Python Language Fundamentals

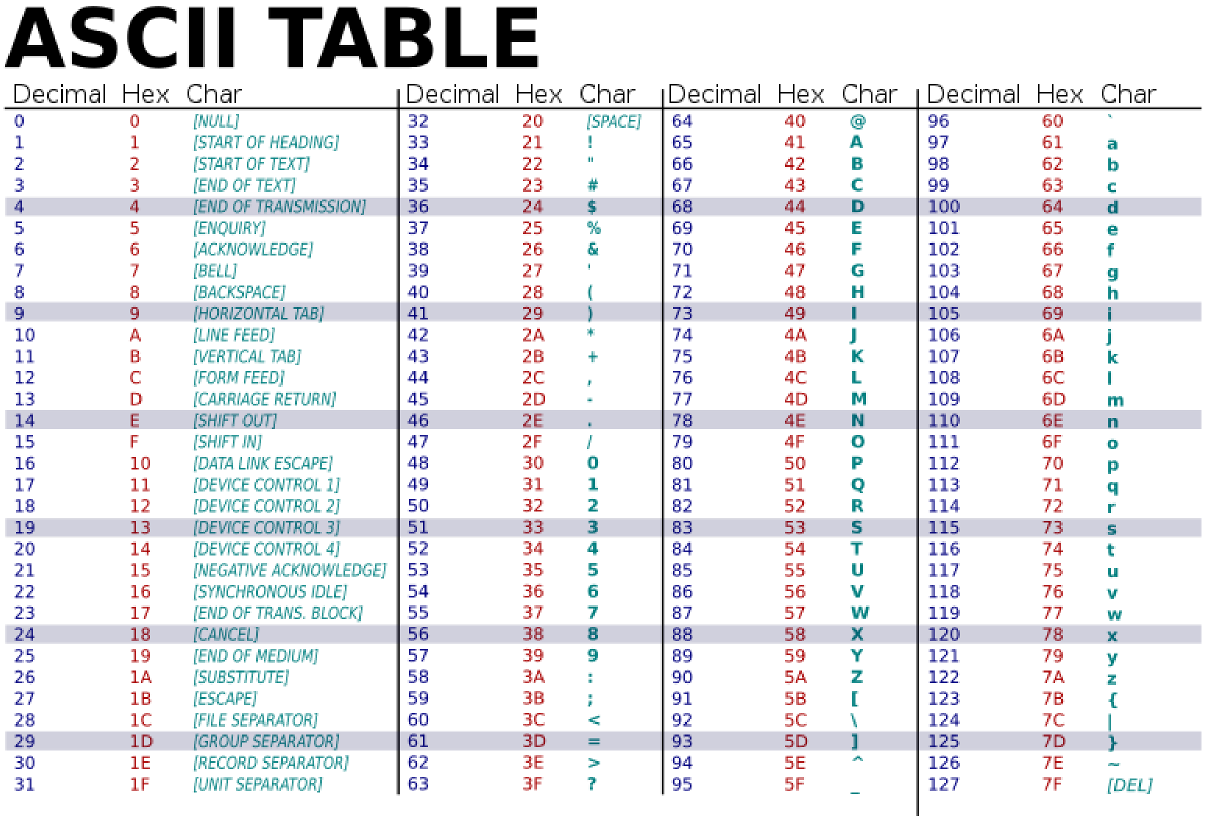
1. Character set of python
2. Tokens of Python
3. Input/Output statements

Character set of python

Character set defines encoding and decoding standards.

1. ASCII → 256
2. UNICODE → 1,00,000

ASCII stands for American Standard Code for Information Interchange. ASCII supports 256 characters, which includes characters in English (a-z,A-Z), digits (0-9) and special characters.



ASCII TABLE

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	[END 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]

UNICODE is superset of ASCII, It not only support characters in English it also support characters in other languages.

```
>>> name="naresh"
>>> print(name)
```

```
naresh
>>> studname="नरेश"
>>> print(studname)
नरेश
>>> empname="నరేష్"
>>> print(empname)
నరేష్
>>> print('\u1F600')
```

Upper case Letters
A → 65
B → 66
C → 67
Z → 90
Lower Case Letters
a-97
b-98
..
Z→122

Digits
0-48
1 – 49
9 → 57

Python Tokens

A smallest individual unit within program is called token.

1. Keywords
2. Identifiers
3. Literals
4. Data types
5. Operators

Keywords

Keywords are language related words.

Keywords are reserved words. The meaning of these words reserved by python language cannot used as identifiers or user defined words.
Each keyword is used for specific purpose.

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try',
'while', 'with', 'yield']
>>> len(keyword.kwlist)
35
>>> keyword.softkwlist
['_', 'case', 'match', 'type']
```

Soft keywords are introduced in python 3.11 version.
These keywords can be used as identifiers.

Identifiers

Identifiers are user defined words, which are used to identify programming elements.

1. Variable name
2. Function names
3. Class name
4. Constant names
5. Program name/module name
6. Package name

Identifier is a word, this word is created using alphabets (a-z,A-Z), digits (0-9) and special character (_)

Python is case sensitive language and finds the difference between uppercase and lowercase.

Rules

1. Identifier should not be keyword

```
>>> lambda=1
SyntaxError: invalid syntax
>>> else=100
```

```
SyntaxError: invalid syntax
>>> ELSE=100
>>> ELSE
100
>>> PASS=200
>>> PASS
200
```

2. Identifier cannot start with digit

```
>>> n1=100
>>> 2n=200
SyntaxError: invalid decimal literal
>>> address1="ameerpet"
>>> 2address="srnager"
SyntaxError: invalid decimal literal
```

3. Only one special character is allowed _

```
>>> _a=100
>>> __abc=200
>>> _a
100
>>> __abc
200
>>> _=300
>>> _
300
>>> a_=500
>>> a_
500
>>> $amt=100
SyntaxError: invalid syntax
>>> amt$=10
SyntaxError: invalid syntax
```

4. There should not be any space between identifier

```
>>> roll number=1
SyntaxError: invalid syntax
```

```
>>> base of triangle=1.5
SyntaxError: invalid syntax
>>> roll_number=1
```

5. Maximum length of identifier is unlimited

```
>>> aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa=1
>>> aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
1
```

6. Identifiers can be defined in uppercase or lowercase

```
>>> a=100
>>> A=200
>>> a
100
>>> A
200
```

Telegram: codewithsatishgupta
Emailid : pythonbygupta@gmail.com

Python & Full Stack Python @ 5:00 PM (IST) by Mr.Satish Gupta
Day-1 https://youtu.be/0RdZ-i5tjVwDay-2 https://youtu.be/XuD_tHMqDbIDay-3 https://youtu.be/lBwuPoPEswMDay-4 https://youtu.be/ZBCAe39TYQoDay-5 https://youtu.be/45hY-KNvnDA