

Python

Informatica

Renske Smetsers



Hoofdstuk 10: Strings

Na deze les kun je:

- ❑ Uitleggen waar je strings voor gebruikt
- ❑ Strings gebruiken over meerdere regels
- ❑ Indices voor strings gebruiken
- ❑ Strings doorlopen
- ❑ Uitleggen dat strings niet te veranderen zijn
- ❑ String methodes gebruiken: `upper()`, `lower()`, `find()`, `replace()`



Strings

Strings zijn teksten

```
string1 = "appel"      # aangeven met dubbele aanhalingstekens
string2 = 'banaan'     # of met enkele aanhalingstekens
groet = "Hallo iedereen" # string kan meerdere woorden bevatten
string4 = " "          # een spatie is ook een string
string5 = ""           # een string mag leeg zijn (0 letters)
```

#Een string kun je afdrukken.

```
print( string1 )      # Dit print: appel
```

#Een string kun je aan elkaar plakken.

```
print( string1 + string2 ) # Dit print: appelbanaan
```

#kun je herhalen.

```
print( 3 * string1 )   # Dit print: appelappelappel
```



String doorlopen met een for loop

```
fruit = "banaan"

for letter in fruit:
    if letter == "a":
        print( "a gevonden!" )
```

UITVOER:
a gevonden!
a gevonden!
a gevonden!



Newline

Voor het afdrukken op een volgende regel: `\n`

```
print( "eerste regel \n tweede regel" )
```

UITVOER:

eerste regel

tweede regel



Lengte

Voor het bepalen van de lengte van een woord: **len(...)**

```
woord = "fiets"  
lengte = len(woord)  
  
print(  lengte  )
```

UITVOER:

5



Index

- Elk letter in een string heeft een index (of positie)
- Tellen begint bij 0

```
woord = "fiets"
```

```
print( woord[0] )    #drukt eerste letter af (index 0): f
```

```
print( woord[1] )    #drukt tweede letter af (index 1): i
```

```
print( woord[4] )    #drukt 5e letter af (index 4): s
```

```
print( woord[ len(woord) - 1 ] )
```

Hoe druk je de laatste letter af?

Hoe druk je de laatste letter van **elk** woord af?
Tip: met `len(woord)` bepaal je de lengte van een woord



String methodes met . (*punt*)

```
tekst = "FieTs"
```

```
print( tekst ) # drukt af: FieTs
```

```
print( tekst.upper() ) #drukt af: FIETS
```

```
print( tekst.lower() ) #drukt af: fiets
```

```
#drukt positie van T af:
```

```
print( tekst.find("T") ) #drukt af: 3
```

```
#vervangt T met t.
```

```
print(tekst.replace("T", "t"))#drukt af: Fiets
```

```
#knipt uit tekst: eerste index tot laatste
```

```
print( tekst[1:3] )#drukt af: ie
```

Let op: niet t/m



Strings zijn onveranderbaar

- Strings zijn **niet** veranderbaar

```
tekst = "klein"  
print( tekst.upper() ) #drukt af: KLEIN  
print( tekst )        #drukt af: klein
```

- Dus de waarde van tekst kun je niet zomaar veranderen



Strings zijn onveranderbaar

Werkt niet, tekst blijft de waarde *klein* houden:

```
tekst = "klein"  
print( tekst.upper() )    #drukt af: KLEIN  
print( tekst )           #drukt af: klein
```

- ❑ De waarde van een string veranderd niet zomaar!
- ❑ Tekst blijft namelijk de waarde *klein* houden.
- ❑ Hoe verander je alle letters dan wel in hoofdletters?
- ❑ Kopie maken met daarin de aanpassing

```
#maak kopie met daarin hoofdletters  
nieuwetekst = tekst.upper()
```

```
print( nieuwetekst) #drukt KLEIN af  
print( tekst)      #drukt klein af
```



Strings zijn onveranderbaar

- Strings zijn **niet** veranderbaar
- Je kunt wel een kopie met aanpassingen maken

```
tekst = "klein"  
print( tekst ) #drukt klein af  
  
#maak kopie met daarin hoofdletters  
nieuwetekst = tekst.upper()  
  
print( nieuwetekst ) #drukt KLEIN af  
print( tekst ) #drukt klein af
```

ASCII codering

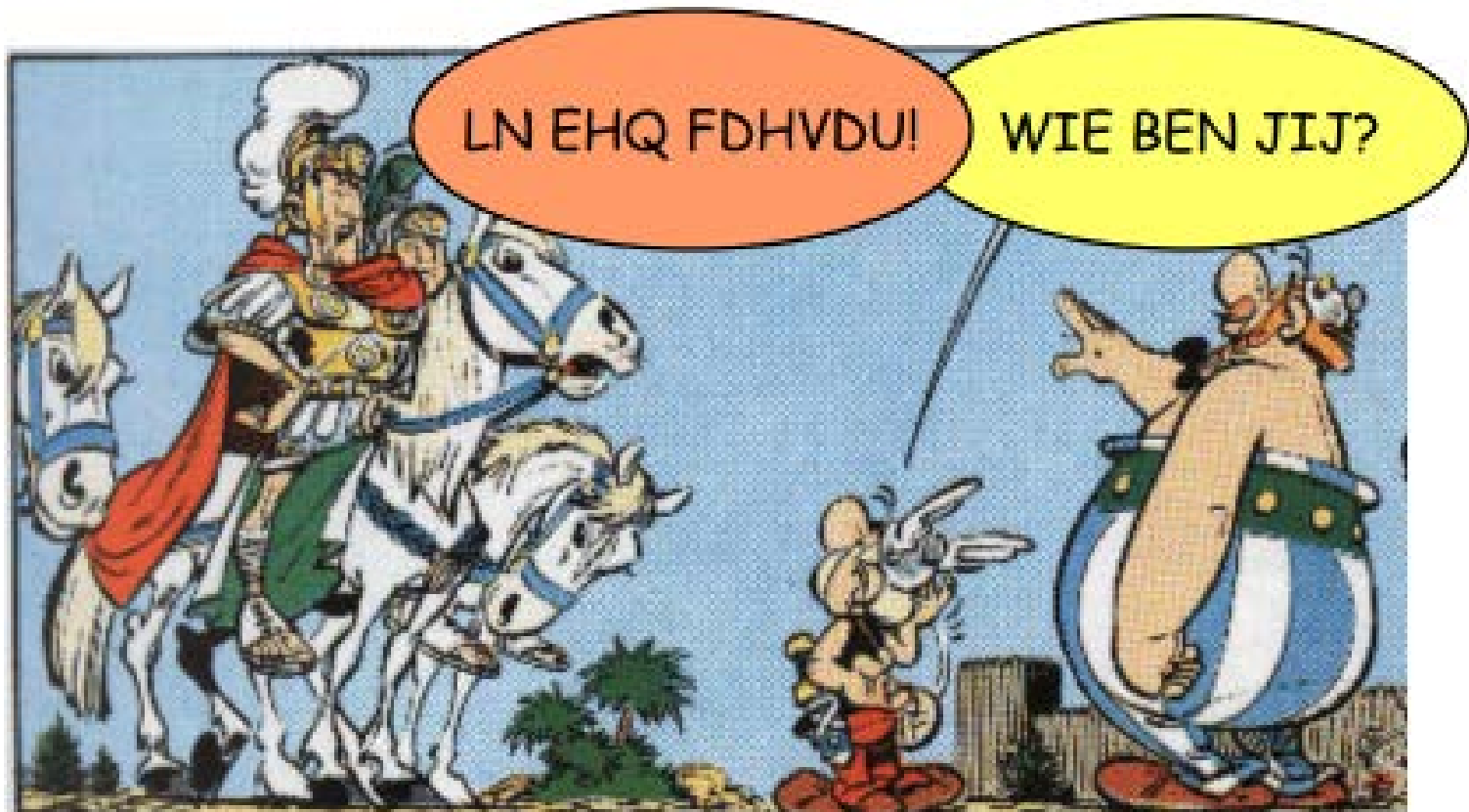
- Letters worden op een computer in ASCII gecodeerd

DC	HX		DC	HX		DC	HX		DC	HX		DC	HX		DC	HX	
32	20		48	30	0	64	40	@	80	50	P	96	60	^	112	70	p
33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
36	24	\$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
40	28	(	56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
41	29	)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
43	2B	+	59	3B	;	75	4B	K	91	5B	[	107	6B	k	123	7B	{
44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	
45	2D	-	61	3D	=	77	4D	M	93	5D	]	109	6D	m	125	7D	}
46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o			

- Bijvoorbeeld: 'A' is 65
- Met `ord()` vraag je de ASCII code op: bv. `ord('A')`
- Met `chr()` vraag je de letter op: bv. `chr(65)`

Caesar Encryptie

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C



Caesar Encryptie

- De Romeinen gebruikten de **Caesarmethode** om boodschappen te versleutelen.
- In dit geheimschrift wordt iedere letter vervangen door de letter die een paar plaatsen verder staat in het alfabet.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C



- Hier een verschuiving van 3 plaatsen.
- $A \rightarrow D$, ... de letter $X \rightarrow A$, $Y \rightarrow B$, $Z \rightarrow C$.



Caesar Encryptie

- ❑ Maak het werkblad
- ❑ Omschrijf je algoritme
- ❑ Programmeren!
 - Afsluitende opgave Strings: 10.5

Klaar? Ga dan verder met:

Week 3:
ma 19-11(VR)

Strings: Theorie en opgaven
Strings: Afsluitende opgaven
Gemengde opgaven (For and While loop)



Afsluiting: Strings

Als het goed is kun je:

- ❑ Uitleggen waar je strings voor gebruikt
- ❑ Strings gebruiken over meerdere regels
- ❑ Indices voor strings gebruiken
- ❑ Strings doorlopen
- ❑ Uitleggen dat strings niet te veranderen zijn
- ❑ String methodes gebruiken: `upper()`, `lower()`, `find()`, `replace()`