Er zijn verschillende soorten structuren voor programmeren. Het kiezen van een taal bepaald al voor een groot deel welke structuur je zult gebruiken. Het is erg handig om te weten welke structuren er zijn en waar welke structuur handig voor is.

Er zijn 4 erg belangrijke soorten structuren die je moet kennen.

Imperatief

Imperatief programmeren is de eenvoudigste vorm. Je schrijft regel voor regel de code en het wordt van boven naar onder uitgevoerd. De code die je zover hebt geschreven is imperatief.

## Functioneel

Bij functioneel programmeren maak je gebruik van functies. Functies zijn blokken code die hergebruikt kunnen worden. Een voorbeeld is een functie zoals kwadrateer, die een waarde aanneemt (input) en een resultaat geeft (output), in dit geval is dat het kwadraat van de ingevoerde waarde.

Functioneel programmeren zal in de theorie vaker tegenkomen.

## Object-georiënteerd

Object-georiënteerd programmeren is een uitbereiding op het imperatief programmeren. Je gaat zoveel denken in objecten, deze werken net zoals in de echte wereld. Een voorbeeld is een boom als object, deze boom heeft bepaalde kenmerken oftewel properties. Een property kan bijvoorbeeld het aantal bladeren zijn of de hoeveelheid water dat de boom vast kan houden.

Je kunt ook overerving hebben oftewel inheritance. Dit betekent dat een object een soort van ander object is. Een voorbeeld is dat een Amerikaanse eik een soort boom is. Deze eik zal dan veel in gemeen hebben met bomen in het algemeen en kan dan gebruikmaken van de omschrijvingen van bomen.

Object-georiënteerd programmeren is al lastiger en zal niet verder behandeld worden in deze methode.

## Logisch

Bij logisch programmeren geef je feiten en regels die gebruikt worden om een bepaald antwoord te geven. Deze structuur komt maar heel weinig voor en zal daarom ook niet verder behandeld worden.