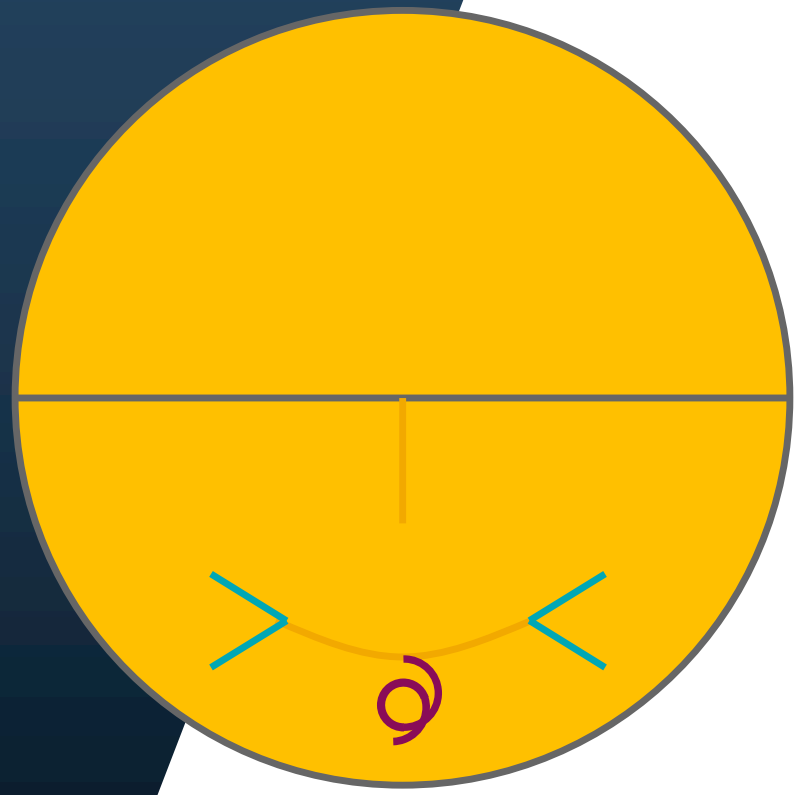


if-else statement

- Commands are exectuted when conditions are met.

```
if (colour == green) {
   Sound(Boing);
}

if (Waiter.ready == Yes) {
   order();
}
```

LOOP

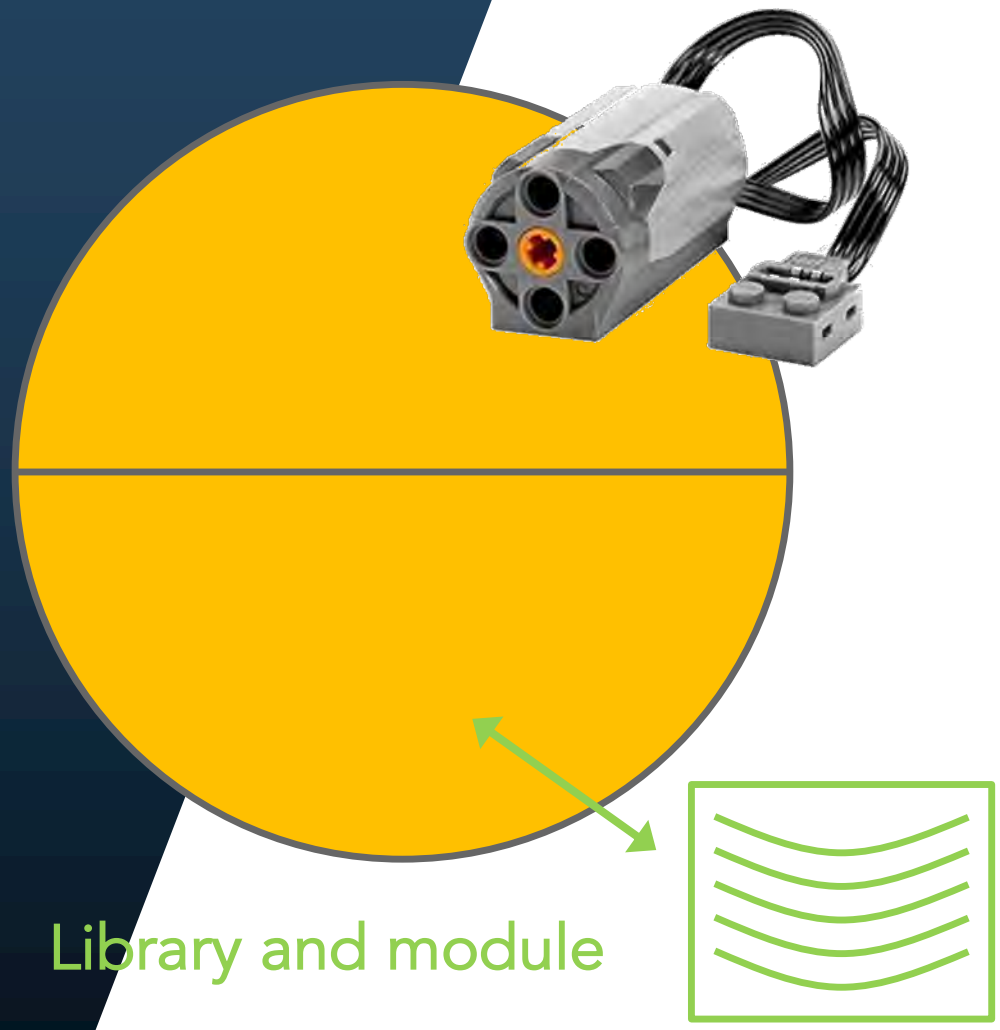


while loop

- Commands are repeated as long as condition is met

```
Var speed = 5;

while (speed <= 100) {
   speed = speed + 5;

   role (time, speed, dir);

}
while (orders < hunger) {
   order();

}
```

LIBRARY/MODULES

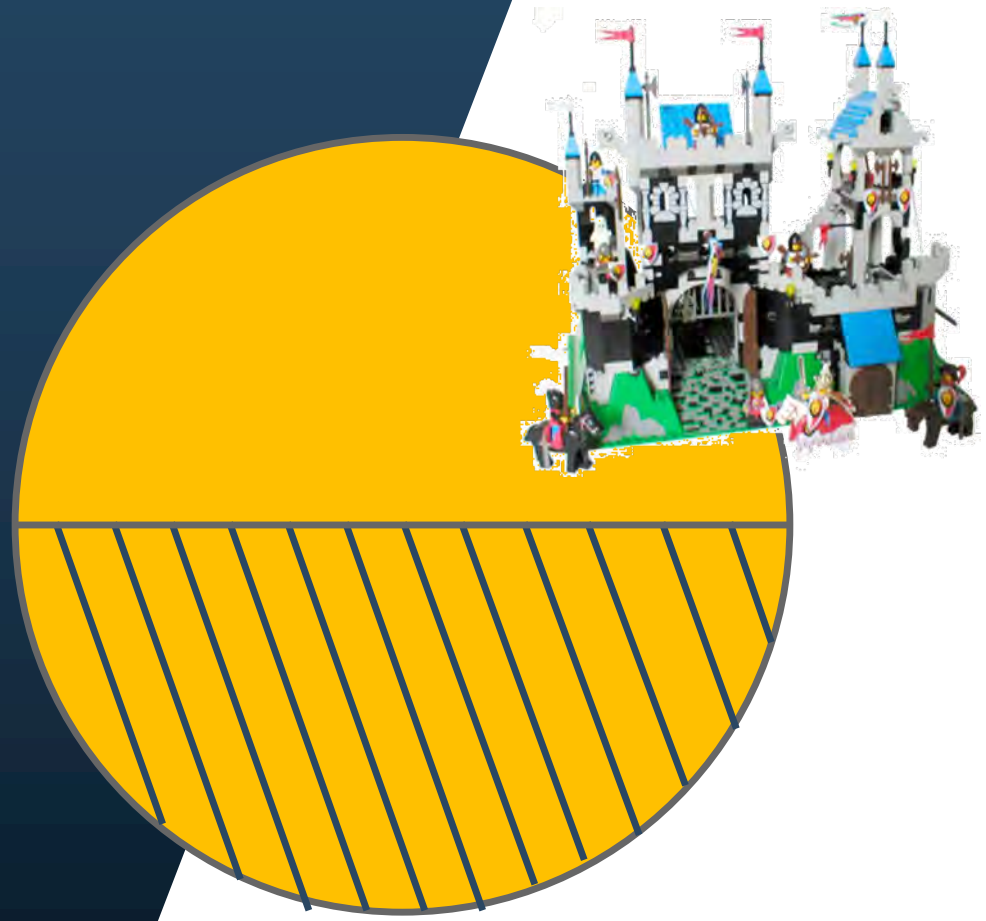


- Collection of functions and commands

…

Libraries are dependent on language- and application …

FRAMEWORK



Framework

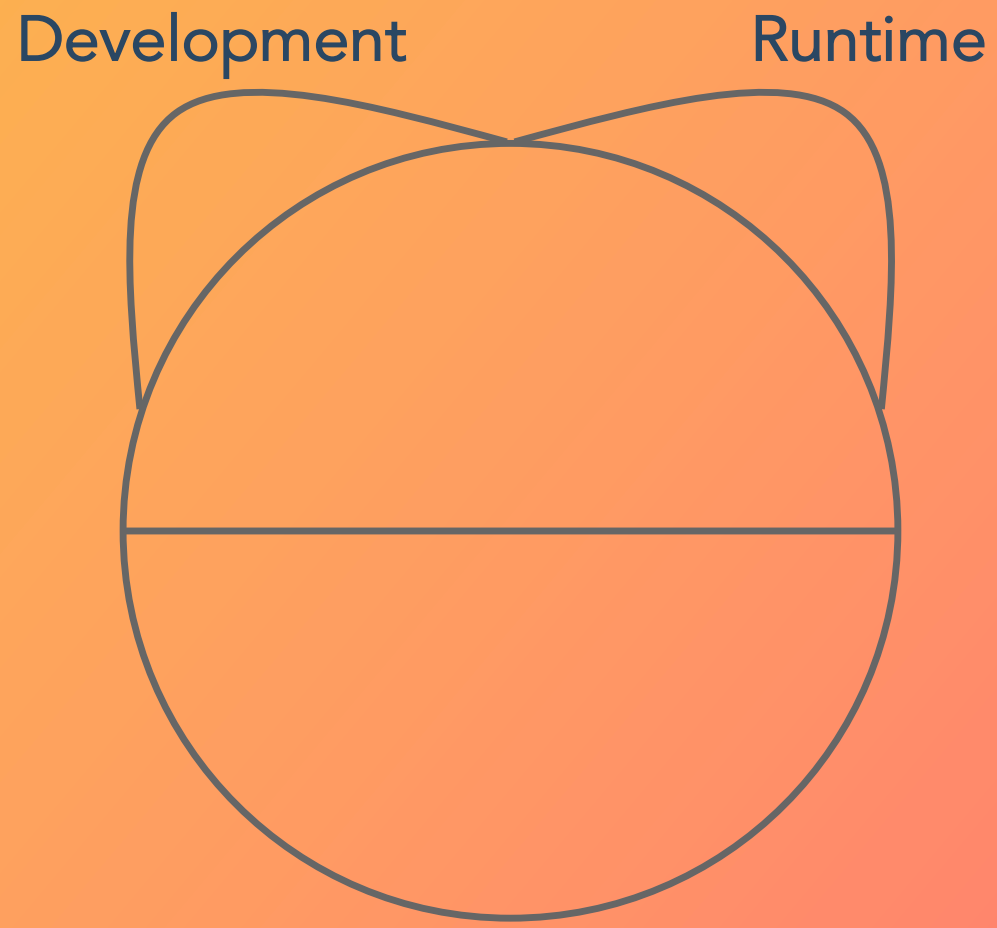
- Collection of functions, commands and rules

…

Framework are dependent on language- and application …

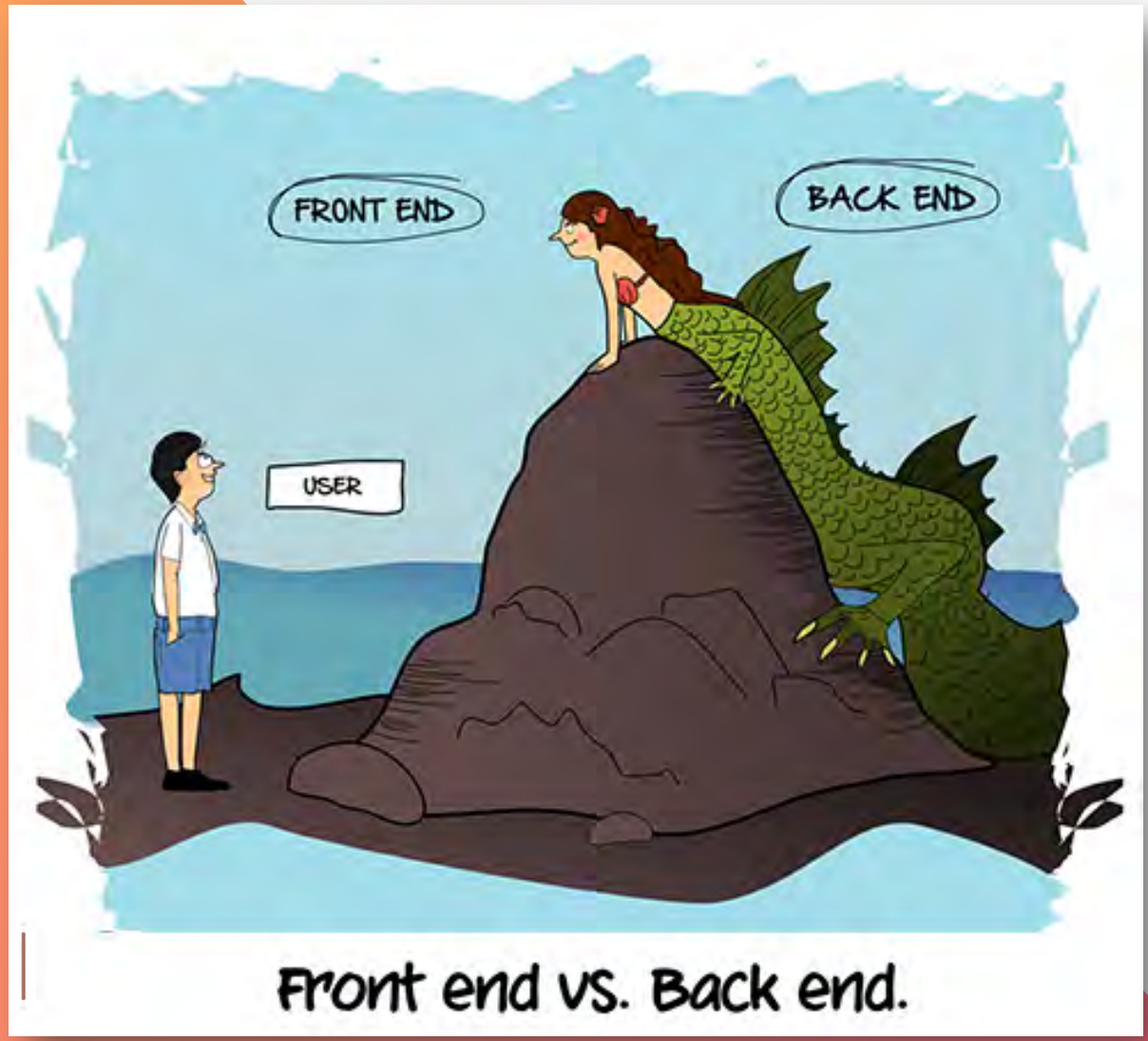
…

SOFTWARE PHASIS



- The software is programmed during the development phase (programmer's task).
- The software is executed at runtime (user task).
- Analogy: Car during manufacturing & during life cycle.

FRONTEND
& BACKEND



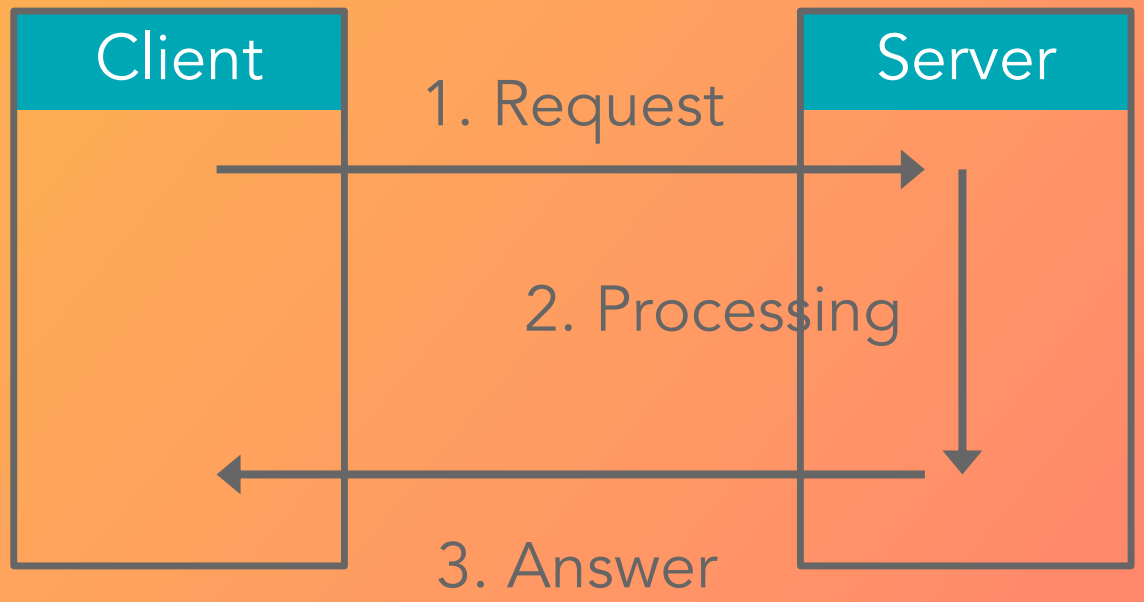
FRONTEND & BACKEND

- Front-end and back-end describe layers of IT systems
- Front-end is closer to the user and to data inputs, can contain logic
- Back-end is closer to data processing
- Pair of terms is context-related
- Analogy: mimic as front-end, thoughts as back-end

INTERFACE

- Interfaces provide defined functions to for accessing the IT system from an environment.
- Graphical interfaces (also GUI, Graphic User Interface) are used by humans.
- API (Application Programming Interfaces) used by other programs
- Analogy: Power cable in socket, stove-plate and pot and lid

CLIENT/SERVER



- Client-Server describes the distribution of tasks between two software systems.
- A physical device can be client and server at the same time.
- The term pair is context-related
- Analogy: Pizza customer is client, employee is server

// AGENDA

1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it

PROGRAMMING ROBOTS

HELLO WORLD

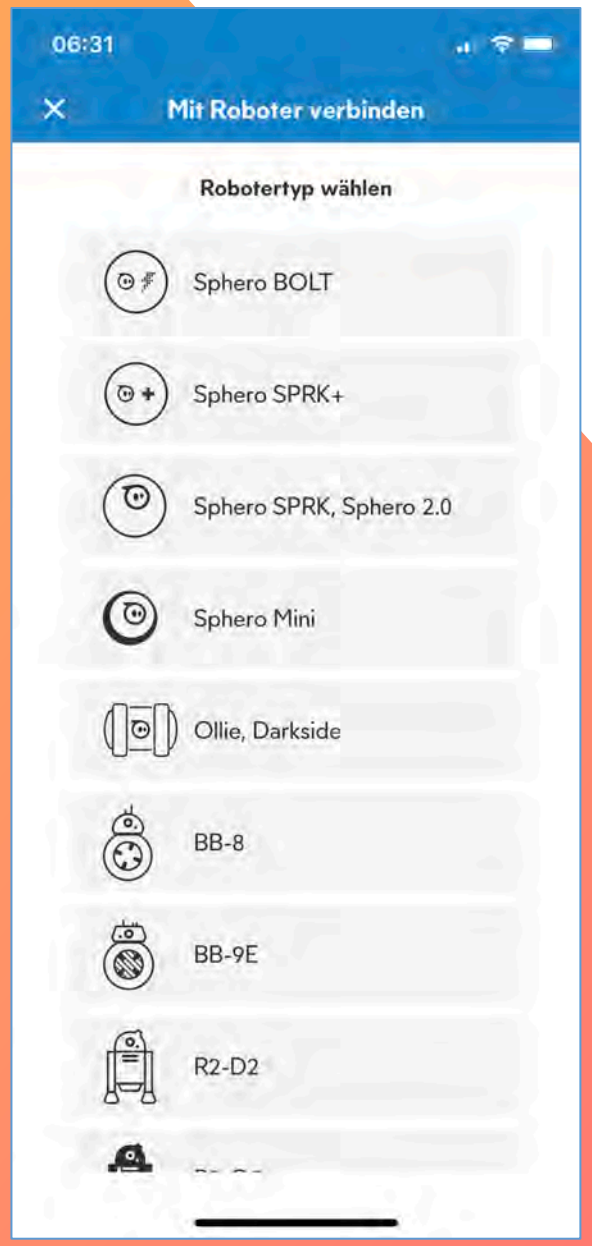
Sphero

BOLT

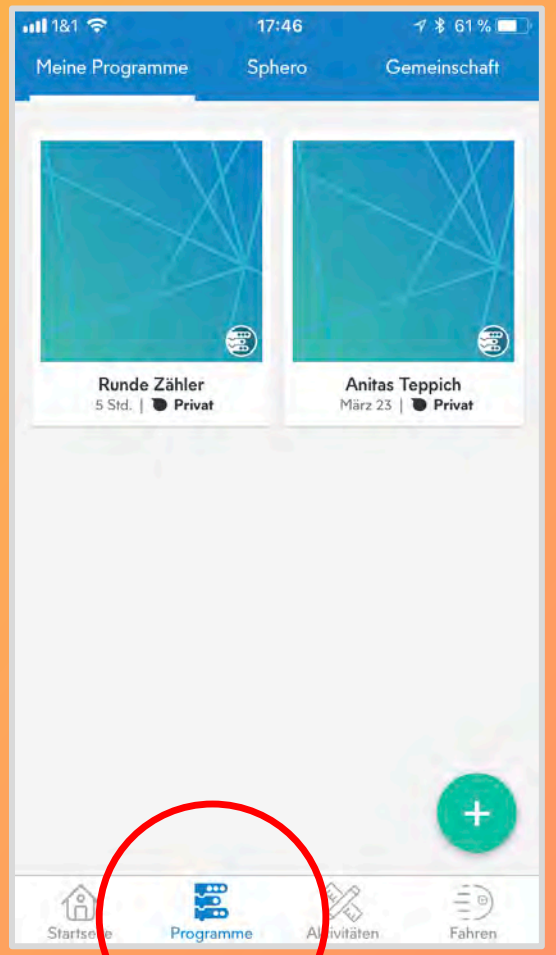
PROGRAMMING ROBOTS

Sphero

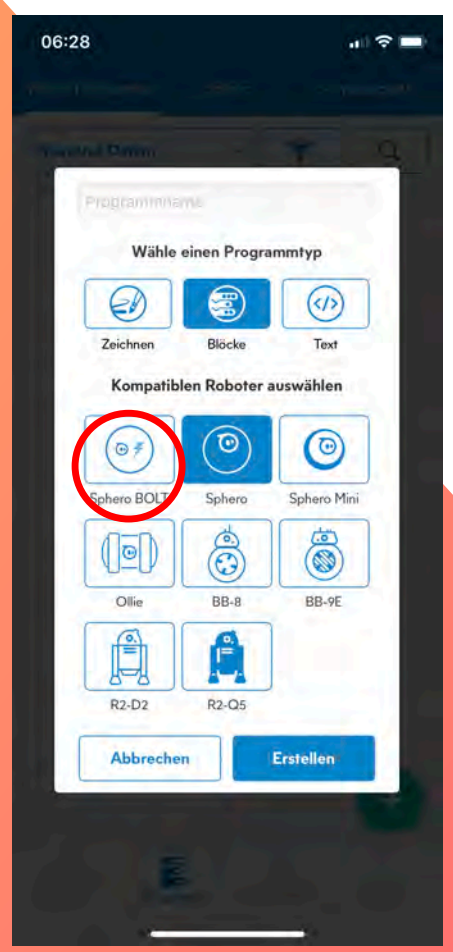
BOLT



PROGRAMMING ROBOTS

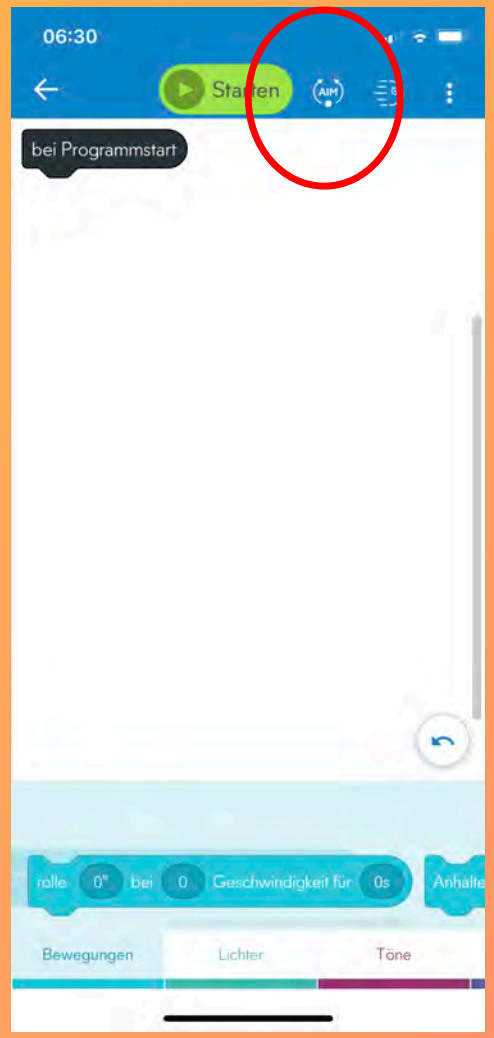


Click on tab Programes

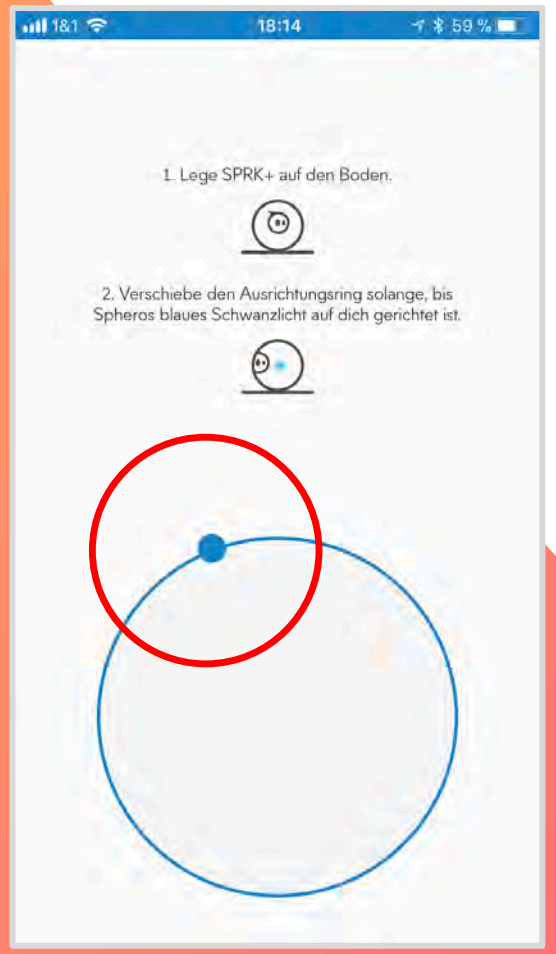


Click „+“ to create a new program and choose “Block“.

PROGRAMMING ROBOTS

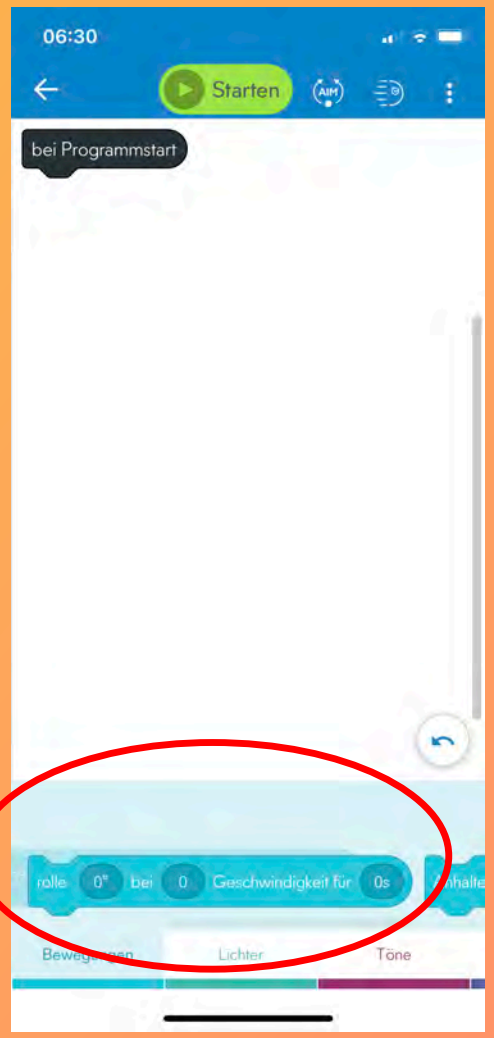


Click „Aim“ to callibrate robot.

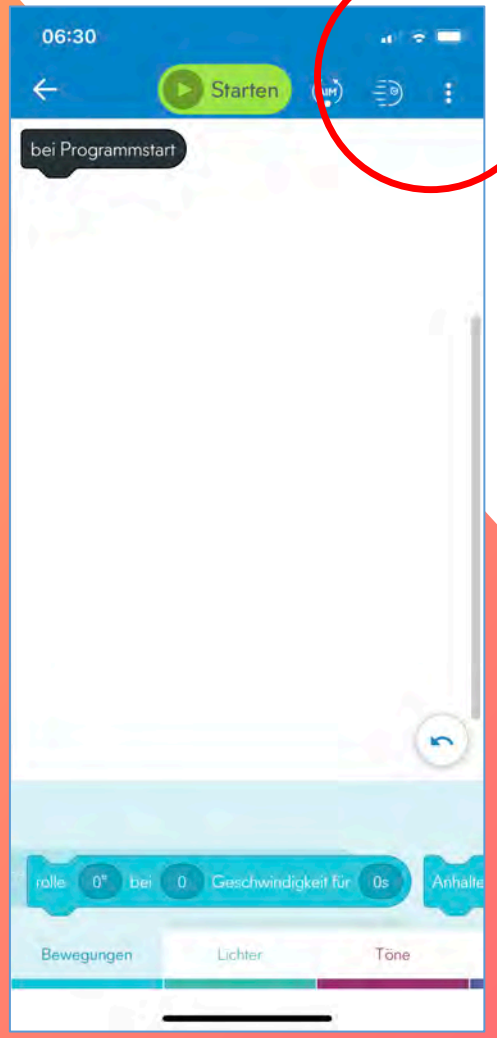


Rotate the blue dot on the smartphone so that your robots blue dot points to you.

PROGRAMMING ROBOTS



Drag & Drop commands into the interface.



Command Bolt in real time.

PROGRAMMING
ROBOTS

Parkour challenge

HELLO WORLD

PROGRAMMING ROBOTS

- Turn the light to green
- Drive along the course
- After the first curve, change the color to red.
- Play a sound after impact

Remember

- *App „Sphero Edu“*
- *Activate bluetooth*
- *Create program*
- *Calibrate via „AIM“*

// AGENDA

1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it

PROGRAMMING IS FEMALE

Ada Lovelace (1815 - 1852)

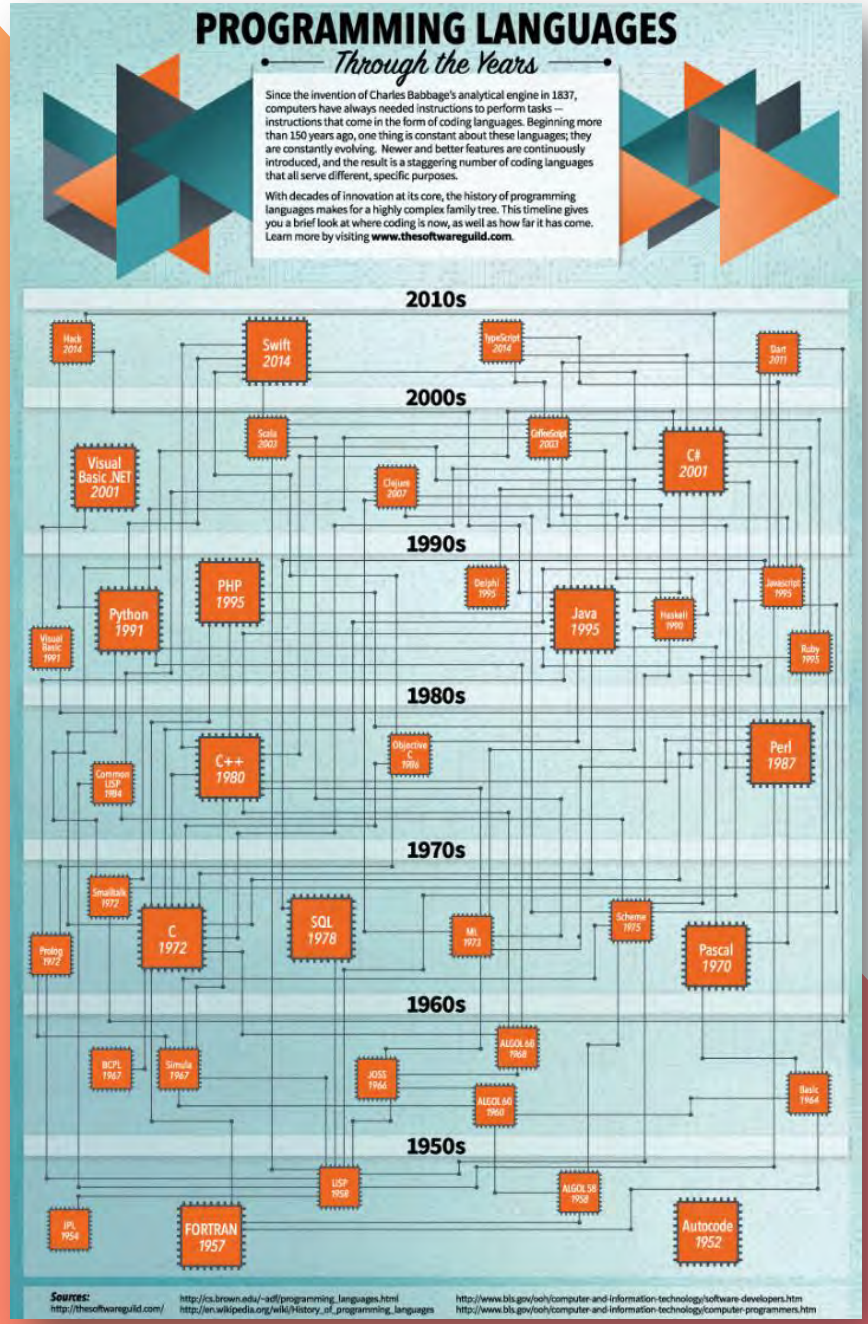
PROGRAMMING IS FEMALE

Grace Brewster Murray Hopper (1906 -1992)

PROGRAMMING IS FEMALE

Margaret Hamilton (1936 -)

PROGRAMMING LANGUAGES IN TIME



PROGRAMMING LANGUAGES
RUBY


```
MyLittleVar = "Hello World!"
5.times{puts MyLittleVar}
```

PROGRAMMING LANGUAGES
PYTHON

HELLO WORLD

```
MyLittleVar = "Hello World!"
for _ in range(5):
    print(MyLittleVar)
```

PROGRAMMING LANGUAGES
JAVASCRIPT

HELLO WORLD

```
var MyLittleVar = "Hello World!";
for (var x = 1; x <= 5; x++)
    {
    alert(MyLittleVar);
    };
```

PROGRAMMING LANGUAGES
JAVA

HELLO WORLD

```
class MyExample {
    public static void main(String[] args)
    {
        String MyLittleVar = "Hello World!";
        for (int x = 0; x < 5; x++)
        {
            System.out.println(MyLittleVar);
        }
    }
}
```

PROGRAMMING LANGUAGES
C

HELLO WORLD

```
#include<stdio.h>
int main() {
	char MyLittleVar[] = "Hello World\n";
	int x = 0;
	for (x = 0; x < 5; x++)
	{
		printf("%s", MyLittleVar);
	}

	return 0;
}
```

PROGRAMMING LANGUAGES ASSEMBLER

HELLO WORLD

```
section .data
    MyLittleVar  db 'Hello World!', 10
    length equ $ - MyLittleVar;

section .data
 start:
       mov cx, 5 ; fill cx-register with 5
 loop:
       mov eax, 4 ; write(stdout, hello, length)
       mov ebx, 1
       mov ecx, MyLittleVar
       mov edx, length
       int 80h

       loop schleife ; jump to ‘loop’ as long as cx > 0 and decrease cx by 1
       mov ebx, 0  ; Call: exit
       mov eax, 1
       int 80h
```

CODING LANGUAGES
MACHINE CODE

HELLO WORLD

```
01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010
01101100 01100100 0001010  01001000 01100101 01101100 01101100 01101111 00100000
01010111 01101111 01110010 01101100 01100100 0001010  01001000 01100101 01101100
01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100 0001010
01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010
01101100 01100100 0001010  01001000 01100101 01101100 01101100 01101111 00100000
01010111 01101111 01110010 01101100 01100100
```

=

Hello World

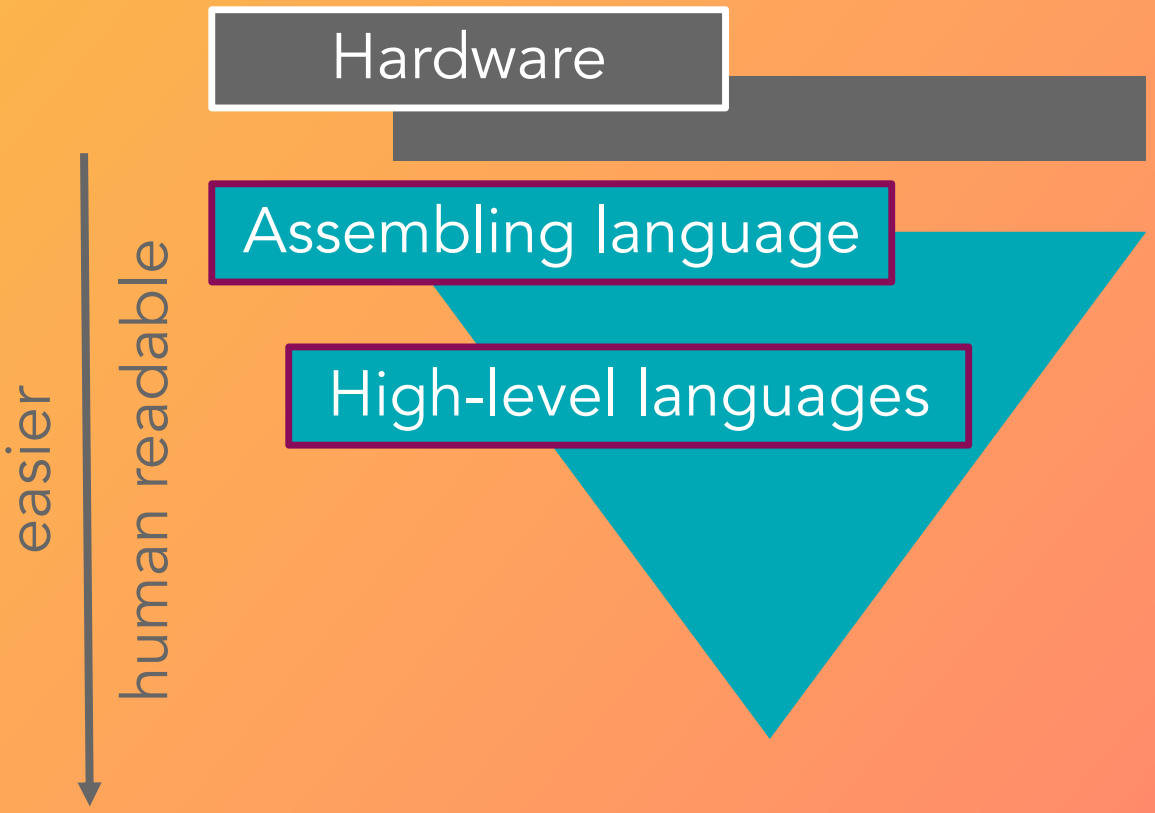
Hello World

Hello World

Hello World

Hello World

SOFTWARE ARCHITECTURE

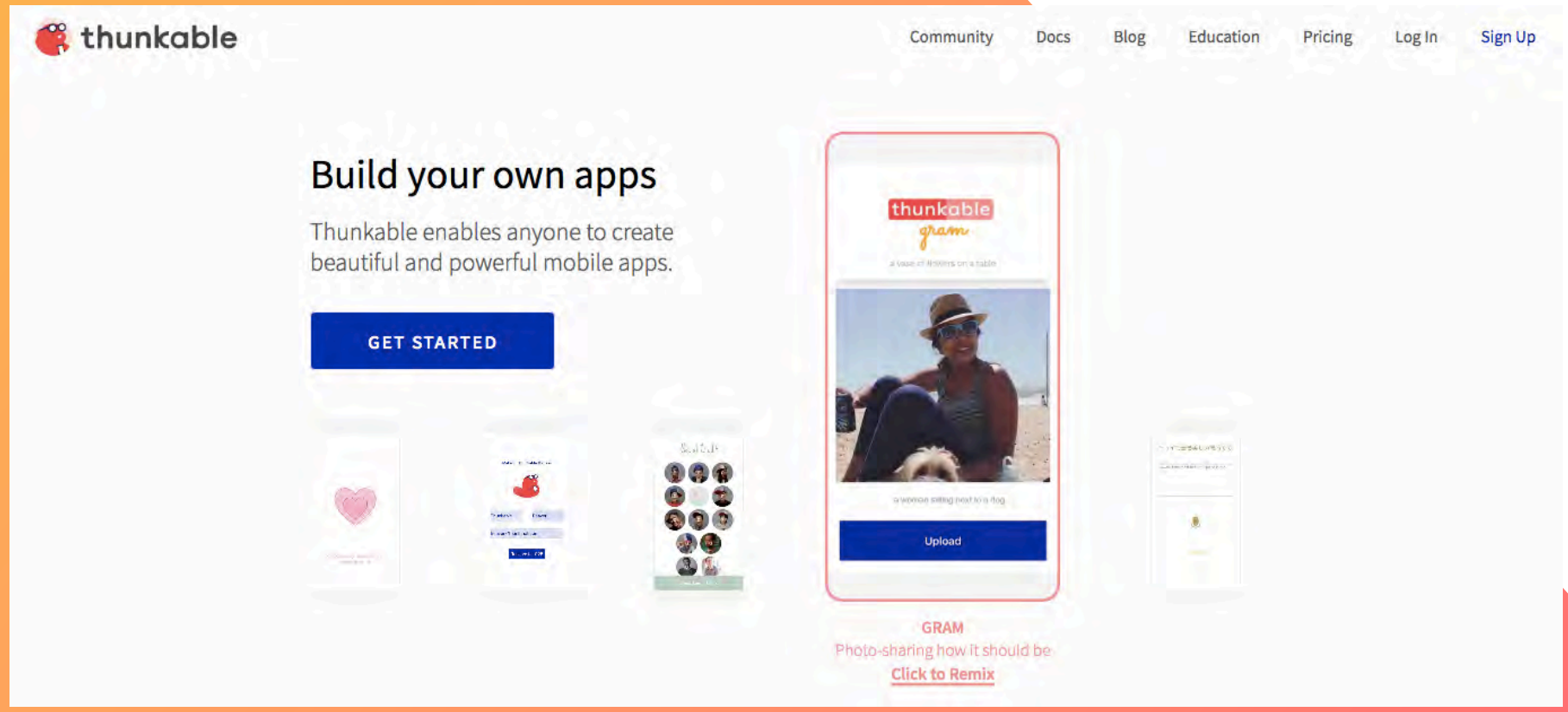


- Source code is written in high-level programming languages
- Assembler code is processed directly by the hardware, difficult to read
- High-level languages are translated into assembler language or are executed in runtime environments

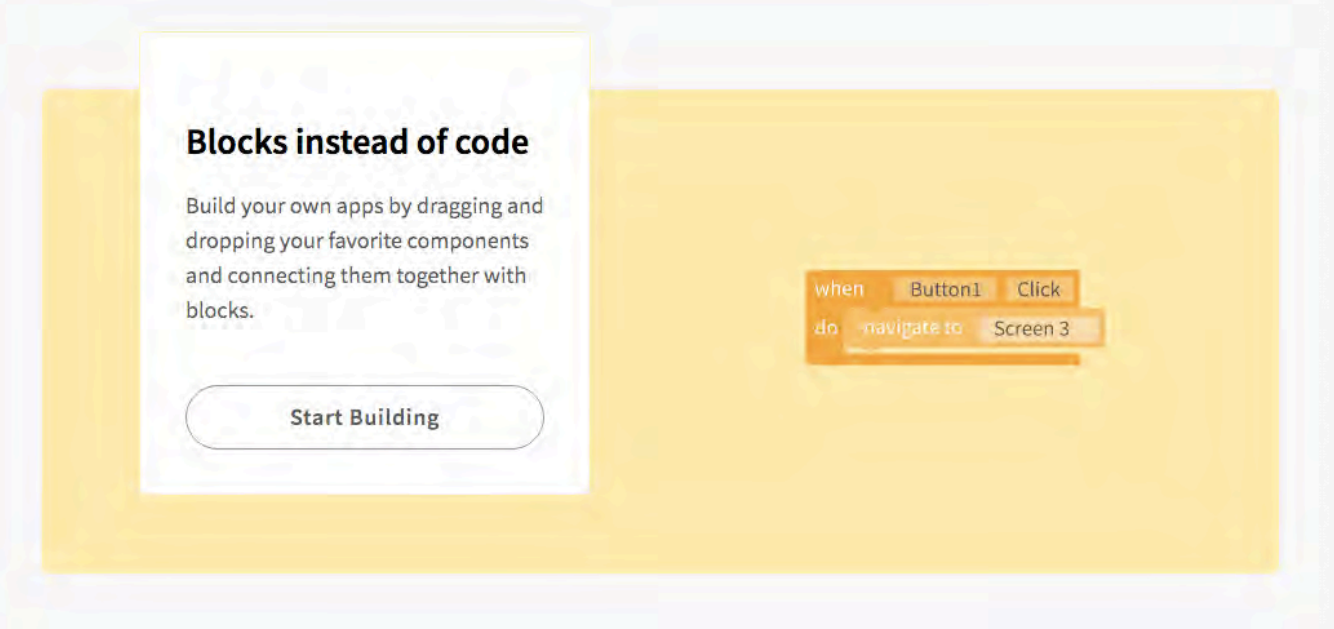
// AGENDA

1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it

APP PROTOTYPING

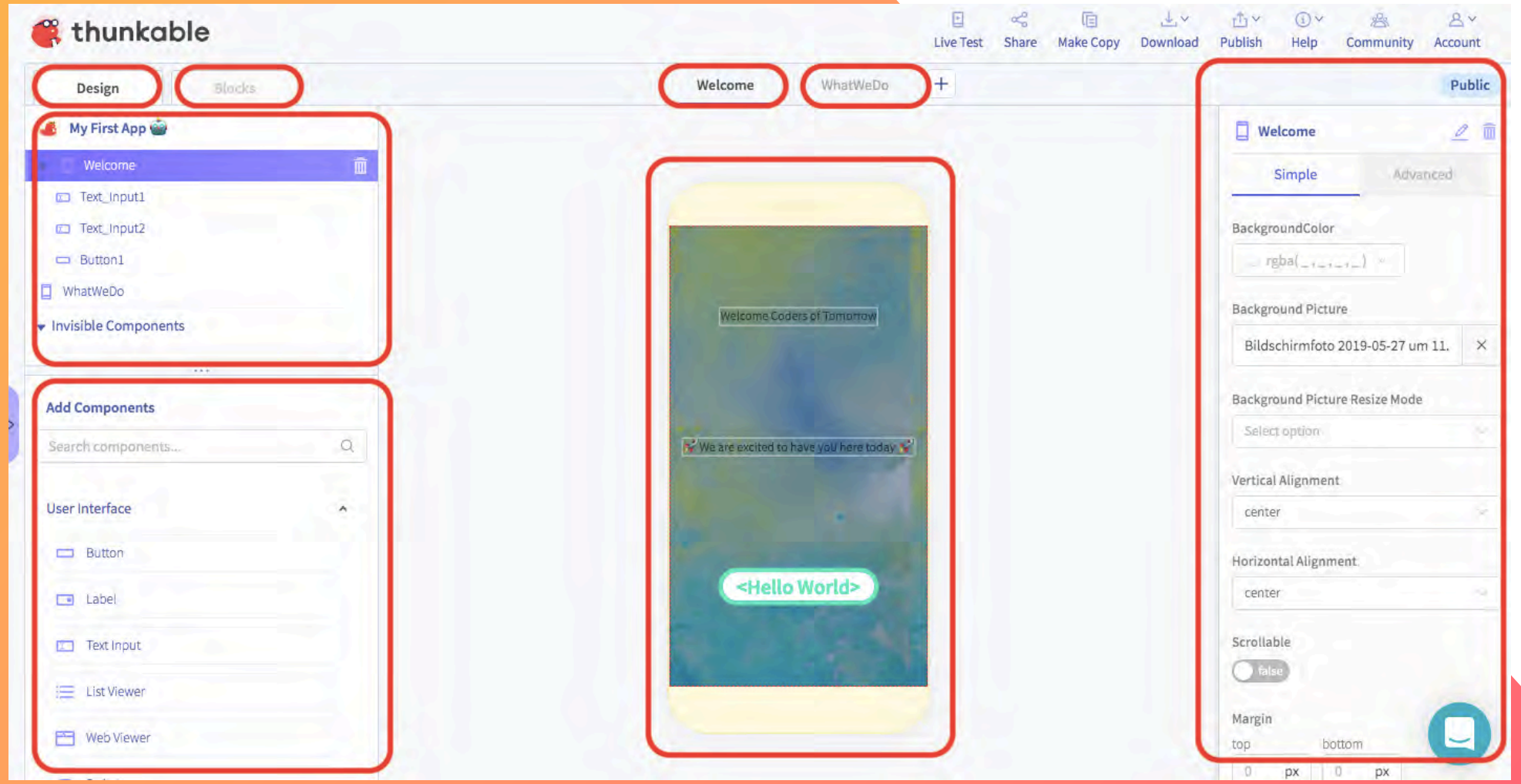


APP PROTOTYPING

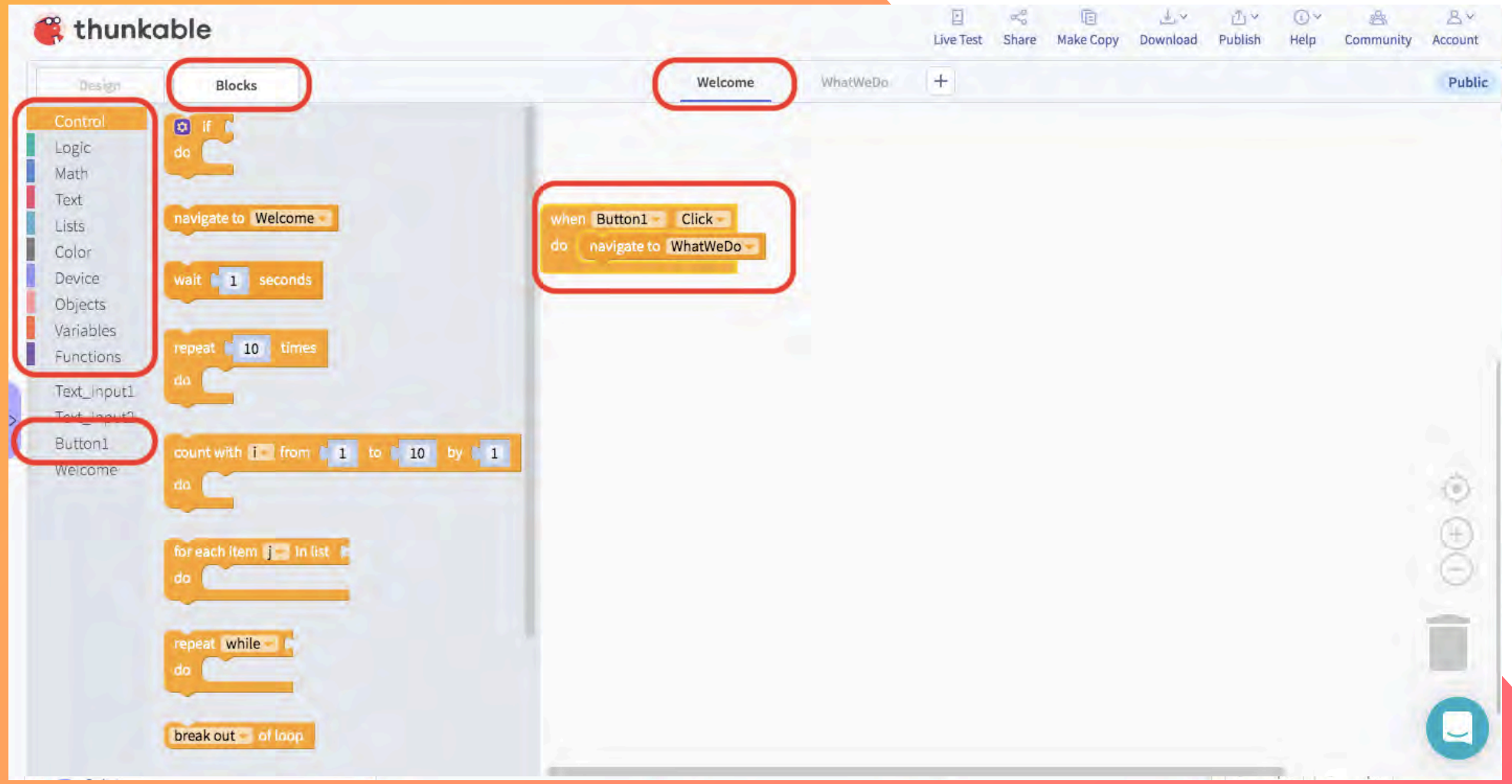


- Block-based visual programming language
- Lets users create programs by manipulating program elements graphically rather than by specifying them textually.

APP PROTOTYPING

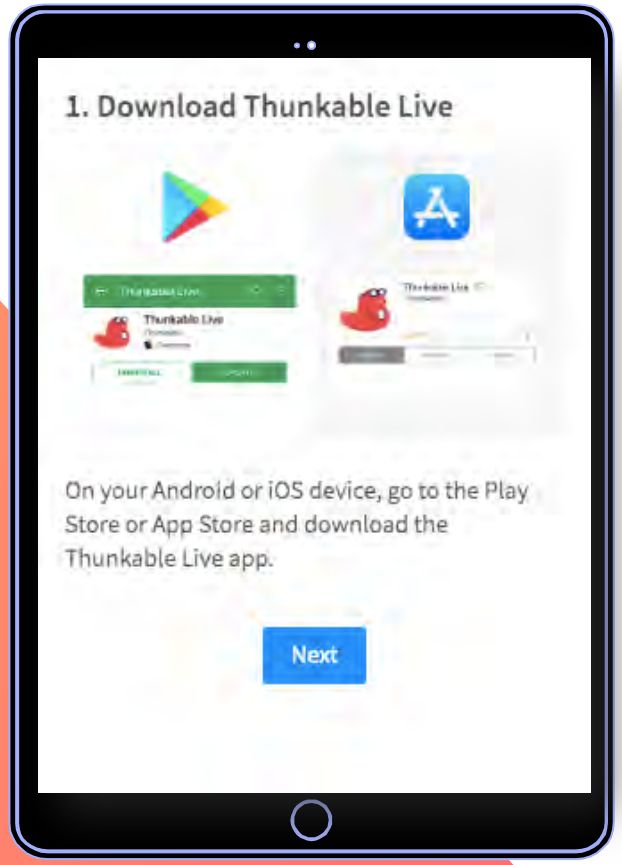


APP PROTOTYPING



LEVEL 1: LETS GET STARTED

- Go to www.thunkable.com and sign up
- Download Thunkable App
- Open the Thunkable Live app and log in
- On your computer, click the "Live Test" button
- When you make changes to your app on the computer, they will update on your mobile device.

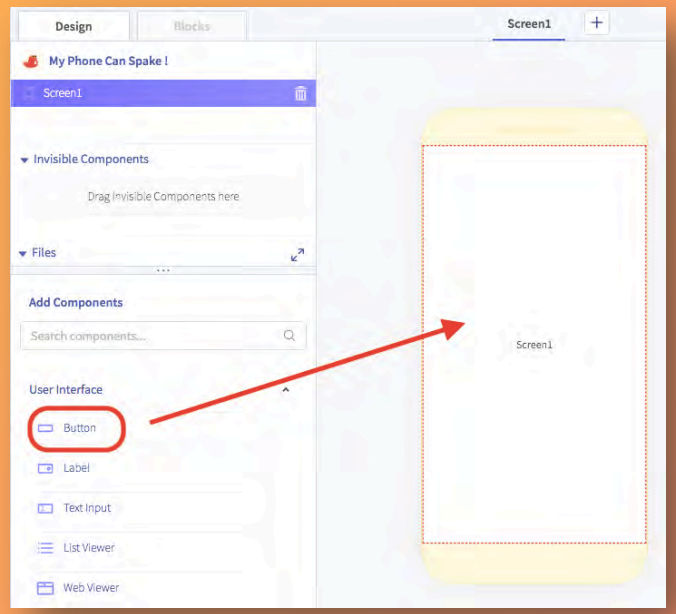


LEVEL 2: MY DEVICE CAN TALK

In 7 Steps you will learn how to make your phone read anything aloud with the click of 1 button.

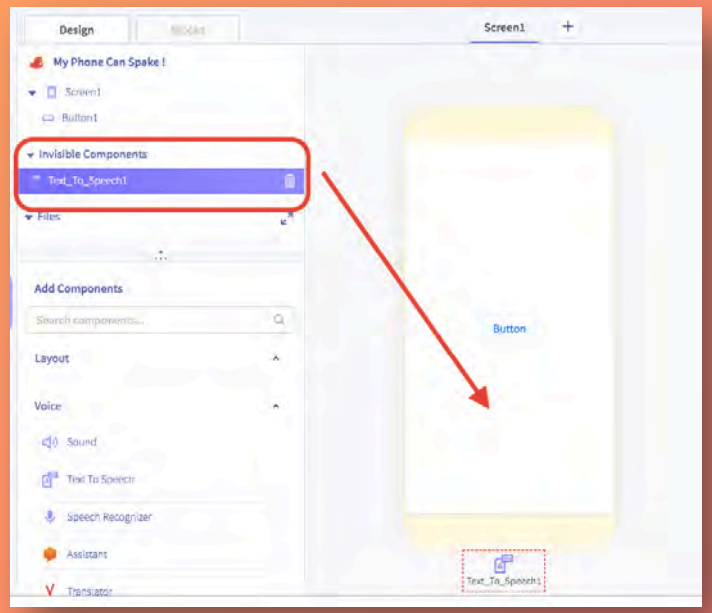
Step 1:

- Drag & drop **Button Component** onto phone



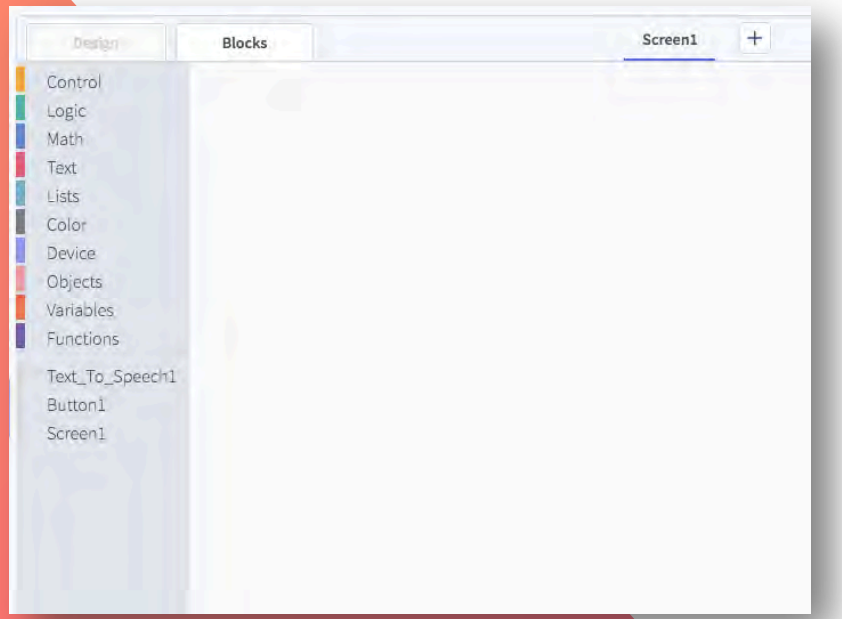
Step 2:

From the Voice section, drag and drop a **Text to Speech Component** into the phone.



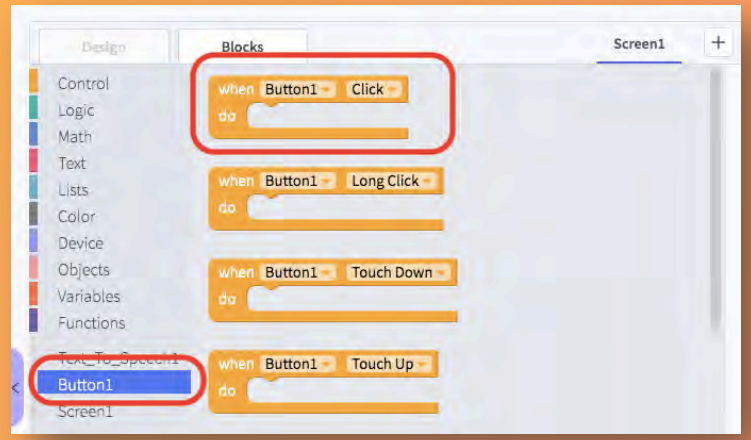
Step 3:

- Click the **Blocks Tab**.
- The blocks tab is where you will program your app.



Step 4:

- Click Button1 to open blocks drawer for this component.
- Drag and drop the **when Button1 Click**



Step 5:

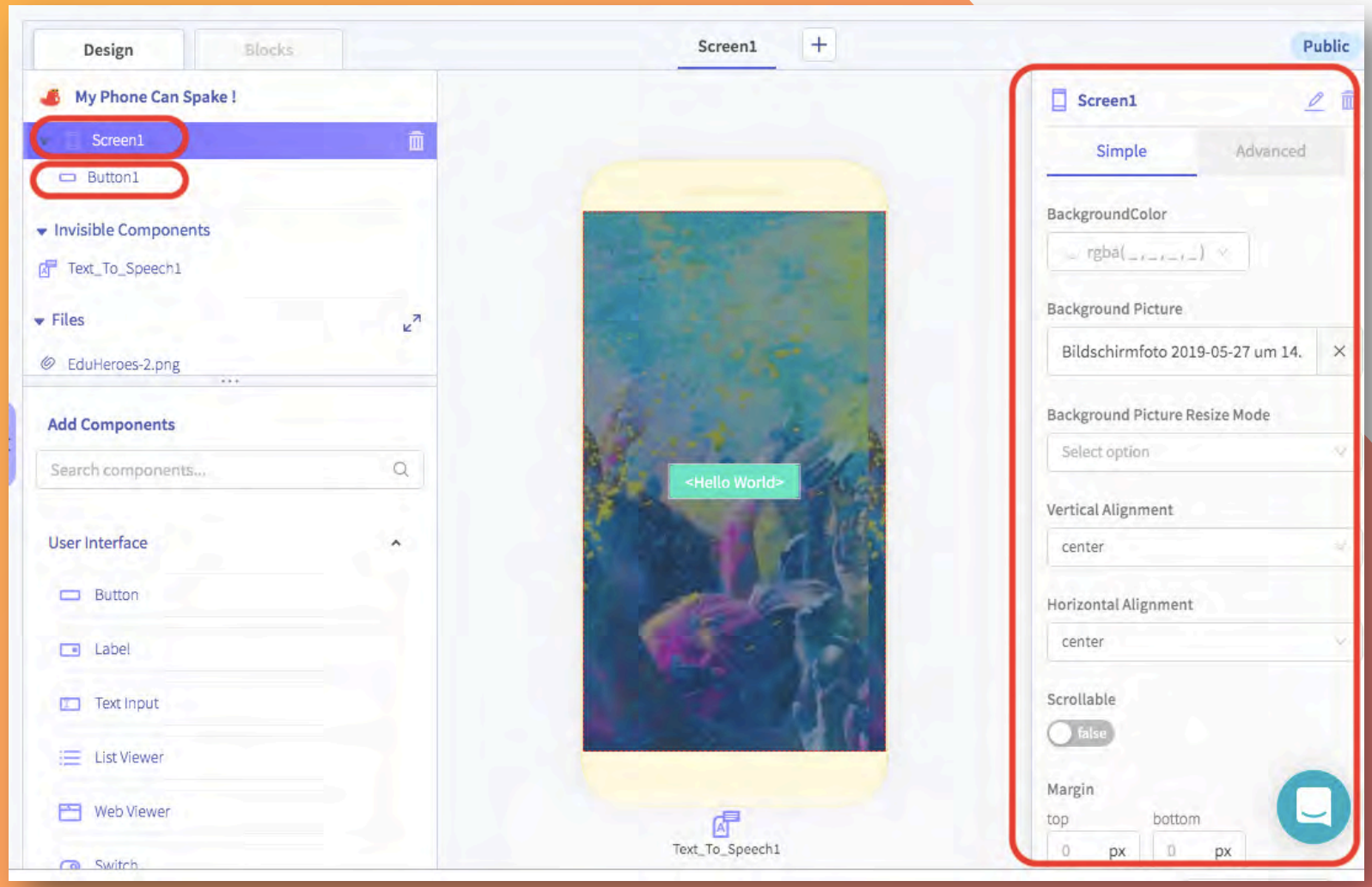
- Click Text_to_Speech1's drawer.
- Drag and drop the **in Text_to_Speech1 call Speak** block into the **when Button1 Click** block.

Step 6:

- Open the Text drawer and select an empty text block.
- Drag and drop the empty text block into the opening of the **in Text_to_Speech1 call Speak** block.
- Now write something!

STEP 7: CLICK LIVE TEST

LEVEL 3: ADD COLOUR !

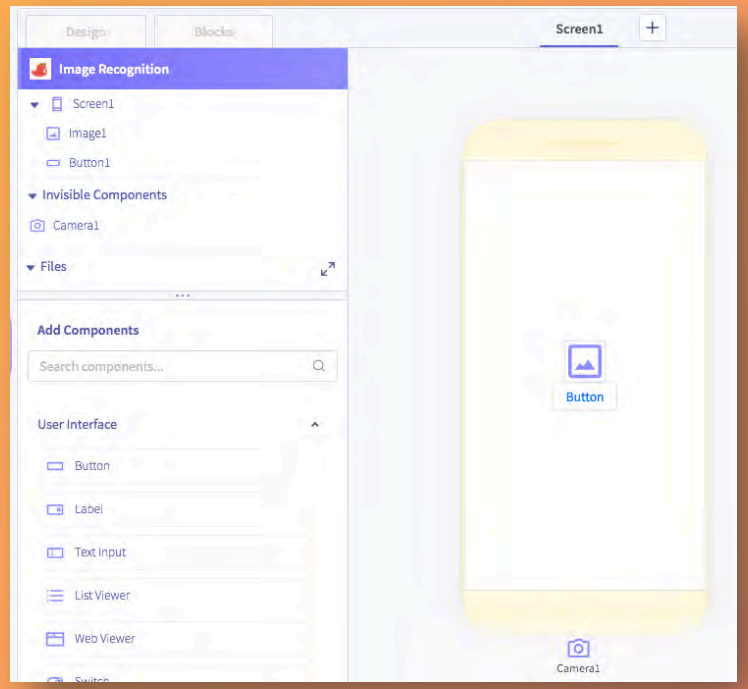


LEVEL 4: IMAGE RECOGNITION

Step 1: Add **Button**

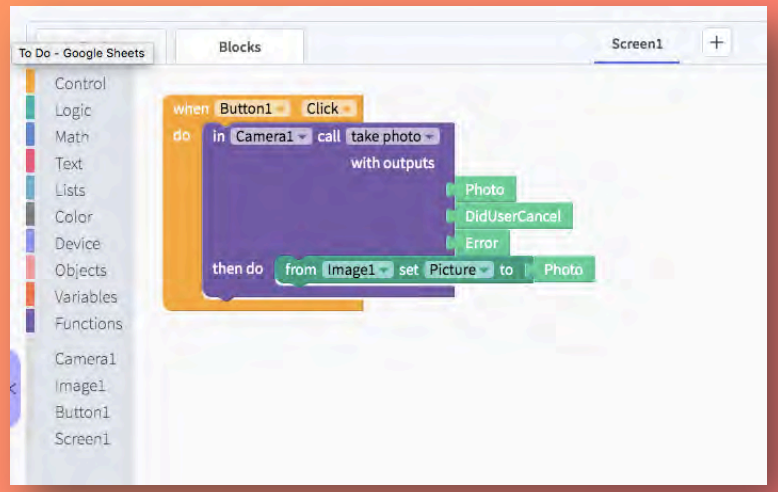
Step 2: Add **Image**. Image will be placed above button icon.

Step 3: Drag & Drop **Camera** component. Camera button will appear under phone screen.



Step 4:

- Open the drawer for Image1.
- Drag and drop the from **Image1 set Picture to** block into the **in Camera1 call TakePhoto** block.
- Next, drag the light green **photo** block from the Camera1 block, and drop it into the opening of the **from Image1 set Picture to** block.



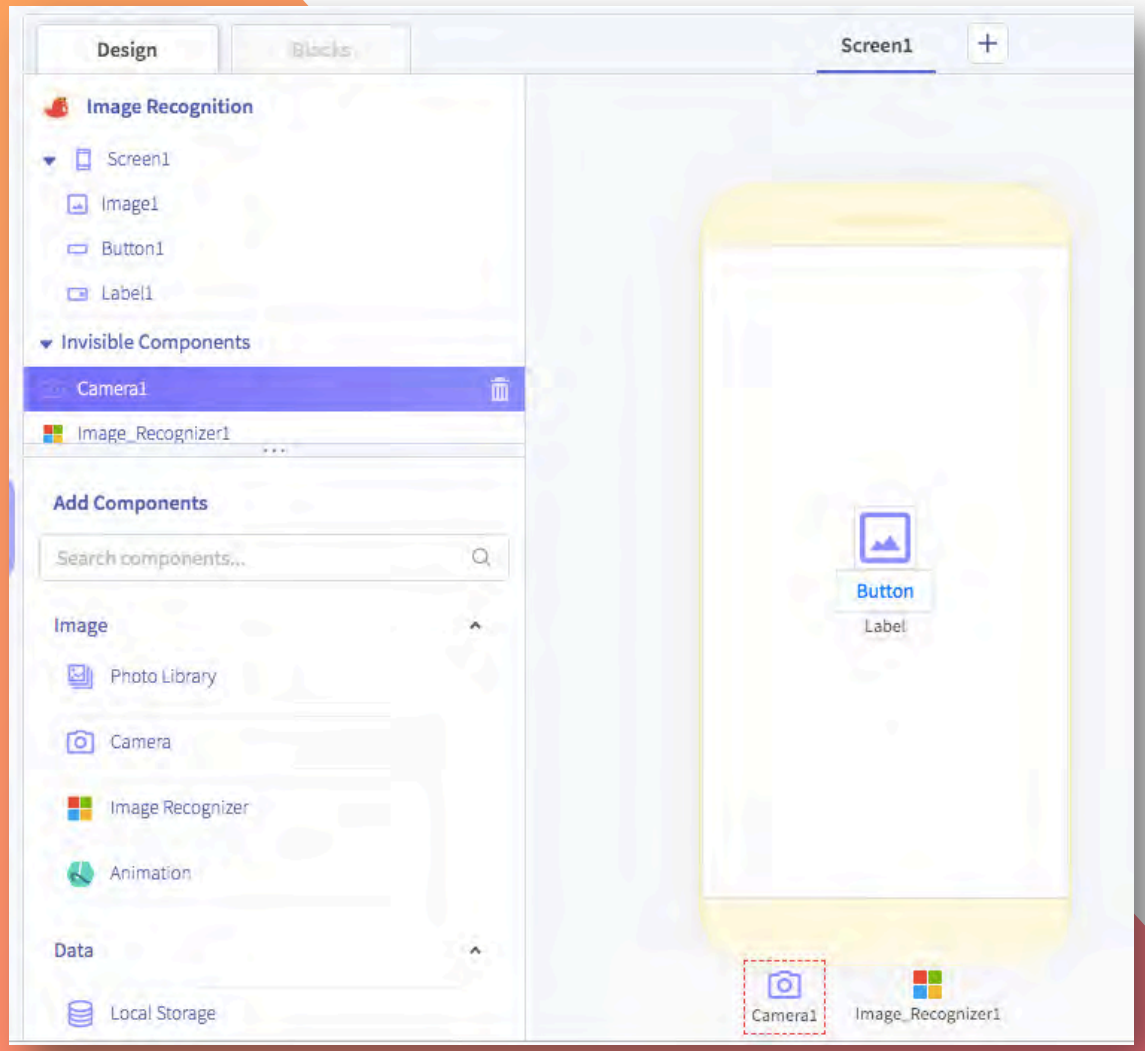
Step 5:

- Open the **Thunkable Live App**.
- Test to see if the image on the screen is set to the picture that you took with the camera.

Step 6: Go back to the **Design Mode**

Step 7: Add a **Label** component and drag it below the button label.

Step 8: From the Image section, drag and drop an **Image Recognizer** component into the phone.

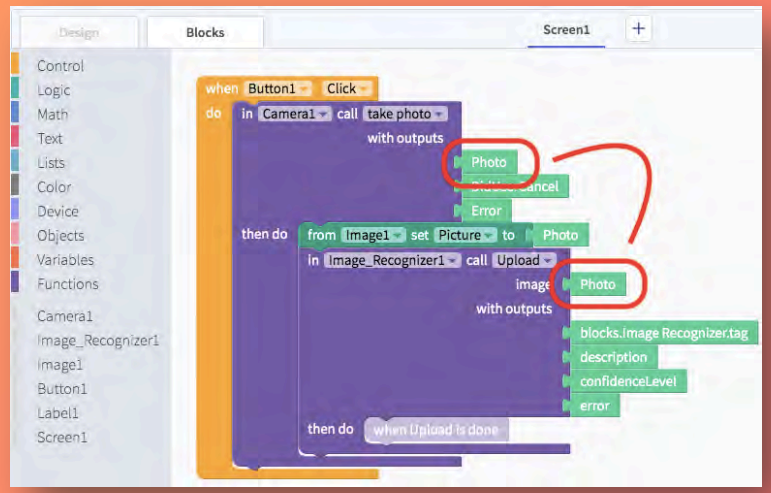


Step 9:

- Back to **Blocks Mode**
- Click the **Image_Recognizer1** drawer.
- Select the **in Image_Recognizer1 call Upload** block, and drop it into the **in Camera1 call TakePhoto** block.

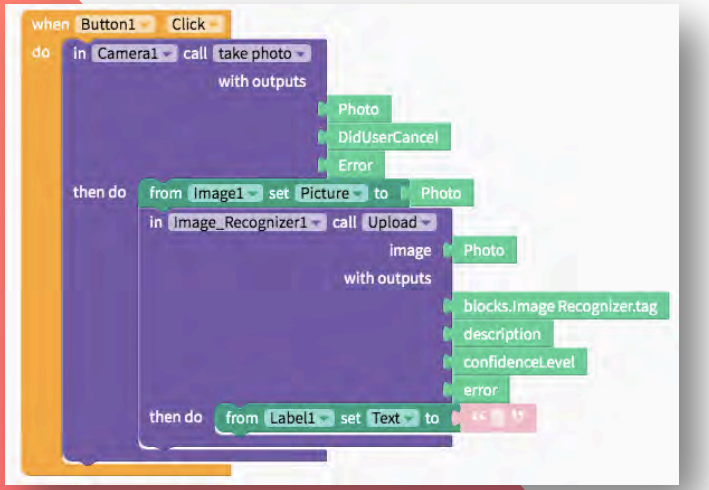
Step 10:

- Drag and drop the **photo** block from the **Camera1** block into the **image** socket on the **Image_Recognizer1** block.



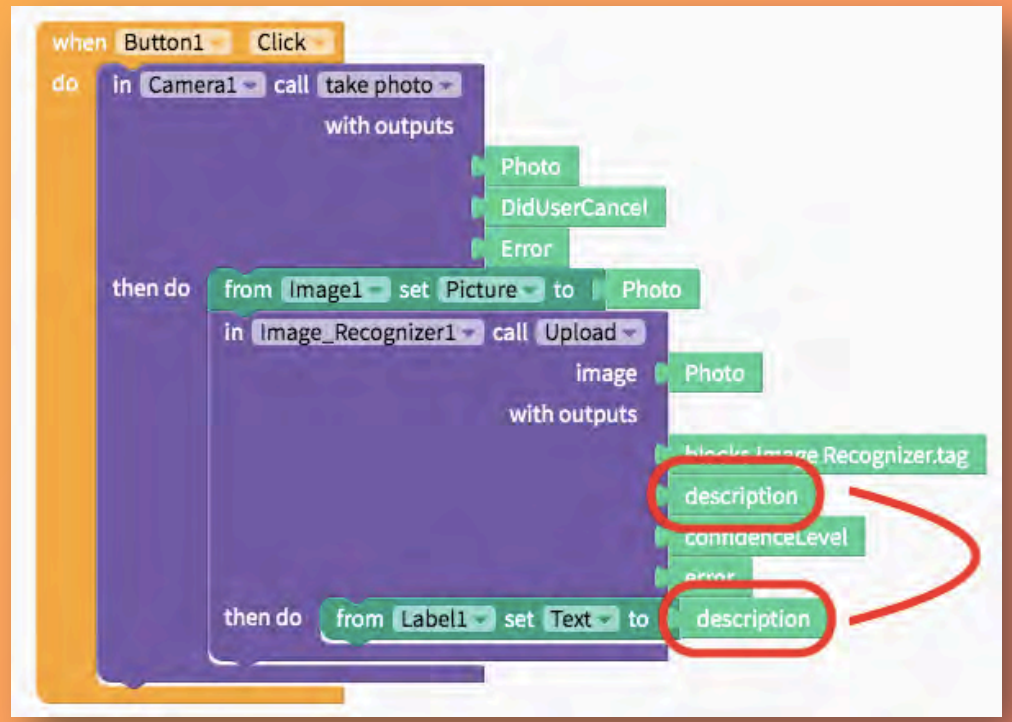
Step 11:

- Open the drawer for **Label1**. Drag and drop a **from Label1 set Text to** block inside the **in Image_Recognizer1 call Upload** block.



Step 12:

- Drag and drop the **description** block from the **Image_Recognizer1** block into the opening of the **from Label1 set Text to** block.



Congratulations you created an AI-based foto app 🚀

// AGENDA

1. <Hello World>
2. Computational Thinking
3. Theory & Terminology
4. Programming Robots
5. Coding & Languages
6. App Protoyping
7. You did it

YOU DID IT!

1. Did we meet your expectations?
2. What went well?
3. What should we change?

// LOOKING AHEAD – 2019/06/13

- A brief introduction to Artificial Intelligence (AI§)
- A deep dive into Python
- Basics of Web Development (HTML, CSS und JS)
- Webscraping
- Excel was yesterday - Exploratory Data Analysis with Pandas
- Automate the boring stuff
- …

LOOKING AHEAD

HELLO WORLD

- Bring your own laptop!
- Are you allowed to install software on your machine?
- Which WiFi network are you using?

Annemieke Frank

Joachim Krois