

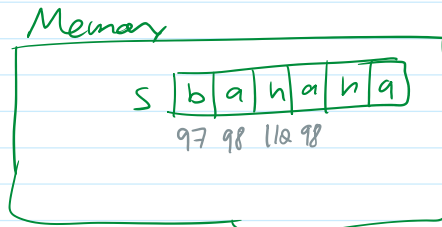
3 Sequence Types

Monday, February 5, 2024 2:01 PM

- Strings

-text representation

S = "banana"



- ASCII
- Unicode.

ASCII control characters			ASCII printable characters			Extended ASCII characters		
00	NULL	(Null character)	32	space	64	@	96	.
01	SOH	(Start of Header)	33	!	65	A	97	a
02	STX	(Start of Text)	34	"	66	B	98	b
03	ETX	(End of Text)	35	#	67	C	99	c
04	EOT	(End of Trans.)	36	\$	68	D	100	d
05	ENQ	(Enquiry)	37	%	69	E	101	e
06	ACK	(Acknowledgement)	38	&	70	F	102	f
07	BEL	(Bell)	39	'	71	G	103	g
08	BS	(Backspace)	40	(72	H	104	h
09	HT	(Horizontal Tab)	41)	73	I	105	i
10	LF	(Line feed)	42	*	74	J	106	j
11	VT	(Vertical Tab)	43	+	75	K	107	k
12	FF	(Form feed)	44	,	76	L	108	l
13	CR	(Carriage return)	45	-	77	M	109	m
14	SO	(Shift Out)	46	.	78	N	110	n
15	SI	(Shift In)	47	/	79	O	111	o
16	DLE	(Data link escape)	48	0	80	P	112	p
17	DC1	(Device control 1)	49	1	81	Q	113	q
18	DC2	(Device control 2)	50	2	82	R	114	r
19	DC3	(Device control 3)	51	3	83	S	115	s
20	DC4	(Device control 4)	52	4	84	T	116	t
21	NAK	(Negative acknowl.)	53	5	85	U	117	u
22	SYN	(Synchronous idle)	54	6	86	V	118	v
23	ETB	(End of trans. block)	55	7	87	W	119	w
24	CAN	(Cancel)	56	8	88	X	120	x
25	EM	(End of medium)	57	9	89	Y	121	y
26	SUB	(Substitute)	58	:	90	Z	122	z
27	ESC	(Escape)	59	;	91	[123	{
28	FS	(File separator)	60	<	92	\	124	
29	GS	(Group separator)	61	=	93]	125	}
30	RS	(Record separator)	62	>	94	^	126	~
31	US	(Unit separator)	63	?	95	_		
127	DEL	(Delete)						
128	Ç		160	á	192	Ł	224	Ó
129	ü		161	í	193	ł	225	ó
130	é		162	ó	194	ł	226	ô
131	à		163	ú	195	ł	227	õ
132	â		164	ñ	196	ł	228	ö
133	ä		165	Ń	197	ł	229	ø
134	å		166	•	198	ł	230	ù
135	ç		167	•	199	ł	231	ú
136	ê		168	¿	200	ł	232	û
137	ë		169	®	201	ł	233	ü
138	è		170	¬	202	ł	234	ý
139	í		171	¼	203	ł	235	ÿ
140	ì		172	½	204	ł	236	ÿ
141	í		173	¾	205	ł	237	ÿ
142	Ā		174	«	206	ł	238	ÿ
143	Ă		175	»	207	ł	239	ÿ
144	Ĕ		176		208	ł	240	ÿ
145	æ		177		209	ł	241	ÿ
146	Æ		178		210	ł	242	ÿ
147	ó		179		211	ł	243	ÿ
148	ô		180		212	ł	244	ÿ
149	õ		181		213	ł	245	ÿ
150	ù		182	Ā	214	ł	246	ÿ
151	ú		183	Ă	215	ł	247	ÿ
152	ÿ		184	Ĕ	216	ł	248	ÿ
153	Ō		185	Ė	217	ł	249	ÿ
154	Ū		186	Į	218	ł	250	ÿ
155	ø		187	Š	219	ł	251	ÿ
156	ē		188	Š	220	ł	252	ÿ
157	ø		189	š	221	ł	253	ÿ
158	x		190	Ÿ	222	ł	254	ÿ
159	ƒ		191	Ÿ	223	ł	255	nbsp

character table functions.

ord(char)

chr(num)

- "Escape Sequences"

- \t tab
- \n newline
- \\ backslash
- \' quote
- \" double quote

Literal : "abcñ" "apple banana"

Operator + *
concat repeat.

- len(s) length of the string.

indexing
-1 ... [num] character at position num

Indexing
string[num] character at position num
starts at zero

• List

a sequence of data elements.
data elements can be any value allowed by Python
(even other Lists)

Literal: [item, item, item, ..., items]

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

Operators:

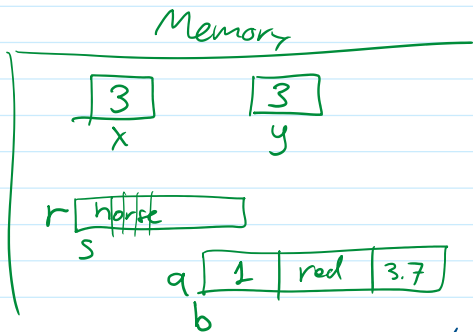
len()

+
concat.

= assignment.

NOTE: the special meaning of =

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



★ assignment = of sequence types is not a copy

Indexing: []

indexing can be used both to retrieve values, and to assign values.

list(x) builds a list from x.

• Other useful functions:

min(l)

max(l)

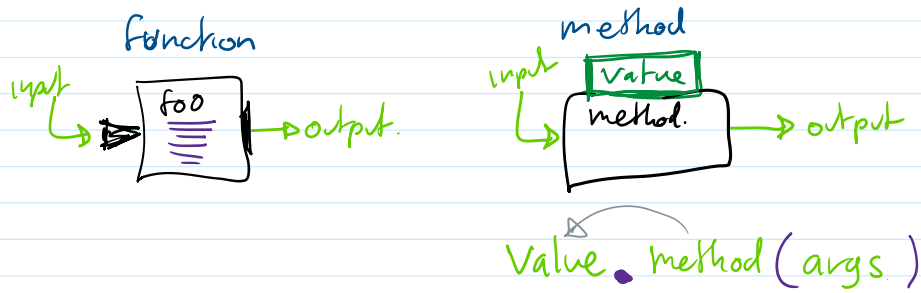
sum(l)

Items must be of same kind

Items must be numbers.

• Useful "methods"

Method: a function attached to a value.



- `count(x)` - returns number of times `x` appears in list.
- `append(x)` - appends item to the end of the list
- `pop()` - removes last element.
- `remove(x)`
- `clear()`

• Lists can be "unpacked"

`item, item, item, ... = list.`

• Tuple

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

- immutable

Literal:

`(item, item, item, ...)`

Operators:

- `len()`
- `[]` but only to read values.
- unpacking is available.

• Set

A set is an unordered collection of unique elements

Literal

`{item, item, item, ...}`

• Operators



• Methods :

- add(item)
- remove(item)
- pop() - removes element at random
- clear

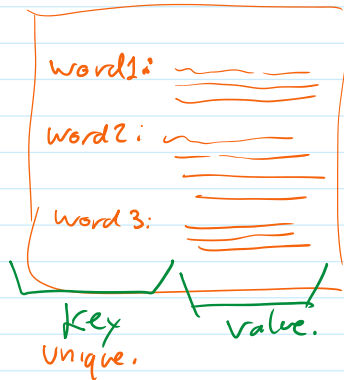
- intersection(set)
- union(set)
- difference(set)

• Dictionary

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

A dictionary is an unordered* collection of $\langle \text{key}, \text{value} \rangle$ pairs, where the keys are unique.

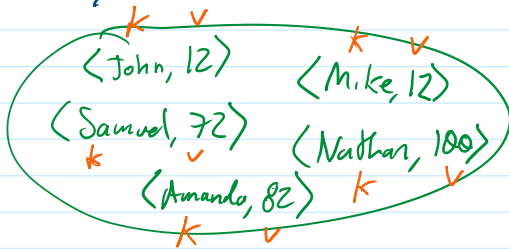
Intuition:



keys are ordered so they are easier to find.

Python Dictionaries are also designed to be searched by key.

e.g.



Literal

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

Operator

[] dict[key]

can be used to - access values
- insert new values.

can be used to - access values
- insert new values.
 $dict[k] = val$

function: $len()$

methods: $pop(key)$ removes a key:value pair
returns the value.

- values can be any valid type.
- keys can be mixed.
- keys must be immutable. values
 - integers
 - floats
 - strings
 - tuple.

• Nesting

Sequence types can contain inside other sequence types:

lists inside dictionaries
dictionaries inside lists
etc
etc.

you still use $[]$ to access data.

• Constructors:

functions to build a value.

$int(x)$ $str(x)$ $list(x)$
 $float(x)$. $tuple(x)$
 $set(x)$
 $dict(x)$

EOF