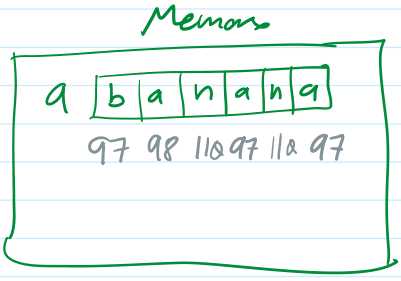
3 Sequence Types

Monday, September 9, 2024 8:22 AM

- STRING

Text representation.

a = "banana"
s = "orange"



- ASCII
- Unicode.

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

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

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Character table functions

ord(char)
chr(num)

- Escape Sequences.

\t tab
\n newline
\\ backlash.
\' single quote
\" double quote

- Literal 'abc' "apple banana"

- Operator `+` concat `*` repeat.

`[ ]` indexing

`string[num]` the character at position 'num'
note: positions start at zero.

- LISTS

a sequence of data elements
data elements can be any python value
(even other lists)

- Literal = `[item, item, item, item, ... , item]`

  - items don't have to be of the same kind.
  - items can be lists themselves
  - the empty list `[ ]` is allowed.

Operators

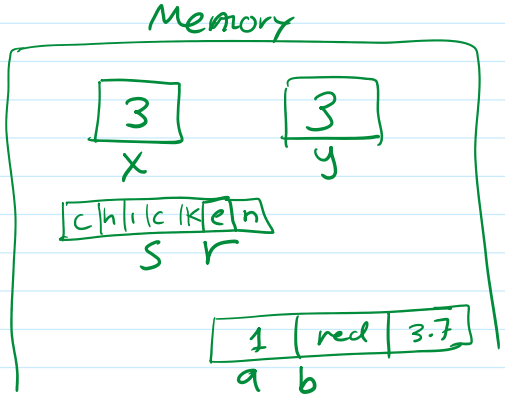
`len( )` length

`+` concatenation.

`=` assignment

Note: the special meaning of =

- `x = 3`
- `y = x`
- `s = "chicken"`
- `r = s`
- `a = [1, "red", 3.7]`
- `b = a`



★ Assignment = of sequence types is not a copy ★

- indexing `[pos]`
  - get element at position pos
  - can also be used to assign element at position pos

- get element at position pos
- can also be used to assign element at position pos

- Some useful functions.

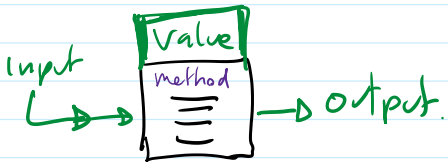
min(l) max(l) — items must be of same type

sum(l) — items must be numbers

- Useful "methods"

Method: a function attached to a value

function



foo(input) value.method(input)

- count(X) return number of occurances of X
- append(X) adds X at the end of the list
- pop() removes last element
- remove(X) removes X
- clear()

- Lists can be "unpacked"

item, item, item, ... = list

- TUPLE

A Tuple is a sequence of items (like a list) but once created, it cannot be changed

- immutable

Literal: (item, item, item, ..., item)

Operators: [] read-only indexing

len() size.

unpacking is available

- SET

A set is an unordered collection of unique items

Literal {item, item, item, ..., item}

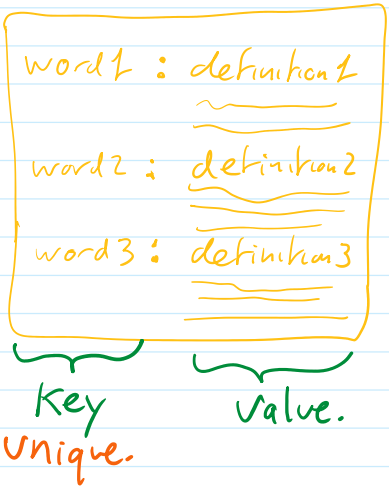
Operators:
- add(item)
- remove(item)
- pop() — removes random item.
- clear()
- intersection(set)
- union(set)
- difference(set)

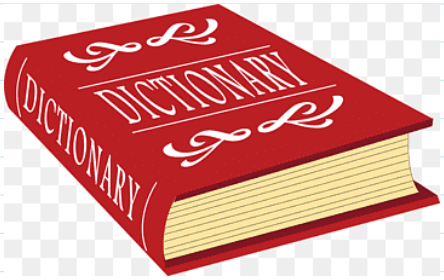
- DICTIONARY

"Map" "Hash-table" "hash-map" "tuple-set"

A "dictionary" is an unordered collection of key & value pairs, where the keys are unique.

Intuition:

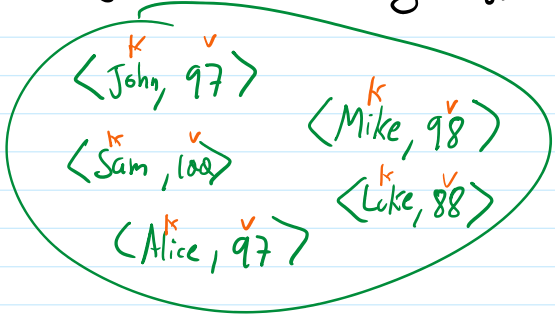




Keys are ordered in a paper dictionary: to make them easier to find.

Python dictionaries are also designed to be searched by key.

e.g: students and their grades.



Literal: {key : value, key : value, key : value, ... }

Operators:

[ ] dict[key]

can be used to both - access values (read, writing)
- add new key-value pairs.

function: len(d)

methods:
- pop(key) — removes key-value pair
  — returns the value

Notes:
- values can be any python type.
- keys can be mixed.
- keys can only be immutable types { integers, floats, strings, tuple

- NESTING

Sequence types can contain other sequence types.

lists inside dictionaries
dictionaries inside lists.
etc.. etc.

You still use [] to access data.

- Demo

- CONSTRUCTORS

functions that build a value.

int(X)
float(X)
str(X)
list(X)
tuple(X)
set(X)
dict(X)

EOF