



IT Essentials

Hoofdstuk 6

Strings

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



6.1 Introductie

```
s1 = "appel"  
s2 = '  banaan  '  
print(s1)  
print(s2)  
print(s1 + s2)  
print(3 * s1)  
print(s2 * 3)  
print(2 * s1 + 2 * s2)
```

Output?

```
appel  
  banaan  
appel banaan  
appelappelappel  
  banaan banaan banaan  
appelappel banaan banaan
```



6.1.1 len()

len(s)

Geeft als returnwaarde de lengte of het aantal elementen van een object. Dit object kan een string, een verzameling, ... zijn.

Output?

<code>print(len('party'))</code>	5
<code>print(len("off you go"))</code>	10
<code>print(len(""))</code>	0
<code>print(len('mango\'s'))</code>	7



Opgave 1

Maak een programma met een regel tekst. Druk de tekst en de lengte van de tekst af.



6.1.2 format

```
print("Sammy has {} balloons.".format(5))
sammy_string = "Sammy loves {} {}, and has {} {}."
print(sammy_string.format("red", "balloons", 1, "pizza"))
print("Sammy ate {0:.1f}% of a pizza!".format(75.765367))
for i in range(3,13):
    print("{:3d} {:4d} {:5d}".format(i, i*i, i*i*i))
```



6.1.3 for-loop

```
s1 = "mango"  
s2 = "banaan"  
for letter in s1:  
    if letter in s2:  
        print(s1, "en", s2, "bevatten", letter)
```



6.2 Multi-line strings

```
string1 = "I was never insane \  
except upon occasions \  
when my heart was touched."
```

```
string2 = """I was never insane  
except upon occasions  
when my heart was touched."""
```

```
string3 = "I was never insane\n\  
except upon occasions\n\  
when my heart was touched."
```

```
string4 = "I was never insane\n" \  
          "except upon occasions\n" \  
          "when my heart was touched."
```



6.3 Tekens in een string

	p	y	t	h	o	n	
	0	1	2	3	4	5	= len(s) - 1
-len(s) =	-6	-5	-4	-3	-2	-1	

6.3.1 String indices

Output?

```
fun = "party"  
print(fun[0])  
print(fun[4])  
print(fun[2])  
print(fun[-2])  
print(fun[-5])  
print(fun[-1])  
print(fun[5])
```

p

y

r

t

p

y

IndexError: string index out
of range



6.3.1 String indices

Output?

```
fun = "party"
x = 3
print(fun[x])           t
print(fun[2 * 2])       y
print(fun[7 // 3])       r
print(fun[-len(fun)])   p
print(fun[len(fun) - x]) r
```



6.3.3 Substrings met stappen

Output?

<code>fun = "party"</code>	
<code>print(fun[1:4])</code>	art
<code>print(fun[1:4:2])</code>	at
<code>print(fun[:2])</code>	pa
<code>print(fun[3:])</code>	ty
<code>print(fun[0::2])</code>	pry
<code>print(fun[-1::-1])</code>	ytrap



Opgave 2

(zie cursus opgave 6.3)

Schrijf een programma dat van een string de indices print van alle klinkers (a, e, i, o, en u).



Opgave 3

(zie cursus opgave 6.4)

Schrijf een programma waarbij je twee string-variabelen aanmaakt en voor ieder teken in de eerste string dat in de tweede string precies hetzelfde teken heeft op precies dezelfde index, druk je het teken en de index af.



6.4 Strings zijn onveranderbaar

```
fruit = "aaldbei"  
fruit[2] = "r"  
print(fruit)
```

`TypeError: 'str' object does
not support item assignment`



6.4 Strings zijn onveranderbaar

```
fruit = "aaldbei"  
fruit = fruit[:2] + "r" + fruit[3:]  
print(fruit)
```

Je kan geen individueel teken (of karakter) uit de string aanpassen, maar moet in de plaats de variabele, die de string bevat, overschrijven.



6.5.1 strip()

`str.strip([chars])`

Return a copy of the string with the leading and trailing characters removed. The *chars* argument is a string specifying the set of characters to be removed. If omitted or None, the *chars* argument defaults to removing whitespace. The *chars* argument is not a prefix or suffix; rather, all combinations of its values are stripped.

```
print( "*" + "    dit is een test    \n    ".strip() + "*" )
pxl_url = "www.pxl.be"
print(pxl_url.strip( 'w.be' ) )
```



6.5.2 upper() and lower()

str.upper()

Return a copy of the string with all the cased characters converted to uppercase.

str.lower()

Return a copy of the string with all the cased characters converted to lowercase.

```
print ( "aBcDeF" .upper ( ) )  
print ( "aBcDeF" .lower ( ) )
```

Opgave 4

- Gegeven de variabele
spreuk = " aBRaCaDAbra "
- Verwijder de spaties vooraan en achteraan.
- Druk de tekst af met allemaal hoofdletters.
- Druk de tekst af met allemaal kleine letters.



Opgave 5

Schrijf een programma om 2 tekstvariabelen naam en voornaam samen te voegen tot 1 variabele. De nieuwe variabele bestaat uit de eerste letter van de voornaam in hoofdletter gevolgd door een punt en een spatie, gevolgd door de achternaam waarbij de 1ste letter in hoofdletters is en de andere letters in kleine letters zijn.



Opgave 6

Schrijf een programma om de middelste of de 2 middelste letters van een woord in hoofdletters te zetten.



6.5.3 find()

`str.find(sub[, start[, end]])`

Return the lowest index in the string where substring *sub* is found within the slice *s[start:end]*. Optional arguments *start* and *end* are interpreted as in slice notation. Return -1 if *sub* is not found.

```
quote = 'Let it be, let it be, let it be'
first = quote.find('be')
print(first)                                # 7
print(quote.find("be", first + 1))          # 18
print(quote.find("be", 9, 18))              # -1
```

Gebruik de find-methode enkel als je de positie (index) van *sub* nodig hebt. Om te controleren of *sub* een substring is van *str* gebruik je de in-operator: "ho" in "python"



Opgave 7

Neem de string "Barefoot on the grass,# listening to our favorite song" en toon hem vanaf, maar niet inclusief, de hashtag (#). Zorg ervoor dat er geen spaties voor de tekst getoond worden.



6.5.4 replace()

`str.replace(old, new[, count])`

Return a copy of the string with all occurrences of substring *old* replaced by *new*. If the optional argument *count* is given, only the first *count* occurrences are replaced.

```
song = 'cold, cold heart'  
print(song.replace('cold', 'hurt'))  
# hurt, hurt heart
```

```
song = 'Let it be, let it be, let it be, let it be'  
print(song.replace('let', "don't let", 2))  
# Let it be, don't let it be, don't let it be, let it be
```



Opgave 8

- Gegeven de variabele
spreuk = "abracadabra"
- Vervang de letters "a" door de letters "o" en druk het resultaat af.
- Tel vervolgens het aantal letters "o" en druk het resultaat af



Opgave 9

Gegeven de tekst "The quick brown fox jumps over the lazy cat.". Vervang eerst iedere "d" door een "th" en vervolgens "cat" door "dog".

Print het resultaat.



6.4 ASCII-tabel

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

In hoofdstuk 8 gaan we dieper in op de representatie van strings binnen de computer en de verschillende coderingen.

ord() en chr()

ord(c)

Given a string representing one Unicode character, return an integer representing the Unicode code point of that character. For example, `ord('a')` returns the integer 97 and `ord('€')` (Euro sign) returns 8364. This is the inverse of [`chr\(\)`](#).

chr(i)

Return the string representing a character whose Unicode code point is the integer *i*. For example, `chr(97)` returns the string 'a', while `chr(8364)` returns the string '€'. This is the inverse of [`ord\(\)`](#).



ord() en chr()

```
print(ord('A'))  
print(ord('a'))  
print(chr(65))  
print(chr(97))  
print("mango" > "mangaan")
```



Opgave 10

(zie cursus opgave 6.5)

Schrijf een functie die een string als parameter krijgt, en die dan een nieuwe string teruggeeft die hetzelfde is als de parameter, maar waarbij ieder teken dat geen kleine letter is vervangen is door een spatie (bijvoorbeeld, de uitdrukking "ph@t l00t" wordt gewijzigd in "ph t l t").

