

Les 3: controle structuren.

Wat is een controle structuur ?

Een controlestructuur is een constructie die toelaat om de normale doorloop van een programma (instructie na instructie na instructie ...) te wijzigen in functie van een voorwaarde die, na evaluatie, leidt tot **true** of **false**.

De waarden true en false worden in PHP voorgesteld (naast de sleutelwoorden true en false zelf) door de waarden 1 (=true) en 0 (=false).

Waarvoor gebruiken we controle structuren?

Controle structuren worden erg vaak gebruikt bij het programmeren. Stel, je maakt een inlog formulier. Wanneer de gebruiker een foutieve gebruikersnaam of paswoord ingeeft moet je deze een foutmelding geven. Wanneer de gebruiker de gegevens correct heeft ingevuld mag deze verder gaan in je website. Dit is een typisch voorbeeld van een controle structuur.

Je controleert als de ingevoerde gegevens correct zijn, indien ja, dan mag de gebruiker verder. Indien niet, moet je de gebruiker op je inlog pagina behouden.

if en else structuur.

if

Syntax:

```
If (expressie)
{
    waar-blok
}
```

of

```
if (expressie)
{
    waar-blok
}
else
{
    onwaar-blok
}
```

elsif

Een uitbreiding op de if-structuur.

Syntax:

```
if (expressie)
{
    waar-blok
}
elseif (expressie)
{
    waar-blok
}
elseif (expressie)
{
    waar-blok
}
...
else
{
    onwaar-blok
}
```

Er kunnen een onbeperkt aantal elseif takken zijn.
Hetzelfde resultaat kan bekomen worden door nested if's!

Vergelijkingsoperatoren.

==	Testen op gelijkheid. Enkel de waarden worden bekeken: 0 == 0.0 !
===	Identiek: zowel waarde als type moeten gelijk zijn!
!= of <>	Niet gelijk.
<	Kleiner
>	Groter
<=	Kleiner of gelijk
>=	Groter of gelijk

Logische operatoren.

Logische operatoren worden gebruikt om combinaties van vergelijkingen te testen.

!	NOT: keert de waarde van een vergelijking om. True wordt false en omgekeerd.
&&	AND: zowel het deel links van de operator als het deel rechts moeten true zijn om als eindresultaat true op te leveren.
	OR: minstens één deel (links en/of rechts) moeten true zijn om als eindresultaat true op te leveren.
and	Zelfde als && maar dan met een lagere prioriteit
or	Zelfde als maar dan met een lagere prioriteit.

Opmerking: &&, ||, and, or zijn allen zuinige operatoren! Zodra in een expressie een uitspraak kan gedaan worden kijkt het systeem niet verder.

Oefeningen.