

---

## if-else

Aan het einde van dit werkblad kun jij:

- voorspellen wat een if-else code zal doen
- goede en foute if-else codes van elkaar onderscheiden

Spelregels:

- Denk niet te diep na over de codes, het gaat er om dat je het snel gaat kunnen
- Schrijf de codes over als daarom gevraagd wordt, juist door opschrijven oefen je goed
- Spieken/overschrijven is zinloos, het gaat juist om het oefenen. Snap je iets niet, vraag het dan aan ons

### ###if-else uitleg

Met een if-else code kun je een keuze maken in Python. Een if-else ziet er zo uit:

```
dier = 'hond'
if dier == 'hond':
    print('Ik ben Bello')
else:
    print('Ik ben Minoes')
```

Let op deze dingen:

- bij de if gebruik je twee keer de is als je wilt vergelijken: == Één isje werkt niet!
- achter de regel met de if hoort een dubbele punt :
- achter de regel met else hoort een dubbele punt :
- regels onder de if beginnen met 2 spaties
- regels onder de else beginnen met 2 spaties
- dat beginnen met spaties noemen we in het Nederlands inspringen, in het Engels is dat “indentation”
- maar één van de twee takken wordt uitgevoerd

Je mag ook een **if** gebruiken zonder **else**. Dan gebeurt er simpelweg niks als de conditie niet waar is. Bijvoorbeeld zo:

```
dier = 'kat'
if dier == 'hond':
    print('Ik ben Bello')
```

Deze code print niks, want dier is niet gelijk aan “hond”.

---

## Opdrachten

1) Je krijgt een aantal codes. Wat wordt er geprint?

Voorbeeld:

```
ingredient = 'melk'
if ingredient == 'melk':
    print('gieten')
else:
    print('schudden')
```

ingredient is “melk”, dus de conditie is waar, en de code print: “gieten”.

Nu jij!

```
ingredient = 'pindakaas'
if ingredient == 'melk':
    print('gieten')
else:
    print('schudden')
```

```
dier = 'reptiel'
if dier == 'reptiel':
    print('legt', 'een', 'ei')
```

```
dier = 'hagedis'
if dier == 'reptiel':
    print('legt', 'een', 'ei')
else:
    print('geeft', 'melk')
```

```
taal = 'engels'
if taal == 'Engels':
    print('Hello', 'good morning!')
else:
    print('Hallo', 'goedemorgen!')
```

```
taal = 'Nederlands'
if taal == 'Engels':
    print('Hello', 'good morning!')
else:
```

---

```
print('Hallo', 'goedemorgen!')
```

```
taal = 'English'
if taal == 'Engels':
    print('Hello', 'good morning!')
```

2) Er zijn vijf dingen belangrijk bij een if-else. Schrijf er zoveel mogelijk op. Kijk niet terug naar de vorige pagina maar doe het uit je hoofd.

3)

4)

5)

6)

7)

8) Is de code goed of fout?

- De code is goed -> schrijf wat de code print
- De code is fout -> schrijf FOUT
- Extra: schrijf ook **wat** er fout is

Voorbeeld:

```
taal = 'Frans'
if taal == 'Frans':
    print('Bonjour!')
else:
    print('Hello!')
```

De code.... print "Bonjour" Nog een voorbeeld:

```
taal = 'Frans'
if taal == 'Frans'
    print('Bonjour!')
else:
    print('Hello!')
```

De invoer is: Frans. De code.... is FOUT, want de eerste regel mist een :

Dat klinkt misschien heel flauw, maar reken maar dat jij straks op de computer ook dat soort foutjes gaat maken. Zelfs als je al jaren programmeert gebeurt dat nog wel eens.

Nu jij!

---

```
taal = 'Engels'
if taal == 'Frans':
    print('Bonjour!')
else:
    print('Hello!')
```

```
taal = 'Frans'
if taal = 'Frans':
    print('Bonjour!')
```

```
taal = 'Engelse'
if taal = 'Frans':
    print('Bonjour!')
else:
    print('Hello!')
```

```
taal = 'Engelse'
if taal = 'Frans':
    print(Bonjour!)
else:
    print>Hello!)
```

```
taal = 'Duits'
    prnt('Gutenabend!')
else:
    prnt('Goedeavond!')
```

```
taal = 'Duits'
    print('Gutenabend!')
else:
    print('Goedeavond!')
```

#### 4) Foutmeldingen lezen

Je krijgt steeds een foutmelding te zien. Wat denk jij dat er mis is?

```
1.      if input('melk of suiker?') == 'suiker'
                                     ^
SyntaxError: invalid syntax
```

---

```
2.    print('gieten')
      ^
```

IndentationError: expected an indented block

```
3.    if input('melk of suiker?') = 'suiker':
      ^
```

SyntaxError: invalid syntax

```
4.    if input('melk of suiker?) == 'suiker':
      ^
```

SyntaxError: invalid syntax