ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Day 2 - Strings

2022 September 12-16

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Characters & Strings

- Declaring strings can be done in two ways
  - str = ‘Hello World’
  - str = “Hello World”
- Strings are an array of characters
  - Each character can be directly addressed using the same logic as array indexing
    - str[0] is ‘H’, str[4] is ‘o’
  - Negative indexing can be used, when we want to count from the end
  - str[-1] gives “d”, str[-2] gives “l”

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

String Manipulation Operators

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

String Manipulation, Operators

- You already now + and * operators for numbers
- The + operator can be used to concatenate, or join two strings

- s = ‘foo’
- t = “bar”
- s + t gives us “foobar”
- print(s + t) prints out “foobar”

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

String Manipulation, Operators

- The * operator can be used to create multiples of the same string
- If a string is multiplied by an n integer, the result will be n joined copies if the string
  - str * n has the same result as n * str
  - If n is less than 1, the result is an empty string

- s = “foo”
- s * 4 gives us “foofoofoofoo”

- s * 0 and -8 * s both give us “ ”

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

String Manipulation, Operators

- There are two new operators for strings
- The in operator returns True if the first operand is contained within the second, and False otherwise

- s = “foo”
- s in “foobar” returns True
- s in “Python classes” return False

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

String Manipulation, Operators

- The pair of the in operator is not in
- The not in operator returns False if the first operand is contained within the second, and True otherwise

- s = “foo”
- s not in “foobar” returns False
- s not in “Python classes” return True

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Built-In String and Character Functions

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

- Python has built-in functions for working with text
  - ord() – Converts character to an integer
  - chr() – Converts an integer to a character
  - len() – Returns the length of a string
  - str() – Returns a string representation of an object

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

- ord() – Converts character to an integer
  - Computers store data as numbers
  - For text, the easiest scheme is called ASCII
    - ASCII covers the standard Latin alphabet and the commonly used symbols
    - For these, ord() returns their ASCII value
  - For other characters, be it letters from other languages or emojis, ord() gives us the UNICODE value

  - ord(‘a’) returns 97 as the ASCII value of ‘a’ is 97
  - ord(‘1’) returns 49

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

- chr() – Converts an integer to a character
  - The opposite of ord()
  - Works with ASCII and UNICODE

  - chr(97) returns an ‘a’
  - chr(128512) gives an emoji😀

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

- len() – Returns the length of a string

- len(“foo”) returns 3

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

- str() – Returns a string representation of an object
  - Nearly any object can be represented as a string

- str(42.069) returns ’42.069’
- str(True) returns ‘True’
- str(math.e) returns ‘2.718281828459045’

- You can also do operations inside the function
  - str(18 + 24) returns ‘42’

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Functions

- str() – Returns a string representation of an object
  - Nearly any object can be represented as a string

- str(42.069) returns ’42.069’
- str(True) returns ‘True’
- str(math.e) returns ‘2.718281828459045’

- You can also do operations inside the function
  - str(18 + 24) returns ‘42’

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Getting Parts of Strings

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Slicing

- You can get a part of a string with a starting and an end location, by slicing them
- The starting and end locations can be expressed by the indexes of the characters
- The starting is inclusive and the end is exclusive
  - This means that the specified start index will be in the sliced string while the end won't

- s = "foobar"
- s[1:4] gives oob

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Slicing

- If you want to get a slice starting from the start, you don’t need to write the first index
- s[0:5] is the same as s[:5]

- Similar applies if you want the slice from a position till the end
- s[1:len(s)] is the same as a[1:]

- By leaving out both indices, you get back a reference for the original string

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Slicing

- You can also use the negative index numbers
- If you want to only leave off the last character, you can do it like this:
  s[:len(s) - 1] or with negative index: s[:-1]

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Slicing

- If you add a third number to slicing, you get every nth character of the string, starting from the first character of the slice

- s = “foobarbaz”
- s[1:8] returns “oobarba”
- s[1:8:2] will return “obra” as it selected oobarba
- s[1:8:3] returns “oaa” as it selected oobarba

- You can also use leave the start and end indices off as previously
- s[::2] will return “foabz” as it selected foobarbaz

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Putting Variables Into Strings

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Interpolating Variables

- Putting variables into strings can be useful for showing data, or assembling arguments for other functions

- You don’t have to get the string representation of a variable in order to put it in a string

- n = 20
- m = 25
- prod = n*m
- print(‘The product of ’, n, ‘ and ’, m, ‘ is ’, prod);
- print(f‘The product of {n} and {m} is {prod}‘);

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Built-in String Methods

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Built-In Methods

- Most built-in methods have very descriptive names, so We'll give examples, not a description

- The input string s will be "fooBar Baz"

Method	Output
s.replace('r', 'z')	fooBaz Baz
s.capitalize()	Foobar baz
s.lower()	foobar baz
s.upper()	FOOBAR BAZ
s.swapcase()	FOObAR bAZ
s.title()	Foobar Baz
s.count('a')	2

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Built-In Methods

- The input string s will be “fooBar Baz”

Method	Output
s.endswith(‘Baz’)	true
s.startwith(‘foo’)	true
s.find(‘Bar’)	3

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Built-In Methods

- There are a few classifier methods

Method	Description
s.isalnum()	True, if all characters are alphanumeric
s.isalpha()	True, if all characters are alphabetic
s.isdigit()	True, if all characters are numbers

- There are more, but most are nichely used.