2023 N.E.T Python

Today

- 1. Variables
- 2. Data Type 1
 - Int
 - Float
 - Complex
 - Boolean
 - String

3. Operators

- Arithmetic
- Assignment
- Comparison
- Logical
- Identity
- Membership
- 4. Conditional Statement
- 5. Loop
 - For
 - While

Variables

Variable(변수) : 값을 저장하는 장소

변수 선언 방법

Variables

이름 지정 규칙



0

Ovariable < 숫자가 맨 앞

Print

<- 숫자로만 이루어져 있음

He llo

1234

<- 띄어쓰기

<- 키워드

He-llo

<- 하이픈(-)

Variable variable



대소문자 구별 가능 => 변수 이름으로 지정 가능

max_number

_hello

World_

Data Type 1

- int (정수)
- float (소수)
- complex (복소수)

```
var1 = 10
print(type(var1))

var2 = 3.14
print(type(var2))

var3 = 1+2j
print(type(var3))
```



```
<class 'int'>
<class 'float'>
<class 'complex'>
```

Data Type 1

- bool (T / F)
- str (문자열)

```
var4 = True
print(type(var4))

var5 = 'Hello'
print(type(var5))
```



```
<class 'booi'>
<class 'str'>
```

String

```
str = "Hello World!"
print(str)
print(str[0])
print(str[2:5])
print(str[2])
print(str * 2)
print(str + "TEST")
Hello World!
```

Arithmetic Operator

+	a = 2 result = a + 3 print(result)	2
-	a = 10 result = a - 3 print(result)	7
*	a = 3 result = a * 9 print(result)	27
/	a = 7 result = a / 2 print(result)	3.5

%	a = 7 result = a % 2 print(result)	1
**	a = 2 result = a ** 3 print(result)	8
//	a = 7 result = a // 2 print(result)	3

Assignment Operator

```
number = 20
result = number
print(result)

str1 = 'hello world'
str2 = str1
print(str2)

bool1 = True
bool2 = bool1
print(bool2)
20
hello world
True
```

Assignment Operator

=	a = 2 print(a)	2
+=	a = 2 a += 3 print(a)	5
-=	a = 10 a -= 4 print(a)	6
*=	a = 4 a *= 7 print(a)	28

/=	a = 72 a /= 8 print(a)	9
//=	a = 7 a //= 2 print(a)	3
%=	a = 7 a %= 2 print(a)	1
**=	a = 4 a **= 2 print(a)	16

Comparison Operator

==	5 == 3	False
!=	5 != 3	True
>	4 > 3	True
<	3 < 3	False
>=	6 >= 6	True
<=	7 <= 8	True

Logical Operator

and	모두 참일 경우에만 True 리턴	2 < 5 and 3 < 10	True
or	하나라도 참이면 True 리턴	2 < 4 or 7 > 5	True
not	결과가 False이면 True 리턴	not (3 < 4 and 5 < 2)	True

Identity Operator

is	두 변수가 같은 객체메모리면 True 리턴	<pre>x = ["apple", "banana"] y = ["apple", "banana"] z = x print(x is z)</pre>	True
is not	두 변수가 같은 객체메모리가 아니면 True 리턴	x = ["apple", "banana"] y = ["apple", "banana"] print(x is not y)	True

Membership Operator

in	객체 내에 해당 값이 포함된다면 True 리턴	x = ["apple", "banana"] print("banana" in x)	True
not in	객체 내에 해당 값이 포함되지 않는다면 True 리턴	x = ["apple", "banana"] print(("pineapple" not in x)	True

input & print

```
a = input()
print(a)
a = int(input())
print('a :', a)
a, b = input().split()
print("a + b = ", a+b)
a, b = map(int, input().split())
print("a + b =", a+b)
abc = input("""input : """)
print(abc)
```



```
hi
hi
8
a:8
12
a+b=12
12
a+b=3
input:string
string
```

а	b	a and b	a or b
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

ASCII: 미국에서 표준화가 추진된 정보교환용 7비트 부호

Dec Hx Oct Char Dec Hx Oct Html Chr Pf Cf Cf Hx Oct Html Chr Pf Cf Cf Hx Oct Html Chr Pf Cf Cf Hx Oct Html Chr Pf Cf	5; a 7; a 8; b 9; c 00; d 01; e 02; f 03; g 04; h 05; i 06; j 07; k
1 1 001 SOH (start of heading) 2 2 002 STX (start of text) 3 3 003 ETX (end of text) 3 4 22 042	7; a 3; b 9; c 00; d 01; e 02; f 03; g 04; h 05; i 06; j 07; k
2 2 002 STX (start of text) 34 22 042 c#34;" 66 42 102 c#66; B 98 62 142 c#53 3 003 ETX (end of text) 35 23 043 c#35; # 67 43 103 c#67; C 99 63 143 c#5	3; b 9; c 00; d 01; e 02; f 03; g 04; h 05; i 06; j 07; k
3 3 003 ETX (end of text) 35 23 043 # # 67 43 103 C C 99 63 143 U	9; c 00; d 01; e 02; f 03; g 04; h 05; i 06; j 07; k 08; 1
4 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 36 24 044 «#36; \$ 36 24 044 «#36; \$ 37 25 045 «#37; \$ 38 26 046 «#38; \$ 38 26 046 «#38; \$ 39 27 047 «#39; \$ 40 28 050 «#40; (40 28 050 «#40; (41 29 051 «#41;) 42 2A 052 «#42; \$ 43 110 «#72; H 44 112 «#74; J 45 2D 055 «#44; , 46 2E 056 «#44; , 47 2F 057 «#47; / 48 E 116 «#78; M 48 110 6E 156 «#1 49 31 061 «#49; 1 49 31 061 «#49; 1 49 31 061 «#49; 1 49 31 061 «#49; 1 48 52 022 CC2 (device control 2) 49 32 062 «#50; 2 40 44 104 «#68; D 40 106 64 144 «#1 40 106 64 106 «#6; E 40 106 (#48; D 40 106 (#48; D 40 106 (#48; D 40 107 (#70; F 40 107 (#70; F 40 107 (#70; D 40 108 (#70; F 40 108 (#70; H 40 108 (#70;	00; d 01; e 02; f 03; g 04; h 05; i 06; j 07; k
5 5 005 ENQ (enquiry) 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 37 25 045 6#37; % 38 26 046 6#38; 6 70 46 106 6#70; F 102 66 146 6#1 70 46 106 6#70; F 102 66 146 6#1 70 48 110 6#70; I 102 66 146 6#1 70 48 110 6#70; I 103 67 147 6#1 104 28 050 6#40; (72 48 110 6#72; II 105 69 151 6#1 105 69 151 6#1 107 68 153 6#1 107 68 153 6#1 108 6C 154 6#1 109 6D 155 6#1 109 6D 155 6#1 110 65 145 6#1 100 100 6#70; F 100 6#106 6#	01; e 02; f 03; g 04; h 05; i 06; j 07; k
6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 38 26 046 6#38; 6 38 26 046 6#38; 6 39 27 047 6#39; 1 70 46 106 6#70; F 102 66 146 6#1 71 147 107 6#71; G 103 67 147 6#1 104 68 150 6#1 104 28 050 6#40; (72 48 110 6#72; H 104 68 150 6#1 17 14 107 6#71; G 103 67 147 6#1 104 68 150 6#1 104 68 150 6#1 105 69 151 6#1 106 6A 152 6#1 107 44 112 6#74; J 108 6C 154 6#1 17 14 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 18 2 52 122 6#82; R 114 72 162 6#1	02; f 03; g 04; h 05; i 06; j 07; k
7 7 007 BEL (bell) 39 27 047 6#39;	03; g 04; h 05; i 06; j 07; k 08; l
8 8 010 BS (backspace) 40 28 050 c#40; (72 48 110 c#72; H 104 68 150 c#1 10 A 012 LF (NL line feed, new line) 42 2A 052 c#42; * 74 4A 112 c#74; J 106 6A 152 c#1 11 B 013 VT (vertical tab) 43 2B 053 c#43; + 75 4B 113 c#75; K 107 6B 153 c#1 12 C 014 FF (NP form feed, new page) 44 2C 054 c#44; 76 4C 114 c#76; L 108 6C 154 c#1 13 D 015 CR (carriage return) 45 2D 055 c#45; - 77 4D 115 c#77; W 109 6D 155 c#1 14 E 016 SO (shift out) 46 2E 056 c#46; . 78 4E 116 c#78; N 100 6E 156 c#1 15 F 017 SI (shift in) 47 2F 057 c#47; / 79 4F 117 c#79; 0 111 6F 157 c#1 16 10 020 DLE (data link escape) 48 30 060 c#48; 0 80 50 120 c#80; P 112 70 160 c#1 18 12 022 DC2 (device control 2) 50 32 062 c#50; 2 82 52 122 c#82; R 114 72 162 c#1	04; h 05; i 06; j 07; k 08; l
9 9 011 TAB (horizontal tab) 41 29 051 6#41;) 73 49 111 6#73; I 105 69 151 6#1 10 A 012 LF (NL line feed, new line) 42 2A 052 6#42; * 74 4A 112 6#74; J 106 6A 152 6#1 11 B 013 VT (vertical tab) 43 2B 053 6#43; + 75 4B 113 6#75; K 107 6B 153 6#1 13 D 015 CR (carriage return) 45 2D 055 6#45; - 76 4C 114 6#76; L 108 6C 154 6#1 14 E 016 SO (shift out) 46 2E 056 6#46; . 78 4E 116 6#78; N 100 6E 156 6#1 15 16 10 020 DLE (data link escape) 48 30 060 6#48; O 80 50 120 6#80; P 112 70 160 6#1 18 12 022 DC2 (device control 2) 50 32 062 6#50; 2 82 52 122 6#82; R 114 72 162 6#1	05; i 06; j 07; k 08; l
10 A 012 LF (NL line feed, new line) 42 2A 052 c#42; * 43 2B 053 c#43; + 45 2B 053 c#43; + 47 2F 057 c#44; , 75 4B 113 c#74; J 106 6A 152 c#1 107 6B 153 c#1 11 021 DC1 (device control 1) 49 31 061 c#49; L 81 51 121 c#81; Q 113 71 161 c#1 18 12 022 DC2 (device control 2) 42 2A 052 c#42; * 74 4A 112 c#74; J 106 6A 152 c#1 107 6B 153 c#1	06; j 07; k 08; l
11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 43 2B 053 «#43; + 44 2C 054 «#44; , 45 2D 055 «#45; - 46 2E 056 «#46; . 47 2F 057 «#47; / 48 30 060 «#48; 0 48 30 060 «#48; 0 49 31 061 «#49; 1 50 32 062 «#50; 2 82 52 122 «#82; R 114 72 162 «#1	07; k
12 C 014 FF (NP form feed, new page) 44 2C 054 c#44; , 76 4C 114 c#76; L 108 6C 154 c#1 13 D 015 CR (carriage return) 45 2D 055 c#45; - 77 4D 115 c#77; M 109 6D 155 c#1 14 E 016 SO (shift out) 46 2E 056 c#46; . 78 4E 116 c#78; N 110 6E 156 c#1 15 F 017 SI (shift in) 47 2F 057 c#47; / 79 4F 117 c#79; O 111 6F 157 c#1 17 11 021 DC1 (device control 1) 49 31 061 c#49; L 81 51 121 c#81; O 113 71 161 c#1 18 12 022 DC2 (device control 2) 50 32 062 c#50; 2 82 52 122 c#82; R 114 72 162 c#1	08; 1
13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 45 2D 055 c#45; - 46 2E 056 c#46; . 47 2F 057 c#47; / 48 30 060 c#48; 0 49 31 061 c#49; 1 50 32 062 c#50; 2 77 4D 115 c#77; M 109 6D 155 c#1 78 4E 116 c#78; N 110 6E 156 c#1 79 4F 117 c#79; 0 111 6F 157 c#1 80 50 120 c#80; P 112 70 160 c#1 81 51 121 c#81; 0 113 71 161 c#1 82 52 122 c#82; R 114 72 162 c#1	
14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 46 2E 056 . . 47 2F 057 / / 48 30 060 0 0 49 31 061 1 1 49 31 061 1 1 50 32 062 2 2 82 52 122 R R 114 72 162 	
15 F 017 SI (shift in) 47 2F 057 6#47; / 79 4F 117 6#79; 0 111 6F 157 6#1 16 10 020 DLE (data link escape) 48 30 060 6#48; 0 80 50 120 6#80; P 112 70 160 6#1 17 11 021 DC1 (device control 1) 49 31 061 6#49; 1 81 51 121 6#81; 0 113 71 161 6#1 18 12 022 DC2 (device control 2) 50 32 062 6#50; 2 82 52 122 6#82; R 114 72 162 6#1	
16 10 020 DLE (data link escape) 48 30 060 c#48; 0 80 50 120 c#80; P 112 70 160 c#1 17 11 021 DC1 (device control 1) 49 31 061 c#49; 1 81 51 121 c#81; 0 113 71 161 c#1 18 12 022 DC2 (device control 2) 50 32 062 c#50; 2 82 52 122 c#82; R 114 72 162 c#1	
17 11 021 DC1 (device control 1)	
18 12 022 DC2 (device control 2) 50 32 062 6#50; 2 82 52 122 6#82; R 114 72 162 6#1	
	14; r
19 13 023 DC3 (device control 3) 51 33 063 3 3 83 53 123 S 5 115 73 163 	
20 14 024 DC4 (device control 4) 52 34 064 6#52; 4 84 54 124 6#84; T 116 74 164 6#1	
21 15 025 NAK (negative acknowledge) 53 35 065 6#53; 5 85 55 125 6#85; U 117 75 165 6#1	17; u
22 16 026 SYN (synchronous idle) 54 36 066 6 6 86 56 126 V V 118 76 166 	18; 🔻
23 17 027 ETB (end of trans. block) 55 37 067 6#55; 7 87 57 127 6#87; W 119 77 167 6#1	
24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 	20; X
25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 	21; Y
26 1A 032 SUB (substitute) 58 3A 072 : 90 5A 132 Z Z 122 7A 172 	100
27 1B 033 ESC (escape) 59 3B 073 ; ; 91 5B 133 [[123 7B 173 	23; {
28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 	24;
29 1D 035 GS (group separator) 61 3D 075 4#61; = 93 5D 135 4#93;] 125 7D 175 4#1	
30 1E 036 RS (record separator) 62 3E 076 6#62; > 94 5E 136 6#94; ^ 126 7E 176 6#1	
31 1F 037 US (unit separator) 63 3F 077 4#63; 2 95 5F 137 4#95; _ 127 7F 177 4#1	26; ~

A ~ Z	65 ~ 90
a~z	97 ~ 122

• If문

```
a = 1
if (a == 0):
    print('a =', 0)
elif(a == 1):
    print('a =', 1)
else:
    print('Nothing')
```



$$a = 1$$

```
1 flag = True
2
3 if flag:
4 print("참")
5 else:
6 print("거짓")
```





• for문

```
for <variable> in <list> : <block>
```

```
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number +1
    if mark >= 60:
        print("%d번 학생은 합격입니다." % number)
    else:
        print("%d번 학생은 불합격입니다." % number)
```



1번 학생은 합격입니다. 2번 학생은 불합격입니다. 3번 학생은 합격입니다. 4번 학생은 불합격입니다. 5번 학생은 합격입니다.

```
for <variable> in range (start, end) :
     <block>
```

```
for i in range (2,10):
    for j in range(1,10):
        print(i*j,end=" ")
    print('')
```

```
2 4 6 8 10 12 14 16 18
3 6 9 12 15 18 21 24 27
4 8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
```

• while문

```
while <expression> : <block>
```

```
cnt = 0
while(cnt < 10):
    print(cnt)
    cnt = cnt + 1</pre>
```

break

```
s = "C2H6O"
index = 1
for i in range(len(s)):
    if s[i].isdigit():
        index = i
        print(index)
```



1

```
s = "C2H60"
index = 1
for i in range(len(s)):
    if s[i].isdigit():
        index = i
        print(index)
        break
```



1

continue

```
for i in range(11):
    if i % 2 == 0:
        continue
    print(i)

1
3
5
7
9
```

List

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print(list)
print(list[0] )
print(list[1:3])
print(list[2:])
print(tinylist * 2)
print(list + tinylist)
```

```
['abcd', 786, 2.23, 'john', 70.2]
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```

Tuple

```
tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
tinytuple = (123, 'john')

print(tuple)
print(tuple[0])
print(tuple[1:3])
print(tuple[2:])
print(tinytuple * 2)
print(tuple + tinytuple)
```

```
('abcd', 786, 2.23, 'john', 70.2)
abcd
(786, 2.23)
(2.23, 'john', 70.2)
(123, 'john', 123, 'john')
('abcd', 786, 2.23, 'john', 70.2, 123, 'john')
```

Dictionary

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print(list)
print(list[0] )
print(list[1:3])
print(list[2:])
print(tinylist * 2)
print(list + tinylist)
```

```
['abcd', 786, 2.23, 'john', 70.2]
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```