

Inleiding tot statements

In deze lezing zullen we een kort overzicht geven van Python-statements. Deze lezing zal de nadruk leggen op verschillen tussen Python en andere talen zoals C++ of Java.

Er zijn twee redenen waarom we deze benadering gebruiken om de context van Python-statements te leren:

- 1.) Als je uit een andere taal komt, zal dit je begrip van Python snel versnellen.
- 2.) Door uitspraken te leren, kunt u in de toekomst gemakkelijker andere talen lezen.

Python versus andere programmeertalen

Laten we een eenvoudige verklaring maken die zegt: “Als a groter is dan b, wijs 2 toe aan a en 4 aan b”

Bekijk deze twee if-statements eens (we zullen binnenkort leren over het bouwen van if-statements).

Versie 1 (Andere talen)

```
if (a>b){  
    a = 2;  
    b = 4;  
}
```

Versie 2 (Python)

```
if a>b:  
    a = 2  
    b = 4
```

Je zult merken dat Python minder rommelig (cluttered) en veel leesbaarder is dan de eerste versie. Hoe regelt Python dit?

Laten we de belangrijkste verschillen doornemen:

Python verwijderd `()` en `{}` door twee hoofdfactoren op te nemen: een *dubbele punt* en *spatie*. De verklaring wordt afgesloten met een dubbele punt en er wordt spaties gebruikt (inspringing) om te beschrijven wat er gebeurt in het geval van de verklaring.

Een ander groot verschil is het ontbreken (lack) van puntkomma's in Python. Puntkomma's worden in veel andere talen gebruikt om het einde van een instructie aan te duiden, maar in Python is het einde van een regel hetzelfde als het einde van een instructie.

Laten we tot slot, om dit korte overzicht van verschillen te beëindigen, de inspringing-syntaxis (indentation) in Python versus andere talen eens nader bekijken:

Inspringing (Indentation)

Hier is wat pseudo-code om het gebruik van spatie en inspringing in Python aan te geven:

Andere talen

```
if (x)
    if(y)
        code-statement;
else
    another-code-statement;
```

Python

```
if x:
    if y:
        code-statement
else:
    another-code-statement
```

Merk op hoe Python zo sterk wordt aangedreven door code-inspringing (code-indentation) en spatie. Dit betekent dat leesbaarheid van de code een kernonderdeel is van het ontwerp van de Python-taal.

Laten we nu dieper duiken door dit soort uitspraken in Python te coderen!

Tijd om te coderen!