

WEEK 3 HOUR 3

TODAY'S TERMINOLOGY



Summary

- Constants
- Operations with strings
- Indexing
- Slicing
- Slicing with stride

CONSTANTS



Definition

Variables that we are expecting to not change their values throughout the execution of the code



Warning

The convention we will follow is to define constants with capital snake_case. For example:
`NUM_STUDENTS`



TEENAGER DEMO



STRING COMPARISONS

Operator	Symbol	Example Expression	Example Result (Type: bool)
Less Than	<	'Jen' < 'Paul'	True
Greater Than	>	'Jen' > 'Paul'	False
Equal To	==	'A' == 'a'	False
Greater Than or Equal To	>=	'a' >= 'a'	True
Less Than or Equal To	<=	'a' <= 'b'	True
Not Equal To	!=	'Jen' != 'Paul'	True
Contains	in	'Jen' in 'Jennifer'	True

↗ Note

In strings: special chars < digits < upper-case letters < lower-case letters

FUN FACT

(i.e. not part of CSC108)

- Python manages characters following the ASCII code

Hex	Value																
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p		
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q		
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r		
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s		
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t		
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u		
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v		
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w		
08	BS	18	CAN	28	(38	8	48	H	58	X	68	h	78	x		
09	HT	19	EM	29)	39	9	49	I	59	Y	69	i	79	y		
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z		
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[6B	k	7B	{		
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C			
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D]	6D	m	7D	}		
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~		
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL		

② How does Python store a string in memory?

-
-
-
-
-
-
-
-

STRING INDEXING

◊ Definition

Gives us an specific character within a string

⚠ Attention

The notation is using square brackets, with an integer inside, to the right of the variable name

X



makeameme.org

< >

STRING INDEXING

```
>>>str1 = 'This is my string example'  
>>>str1[0]  
'T'  
>>>str1[1]  
'h'  
>>>str1[-1]  
'e'  
>>>str1[-2]  
'l'
```

Copy

STRING INDEXING

```
"""
str2 = 'HelloWorld'

      H   e   l   l   o   W   o   r   l   d   H   e   l   l   o   W   o   r   l   d
-10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7   8   9
"""


```

Copy

STRING SLICING

Definition

Since we can index for an specific character, we can also *slice* the string by indexing a subset of characters.

STRING SLICING



Attention

The notation is similar to indexing, but we use a colon to state from which two characters we want to slice the string. Note that the second index is **non-inclusive**

X



makeameme.org

< >

STRING SLICING

```
>>>str1 = 'This is my string example'  
>>>str1[0:4]  
'This'  
>>>str1[1:len(str1)]  
'his is my string example'  
>>>str1[10:]  
' string example'  
>>>str1[:8]  
'This is '
```

Copy



STRING SLICING WITH STRIDE

```
>>>str1 = 'abcdefghijklmnopqrstuvwxyz'  
>>>str1[0:4:2]  
'ac'          # slice from index 0 to 3 ('abcd') and take every  
two indices until the end (index 0 -> 'a' and index 2 -> 'c')  
>>>str1[1:len(str1):5]  
'bglqv'      # slice from index 1 until the end  
('bcdefghijklmnopqrstuvwxyz') and take every five indices until the  
end (index 0 -> 'b', index 5 -> 'g', index 10 -> 'l', index 15 ->  
'q', and index 20 -> 'v')
```

Copy



STRING SLICING WITH NEGATIVE STRIDE

```
>>>str1[10::-1]
'kjihgfedcba' # the slice will begin at index 10 ('k'), but since
the stride is negative, the new end is actually the beginning of
the string, so it takes every index in that direction
```

Copy

WORKSHEET

STRING OPERATIONS

