

# Start to Program: Python, Les 3

## Indexeren en opdelen van Strings

### Lengte van een string tellen

Er zijn ook een hele hoop methode die je meer informatie kunnen geven over je string. De lengte van een string bepalen bijvoorbeeld.  
Dit doe je zo:

```
print(len(mijn_var))
```

Let wel op! Deze methode zal ALLES in je string tellen! Dus ook de spaties, speciale tekens,...

### Bepaalde positie van eens string = indexering

De positie van een teken wordt ook de index genoemd.

LET OP! Bij programmeren telt het getal 0 wel degelijk mee. De index begint dus bij 0!

indexering van 'hello world'

```
h e l l o   w o r l d
0 1 2 3 4 5 6 7 8 9 10
```

Je kan ook omgekeerd te werk gaan. Door op te vragen welk teken op een bepaalde index staat.

```
print(hw[3])
Geeft l weer.
```

Als je een teken nodig hebt dat ergens achteraan de string staat kan je ook negatief te werk gaan. Hier begint de index niet met 0 maar ineens met -1.

```
h e l l o   w o r l d
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1
```

Tel hoeveel tekens je naam, voor- en achternaam, bevat.  
Print het 5de teken in je naam en het derde laatste teken.

## Slicing of opdelen van strings

Door een string te slicen kan je een deel van de string opvragen.  
Hierbij maak je ook gebruik van indexering.  
Dit doe je als volgt:

```
print(mijn_var[x:y])
```

- mijn\_var □ de string in een variabele waar je eens stuk uit wil halen
- x □ de index vanaf waar je begint op te delen. Deze is **INCLUSIEF**
- y □ de index tot waar je wil opdelen. Deze is **EXCLUSIEF**

Je kan vanaf het begin van de string tot een bepaalde index splitsen:

```
print(mijn_var[:y])
```

OF vanaf een index tot het einde van de string splitsen:

```
print(mijn_var[x:])
```

Dit werkt ook met de negatieve indexering.

Je hebt nu telkens een slice gemaakt met een sprong van 1. Er worden geen letters overgeslagen. Je kan ook met sprongen werken. Daarvoor geef je een 3de waarde in.

```
print(mijn_var[x:y:z])
```

z □ gaat elke “z-ste” letter nemen. Dus niet z aantal letter tussen laten.

Dit kan je ook gebruiken om tekst om te draaien. Daarvoor geef je -1 als z waarde in. Hier had je al een manier voor gezien, dit is dus een alternatieve manier. Maar hier kan je ook in één keer een slice nemen. Wat de code korter maakt.  
Bij de andere negatieve getallen ga je weer letter overslaan én de string omdraaien.

Opdrachtje slice index:

Maak een variabele van je volledige naam.

Isoleer je voor- en achternaam.

Print deze niet alleen, maar sla deze ook op in nieuwe variabelen.

Zonder je naam opnieuw te typen.

Draai je naam om en print hem in stappen van 4.

## Tellen en specifieke indexering

Je kan tellen hoeveel keer een specifiek teken voorkomt in een string. Daarvoor gebruik je de methode `str.count()`

```
print(mijn_var.count('x'))
```

Deze methode wordt ook gebruikt om volledige stukken string of woorden te tellen.

```
print(mijn_var.count('string'))
```

LET OP! Python is hoofdletter gevoelig! De tekens worden exact opgezocht en gaan dus enkel de kleine letters of hoofdletters tellen.

Als je een string wil opdelen vanaf een bepaalde letter moest je de index van deze letter weten. Het kan van pas komen om deze index te achterhalen voor je opdeelt. Dit doe je met de methode `str.find()`. De methode kan ook gebruikt worden om hele woorden of stukken string te vinden.

```
print(mijn_var.find('x'))  
print(mijn_var.find('string'))
```

Standaard gaat de methode op zoek naar de index waar het teken of stuk string voor de eerste keer voorkomt.

Je kan ook de startpositie/index aanpassen zodat er pas gezocht wordt na een bepaalde index. Of zelfs aanduiden tussen welke 2 indexen er moet gezocht worden.

```
print(mijn_var.find('x', start_index))
```

OF

```
print(mijn_var.find('x', start_index, eind_index))
```

Je kan hier ook negatieve indexering gebruiken of zelfs een combinatie van positieve en negatieve indexering.

Je print nu telkens de uitkomst, maar je kan deze ook rechtstreeks opslaan in variabelen ipv te printen. Zo kan je alles 'variabel' maken ipv een directe invoer.

Hoeveel keer komt de letter e voor in: 'De kat krabt de krollen van de trap.'  
Hoeveel spaties zijn er? Hoeveel keer komt 'de' voor?

Hoe achterhaal je de index van de 'a' in 'van'? Doe dit in verschillende stappen en print alleen het definitieve antwoord af. Tip maak gebruik van variabelen.

# Datatypes: Number

Het datatype number wordt gebruikt voor alles wat met getallen te maken heeft. Hiermee kan je dan berekeningen maken.

Je plaatst deze niet tussen aanhalingstekens, want dan zal Python dit als een string bekijken.

## integers en floats

In het datatype number kan je onderscheid maken tussen integers en floats.

**Integers** ▫ hele getallen. Niets achter de komma.

**Floats** ▫ komma getallen. Wel waardes achter de komma.

LET OP! Een komma wordt weergegeven als een punt, Amerikaanse notatie

Je moet dit niet extra declareren bij een variabele. Python zal weten wanneer iets een integer of een float is.

Let wel op. Een integer is niet hetzelfde als een float. Dus 3 is niet hetzelfde als 3.0

Je kan wel wisselen tussen de 2 types.

Van integer naar float en omgekeerd.

Hiervoor gebruik je de methode's *int()* en *float()*

Een string kan ook in een getal worden omgezet en omgekeerd.

Hiervoor moet de inhoud van de string wel een getal zijn natuurlijk!

Bij het omzetten van datatypes moet je wel goed opletten. Er zijn een paar omzettingen die niet helemaal logisch verlopen en een paar dingen die helemaal niet gaan lukken. Schrijf in commentaar je bevindingen.

Zet om, sla je uitkomst eventueel op in een nieuwe variabele en print ze.

1. 57 naar een float.

2. 390,8 en 390,2 naar een integer.

Maak de volgende opgaven eerst letterlijk, zoek nadien een oplossing:

3. Maak een var met je naam en een var met een willekeurige int en float.

- Print de zin: Gefeliciteerd, *naam*! Je hebt vandaag *int* lijnen code geschreven.
- Print: *naam* heeft *float* punten.

4. Maak 2 variabelen met een geheel getal als string. Sla het verschil tussen deze strings op in een nieuwe variabele. Print deze af.
5. Maak 2 variabelen met een kommagetal als string. Sla het verschil tussen deze strings op in een nieuwe variabele. Print deze af.
6. Gebruik 1 van de vorige variabelen met kommagetal. Converteer deze naar een integer.

### Wiskundige berekeningen

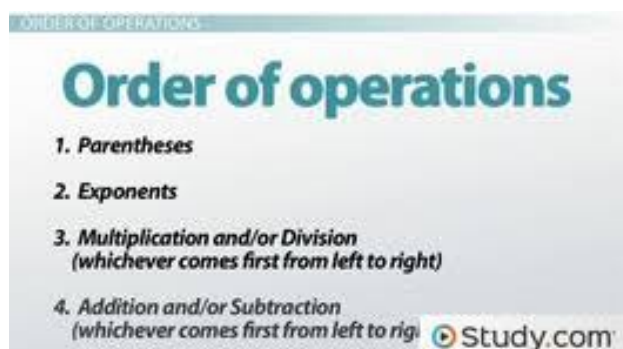
Je kan natuurlijk veel meer met getallen/numbers dan omzettingen tussen de verschillende datatypes.

Het meest voor de hand liggende zijn de wiskundige berekeningen.

berekening	uitkomst
$X + Y$	optellen
$X - Y$	afrekken
$X * Y$	vermenigvuldigen
$X / Y$	delen
$- X$	Van negatief naar positief en omgekeerd.
$X // Y$	Rond de uitkomst van de deling af naar een geheel getal.
$X \% Y$	Modulo, geeft de restwaarde van een deling tussen de opgegeven getallen.
$X ** Y$	machtsverheffing

Er zijn nog andere mogelijkheden, maar die kunnen niet met een symbool worden uitgewerkt. Daarvoor ga je methodes moeten gebruiken. Zoals bijvoorbeeld de wortel van een getal. Later meer hierover.

Deze wiskundige houden zich aan dezelfde regels zoals die ooit op de schoolbanken geleerd hebt. Je kan dus ook met haakjes werken voor complexere berekeningen. Twijfels? Geen nood hier is een overzicht.



**When doing math  
remember, P.E.M.D.A.S.**

**Please Excuse My Dope Ass  
Swag.**

Je kan op verschillende manieren de bewerkingen toepassen.

- Rechtstreeks in de variabele of print.
- 2 variabelen met het datatype integer of float

MAAR, je kan ook rechtstreeks bewerking uitvoeren en bij een bestaande variabele en uitkomst overschrijven in dezelfde variabele.

Dit doe je door een = teken te plaatsen.

Berekening x is de var en y een getal	Uitkomst zal x overschrijven
$X += Y$	optellen
$X -= Y$	afrekken
$X *= Y$	vermenigvuldigen
$X /= Y$	delen
$X //= Y$	Rond de uitkomst van de deling af naar een geheel getal.
$X \% = Y$	Modulo, geeft de restwaarde van een deling tussen de opgegeven getallen.
$X ** = Y$	machtsverheffing

### Modulo

Terwijl de meeste bewerkingen voor de hand liggend zijn is de modulo een beetje een speciaal geval.

Je kan je afvragen wat hier het nu van is.

$X \% Y$

De modulo zal je dan altijd een waarde geven die nooit groter of gelijk is aan Y.

Er zal steeds een uitkomst zijn tussen 0(inclusief) en Y(exclusief).