



# PSD

## Inleiding

Computers en andere digitale apparatuur is tegenwoordig niet meer weg te denken in de huidige samenleving. Zonder programma's zijn deze apparaten echter niets waard. Het zijn de programma's die het mogelijk maken dat deze apparaten van alles kunnen.

## Programma's

Programma's zijn een serie instructies in een taal die de computer begrijpt en door de computer kunnen worden uitgevoerd. Zo'n verzameling instructies vormen een programma ofwel een algoritme.

## Algoritmen

Een Algoritme is een eindige serie instructies die vanuit een gegeven begintoestand naar een beoogd doel leiden. Dit doel kan zijn het uitvoeren van een bepaalde handeling of het oplossen van een probleem.

De eigenschappen van algoritmen zijn:

- Er moet een eindig aantal stappen worden doorlopen.
- De volgorde (**sequentie**) waarin de stappen moeten worden doorlopen kan van belang zijn.
- Soms moeten stappen worden herhaald (**iteratie**).
- Soms moeten er keuzes worden gemaakt (**selectie**) hoe het algoritme verder moet gaan.

## ***Reeksen van logische procedures om problemen op te lossen in een eindig aantal stappen.***

Stel men wil een ei bakken. Op zichzelf iets vrij eenvoudigs, maar als we het gaan opschrijven als algoritme dan krijgen we het volgende:

1. Vuur aansteken
2. Zet pan op het vuur
3. Voeg boter in de pan toe
4. Boter laten smelten
5. Breek ei en plaats deze in de pan
6. Bak het ei gedurende 3 minuten
7. Vuur uitmaken



# PSD

Het algoritme bestaat uit 7 **deelalgoritmen**, welke als **elementair** zijn. Hieronder verstaan men dat iedereen weet wat er mee wordt bedoeld en de omschrijving is **eenduidig** (*precies één betekenis*).

Bij dit algoritme wordt er vanuit gegaan dat de persoon al weet te werken met een gasfornuis. Dit is voorkennis. Zou de persoon deze kennis niet bezitten dan zou het algoritme verfijnd moeten worden.

De stap "vuur aansteken" zou verder gedetailleerd kunnen worden als volgt:

- Neem de aansteker uit de lade
- Draai het gas open
- Druk op de aansteker
- Houdt de aansteker bij het gas

Er wordt hier er vanuit gegaan dat er een aansteker gebruikt wordt, maar dit had even goed lucifers kunnen zijn of een fornuis met een elektrische ontsteking.

## Talen en schema's

Doordat er verschillende programmeertalen zijn is de behoefte ontstaan naar een algemene manier om algoritmen/programma's weer te geven. Hiervoor wordt het **Programma Structuur Diagram (Nassie-Shneiderman diagram)** gebruikt. Er bestaan nog meer schema technieken, zoals "Flow chart" die ook algoritmen schematisch weergeven.

De **PSD** schema's bestaan uit:

- Rechthoeken
- Driehoeken
- Tekst

Door dat schematechnieken als PSD onafhankelijk zijn van een programmeertaal, kunnen de schema's vertaald worden naar elke programmeertaal.



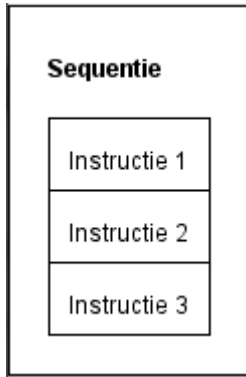


# PSD

## PSD structuren

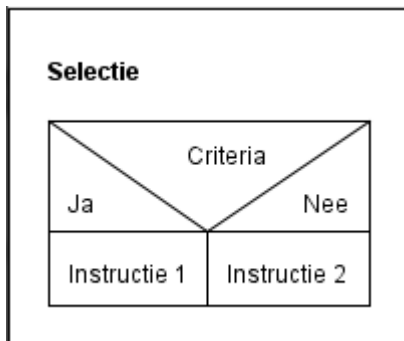
Er zijn drie basisstructuren waar alle programma's uit bestaan. Deze structuren zijn:

### 1. Sequentie



In elke recht hoek staat één instructie. Instructies worden in volgorde, van boven naar beneden, achter elkaar uitgevoerd.

### 2. Selectie



Selectie-structuren ook wel keuze structuren (*als ... dan ... anders ...*) worden gebruikt om het algoritme een bepaalde weg te laten volgen.

Er zijn verschillende varianten van deze structuur mogelijk in de meeste programmeertalen.

Wanneer er meerder selectie structuren achterelkaar of genest gebruikt moeten worden gebruikt men een case ofwel switch structuur. Deze maken het gebruik van meerdere opeenvolgende selectie structuren overzichtelijker en beter leesbaar.



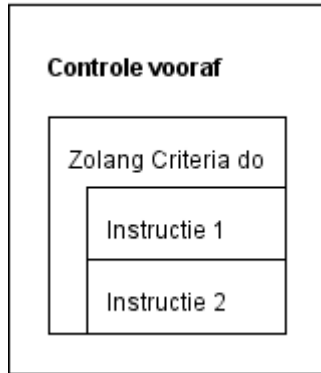
# PSD

## 3. Iteratie

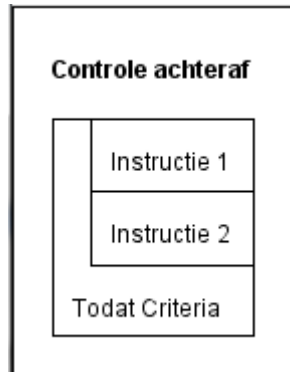
De iteratie-structuur ook wel herhaal structuur, wordt gebruikt om een serie opdrachten te herhalen tot dat een bepaalde voorwaarde niet meer wordt voldaan.

De drie iteratie-structuren zijn:

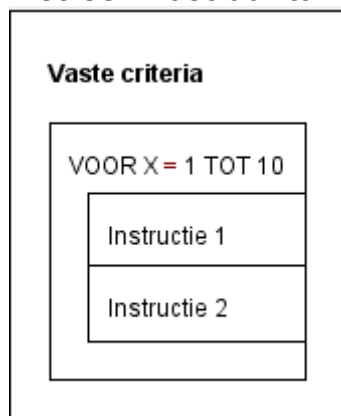
1. Met controle vooraf.



2. Met controle achteraf.



3. Met een vast aantal herhalingen.





# PSD

De 2<sup>de</sup> wordt eigenlijk afgeraden om te gebruiken daar het vaak voor problemen zorgt in de programma's.

## Opdrachten

De tekst die gebruikt wordt in een PSD bestaat uit de opdrachten die mogelijk zijn.

- Variabelen
- LEES
- SCHRIJF
- ZOLANG ... DOE/DOE ...TOTDAT
- VOOR TOT STAP VOLGENDE

## Variabelen

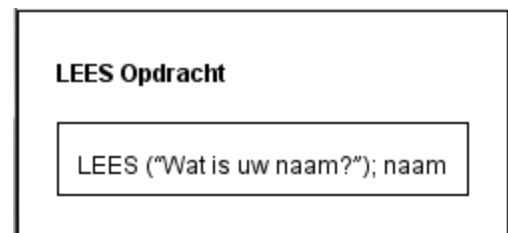
Dit zijn geheugenlocaties in het werkgeheugen van een computer. Variabelen hebben een naam, in letters en cijfers aaneengesloten, die voor elke geheugenlocatie in de computer uniek is (net als een telefoonnummer).

```
var1 = 5  
var2 = 2  
var3 = var1 + var2
```

Het "=" teken is de toekenningsoperator ook wel het toekenningsteken.

## De opdracht **LEES**

De gebruiker van het programma krijgt een vraag/opdracht dat hij/zij moet invoeren. De ingevoerde waarde wordt dan opgeslagen in een variabele in het werkgeheugen van de computer.



```
LEES ("Wat is uw naam?"); naam
```

## De opdracht **SCHRIJF**

Met deze opdracht kan de computer tekst op het scherm laten zien aan de gebruiker.

```
SCHRIJF "Hallo ";naam;". Welkom."
```



# PSD

## De opdracht **ZOLANG ... DOE**

De ZOLANG opdracht wordt gebruikt in de iteratie structuur voor herhaalstructuren met controle vooraf en achter af.

Het geen dat na de ZOLANG opdracht staat is de criteria ook wel voorwaarde waaraan de lus moet voldoen om te blijven herhalen.

ZOLANG A > 10 DOE

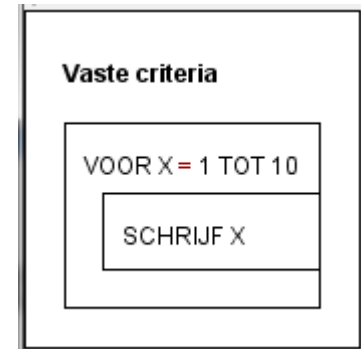


## De opdracht **VOOR TOT**

De VOOR TOT opdracht is een herhaal opdracht waarbij precies wordt aangegeven hoeveel keer een lus herhaald moet worden.

VOOR X = 1 TOT 10  
SCHRIJF X

Deze opdracht zal een lijst op het scherm afdrukken van 1 t/m 10. 1, 2, 3, etc.



## De opdracht **VOOR TOT STAP**

Het verschil met deze opdracht is dat je kunt aangeven in hoeveel stappen de grenswaarde bereikt moet worden.

VOOR X = 1 TOT 10 STAP 2  
SCHRIJF X

Deze opdracht drukt de volgende reeks getallen op het scherm af: 1, 3, 5, 7 en 9.



Bij de het stap deel van de instructie kunnen ook negatieve getallen gebruikt worden, om bijvoorbeeld af te tellen van 10 naar 1. Dus de reeks: 10, 9, 8, ... , 2, 1.

