



07A

ITERATIVES

FOR-LUS

WAT LEREN WE?

- ▶ Wat lussen zijn;
- ▶ `range()`
- ▶ For-lus;
- ▶ ~~For-lus met handmatige collectie;~~
- ▶ ~~Loop-controle;~~
- ▶ Wegwerp-variabelen;
- ▶ Condities in for-lus.



FOR-LUS

▶ Syntax:

```
for <variabele> in <collectie>:  
    <acties>
```

▶ Collectie moet **Iterable** zijn:

- ▶ `string`
- ▶ `range()`
- ▶ `list`
- ▶ `dictionary`
- ▶ ...



FOR-LUS MET STRINGS

```
for letter in 'lus':  
    print(letter + '?')
```

- ▶ Variabele `letter` krijgt achtereenvolgens waarden `'l'`, `'u'` en `'s'`.
- ▶ Telkens `letter` een nieuwe waarde krijgt, wordt de instructie in het for-blok uitgevoerd.

`l?`

`u?`

`s?`



FOR-LUS MET VARIABELE COLLECTIE

```
fruit = input('Lievelingsfruit? ')
m = ' m'
for letter in fruit:
    print(letter + m)
    m += 'm'
```

Lievelingsfruit? *peer*

p m
e mm
e mmm
r mmmm

Lievelingsfruit? *bes*

b m
e mm
s mmm



FOR-LUS MET GETALLENREEKS

- ▶ `range(start)`: Python built-in functie.

```
>>> i = range(10)
```

```
>>> print(i)
```

```
range(0, 10)
```

```
▶ Special Variables  
▼ i = {range} range(0, 10)  
   start = {int} 0  
   step = {int} 1  
   stop = {int} 10
```

```
for i in range(2):
```

```
    print(i)
```

```
0
```

```
1
```



FOR-LUS MET GETALLENREEKS

► `range(start, stop [, step])`

```
for i in range(10, 12):  
    print(i)
```

10

11

```
for i in range(11, 9, -1):  
    print(i)
```

11

10



WEGWERP-VARIABELE

- ▶ 'Teller' wordt niet gebruikt.

```
m = 'm'
for i in range(0,4):
    print(m)
    m += 'm'
```

```
m = 'm'
for _ in range(0,4):
    print(m)
    m += 'm'
```



RANGE IN RUN-TIME

```
aantal_woorden = int(input('Aantal woorden: '))  
  
zin = ''  
  
for _ in range(aantal_woorden):  
    zin += input('Woord: ') + ' '  
  
print(zin)
```

Aantal woorden: *3*

Woord: *jij*

Woord: *bent*

Woord: *mooi*

jij bent mooi



CONDITIES IN FOR-LUS

- ▶ 'lets' tellen, 'enkele gevallen' in volledige collectie.

```
woord = input('woord: ')
klinkers, medeklinkers = 0, 0
for letter in woord:
    if letter in 'aeoui':
        klinkers += 1
    else:
        medeklinkers += 1
print('kl: {} \nmkl: {}'.format(klinkers, medeklinkers))
```

