Verwerking

- Logica
- Loops
- Wiskunde
- Teksten
- Lijsten
- Gebruik
- Variabelen
- Functies

Logica

Voorwaardelijke aanwijzingen

Voorwaardelijke aanwijzingen zijn het belangrijkst voor de programmering. Ze maken het mogelijk om de verschillen tussen gevallen te formuleren zoals:

- Wanneer er een weg naar links is, moet je links afbuigen.
- Wanneer het aantal punten = 100, druk "Goed gedaan!" in.

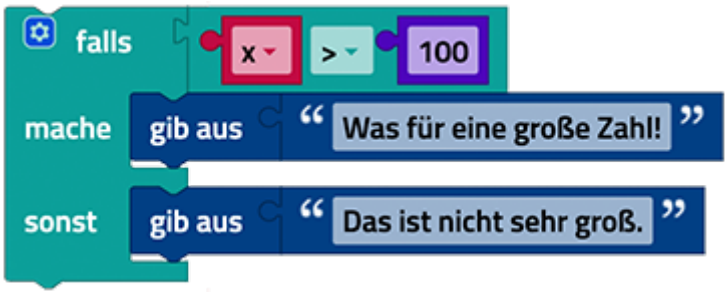
wanneer-blokken

De meest eenvoudige voorwaarde is een **wanneer**-blok:

Wanneer dit wordt uitgevoerd, wordt de waarde van de variabele **x** met 100 vergeleken. Wanneer deze groter is wordt "Wat een hoog cijfer!" uitgegeven. In het andere geval gebeurt er niets.

wanneer-anders-blokken

Ook is het mogelijk om aan te geven dat er iets moet gebeuren wanneer de voorwaarde niet waar is, zoals in dit voorbeeld:



Net als bij het voorgaande blok verschijnt de melding "Wat een hoog cijfer!", wanneer **x** > 100 is. In het andere geval verschijnt de melding "Dat is niet erg hoog.".

Een **wanneer**-blok kan een **anders**-segment hebben, maar niet meer dan een.??????????????????

wanneer-anders-wanneer-blokken

Ook is het mogelijke om meerdere voorwaarden met een enkel **wanneer**-blok te testen wanneer **anders-wanneer**-clausules worden toegevoegd:

Eerst controleert het blok of **x** > 100 is en geeft de melding "Wat een hoog cijfer!", indien dat het geval is. Wanneer dat niet het geval is controleert deze verder of **x** = 42 is. Zo ja, dan verschijnt de melding "Dat is mijn geluksgetal!". In het andere geval gebeurt er niets.

Een **wanneer**-blok kan een willekeurig aantal **anders**-**wanneer**-segmenten hebben. De voorwaarden worden van boven naar onderen toe geanalyseerd, tot aan een voorwaarde is voldaan of er geen voorwaarde meer over is.

wanneer-anders-wanneer-anders-blokken

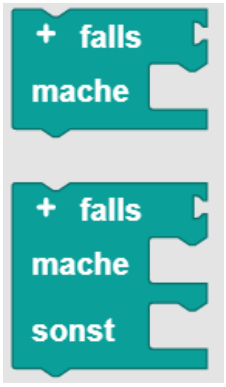
wanneer-blokken kunnen zowel **anders**-**wanneer** als ook **anders**-segmenten hebben:

Het **anders**-segment garandeert dat een actie wordt uitgevoerd, ook wanneer geen van de voorgaande voorwaarden waar is.

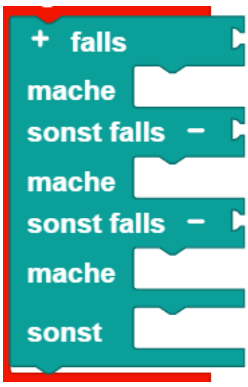
Een **anders**-segment kan na een willekeurig aantal van **anders**-**wanneer**-segmenten optreden, inclusief nul, dan krijg je een heel normaal **wanneer**-**anders**-blok.

Blokmodificatie

In de gereedschapsbalk verschijnt alleen het enkelvoudige **wanneer**-blok en het **wanneer**-**anders-**blok:



Je moet op (+)-symbool klikken om **anders-wanneer**- en **anders**-clausules toe te voegen. Met het (-)-symbool kunnen **anders-wanneer**-clausules weer worden verwijderd:



Let erop dat de vormen van de blokken het toevoegen van een willekeurig aantal **anders-wanneer**-subblokken mogelijk maken, echter slechts een **wanneer**-blok.

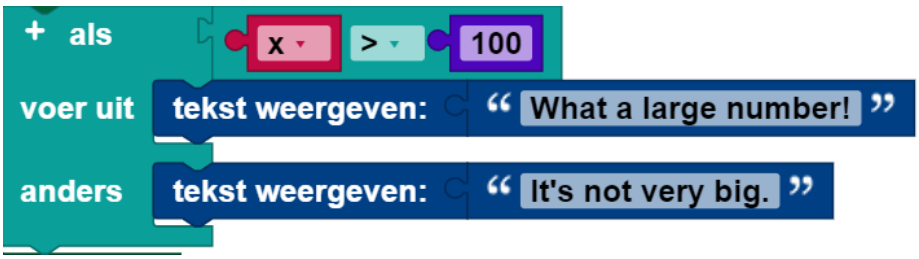
Booleaanse logica

Booleaanse logica is een eenvoudig wiskundig systeem dat twee waarden kent:

- **waar**
- **onwaar**

Logische blokken in ROBO Pro Coding zijn er normaal gesproken voor om voorwaarden en loops te controleren.

Hier een voorbeeld:



Wanneer de waarde van de variabele x groter is dan 100, is de voorwaarde waar en verschijnt de tekst "Wat een hoog cijfer!". Wanneer de waarde van x niet groter is dans 100, is de voorwaarde onwaar en verschijnt de melding "Dat is niet erg hoog.". Booleaanse waarden kunnen ook in variabelen worden opgeslagen en aan functies worden doorgegeven, net als getallen, tekst en lijstwaarden.

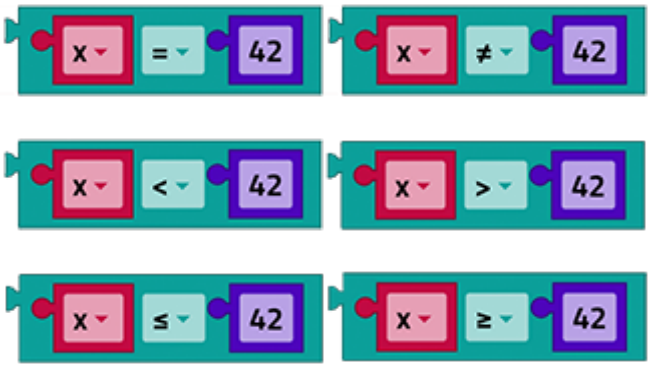
Wanneer een blok een Booleaanse waarde verwacht, wordt een ontbrekende invoer als **onwaar** geïnterpreteerd. Waarden die niet Booleaans zijn kunnen niet rechtstreeks daar worden ingevoerd waar Booleaanse waarden worden verwacht. Het is echter wel mogelijk (maar niet raadzaam) om een waarde die niet Booleaans is in een variabele op te slaan en deze dan in de voorwaardeninvoer in te voegen. Deze methode wordt niet aanbevolen en het gedrag kan in toekomstige versies van ROBO Pro Coding veranderen.

Waarden

Een afzonderlijk blok met een dropdown-lijst, die of **waar** of **onwaar** aangeeft, kan worden gebruikt om een Booleaanse waarde op te vragen:

Vergelijkingsoperatoren

Er zijn zes vergelijkingsoperatoren. Aan iedere vergelijkingsoperator worden twee invoeren (normaal gesproken twee getallen) overgedragen en de vergelijkingsoperator retourneert de melding **waar** of **onwaar**, afhankelijk hoe de invoeren met elkaar worden vergeleken.



De zes operatoren zijn: gelijk aan, niet gelijk aan, kleiner dan, groter dan, kleiner dan of gelijk aan, groter dan of gelijk aan.

Logische operatoren

Het **en**-blok retourneert dan en alleen dan de melding **waar**, wanneer zijn beide ingangswaarden waar zijn.

Het **of**-blok retourneert de melding **waar**, wanneer minimaal een van zijn beide ingangswaarden waar is.

niet

Het **niet**-blok zet zijn Booleaanse invoer om in het tegendeel. Bijvoorbeeld is het resultaat van:

onwaar.

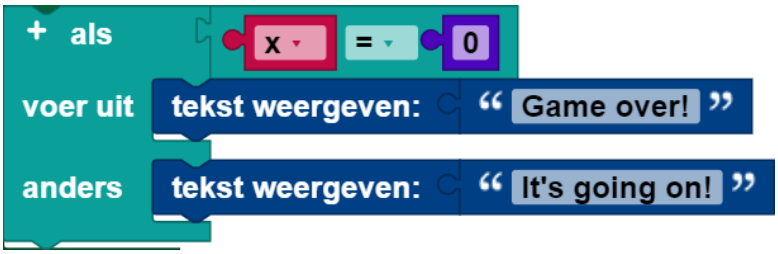
Wanneer niets wordt ingevoerd wordt de waarde **waar** aangenomen, zodat het volgende blok de waarde **onwaar** oplevert:

Er wordt echter niet geadviseerd om niets in te voeren.

trio-operator

De trio-operator gedraagt zich als een miniatuur-**wanneer-anders**-blok. Hij neemt drie ingangswaarden op, waarbij de eerste ingangswaarde de te testen Booleaanse voorwaarde is, de tweede ingangswaarde de waarde die geretourneerd moet worden, wanneer de test als **waar** wordt gezien en de derde ingangswaarde is de waarde die moet worden geretourneerd wanneer de test als onwaar wordt gezien. In het onderstaande voorbeeld wordt de variabele **kleur** op rood gezet, wanneer de variabele **x** minder is dan 10, in het andere geval wordt de variabele **kleur** op groen gezet.

Een trioblok kan altijd door een **wanneer-anders**-blok worden vervangen. De onderstaande twee voorbeelden zijn precies gelijk.



Loops

Het bereik "Besturing" bevat blokken die aansturen of andere blokken, die in hun inhoud zijn opgenomen, worden uitgevoerd.???? Er zijn twee soorten besturingsblokken: **wanneer-anders**-blokken (die op een eigen pagina beschreven worden) en blokken die aansturen hoe vaak de inhoud moet worden uitgevoerd. De laatst worden loops genoemd, omdat hun inhoud, ook loop-systeem of systeem genaamd, (mogelijkerwijs) meerdere malen worden herhaald. Iedere doorloop van een loop wordt als iteratie (herhaling) omschreven.

Blokken voor het aanmaken van loops

permanent herhalen

Het blok **permanent herhalen** voert de code in zijn systeem net zolang uit tot het programma eindigt.

herhaal

Het blok **herhaal** voert de code in zijn systeem net zo vaak uit als aangegeven. Het volgende blok geeft bijvoorbeeld tien keer "Hallo!" uit:

herhaal-net zolang

Stel jezelf een spelletje voor waarbij een speler een dobbelsteen werpt en alle geworpen waarden bij elkaar optelt zolang som lager is dan 30. De onderstaande blokken implementeren dit spel:

1. een variabele genaamd **in totaal** bevat een beginwaarden van 0.
2. De loop begint met een controle of **in totaal** minder is dan 30. Zo ja, dan worden de blokken in het systeem doorlopen.
3. Er wordt een toevalsgetal in een gebied van 1 tot 6 gegenereerd (om een worp met de dobbelsteen te simuleren) en in een variabele genaamd **geworpen** opgeslagen.
4. Het geworpen getal wordt aangegeven.
5. De variabele **in totaal** wordt met de **geworpen** waarde verhoogd.
6. Wanneer het einde van de loop is bereikt gaat de besturing terug naar stap 2.

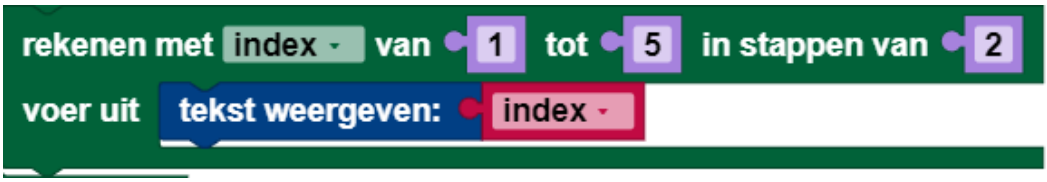
Nadat de loop is beëindigd worden alle opeenvolgende blokken (niet weergegeven) doorlopen. In het voorbeeld eindigt de loop-doorloop nadat een bepaald aantal toevalsgetallen in het bereik van 1 tot 6 zijn uitgegeven en de variabele **in totaal** heeft dan als waarde de som van deze getallen, die tenminste 30 moet zijn.

herhaal-tot

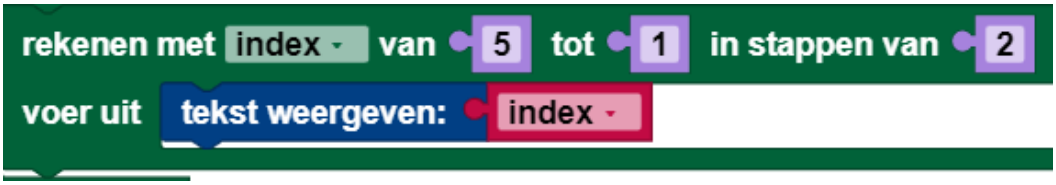
herhaal net zolang-loops herhalen hun systeem, **net zolang** tot aan een voorwaarde is voldaan. **herhaal tot**-loops zijn identiek met dat verschil dat zij hun systeem net zolang herhalen, **tot** aan een bepaalde voorwaarde is voldaan. De onderstaande blokken zijn gelijkwaardig aan het voorgaande voorbeeld omdat de loop draait tot **in totaal** hoger of gelijk aan 30 is.

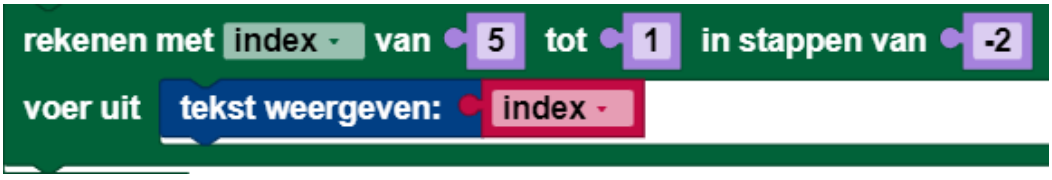
tellen-van-tot

Bij de **tellen-van-tot**-loop verhoogt een variabele de waarde, beginnend met een eerste waarde, eindigend met een tweede waarde en in stappen van een derde waarde, waarbij het systeem voor elke waarde van de variabele eenmaal wordt uitgevoerd. Het volgende programma geeft bijvoorbeeld de getallen 1, 3 en 5 uit.



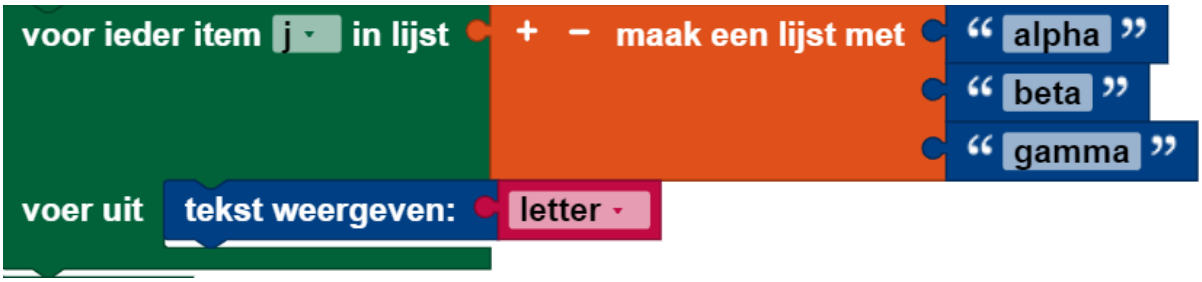
Zoals beide volgende loops laten zien, die elk de getallen 5, 3 en 1 uitgeven, kan deze waarde hoger zijn dan de tweede. Het gedrag is hetzelfde, ongeacht of het incrementele bedrag (derde waarde) positief of negatief is.





voor allemaal

Het blok **voor allemaal** is vergelijkbaar met de **tellen-van-tot**-loop, behalve dat hij in plaats van de loop-variabele in een numerieke volgorde de waarden uit een lijst een voor een gebruikt. Het volgende programma geeft elk element uit de lijst "alfa", "beta", "gamma" uit:



Loop-afbreekblokken

De meeste loops worden net zolang doorlopen tot aan de afbreekvoorwaarde (bij **herhaal**-blokken) is voldaan of totdat alle waarden van de loop-variabelen zijn aangenomen (bij **tellen met**- en **voor allemaal**-lussen). Twee zelden benodigde, maar soms nuttige blokken bieden extra mogelijkheden voor de besturing van het loop-gedrag. Ze kunnen bij iedere soort loop worden gebruikt, ook wanneer het onderstaande voorbeeld het gebruik bij de **voor iedereen**-loop laat zien.

ga-door-met-volgende-iteratie

ga-door-met-volgende-iteratie zorgt ervoor dat de resterende blokken in het loop-systeem worden overgeslagen en de volgende iteratie van de loop begint.

Het volgende programma geeft bij de eerste iteratie van de loop "alfa” aan. Bij de tweede iteratie wordt het blok **ga-door-met-volgende-iteratie** uitgevoerd, waardoor het uitgeven van "beta" wordt overgeslagen. Bij de laatste iteratie wordt "gamma" afgedrukt.

loop-afbreken

Het blok **loop-afbreken** biedt de mogelijk om een loop voortijdig te beëindigen. Het volgende programma geeft bij de eerste iteratie "alfa" weer en wordt bij de tweede iteratie uit de loop afgebroken wanneer de loop-variabele gelijk is aan "beta". Het derde punt in de lijst wordt nooit bereikt.


voor ieder item j in lijst
+ − maak een lijst met
“ alpha ”
“ beta ”
“ gamma ”
voer uit
+ als
letter
=
“ beta ”
voer uit
uit lus breken
anders
tekst weergeven:
letter

Wiskunde

De blokken uit de categorie Wiskunde worden gebruikt om berekeningen te maken. De resultaten van de berekeningen kunnen bijvoorbeeld als waarden voor variabelen worden gebruikt. De meeste wiskunde-blokken hebben betrekking op algemene wiskundige berekeningen en moeten zelfverklarend zijn.

Blokken

Getallen

Gebruik het cijferblok om een willekeurig getal in je programma in te voegen of een variabele van dit getal als waarde toe te wijzen. Dit programma wijst het getal 12 toe aan de variabele **leeftijd** :

Eenvoudige berekeningen

Dit blok heeft de structuur waarde-operator-waarde. Als operatoren zijn de rekenwijzen **+**, **-**, **÷**, **×** en **^** beschikbaar. De operator kan via het dropdown-menu worden geselecteerd. Hij kan rechtstreeks op getallen of ook op waarden van variabelen worden toegepast. Voorbeeld:

it blok geeft het resultaat 144 (12^2) weer.

Speciale berekeningen

Dit blok gebruikt de, via het dropdown-menu geselecteerde, rekenwijze op het daarachter geplaatste getal of op de waarden van de daarachter geplaatste variabele. De beschikbare berekeningen zijn:

- vierkantswortel,,
- bedrag,
- natuurlijke logaritme,
- decimale logaritme,
- exponentiële functie met de basis e (e^1, e^2,...),
- exponentiële functie met de basis 10 (10^1, 10^2,...),
- voortekenwissel (vermenigvuldiging met -1).

e is hierbij de wiskundige constante. Dit blok trekt de vierkantswortel uit 16 en zet de variabele **i** op het resultaat.

Trigonometrische functies

Dit blok werkt hetzelfde als het hiervoor beschreven blok, met dat verschil dat als berekeningen de trigonometrische functies sinus, cosinus, tangens en hun omkeerfuncties worden gebruikt. Het aangegeven getal

of waarde van de aangegeven variabele wordt aldus in de in het dropdown-menu geselecteerde functie ingezet en het resultaat kan dan in het programma verder worden verwerkt. Bovendien is er nog het blok **arctan2 of X: ... Y: ...**, die het mogelijk maakt, met behulp van twee reële getallen (in te zetten als X en Y) een functiewaarde van de arctan2 in het bereik van 360° uit te laten geven.

Veel gebruikte constanten

Dit blok werkt net als het cijferblok, alleen wordt hier het getal niet zelf aangegeven. In plaats daarvan zijn veel gebruikte constanten (bijv. ?) vooraf opgeslagen. De constante kan via het dropdown-menu worden geselecteerd.

Rest van een deling

Het blok **rest van ...** wordt gebruikt om de rest van een deling uit te geven. Dit programma wijst de variabele **Rest** van de rest van de deling 3:2, dus 1, toe:

Afronden

Met het blok **afronden ...** kan een aangegeven decimaal getal of de waarde van een aangegeven variabele op een heel getal worden afgerond. Daarbij kun je in het dropdown-menu uit drie opties kiezen:

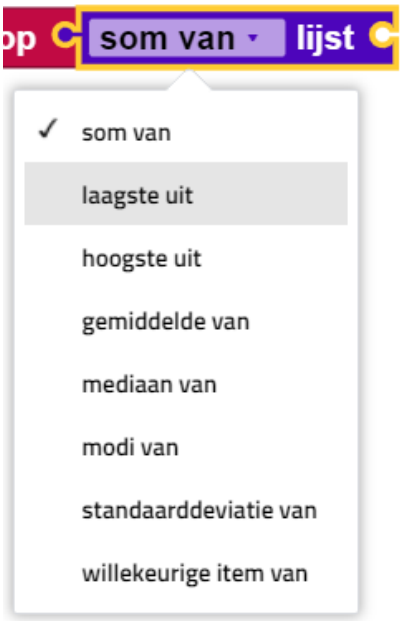
- met "afronden" commercieel afgerond (bijv. 4,5 wordt 5)
- met "afronden naar boven" wordt afgerond (bijv. 5,1 wordt 6)
- met "afronden naar beneden" wordt afgerond (5,9 wordt bijv. 5).

Analyse van lijsten

Met het blok **... van de lijst** kun je

- met "totaal" het totaal van alle waarden in een lijst,
- met "min" de laagste waarde van een lijst,
- met "max" de hoogste waarde van een lijst,
- met "gemiddelde waarde" de gemiddelde waarde van alle waarden in een lijst,
- met "mediaan" de mediaan van een lijst,
- met "modale waarde" de vaakst voorkomende waarde in een lijst,
- met "standaardafwijking" de standaardafwijking van alle waarden in een lijst,
- met "toevalswaarde" een toevallige waarde uit een lijst

laten uitgeven. Al deze opties kun je via het dropdown-menu van het blok selecteren.



De invoerwaarde beperken

Het blok **beperken ... van ... tot ...**maakt het mogelijk om de invoerwaarden tot een bepaalde interval te beperken. Voordat een invoerwaarde verder worden verwerkt, wordt getest of deze binnen een vastgelegde interval ligt. Er zijn drie opties hoe een ingevoerde waarde wordt gebruikt:

- De waarde valt binnen de interval, wordt dus onveranderd doorgegeven.
- De waarde ligt onder de onderste grens van de interval, dus wordt deze onderste grens doorgegeven.
- De waarde ligt boven de bovenste grens van de interval, dus wordt deze bovenste grens doorgegeven.

In dit voorbeeld wordt het blok gebruikt om de waarde van de variabele **toerental** tot de door de motor ondersteunde toerentallen te beperken:

Toevallige waarden genereren

De beide blokken **toevallig getal van ... tot...** en **toevallige breuk** geven een toevallige waarde uit. Daarbij geeft het **toevallige getal van ... tot...- blok** een getal uit de gedefinieerde interval uit. Het blok **toevallige breuk** geeft daarentegen een waarde tussen 0,0 (ingesloten) en 1,0 (uitgesloten) uit.

Teksten

Voorbeelden voor teksten zijn:

"Ding 1"
"12. maart 2010"
"" (lege tekst)
Tekst kan letters (groot of klein geschreven), getallen, leestekens, andere symbolen en spaties bevatten.

Blokken

Opstellen van tekst

Het onderstaande blok genereert de tekst "Hallo" en slaat deze op in de variabelen onder **Groet:**

Het blok **stel tekst op vanuit** combineert de waarde van de variabele **Groet** en de nieuwe tekst "wereld" tot de tekst"Hallowereld". Let erop dat er tussen de beide teksten geen spatie staat, omdat in beide oorspronkelijke teksten ook geen spatie stond.

Om het aantal tekstinvoeren te verhogen moet je het (+)-symbool aanklikken. Om de laatste invoer te verwijderen moet je het (-)-symbool aanklikken.

Tekst wijzigen

Het blok **bij ... toevoegen** voegt de aangegeven tekst aan de aangegeven variabele toe. In dit voorbeeld verandert hij de waarde van de variabele **Groet** van "Hallo" in "Hallo, daar!":

Lengte van de tekst

Het blok **lengte van** telt het aantal tekens (letters, getallen, enz.) die in een tekst zitten. De lengte van "Wij zijn #1!" is 12, en de lengte van de lege tekst is 0.

Controleren op lege tekst

Het component **is leeg** controleert of de aangegeven tekst leeg is (de lengte 0 heeft). Het resultaat is in het eerste voorbeeld **waar** en in het tweede voorbeeld **onwaar**.

Zoeken naar tekst

Deze blokken kunnen worden gebruikt om te controleren of een tekst in een andere tekst voorkomt en zo ja, waar deze voorkomt. Zo wordt bijvoorbeeld gevraagd wanneer voor het eerst een "a" in "Hallo" voorkomt, het resultaat is 2.

Hier wordt gevraagd wanneer voor het laatst een "a" in "Hallo” voorkomt, wat eveneens een 2 oplevert:

Ongeacht of het eerst of laatst voorkomende wordt gekozen, levert dit blok het resultaat 0 op, aangezien "Hallo" geen "z" bevat.

Extraheren van tekst

Extraheren van een afzonderlijk teken

Dit geeft "b", de tweede letter in "abcde":

Dit geeft "d", de voorlaatste letter in "abcde":

Dit geeft "a", de eerste letter in "abcde":

Dit geeft "e", de laatste letter in "abcde":

Dit bevat de 5 letters in "abcde" met dezelfde waarschijnlijkheid:

Geen van hen verandert de tekst van waaruit wordt geëxtraheerd.

Extraheren van een tekstgebied

Met het blok **in tekst ... lever tekenketting** kan een tekstgebied worden geëxtraheerd, dat hetzij met:

- letter #
- letter # vanaf einde
- eerste letter

start en met:

- letter #
- letter # vanaf einde
- laatste letter

eindigt.

In het volgende voorbeeld wordt "abc" geëxtraheerd:

Aanpassen van groot/klein schrijven van de tekst

Dit blok genereert een versie van de invoertekst die hetzij

- GROOT GESCHREVEN (alle letters als hoofdletters) of
- klein geschreven (alle letters zijn klein geschreven) of
- substantief (eerste letter hoofdletter, alle andere letters klein geschreven).

Het resultaat van het volgende blok is "HALLO":

Dit is niet van toepassing op niet-alfabetische tekens. Let erop dat dit blok op tekst in talen zonder groot- en klein geschreven letters, zoals bijv. het Chinees, niet werkt.

Trimmen (verwijderen) van spaties

Het volgende blok verwijdert, afhankelijk wat in het dropdown-menu (klein driehoekje) wordt ingesteld, spaties:

- aan het begin van de tekst
- aan het einde van de tekst
- aan beide kanten van de tekst

Het resultaat van het volgende blok is "Hi jij":

Dit heeft geen invloed op spaties middenin de tekst.

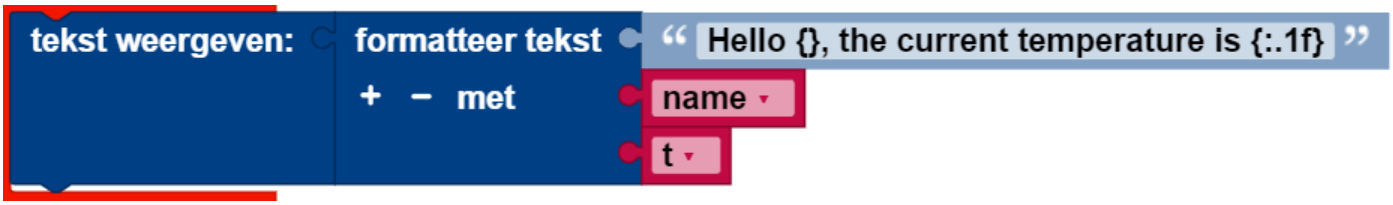
Tekst uitvoeren

Het blok **geeft uit** zorgt er voor dat de invoerwaarde op het consolevenster wordt weergegeven:

In geen geval wordt het naar de printer gestuurd, zoals de naam misschien laat vermoeden.

Tekst weergeven met formattering

Met het blok **formatteer tekst** kunnen teksten met variabeleninhoud geformatteerd worden weergegeven. Daarbij worden alle wildcards **{}** in de tekst door de inhoud van de na de tekst bijgevoegde variabelen vervangen. Binnen de accolades kan een formattering worden aangegeven. De formattering **{:.1f}** geeft bijv. alleen de eerste decimaal achter de komma in de variabele **t** weer.



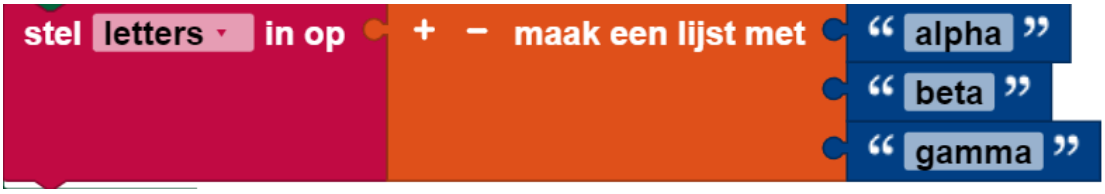
Lijsten

Net als in de alledaagse taal staat ook in ROBO Pro Coding een lijst van een geordende verzameling van elementen, zoals bijv. een "To-do"-lijst of een inkooplijst. Elementen in een lijst kunnen van een willekeurig type zijn en dezelfde waarde kan meerdere malen in een lijst voorkomen.

Opstellen van een lijst

maak lijst met

Met het blok **maak lijst met** kun je de beginwaarden in een nieuwe lijst aangeven. In dit voorbeeld wordt een lijst van woorden gemaakt en in een variabele genaamd **Letters** opgeslagen.

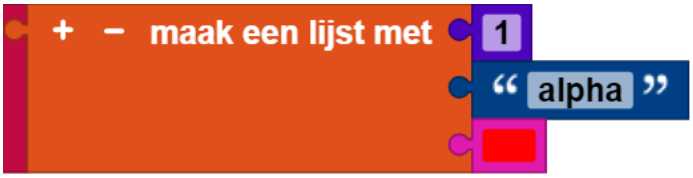


Wij omschrijven deze lijst als ["alfa", "beta", "gamma"].

Dit geeft het opstellen van een lijst van **getallen** weer:

Zo wordt een lijst van **kleuren** opgesteld:

Het is minder gebruikelijk, maar het is mogelijk om een lijst met waarden van verschillende te maken:



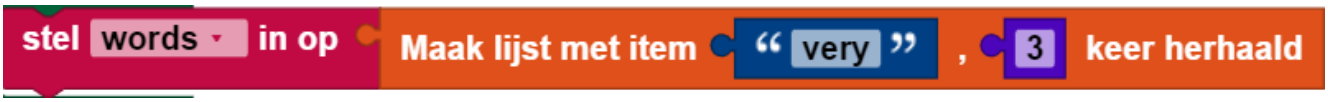
Aantal ingangen veranderen

Om het aantal ingangen te veranderen, moet je op het tandwielsymbool klikken of deze aanraken. Daardoor wordt een nieuwe venster geopend. Je kunt subblokken van elementen van de linker kant van het venster naar het lijstenblok aan de rechter kant trekken, om een nieuwe ingang toe te voegen.

Terwijl het nieuwe element in dit voorbeeld onderaan is toegevoegd, kan het overal worden toegevoegd. Op vergelijkbare wijze kunnen ongewenste subblokken van elementen uit het lijstenblok naar links worden getrokken.

Lijst met element opstellen

Met het blok **stel lijst met element op** kun je een lijst opstellen die het aangegeven aantal kopieën van een element bevat. De onderstaande blokken zetten bijvoorbeeld de variabele **woorden** op de lijst ["heel", "heel", "heel"].



De lengte van een lijst controleren

is leeg

De waarde van een **is leeg**-blok is **waar**, wanneer de invoer de lege lijst is en **onwaar**, wanneer het iets anders is. Is deze invoer **waar**? De waarde van het volgende blok zou **onwaar** zijn, omdat de variabele kleuren niet leeg is: Deze heeft drie elementen.

Let op de gelijkenis met het **is leeg**-blok voor tekst.

Lengte van

De waarde van het blok **lengte van** is het aantal elementen dat in de als invoer gebruikte lijsten is opgenomen. De waard van het volgende blok zou bijvoorbeeld 3 zijn, omdat **kleur** drie elementen heeft.

Let erop dat het blok **lengte van** aangeeft hoeveel elementen in de lijst zijn opgenomen en niet hoeveel verschillende elementen erin inzetten. Zo heeft bijvoorbeeld de volgende de waarde 3, hoewel **woorden** uit drie kopieën uit dezelfde tekst bestaat.

Let op de gelijkenis met het **lengte van**-blok voor tekst.

Zoeken van elementen in een lijst

Deze blokken vinden de positie van een element in een lijst. Het volgende voorbeeld heeft de waarde 1, omdat het eerste voorkomen van "heel" aan het begin van de woordenlijst staat (["heel", "heel", "heel"]).

Het resultaat van de volgende is 3, omdat het laatste voorkomen van "heel" in **woorden** op positie 3 staat.

Wanneer het element nergens in de lijst voorkomt, is het resultaat van de waarde 0, zoals in dit voorbeeld:

Deze blokken werken hetzelfde als de blokken voor het vinden van letters in tekst.

Openen van elementen in een lijst

Openen van een afzonderlijk element

Denk aan de definitie van de lijst **kleuren**:

Het volgende blok bevat de kleur blauw, omdat dat het tweede element in de lijst is (vanaf links beginnend te tellen):

Deze bevat groen omdat dat het tweede element is (vanaf rechts beginnend te tellen):

Deze bevat het eerste element, rood:

Deze bevat het laatste element, geel:

Dit selecteert toevallig een element uit de lijst, waarbij met dezelfde waarschijnlijkheid een van de elementen, rood, blauw, groen of geel wordt weergegeven.

Openen en verwijderen van een element

Via het dropdown-menu wordt het blok **uit lijst ... openen** in het blok **uit lijst ... openen en verwijderen** gewijzigd die dezelfde weergave levert, maar ook de lijst verandert:

Dit voorbeeld zet de variabele **eerste letter** op "alfa" en laat de resterende letters (["beta", "gamma"]) in de lijst.

Een invoer verwijderen

Wanneer je in het dropdown-menu **verwijderen** kiest, verdwijnt de lip links van het blok:

Daarmee wordt het eerste element uit **letters** verwijderd.

Een sublijst openen

Het blok **uit lijst ... Sublijst openen** lijkt op het blok in **uit lijst ... openen** met dat verschil dat deze een uittreksel van een sublijst maakt en niet een afzonderlijk element. Er zijn meerdere opties, om het begin en het einde van de sublijst aan te geven:

In dit voorbeeld wordt een nieuwe lijst **eerste letter** opgesteld. Deze nieuwe lijst heeft twee elementen: ["alfa", "beta"].

Let erop dat dit blok de oorspronkelijke lijst niet verandert.

Toevoegen van elementen aan een lijst

Elementen van een lijst verwijderen

Het blok **in lijst ... vervang** vervangt het element op een bepaalde plaats in een lijst door een ander element.

De betekenis van de afzonderlijke dropdown-opties vindt je in het vorige hoofdstuk.

Het volgende voorbeeld zorgt voor twee dingen:

1. De lijst **woorden** wordt met 3 elementen opgesteld: ["heel", "heel", "heel"].
2. Het derde element in de lijst wordt door "goed" vervangen. De nieuwe waarde van **woorden** is ["heel", "heel", "goed"]

Elementen op een bepaalde plaats in een lijst toevoegen

Het blok **in lijst ... toevoegen bij** wordt via het dropdown-menu van het **in lijst ... vervang**-blok geopend:

Deze voegt een nieuw element op de aangegeven plaatst in de lijst toe, en wel voor het element, dat eerder op deze plaats stond. Het volgende voorbeeld (dat op een eerder voorbeeld is gestoeld) doet drie dingen:

1. De lijst **woorden** wordt met 3 elementen opgesteld: ["heel", "heel", "heel"].
2. Het derde element in de lijst wordt door "goed" vervangen. De nieuwe waarde van **woorden** is daarmee ["heel", "heel", "goed"].
3. Het woord "Wees" wordt aan het begin van de lijst toegevoegd. De uiteindelijke waarde van **woorden** is daarmee ["Wees", "heel", "heel", "goed"].

De tekensketting verdelen en in lijsten samenvoegen

Een lijst vanuit een tekst opstellen.

De bouwsteen **stel lijst op vanuit tekst** knipt de aangegeven tekst met behulp van een begrenzingsteken in delen:

In het bovenstaande voorbeeld wordt een nieuwe lijst geretourneerd, die drie tekststukken bevat: "311", "555" en "2368".

Een tekst vanuit een lijst opstellen.

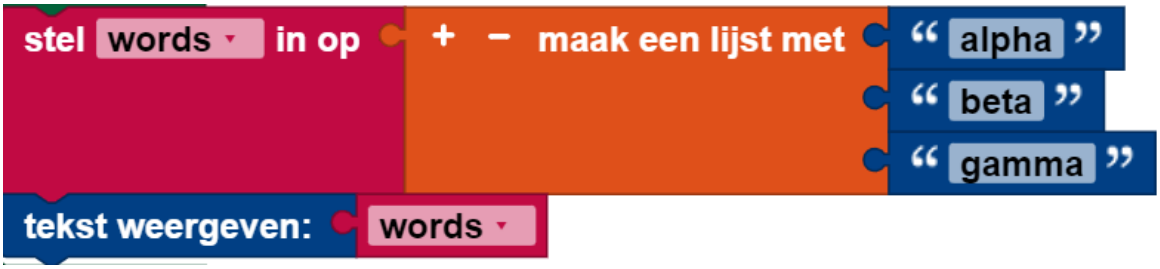
De bouwsteen **stel een tekst op uit lijst** voegt een lijst met behulp van een scheidingsteken tot één enkele tekst samen:

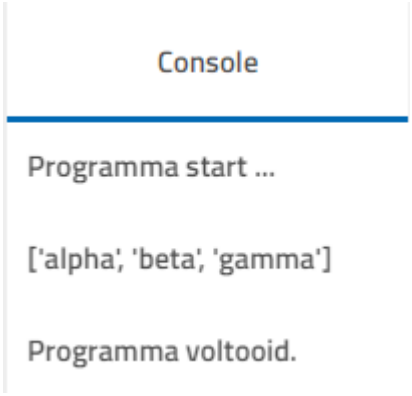
In het bovenstaande voorbeeld wordt een nieuwe tekst met de waarde geretourneerd: "311-555-2368".

Soortgelijke blokken

Afdrukken van een lijst

Met de bouwsteen **afdrukken** in de categorie Tekst kun je lijsten afdrukken. Het resultaat van het onderstaande programma is de afgebeelde console-afdruk:





Iets voor elke element in een lijst uitvoeren

Het blok **voor-iedereen** in de categorie Besturing voert een handeling voor elk element in de lijst uit. Dit blok drukt bijvoorbeeld elk element in de lijst apart af:

Daardoor worden de elementen niet uit de oorspronkelijke lijst verwijderd.

Zie ook de voorbeelden voor de loop-afbreekblokken.

Gebruik

De categorie Gebruik bevat bij ROBO Pro Coding de onderstaande soorten blokken:

- kleurkeuze
- wachten
- Python Code
- starten
- uitvoeren van functie

kleurkeuze

Dit blok dient als invoerwaarde, wanneer naar een kleur wordt gevraagd (bijvoorbeeld voor de kleurcompensatie door de camera). Door de kleur aan de klikken of aan te raken kan uit een kleurenpalet één van de 70 kleuren worden geselecteerd.

wachten

Wachten tot de tijd verstreken is

Het blok **wacht [] ...** voorkomt dat het programma gaat draaien voordat de wachttijd is verstreken. Daarbij kan in het dropdown-menu (kleine driehoekje) de tijdeenheid en in het invoerveld daarachter de gewenste lengte van de pauze worden gekozen.

Wachten met voorwaarde

Bij het blok **wachten tot** is de pauze niet gekoppeld aan de tijd maar of er aan een voorwaarde wordt voldaan (bijvoorbeeld of een toets is ingedrukt) De voorwaarde wordt aan het **wachten tot**-blok gehangen.

Python Code

Wanneer je een bestaande Python-code in ROBO Pro Coding wilt integreren kun je die in het **Python Code**-blok toevoegen. Het programma voert dan alles uit wat in het blok in Python is geschreven.

starten

Ook het blok **start wanneer** is aan een voorwaarde gekoppeld. Pas wanneer aan deze voorwaarde is voldaan wordt het in het bloksysteem staande programma gestart.

Uitvoeren van functie

Met het **voer functie ... in een thread uit** kan de geselecteerde functie in een afzonderlijke thread worden uitgevoerd. Deze maatregel kan er in vele gevallen voor zorgen dat een programma steeds op invoeren kan reageren en sneller wordt uitgevoerd.

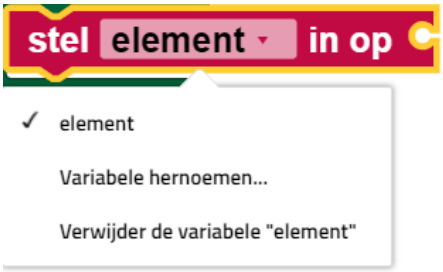
Variabelen

Wij gebruiken het begrip variabele zo zoals in de wiskunde en andere programmeertalen wordt gebruikt: een benoemde waarden die kan worden veranderd (gevarieerd). Variabelen kunnen op verschillende manieren worden opgesteld.

- Sommige blokken zoals **tel met** en **voor iedereen** gebruiken een variabele en definiëren hun waarden. Een traditioneel informaticabegrip voor dergelijke variabelen luidt loopvariabelen.
- Gebruikersgedefinieerde functies (ook "procedures" genoemd) kunnen invoeren definiëren, waardoor variabelen worden gegenereerd, die alleen binnen deze functie kunnen worden gebruikt. Dergelijke variabelen worden traditioneel als "parameters" of "argumenten" aangeduid.
- Gebruikers kunnen te allen tijde variabelen met het **instel**-blok wijzigen. Deze worden traditioneel als "algemene variabelen" aangegeven. Zij kunnen overal in de code van ROBO Pro Coding worden gebruikt.

Dropdown-menu

Wanneer je op het dropdown-symbool (kleine driehoekje) op een variabele klikt, verschijnt het onderstaande menu:



Het menu biedt onderstaande mogelijkheden.

- de weergave van de namen van alle beschikbare, in het programma gedefinieerde variabelen.
- "Variabele hernoemen...", dat wil zeggen het wijzigen van de naam van deze variabele, ongeacht waar deze in het programma verschijnt (wanneer deze optie wordt gekozen wordt de vraag om een nieuwe naam geopend)
- "Variabele wissen...", dat wil zeggen dat alle blokken worden verwijderd waar deze variabele naar verwijst, ongeacht waar deze in het programma voorkomen.

Blokken

Vastleggen

Het **instel**-blok wijst een waarde aan een variabele toe en maakt de variabele aan, mocht deze nog niet bestaan. Als voorbeeld wordt zo de waarde van de variabele **leeftijd** op 12 gezet:

Openen

Het **open**-blok levert de in een variabele opgeslagen waarden, zonder deze te veranderen:

Het is mogelijk, maar een slecht idee, om een programma te schrijven, waarin een **open**-blok voorkomt zonder een dienovereenkomstig instel-blok.

Wijzigen

Het **wijzig**-blok voegt een getal aan een variabele toe.

Het **wijzig**-blok is een afkorting voor de volgende constructie:

Voorbeeld

Bekijk de onderstaande voorbeeldcode eens:

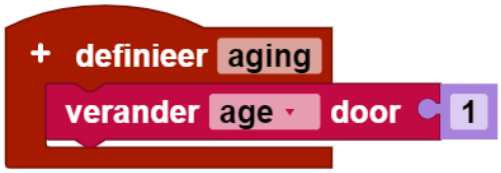
De eerste rij blokken genereert een variabele genaamd **leeftijd** en **stelt** de beginwaarde in op het getal 12. De tweede rij blokken **opent** dan de waarde 12 , telt daar 1 bij op en slaat het totaal (13) in de variabelen op. In de laatste regel wordt de volgende melding getoond: "Hartelijk gefeliciteerd! Je bent nu 13".

Functies

Functies hebben ten doel om delen van de code opnieuw bruikbaar te maken en daardoor de code in zijn geheel te structureren. Wanneer een functieblok wordt gevuld, verschijnt in het functiemenu een nieuwe blok dat dezelfde naam draagt als juist dit functieblok. Het is nu mogelijk om in het hoofdprogramma alleen nog het blok met de naam van de functie te gebruiken. Wanneer het programma wordt doorlopen, leidt dit blok naar de code in de gelijknamige functie en werkt deze af.

Enkelvoudige functie

Met het enkelvoudige blok kan een functie wordt ingesteld die de in het tekstveld ingevoerde naam draagt. De functie kan willekeurig veel variabelen bevatten die met behulp van het tandwielsymbool kunnen worden toegevoegd. De functie **ouder worden** telt 1 op bij de variabele **leeftijd** op:



De functie kan in het hoofdprogramma worden gebruikt:

Functie met geretourneerde waarde

Dit blok maakt het mogelijk om een functie met geretourneerde waarde op te stellen. Deze geretourneerde waarde kan in het hoofdprogramma verder worden gebruikt. Hier een voorbeeld: