

# 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 laatste 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.

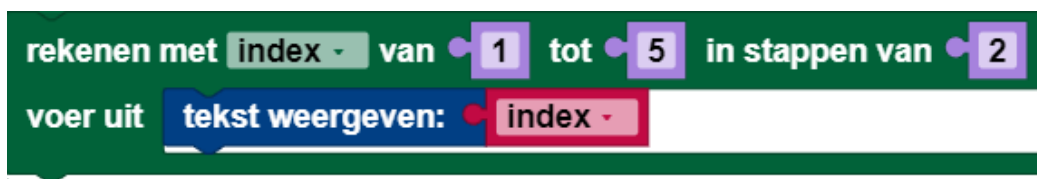
## herhaal-tot

**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 uitvoeren van "beta" wordt overgeslagen. Bij de laatste iteratie wordt "gamma" afgedrukt.



### loop-afbreken

Het blok **loop-afbreken** biedt de mogelijkheid 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.



Revision #6

Created 17 November 2021 20:58:04 by Admin

Updated 8 November 2024 15:12:06 by phuesing