

# Inleiding met 06 nummers

## WTF is een reguliere expressie?

Een reguliere expressie (regex) is een reeks symbolen en speciale karakters die wordt gebruikt om patronen in tekst te vinden en/of aan te passen. Het is als een krachtige zoekopdracht die je kunt gebruiken om te zoeken naar specifieke stukjes tekst in een grotere tekst.

Stel je bijvoorbeeld voor dat je een telefoonboek hebt en je wilt alle telefoonnummers vinden die beginnen met "06". Met behulp van een reguliere expressie kun je een patroon definiëren dat overeenkomt met deze telefoonnummers, ongeacht wat er daarna komt. Je zou kunnen zoeken naar het patroon "06\d{8}", waarbij "\d{8}" overeenkomt met acht cijfers na "06". Dit patroon zou alle telefoonnummers vinden die beginnen met "06" gevolgd door acht willekeurige cijfers.

Reguliere expressies kunnen ook worden gebruikt om gegevens te valideren, zoals het controleren of een e-mailadres een geldig formaat heeft of dat een wachtwoord aan bepaalde criteria voldoet.

Reguliere expressies kunnen ingewikkeld lijken, omdat ze verschillende speciale karakters en symbolen bevatten. Elk symbool heeft een speciale betekenis en wordt gebruikt om bepaalde patronen te beschrijven. Bijvoorbeeld, "\d" komt overeen met een cijfer en "\w" komt overeen met een letter of cijfer.

Het begrijpen van reguliere expressies kan nuttig zijn omdat het je in staat stelt om geavanceerde zoek- en vervangbewerkingen uit te voeren, tekstpatronen te analyseren en gegevens te valideren. Er zijn verschillende programmeertalen en tools die reguliere expressies ondersteunen, waaronder Python, JavaScript en vele andere.

Hoewel reguliere expressies krachtig zijn, kan het even duren om ze volledig onder de knie te krijgen. Het is belangrijk om te oefenen en te experimenteren met verschillende patronen om vertrouwd te raken met hun gebruik en functionaliteit.

## Opdracht

Gebruik de volgende Python code als template.

```
import re

def validate_phone_number(phone_number):
```

```
pattern = r"....."

if re.match(pattern, phone_number):
    return True
return False

# Voorbeeldgebruik:
phone_number = input("Voer een 06-nummer in: ")
if validate_phone_number(phone_number):
    print("Het ingevoerde nummer is geldig.")
else:
    print("Het ingevoerde nummer is ongeldig.")
```

Op de plaats van de puntjes maak jij een reguliere expressie.

Lees de tekst goed dan kom je een eind.

Zorg er ook voor dat je nummer begint met 06, dus het eerst wat je moet vinden in een valide nummer is 06. Omdat aan te geven in een regex kan je een `^` gebruiken. `^06` betekent dus "*begint met*"

Het einde van het nummer moet eindigen met 10 cijfers, dus na de 10 cijfers mag niets meer komen. Een eind teken is de `$`.

Dus `^06\d{2}$` betekent; *begin met 06 en eindig met een decimaal nummer van precies 2 posities*.

Maak de code af en plaats een regex die een 06-nummer valideert.

Plaats daarvoor op de regel met

```
pattern = r"....."
```

op de plaats van de puntjes de reguliere expressie en test je code!

## Inleveren

- Schermafdruk waarin je laat zien dat je code werkt; geef een juist 06-nummer en een paar foutieve nummers die bijna goed lijken.
- Aangepaste code met jouw naam in commentaar.

--

