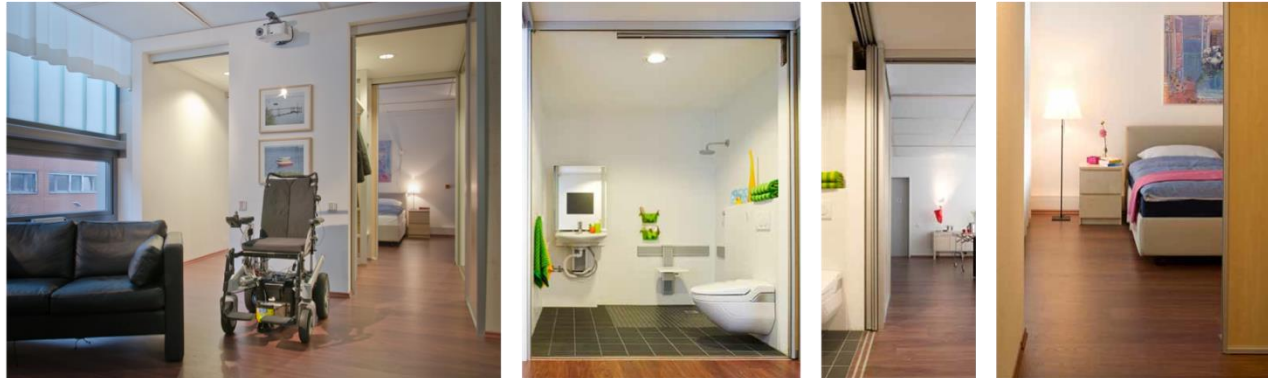


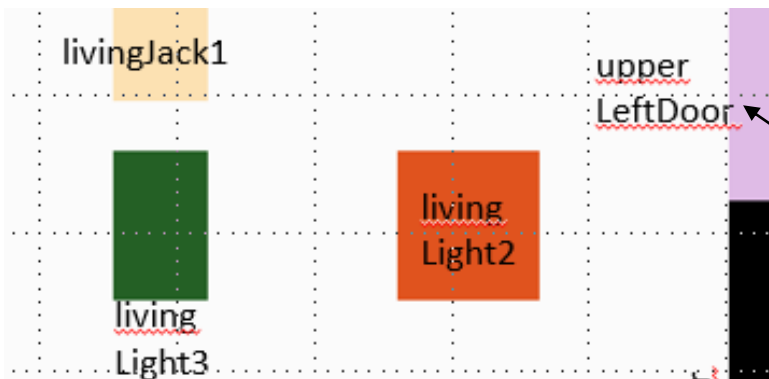
# Das BAALL programmieren



# Das BAALL programmieren

Auf den folgenden Seiten findet ihr eine Skizze vom BAALL und eine Liste aller Objekte und deren Status im BAALL auf die wir zugreifen können. Zugreifen bedeutet, dass wir den Status einzelner Objekte auslesen und ändern können.

Auszug aus der Skizze:



Objekt Name,  
z.B. basin

oder  
upperLeftDoor

Auszug aus der Liste:

**Status Wert, z.B. 108**  
 basin: 108  
 bathroomLight: off  
 bathroomToiletHeight: 111  
 bathroomdoor: on  
 beaconActive: off  
 bed1: 108  
 bed4: 96  
 bedroomJack1: off  
 bedroomJack2: off  
 bedroomLight1: off  
 bedroomLight2: off  
 bedroomTVActive: off  
 bulblamp: 0:0:0  
 corridorLight: off

# Aufgabe

Aufgabe: Schaut euch die BAALL Skizze und die Liste an – achtet dabei besonders auf die Werte, die der Status der Objekte annehmen kann. Welche Objekte aus der Liste würdet Ihr den Klassen „Switch“, „Dimmer“, „RGB Farbe“, „Sensoren“ und „TV Programm“ zuordnen? Warum? Notiert ein paar Beispiele.

„Switch“: z.B. bathroomdoor, \_\_\_\_\_

„Dimmer“: z.B. basin, \_\_\_\_\_

„RGB Farbe“: z.B. bulblamp, \_\_\_\_\_

„Sensoren“: z.B. LuxOutside, \_\_\_\_\_

„TV Programm“: \_\_\_\_\_

# Das BAALL programmieren




Es gibt verschiedene „Klassen“ von Objekten im BAALL die wir unter „Switch“, „Dimmer“, „RGB Farbe“, „Sensoren“ und „TV Programm“ zusammengefasst haben. „Switch“ bedeutet zum Beispiel, dass das Objekt nur zwei Zustände haben kann: An/Aus, Auf/Zu oder 1/0.

Den entsprechenden Block findet ihr in BEESM unter BAALL und dann unter „Verbindung“.



.....

# Status holen

- Im Setup:
  - Wir verbinden uns mit dem BAALL Server 
- In der loop:
  - Wir legen eine Variable an. In dieser speichern wir den Statuswert, den wir holen. Zum Beispiel setzen wir die Variable auf: 
  - Wir geben den Wert der Variable auf dem Display aus.
  - **!!! Setzt in der loop ein delay von mehreren Sekunden, z.B.**  **!**  
(Wir müssen schließlich den Status nicht in jeder Millisekunde abrufen)

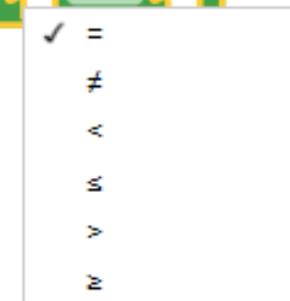
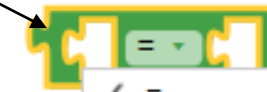
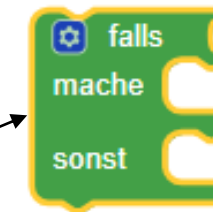
# Aufgaben

- Hole den aktuellen Status und gebe diesen auf dem Display aus:
  - vom Waschbecken (basin)! \_\_\_\_\_
  - von der dimmbaren Lampe (livingLight3)! \_\_\_\_\_
  - vom Schlafzimmerlicht (bedroomLight1 oder 2)! \_\_\_\_\_
  - Von der Temperatur draußen (tempOutside)! \_\_\_\_\_
  - Von der Emotion, die der Badezimmerspiegel gerade „erkennt“! \_\_\_\_\_  
( → Benutze hierzu den Block „Erkenne Gesichtszüge“)
  - Und was euch noch interessiert...

# Bedingungen

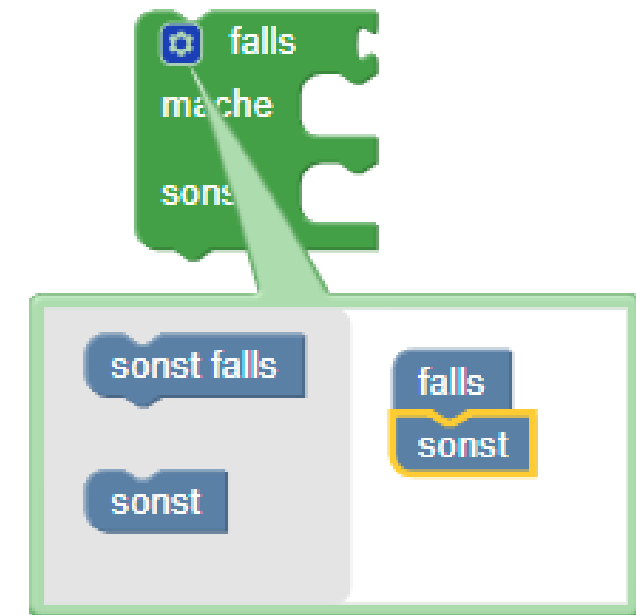
- Bedingungen dienen zum Steuern des Programmflusses. Sie sind so etwas wie „Falls dies zutrifft mache dies, sonst das“.

- Dazu benötigen wir einen **Bedingungsblock** ...
- ...und einen **Vergleichsblock**. Es gibt einfache Vergleiche, wie „kleiner“, „größer“ und „gleich“
- Damit überprüfen wir, ob der Vergleich zweier Werte wahr ist oder falsch. Ist  $1 = 1$  ? Ist er wahr, werden die Anweisungen hinter „mache“ ausgeführt.
- Beide Blöcke findet ihr unter „Logik“



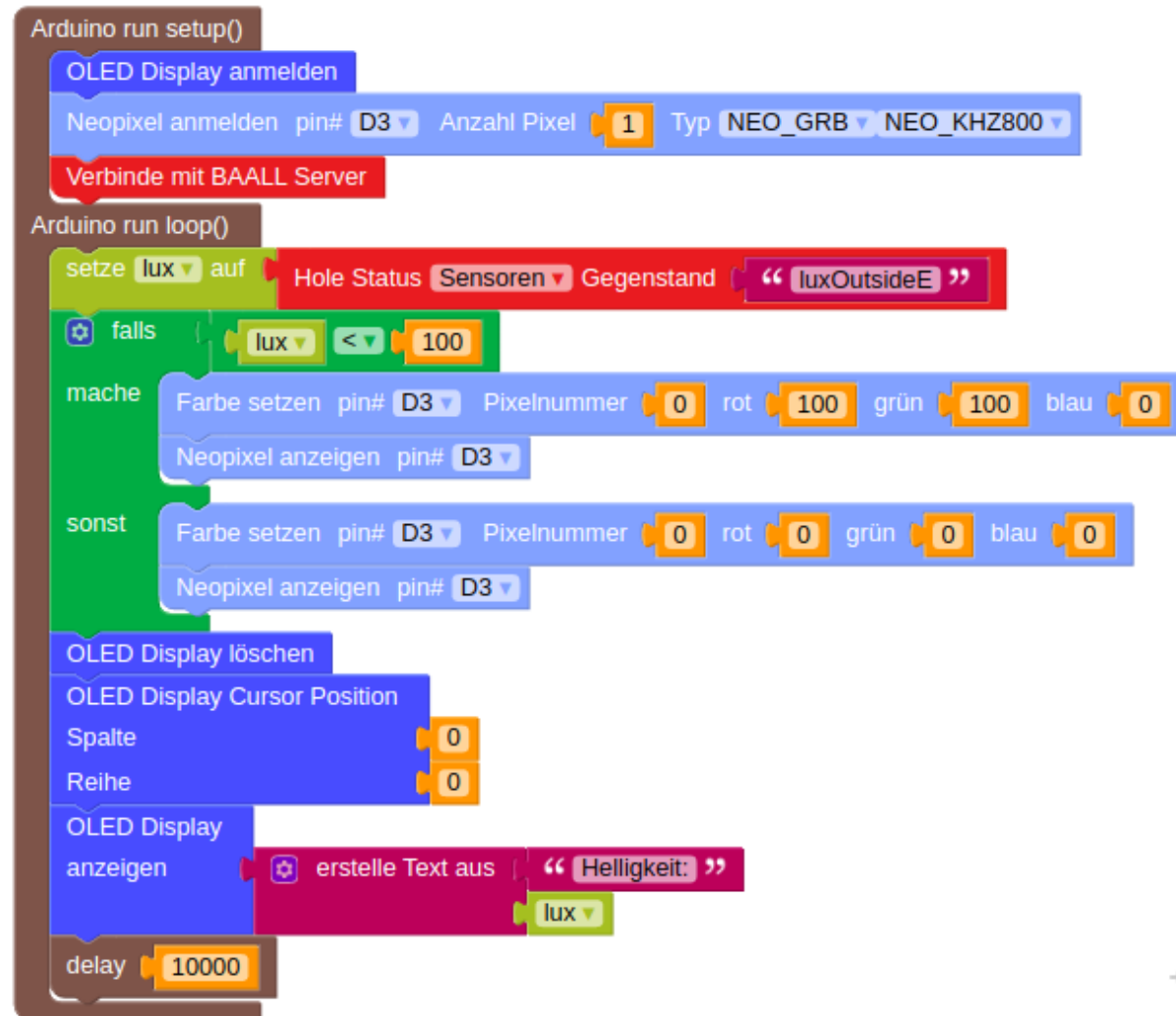
# Aufgabe: Bedingungen

- Wenn die Nachtlampe (bedroomJack1) ausgeschaltet ist, soll auf dem Display eine entsprechende Nachricht ausgegeben werden (z.B. „Es ist Bett-Zeit“).
- Wenn die Nachtlampe angeschaltet ist (sonst), soll eine andere Nachricht auf dem Display erscheinen.
- Es reicht, wenn wir den Status alle paar Sekunden überprüfen. **Also den Delay in der loop nicht vergessen!!**





# Intelligente Lichtsteuerung



Was macht dieses Programm?

---

---

---

---

---

---