KU LEUVEN

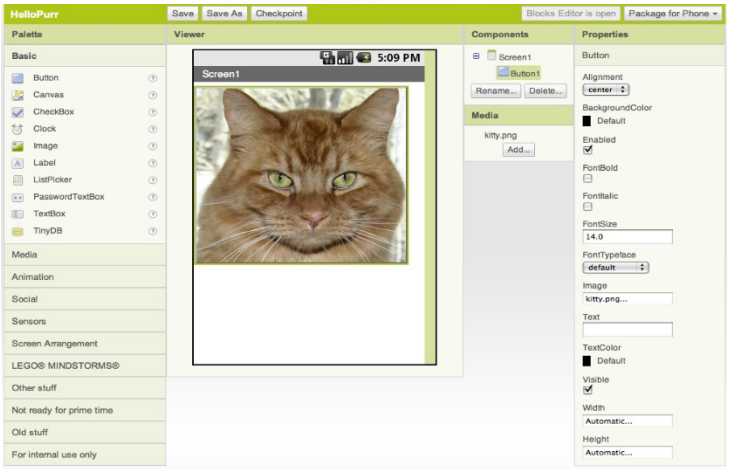
Computationeel denken hoorcollege 2

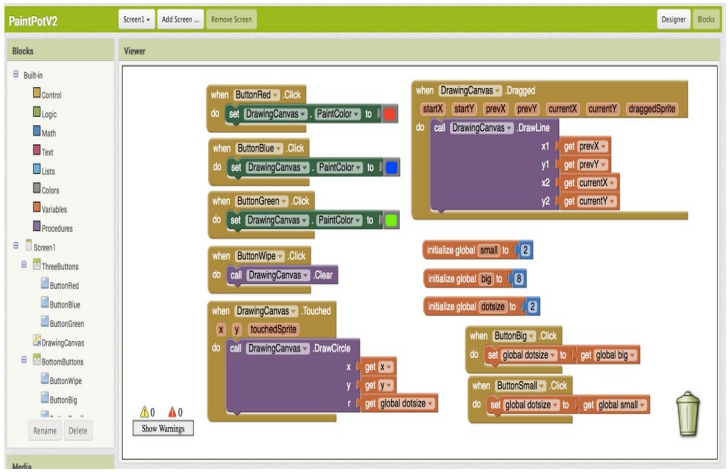
Vincent Naessens

AppInventor::twee views

Designer view

Block view



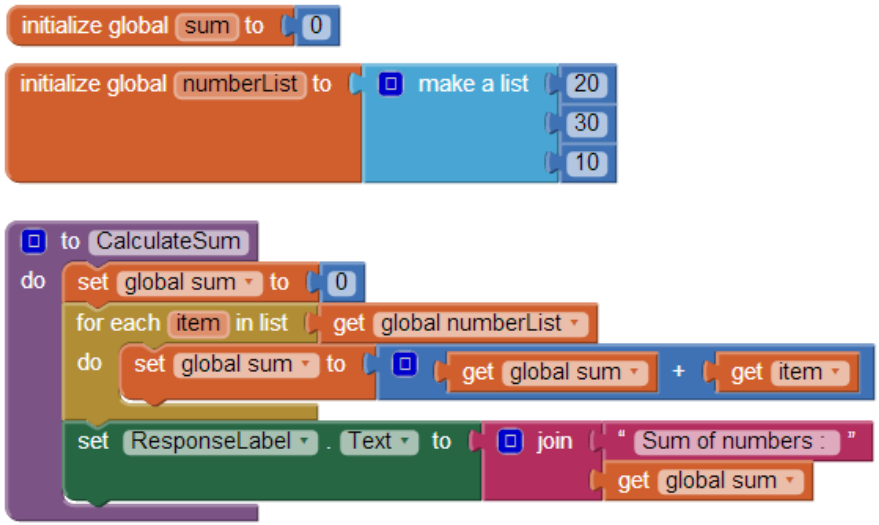


GUI = Graphical User Interface

De code achter ‘de knoppen’

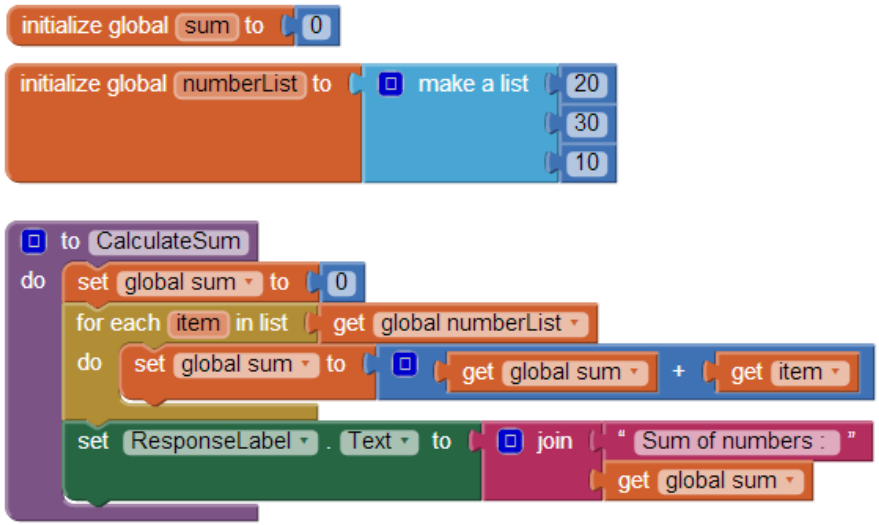
AppInventor::Block view

- **`Math`** ☾ **wiskundige berekeningen** **`(optelling, deling…)`**
- **`Variables`** ☾ **opslag van gegevens in het geheugen**
- `Control` ☾ controlestructuren `(if … then … else …)`
- `Logic` ☾ vergelijkingstesten `( <, >, ==, …)`



AppInventor::Block view

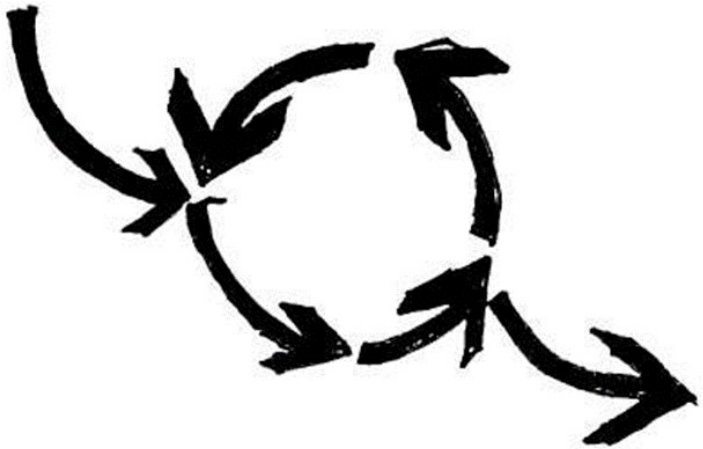
- `Math` ☾ wiskundige berekeningen `(optelling, deling…)`
- `Variables` ☾ opslag van gegevens in het geheugen
- **`Control`** ☾ **controlestructuren `(if … then … else …)`**
- **`Logic`** ☾ **vergelijkingstesten `( <, >, ==, …)`**



Decision blocks

decision blocks

repitition blocks



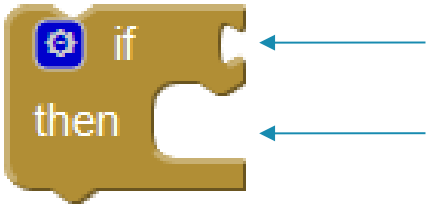
Decision blocks

- Verschillende types
  - `String` ☾ “Jules” “Ik ben hier” “Joske” “Nerd”
  - `Number` ☾ [1] [245] [45.3432] [47]
  - `Boolean` ☾ `true` `false`

- Beslissingen nemen
  - `true` en `false` zijn `Boolean` waarden
  - Deze waarden worden gebruikt bij het maken van beslissingen

Decision blocks

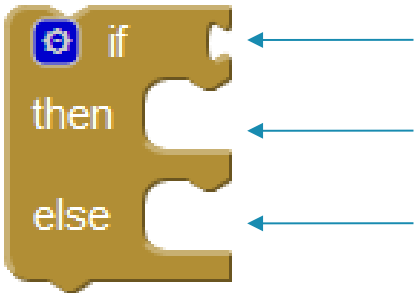
- `if then` blok



als dit `true` is

voer dan de instructies uit die hier staan

- `if then else` blok



als dit `true` is

voer dan de instructies uit die hier staan

In het andere geval, voer de instructies uit die hier staan

Decision blocks

- `if then else if` blok

als dit `true` is

voer dan de instructies uit die hier staan

in het andere geval, als dit `true` is

voer dan de instructies uit die hier staan

in het andere geval, als dit `true` is

voer dan de instructies uit die hier staan

in het andere geval, voer de instructies uit die hier staan

Relational operators

- Een relationele operator vergelijkt twee waarden, en geeft als resultaat een Boolean (`true` of `false`) terug.

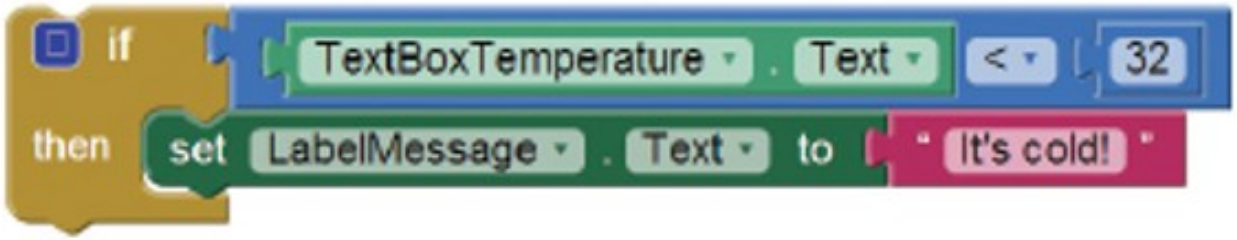
  - `7 = 6` ☾ `false`
  - `3 + 5 < 10` ☾ `true`
  - `7 - 4 > 8` ☾ `false`

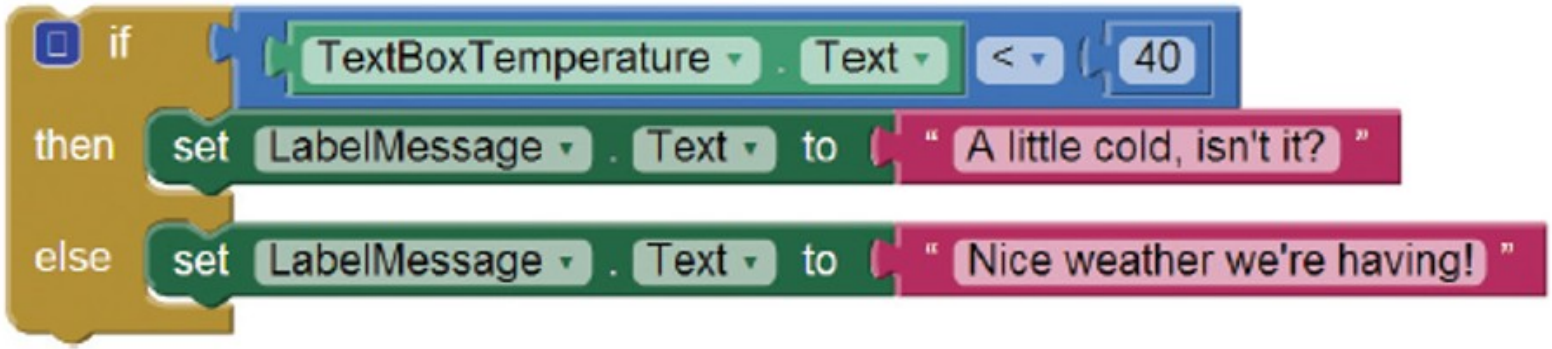
Is the length variable greater than the width variable?

Is the sales variable greater than or equal to 5,000?

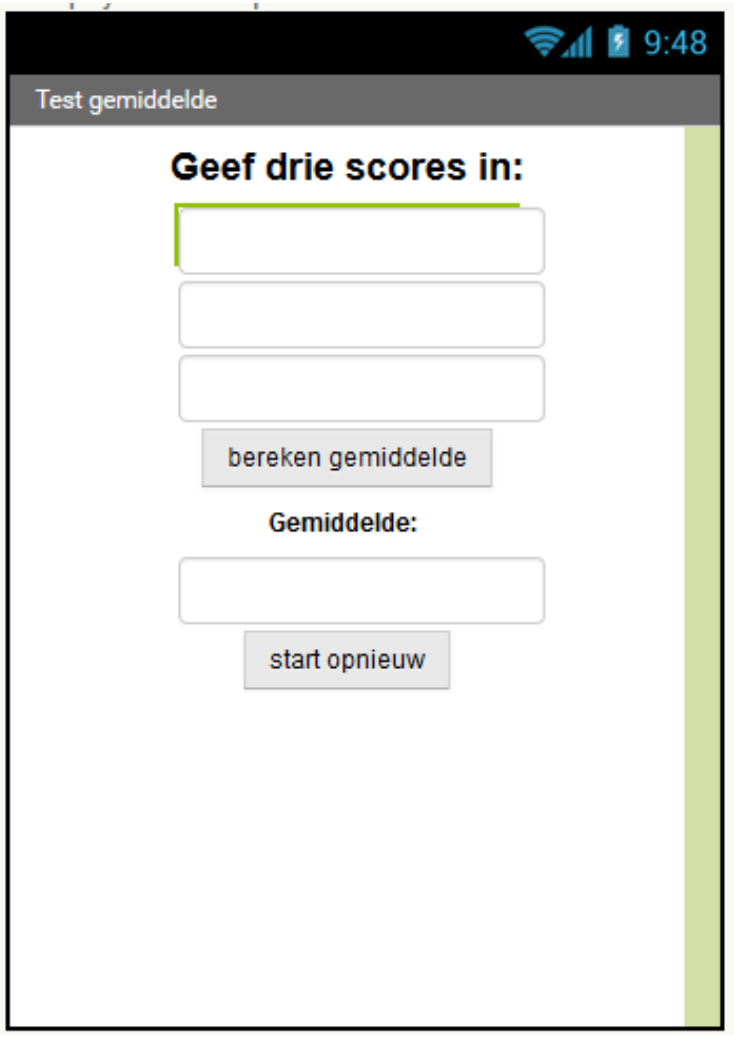
Is the TextBoxTemperature component's Text property less than 32?

Decision blocks



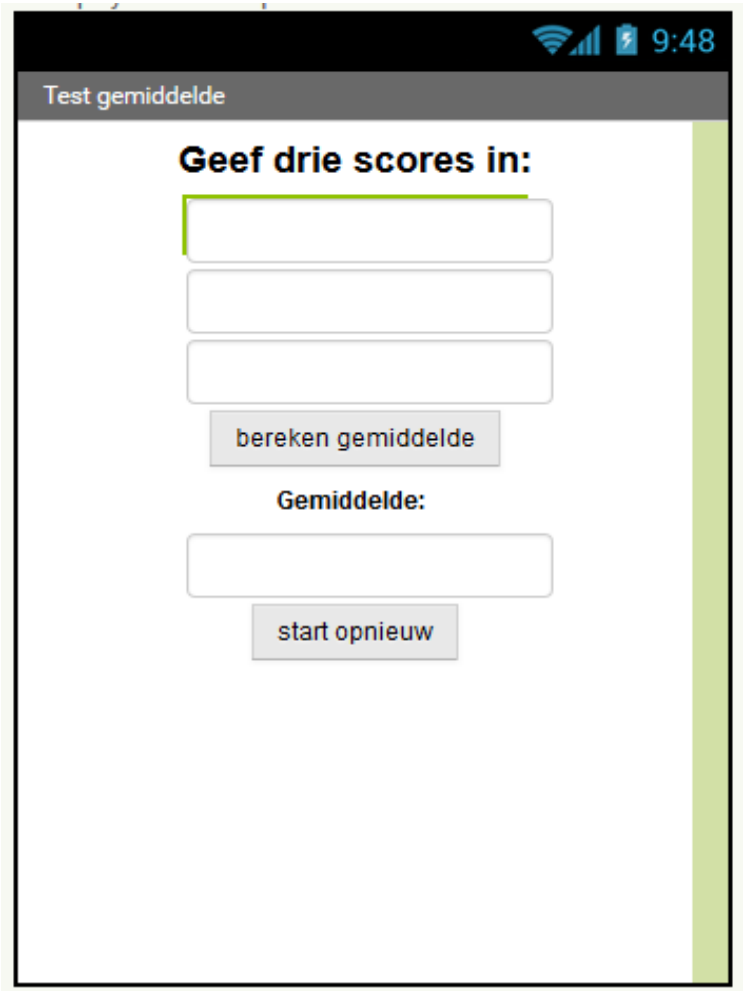


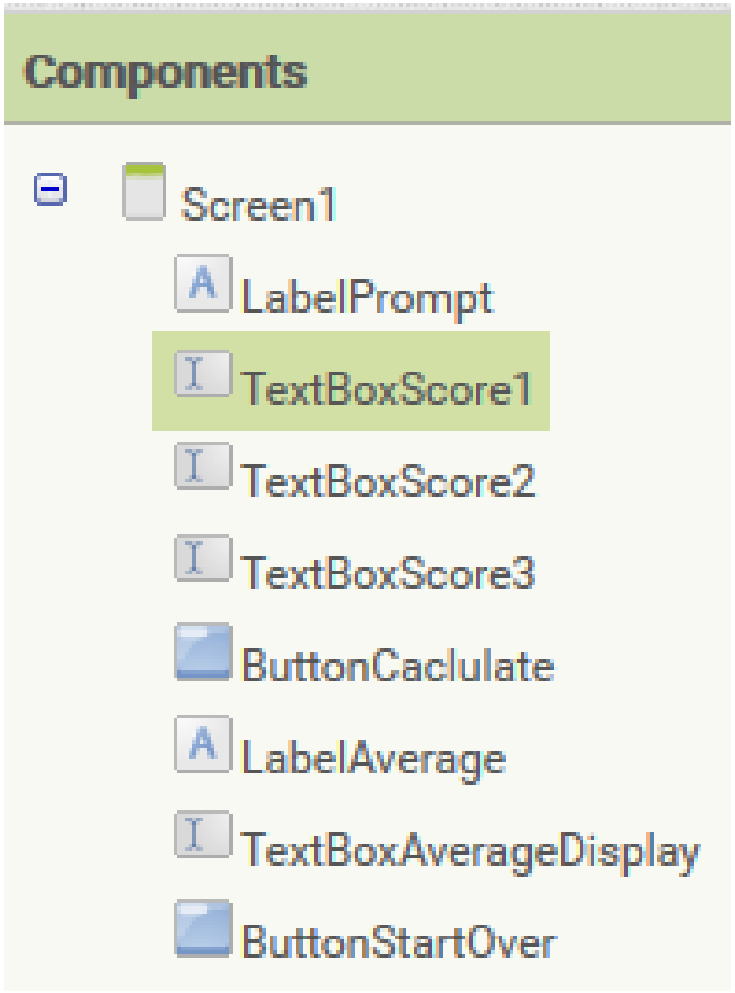
Average App



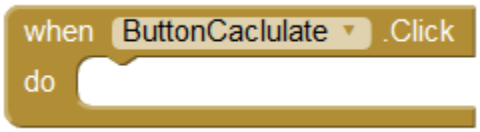
Bij deze app geeft de gebruiker 3 getallen in. Bij het indrukken van de knop wordt het gemiddelde van de 3 getallen berekend, en in de onderste TextBox geplaatst. Als het gemiddeld hoger is dan 95, dan wordt "Mooi gedaan" teruggegeven aan de gebruiker in het bovenste label. Bij het indrukken van de onderste knop worden alle velden opnieuw geïnitialiseerd.

Average App


9:48
Test gemiddelde
Geef drie scores in:
bereken gemiddelde
Gemiddelde:
start opnieuw


Components
Screen1
LabelPrompt
TextBoxScore1
TextBoxScore2
TextBoxScore3
ButtonCaclulate
LabelAverage
TextBoxAverageDisplay
ButtonStartOver

Average App



- STAP 1: maak een locale variabele `gemiddelde` aan
- STAP 2: bereken het `gemiddelde`
- STAP 3: plaats het `gemiddelde` in de juiste TextBox
- STAP 4: print *Mooi gedaan!* af indien `gemiddelde > 95`

Average App

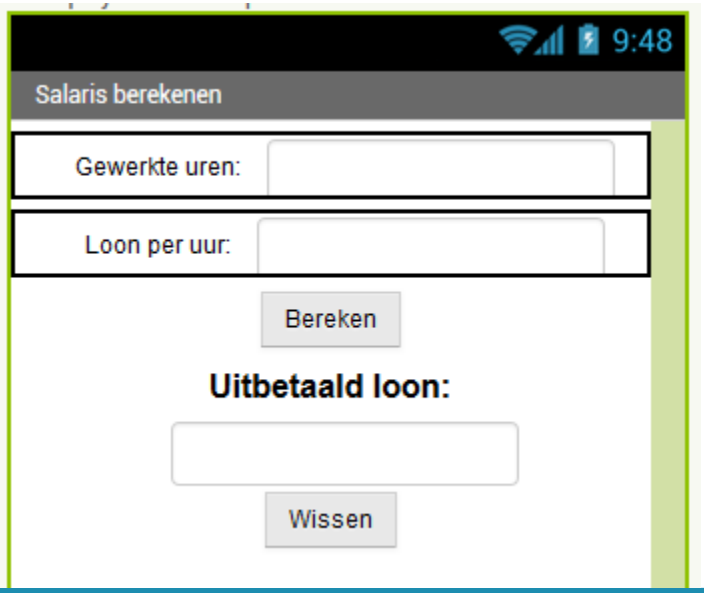
```
when ButtonCaclulate .Click
do
```

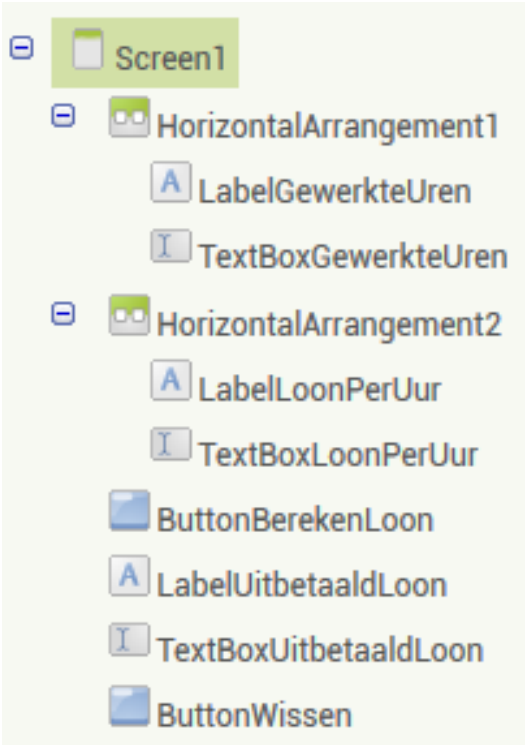
```
when ButtonCaclulate .Click
do
    initialize local Gemiddelde to 0
    in
        set Gemiddelde to ( TextBoxScore1 . Text + TextBoxScore2 . Text + TextBoxScore3 . Text ) / 3
        set TextBoxAverageDisplay . Text to get Gemiddelde
        if get Gemiddelde > 95
        then set LabelPrompt . Text to " Mooi gedaan! "
```

```
when ButtonStartOver .Click
do
    set TextBoxScore1 . Text to " "
    set TextBoxScore2 . Text to " "
    set TextBoxScore3 . Text to " "
    set TextBoxAverageDisplay . Text to " "
    set LabelPrompt . Text to " geef drie scores in "
```

Salary App

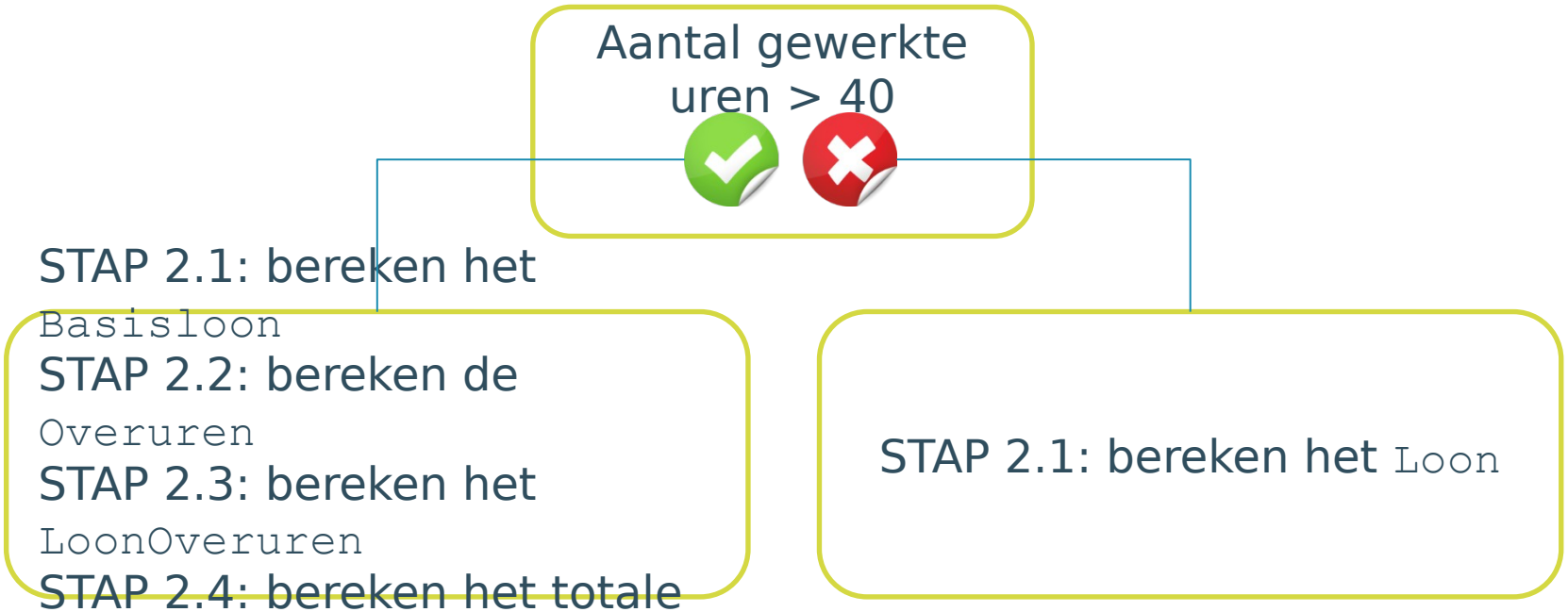
Een werknemer kan zijn gewerkte uren ingeven, en ook het bedrag per uur dat hij betaald wordt. Het berekenen van het loon hangt af van het aantal gewerkte uren. Indien een werknemer meer dan 40 uren werkt, dan worden de meer gewerkte uren aan een tarief van 1.5 uitgekeerd.



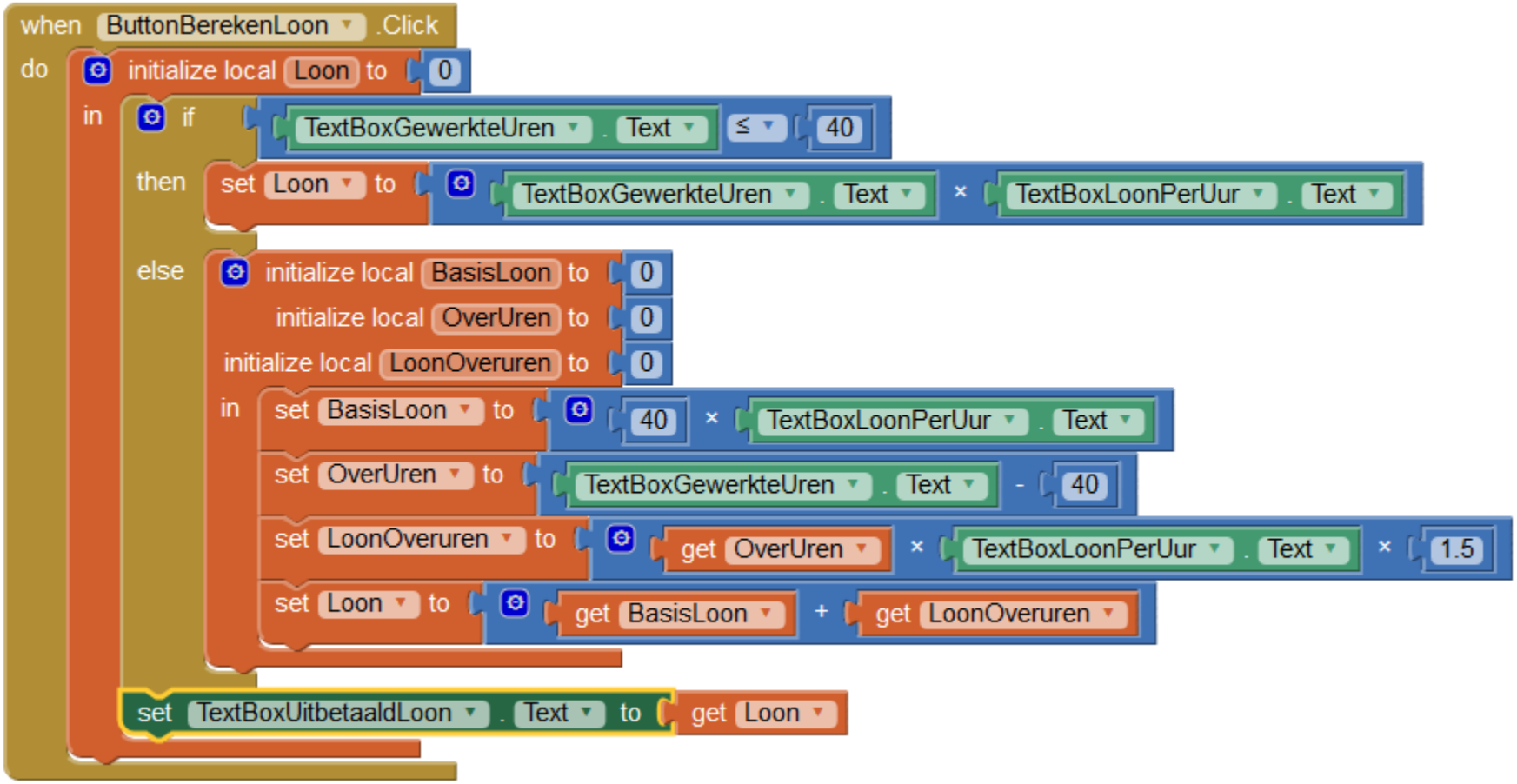


Salary App

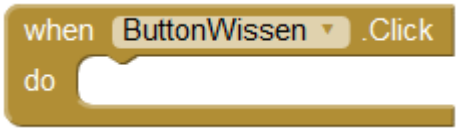
- STAP 1: Initialiseer een variabele Loon
- STAP 2: Bereken het Loon
- STAP 3: plaats het Loon in de TextBox

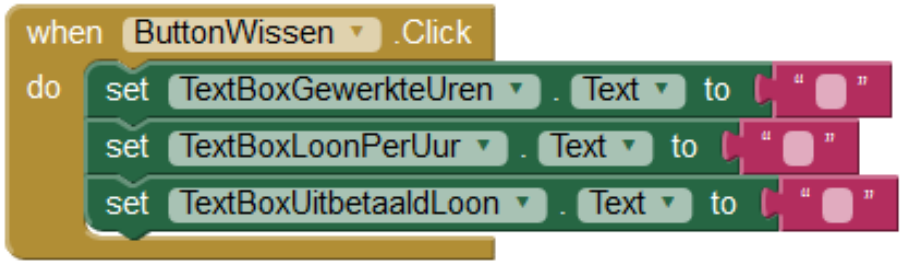


Salary App


when ButtonBerekenLoon .Click
do
initialize local Loon to 0
in
if TextBoxGewerkteUren . Text ≤ 40
then
set Loon to TextBoxGewerkteUren . Text × TextBoxLoonPerUur . Text
else
initialize local BasisLoon to 0
initialize local OverUren to 0
initialize local LoonOveruren to 0
in
set BasisLoon to 40 × TextBoxLoonPerUur . Text
set OverUren to TextBoxGewerkteUren . Text - 40
set LoonOveruren to get OverUren × TextBoxLoonPerUur . Text × 1.5
set Loon to get BasisLoon + get LoonOveruren
set TextBoxUitbetaaldLoon . Text to get Loon

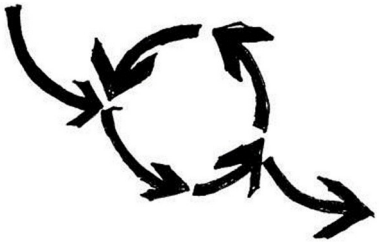
Salary App


when ButtonWissen .Click
do


when ButtonWissen .Click
do
set TextBoxGewerkteUren . Text to
set TextBoxLoonPerUur . Text to
set TextBoxUitbetaaldLoon . Text to

Repetition blocks

Repetition blocks

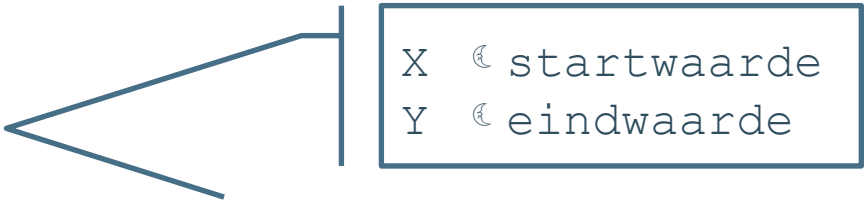


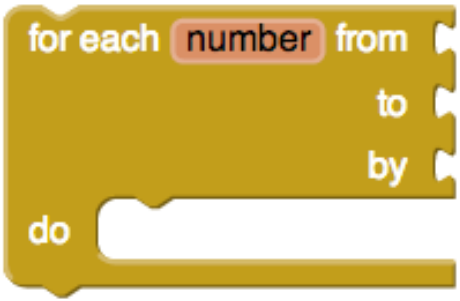
- `while test` lus

zolang dit `true` is

voer dan de instructies uit die hier staan

- `for each` lus



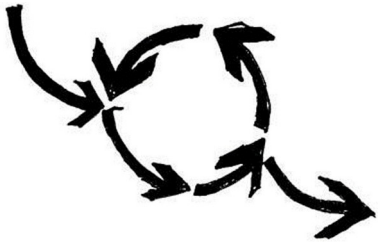


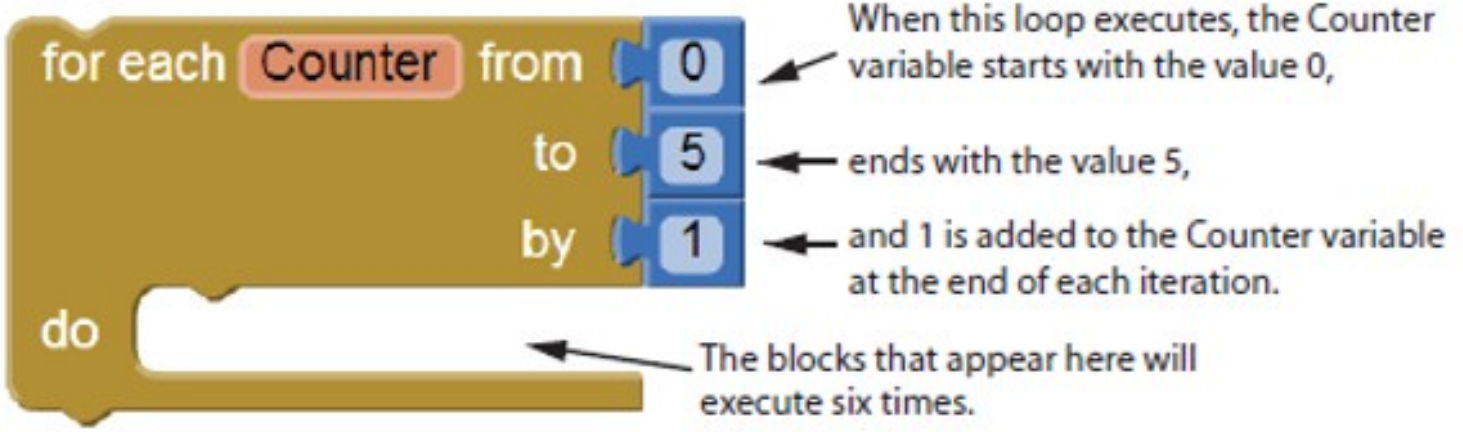
voor elke waarde van `number` van `X` tot `Y`

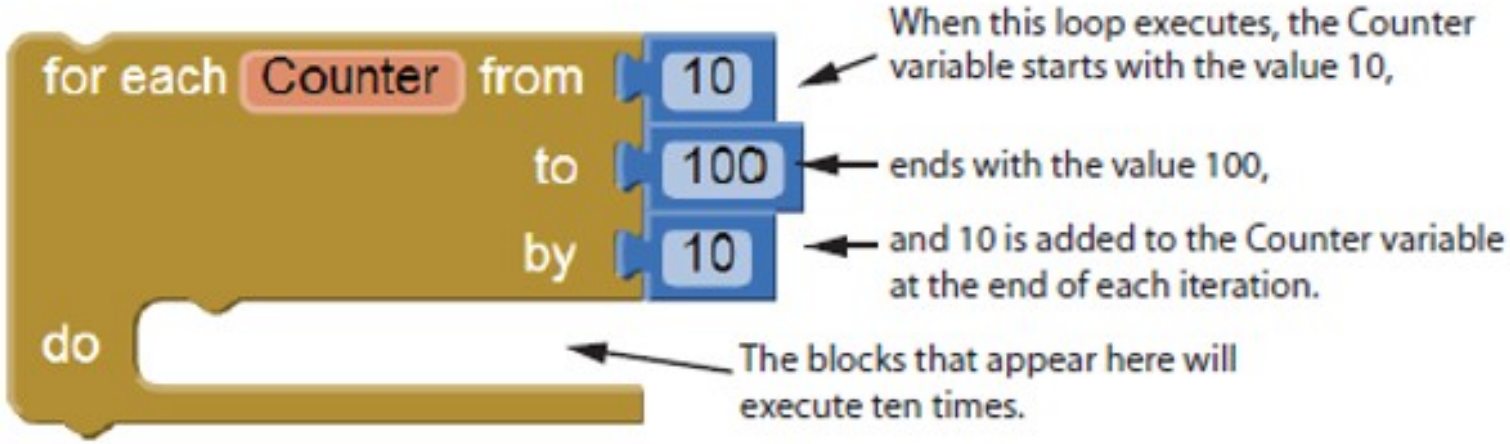
waarbij `number` telkens met `Z` wordt verhoogd

voer dan de instructies uit die hier staan

Repetition blocks

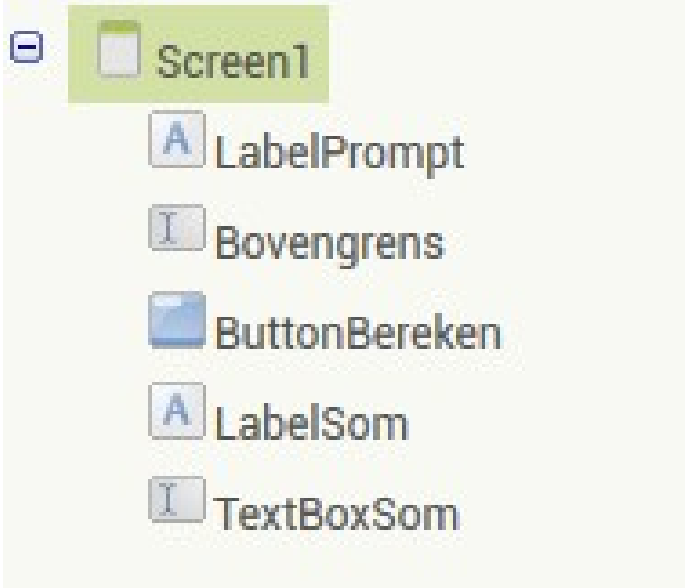




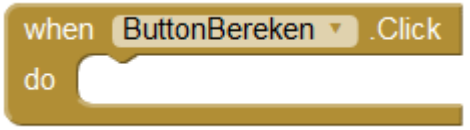


Sum Numbers App

Bereken de som van een aantal getallen.

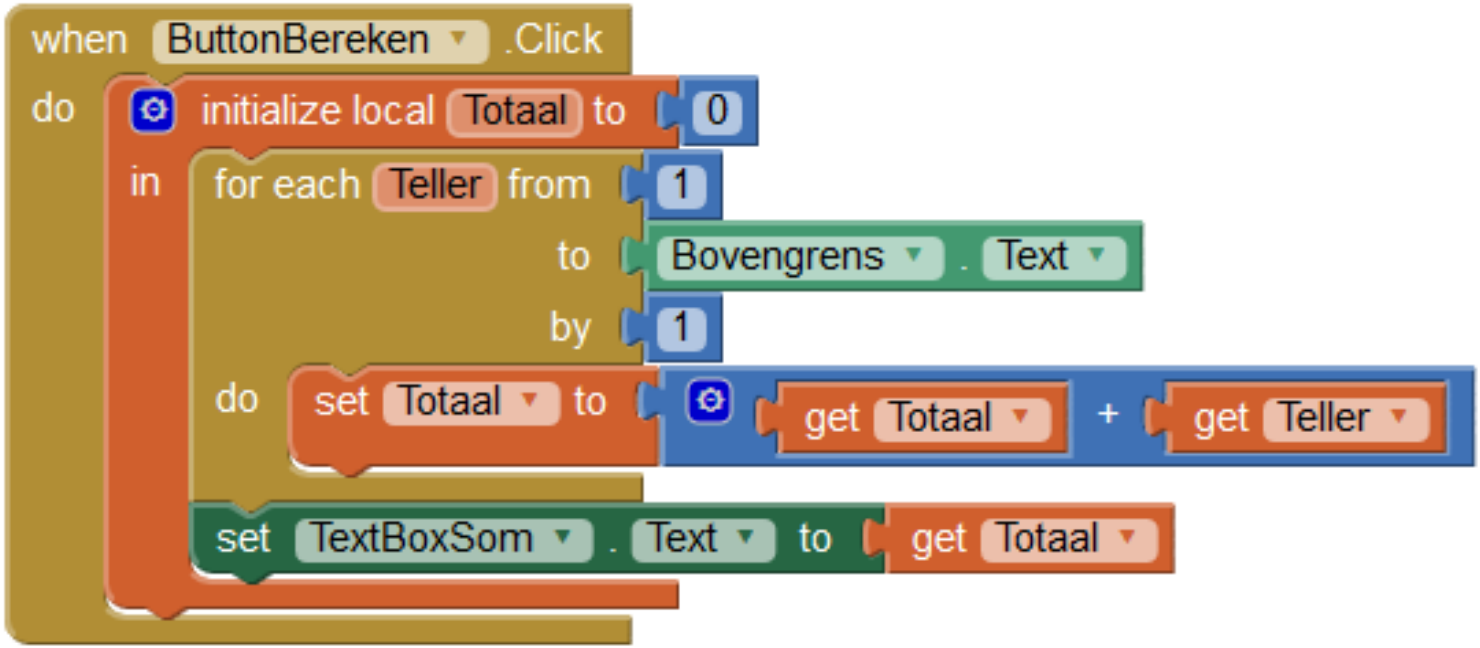


Sum Numbers App



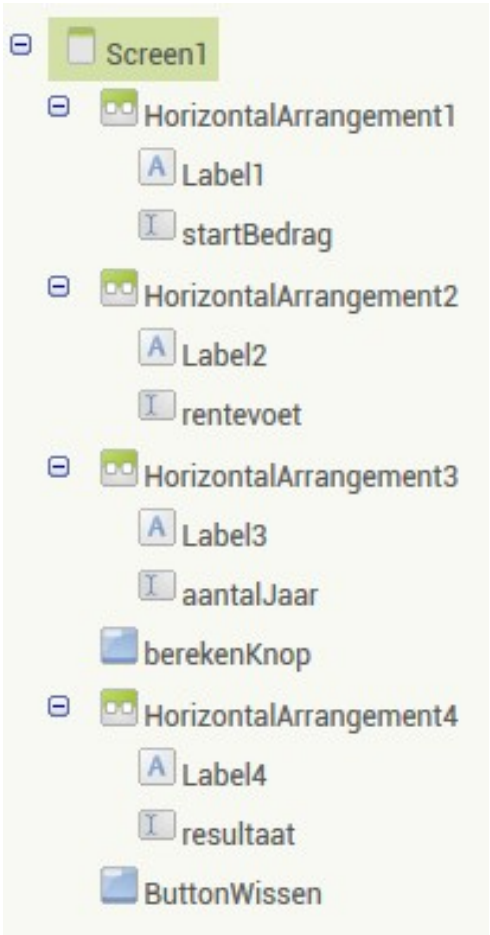
- STAP 1: initialiseer de variabele `Totaal`
- STAP 2: initialiseer de `For Each` lus
- STAP 3: implementeer wat uitgevoerd wordt binnen de lus
- STAP 4: kopieer het `Totaal` in de `TextBox`

Sum Numbers App

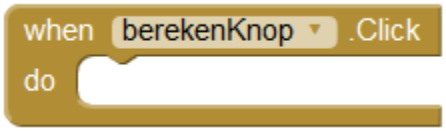


Intrest App

Ontwikkel een app die de af te halen `Balans` *berekent van een* `startBedrag` *dat gedurende een* `aantalJaar` *op de bank staat tegen een bepaalde* `rentevoet`*.*



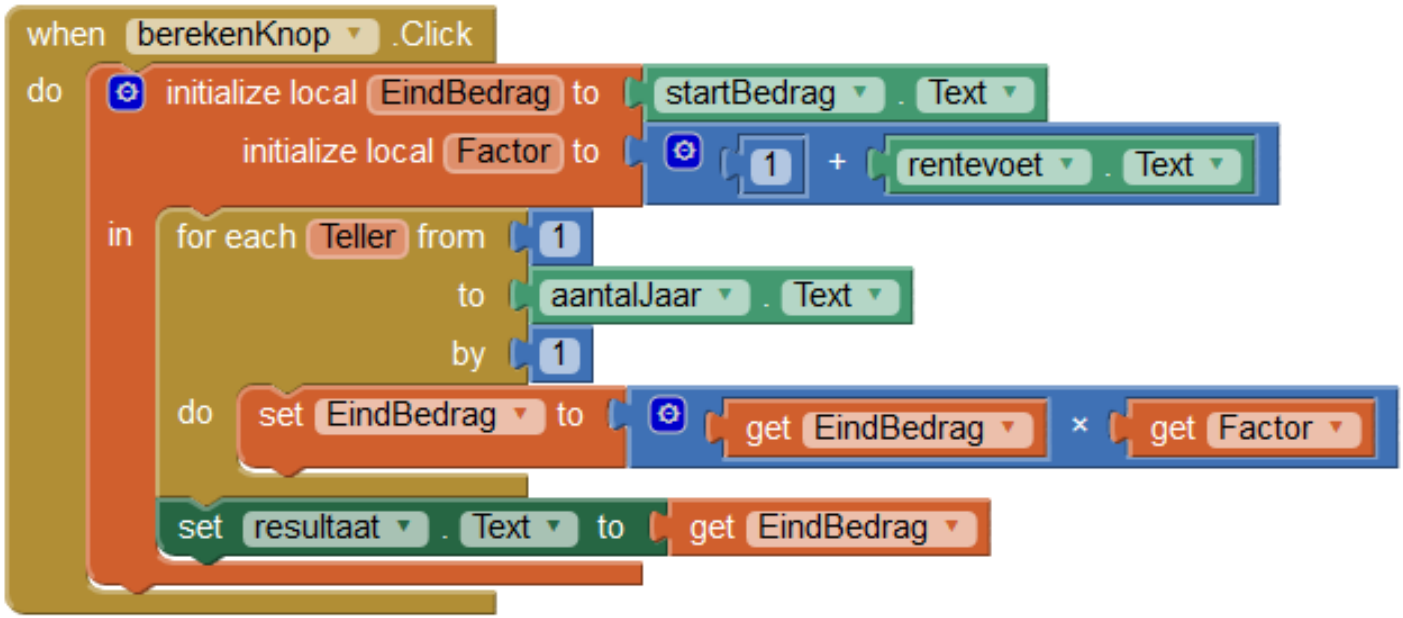
Intrest App



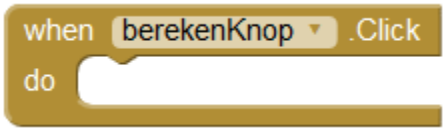
- STAP 1: initialiseer het `EindBedrag`
- STAP 2: initialiseer de `Factor`
- STAP 3: verhoog het `EindBedrag` gedurende elk jaar
- STAP 4: kopieer het `EindBedrag` naar de `TextBox`

Intrest App


when berekenKnop .Click
do

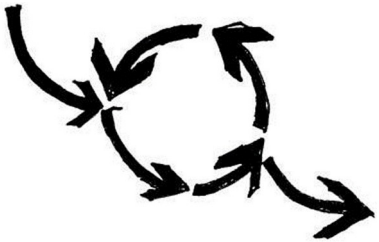

when berekenKnop .Click
do
initialize local EindBedrag to startBedrag . Text
initialize local Factor to 1 + rentevoet . Text
in
for each Teller from 1
to aantalJaar . Text
by 1
do
set EindBedrag to get EindBedrag × get Factor
set resultaat . Text to get EindBedrag

Intrest App

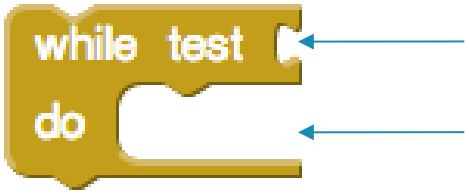

when berekenKnop .Click
do


when berekenKnop .Click
do
initialize local EindBedrag to startBedrag . Text
initialize local Factor to 1 + rentevoet . Text
initialize local Teller to 1
in
while test get Teller ≤ aantalJaar . Text
do
set EindBedrag to get EindBedrag × get Factor
set Teller to get Teller + 1
set resultaat . Text to get EindBedrag

Repetition blocks



- `while test` lus

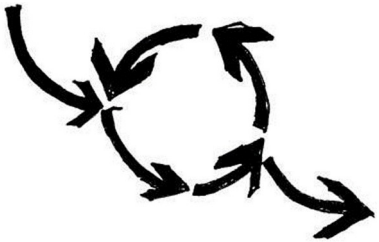


zolang dit `true` is

voer dan de instructies uit die hier staan

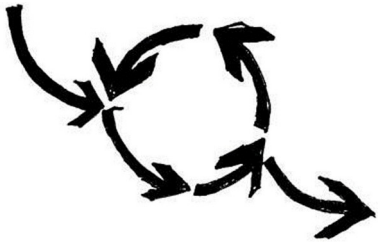
  - Een `while` lus heeft twee delen:
    - [TEST] Een `Boolean` expressie die getest wordt.
    - [DO] Een verzameling instructies die wordt uitgevoerd zolang de `Boolean` expressie `true` is.

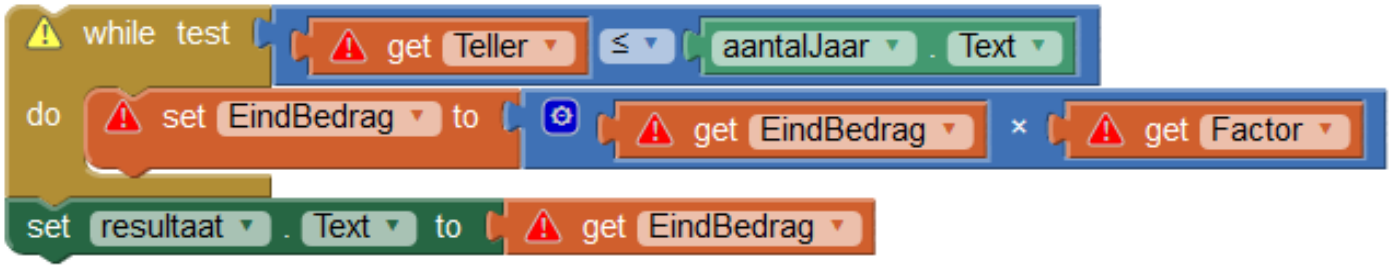
Repetition blocks



- `while test` lus
  - De [test] socket
    - Als `true` dan wordt de [do] socket uitgevoerd.
    - Als `false` dan eindigt de lus.
  - De [do] socket
    - Het aantal keer dat deze wordt uitgevoerd hangt af van de [test]
    - Soms wordt de [do] socket nooit uitgevoerd
  - Infinite loops!!!
    - Lussen moeten op een bepaald moment eindigen
    - In het andere geval kan het programma vastlopen

Repetition Blocks

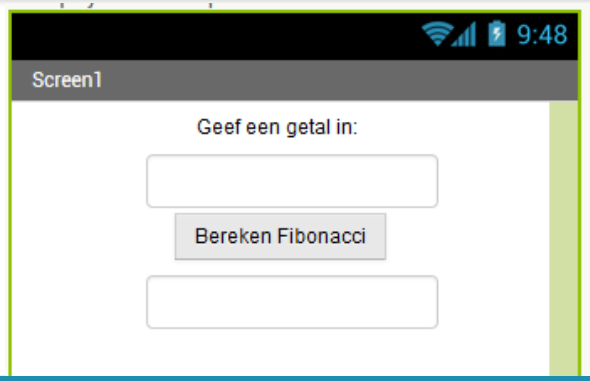


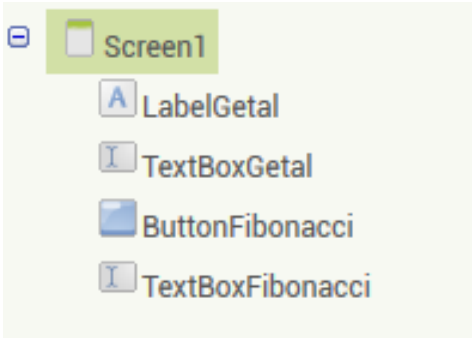


Fibonacci App

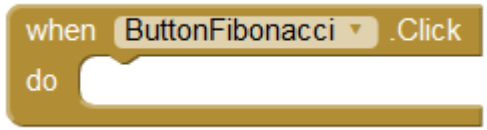
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377

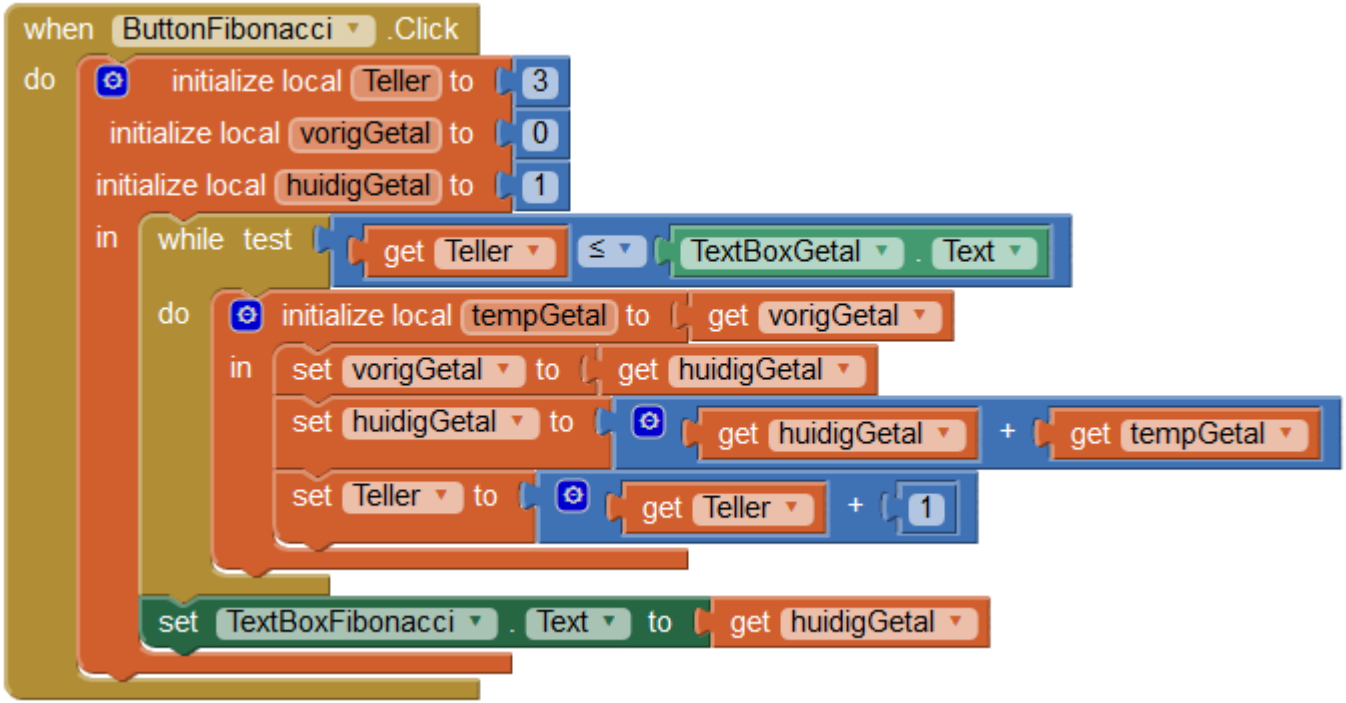
0+1=1 3+5=8 21+34=55





Fibonacci App





Apps of the day

App(s) of the day

- App 1: ontwikkel een programma dat de BMI (body mass index) berekent van een persoon. De gebruiker geeft zijn lengte (in meter) en zijn gewicht (in kg). Het BMI wordt als volgt berekend: `BMI = gewicht/(lengte x lengte)`. Voeg een melding toe van de categorie waartoe de persoon behoort:

BMI index	Categorie
< 18.5	Ondergewicht
18.5 – 25.0	Gezond gewicht
25.0 – 30.0	Overgewicht
> 30	Obesitas

- App 2: Ontwikkel een app die een testscore leest, en vervolgens de graad toont. De volgend graadschaal wordt gebruikt:

Test score	Graad
< 60	F
60 – 69	D
70 – 79	C
80 – 89	B
90 of meer	A