



UNIVERSITAS SAM RATULANGI MANADO
FAKULTAS TEKNIK, JURUSAN TEKNIK ELEKTRO
Program Studi S-1 Teknik Informatika

Strings

Mata Kuliah: Algoritma & Logika Informatika (IFC3504)

Alwin M. Sambul, S.T., M.Eng., Ph.D.

Apa itu String?

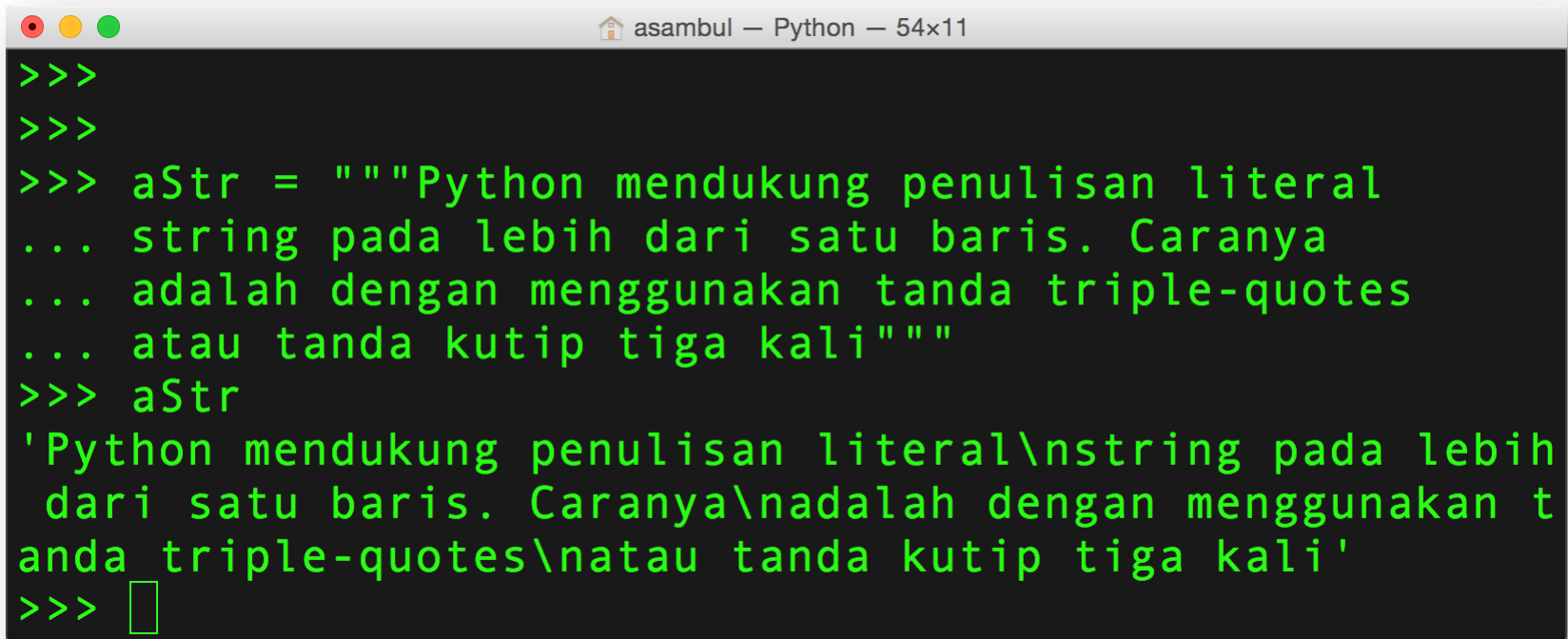
2

- **String** adalah deretan karakter.
- **String literal** ditulis:
 - ▣ Di antara “ ” (kutip ganda)
 - ▣ Di antara ‘ ’ (kutip tunggal)
- **Contoh:**
 - ▣ “Hello World”
 - ▣ ‘Hello World’
 - ▣ “Hi. I’m Python”

Literal String Multi-baris

3

- Python mendukung penulisan literal string multi-baris
- Gunakan triple-quotes (tanda kutip 3 kali)



```
asambul — Python — 54x11
>>>
>>>
>>> aStr = """Python mendukung penulisan literal
... string pada lebih dari satu baris. Caranya
... adalah dengan menggunakan tanda triple-quotes
... atau tanda kutip tiga kali"""
>>> aStr
'Python mendukung penulisan literal\nstring pada lebih
dari satu baris. Caranya\nadalah dengan menggunakan t
anda triple-quotes\natau tanda kutip tiga kali'
>>> □
```

Elemen2 String dan Index

4

- Elemen string diakses dengan menggunakan index dan tanda kurung siku [].
- Penomoran Index dimulai dari:
 - ▣ 0 jika dari depan
 - ▣ -1 jika dari belakang

U	N	S	R	A	T
1	2	3	4	5	6
-6	-5	-4	-3	-2	-1

Mengakses Elemen String

5

```
asambul — Python — 54x10
>>> a = "UNSRAT"
>>> a[0]
'U'
>>> a[1]
'N'
>>> a[-1]
'T'
>>> a[-2]
'A'
>>> 
```

Sub-string

6

asambul — Python — 54x10

```
>>> a = "UNSRAT"
>>> a[:3]
'UNS'
>>> a[-3:]
'RAT'
>>> a[1:4]
'NSR'
>>> a[-4:-1]
'SRA'
>>> 
```

Iterasi Elemen2 String

7

```
asambul — Python — 54x18
>>>
>>>
>>> a = "INFORMATIKA"
>>> for i in range(len(a)):
...     print(a[i])
...
I
N
F
O
R
M
A
T
I
K
A
>>> □
```

Gunakan statemen kendali iteratif **for**, statemen **range**, dan **len**

Menyambung String (operator +)

8

```
asambul — Python — 54x7
>>> a = "Informatika"
>>> b = "Unsrat"
>>> a + b
'InformatikaUnsrat'
>>> a + " " + b
'Informatika Unsrat'
>>> █
```


Representasi String: kode ASCII

0	NUL	1	SOH	2	STX	3	ETX	4	EOT	5	ENQ	6	ACK	7	BEL
8	BS	9	HT	10	LF	11	VT	12	FF	13	CR	14	SO	15	SI
16	DLE	17	DC1	18	DC2	19	DC3	20	DC4	21	NAK	22	SYN	23	ETB
24	CAN	25	EM	26	SUB	27	ESC	28	FS	29	GS	30	RS	31	US
32		33	!	34	"	35	#	36	\$	37	%	38	&	39	'
40	(41)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[92	\	93]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124		125	}	126	~	127	DEL

ASCII = American Standard Code for Information Interchange

Fungsi ord()

10

Mendapatkan nilai ASCII dari karakter tertentu

```
asambul — Python — 54x11
>>>
>>>
>>> ord("A")
65
>>> ord("B")
66
>>> ord("a")
97
>>> ord("b")
98
>>> □
```

Fungsi chr()

11

Mendapatkan karakter dari nilai ASCII tertentu

```
asambul — Python — 54x11
>>>
>>>
>>> chr(65)
'A'
>>> chr(66)
'B'
>>> chr(97)
'a'
>>> chr(98)
'b'
>>>
```

Pengkodean Unicode

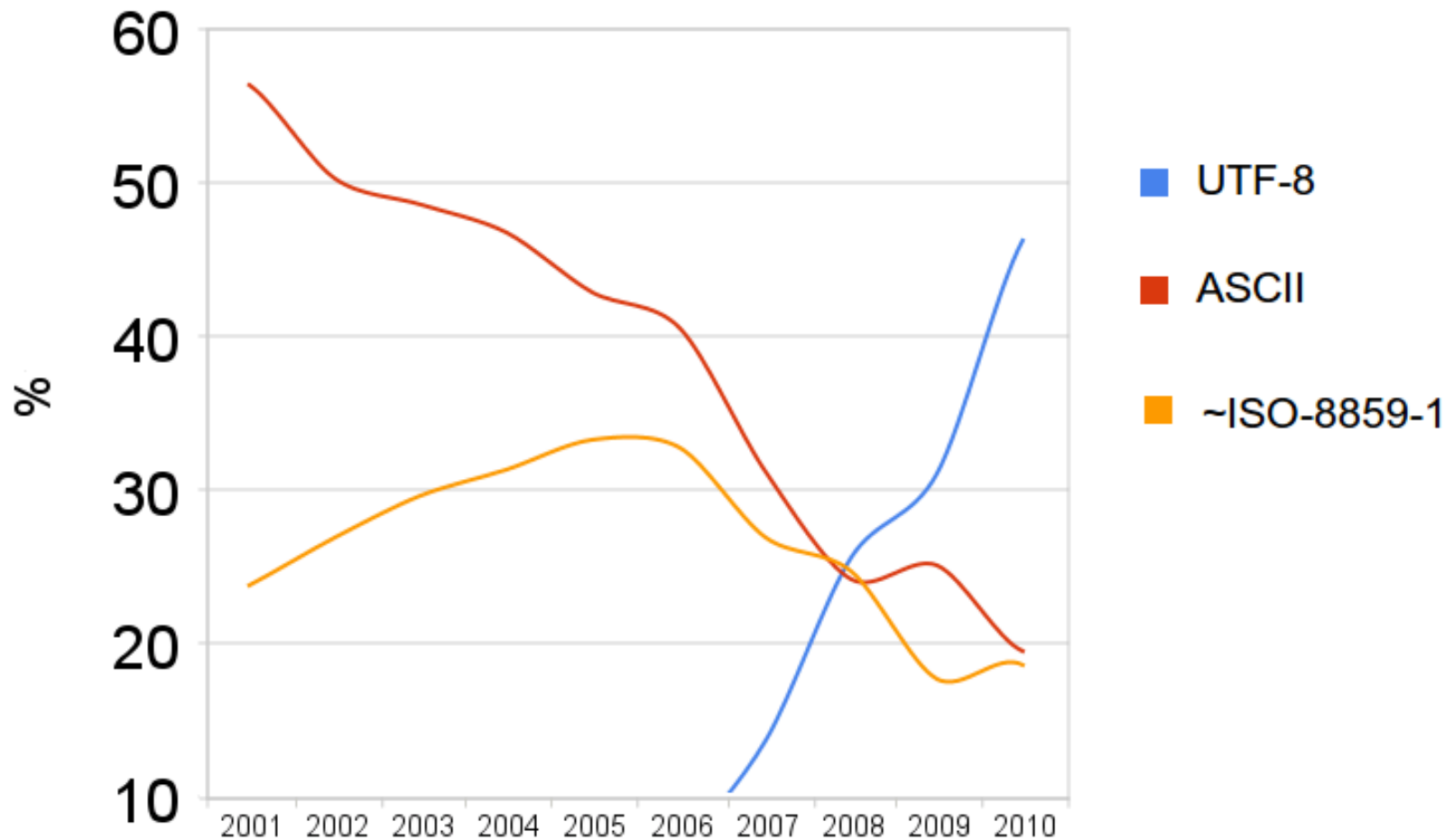
12

- ASCII = 128 karakter
- **Unicode** adalah pengkodean karakter sebagai perluasan dari kode ASCII
 - ▣ Standarisasi dikelola oleh Unicode Consortium
- UTF-8 mencapai lebih dari 1 juta karakter

Pentingnya mempelajari Unicode

13

Growth of UTF-8 on the Web



Penulisan karakter Unicode

14

Contoh: karakter “A”

ord(“A”) → 65

Jika dinyatakan dalam biner 2 byte (16 bit):

Biner: 0000 0000 0100 0001

Hex: 0 0 4 1

Maka karakter “A” dapat dituliskan dalam format unicode dengan menggunakan escape sequence `\u` dan kode hexa:

print(“\u0041”)

Karakter Khusus

15

- Contoh: Bagaimana cara menampilkan simbol Euro (€) dalam string?
- Untuk menuliskan karakter2 khusus, gunakan:
 - ▣ Escape sequence: `\u` bersama kode hexa
 - ▣ Fungsi `chr(<kode desimal>)`

Contoh →

Contoh: Simbol Euro (€)

16

```
asambul — Python — 54x11
>>>
>>>
>>>
>>>
>>> print("Simbol Euro: ", "\u20ac")
Simbol Euro: €
>>> print("Simbol Euro: ", chr(8364))
Simbol Euro: €
>>> print("Gaji saya: ", "\u20ac", format(10000, ', '))
Gaji saya: € 10,000
>>> □
```

Bilangan desimal **8364** berpadanan dengan bilangan hexa **20ac**

Karakter khusus terdaftar

17

Setiap karakter memiliki nama. Gunakan `\N{nama}`.

```
asambul - Python - 54x15
>>> print("\N{registered sign}")
®
>>> print("\N{greek small letter alpha}")
α
>>> print("\N{greek small letter beta}")
β
>>> print("\N{copyright sign}")
©
>>> >>> print("\N{micro sign}")
μ
>>> print("\N{yen sign}")
¥
>>> print("\N{square root}")
√
>>> □
```

Daftar lengkap Unicode

18

Code Charts

Nameslist Charts

- [Help](#)
- 0000 [C0 Controls and Basic Latin](#) 007F
- 0080 [C1 Controls and Latin-1 Supplement](#) 00FF
- 0100 [Latin Extended-A](#) 017F
- 0180 [Latin Extended-B](#) 024F
- 0250 [IPA Extensions](#) 02AF
- 02B0 [Spacing Modifier Letters](#) 02FF
- 0300 [Combining Diacritical Marks](#) 036F
- 0370 [Greek and Coptic](#) 03FF
- 0400 [Cyrillic](#) 04FF
- 0500 [Cyrillic Supplement](#) 052F
- 0530 [Armenian](#) 058F
- 0590 [Hebrew](#) 05FF
- 0600 [Arabic](#) 06FF

C0 Controls and Basic Latin 0000 007F

0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	esc	001C	001D	001E	001F
0020	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
p	q	r	s	t	u	v	w	x	y	z	{		}	~	007F

C0 Controls and Basic Latin

C0 controls

Alias names are those for ISO/IEC 6429:1992. Commonly used alternative aliases are also shown.

0000 C_0 <CONTROL>
= NULL

0001 C_1 <CONTROL>
= START OF HEADING

0002 C_2 <CONTROL>
= START OF TEXT

<http://unicode.org/charts/nameslist/>

Praktek

20

Metode-metode String

Metode-metode String

21

Metode2 yang berkaitan dengan huruf besar/kecil:

1. **capitalize():** Membuat karakter pertama menjadi huruf besar.
2. **title():** Membuat karakter pertama di setiap kata menjadi huruf besar.
3. **swapcase():** Menukar case (besar jadi kecil, kecil jadi besar)
4. **upper():** Semua karakter dijadikan huruf besar
5. **lower():** Semua karakter dijadikan huruf kecil

Contoh:

22

asambul — Python — 54x12

```
>>> s = "informatika UNSRAT"
>>> s.capitalize()
'Informatika unsrat'
>>> s.title()
'Informatika Unsrat'
>>> s.swapcase()
'INFORMATIKA unsrat'
>>> s.upper()
'INFORMATIKA UNSRAT'
>>> s.lower()
'informatika unsrat'
>>> □
```

Metode `count()`

23

Metode `count()` menghitung kemunculan karakter/teks tertentu dalam sebuah teks

```
asambul — Python — 54x12
>>> s = "informatika UNSRAT"
>>> s.capitalize()
'Informatika unsrat'
>>> s.title()
'Informatika Unsrat'
>>> s.swapcase()
'INFORMATIKA unsrat'
>>> s.upper()
'INFORMATIKA UNSRAT'
>>> s.lower()
'informatika unsrat'
>>> □
```

Menghilangkan whitespace

24

- `lstrip()`: menghilangkan whitespace di kiri
- `rstrip()`: menghilangkan whitespace di kanan
- `strip()`: menghilangkan whitespace di kiri dan kanan

```
asambul — Python — 54x8
>>> s = "    informatika    "
>>> s.lstrip()
'informatika    '
>>> s.rstrip()
'    informatika'
>>> s.strip()
'informatika'
>>> □
```


Metode find()

25

Metode find() mencari keberadaan satu string dalam string yang lain. Jika string ditemukan, index-nya akan diberikan. Jika tidak ditemukan, nilai -1 akan diberikan.

```
asambul — Python — 54x8
>>> s = "informatika"
>>> s.find("i")
0
>>> s.find("o")
3
>>> s.find("z")
-1
>>> □
```