



# Haskell versus PHP

## Een korte introductie van beide talen

Haskell en PHP zijn twee totaal verschillende programmeertalen. PHP is een imperatieve programmeertaal ontworpen in 1994 voor het maken van dynamische webpagina's en is doorheen de jaren één van de populairste programmeertalen geworden op het web. Haskell is een functionele programmeertaal ontworpen in 1990 voor algemeen gebruik en is populair in academische kringen.

## Functioneel versus imperatief

Eén van de fundamentele verschillen tussen Haskell en PHP is het gebruikte programmeerconcept. Net als de meeste (populaire) programmeertalen is PHP een imperatieve taal, dit betekent dat de code die geschreven wordt instructies zijn die meteen uitgevoerd kunnen worden. Imperatieve talen zijn gebaseerd op het Turingmodel: het programma wordt stap voor stap uitgevoerd, elke stap doet iets met het programma, het bestaat in feite uit een serie bevelen voor de computer.

Daarentegen gebruikt Haskell het functionele programmeerconcept, gebaseerd op Lambdacalculus (vandaar het logo van Haskell). Dit concept is sterk gevestigd in de wiskunde. Ook gebruikt Haskell luie evaluatie, wat inhoudt dat functies nooit neveneffecten hebben, dat het programma 'puur' blijft. Een voorbeeld hiervan is het feit dat variabelen niet gewijzigd kunnen worden eens ze gedeclareerd zijn in Haskell.

## Programmeren in een dwangbuis

Haskell dwingt je om iets te programmeren dat volledig werkt, er kan niet gewoon iets in elkaar worden gestoken dat half werkt en waar je later op verder zal bouwen zoals bij PHP. De vrijheid van de programmeur wordt sterk beperkt in Haskell ten gunste van 'puurheid'.

## Hello World

### Haskell:

```
module Main where
main :: IO ()
main = putStrLn "Hello World!"
```

### PHP:

```
<?php echo "Hello World!";
```

## Basic syntax

- Puntkomma's
  - **Haskell:** `let a = 5`
  - **PHP:** `$a = 5;`
- Functies
  - **Haskell:** `func x = x * 2`
  - **PHP:** `function func(x) { return x * 2; }`
- Conditities
  - **Haskell:** `func x = if x > 100 then x else x * 2`
  - **PHP:** `function func(x) { if (x > 100) { return x; } else { return x * 2; }`
- Arrays
  - **Haskell:** `[1,2,3]`
  - **PHP:** `[1,2,3]` of `array(1,2,3)`
- Array waarde by index zoeken
  - **Haskell:** `[1,2,3] !! 0`
  - **PHP:** `[1,2,3][0]`
- Arrays concateneren
  - **Haskell:** `[1,2,3] ++ [4,5,6]`
  - **PHP:** `array_merge([1,2,3],[4,5,6])`

## Gebruik

PHP is één van de populairste programmeertalen op het web, Haskell daarentegen is veel minder populair op elk vlak (er zijn wel web frameworks voor Haskell). Volgens [builtwith.com](http://builtwith.com) gebruiken zo'n 52 miljoen websites PHP en er zijn meer dan 1 miljoen vragen over PHP op [stackoverflow.com](http://stackoverflow.com), over Haskell zijn er maar zo'n 35 duizend vragen. Hierdoor zijn er ook veel meer tools, frameworks, libraries, tutorials enzovoort beschikbaar voor PHP.

## Benchmarks

Er zijn weinig benchmarks beschikbaar die Haskell en PHP vergelijken. De auteur van de Python blog "mypy" heeft wel een benchmark test gedaan tussen Haskell en PHP voor een programma waarbij hij lijnen uit een bestand van 15 duizend lijntjes moest nakijken. Om dit programma uit te voeren had PHP 18 milliseconden nodig en Haskell 8 milliseconden.

## Opinies

Persoonlijk vind ik Haskell geen leuke programmeertaal om mee te werken omdat alles zeer ingewikkeld overkomt, ik verkies PHP.