[CSE1017]
프로그래밍기초

expression

#01. 식

김현하

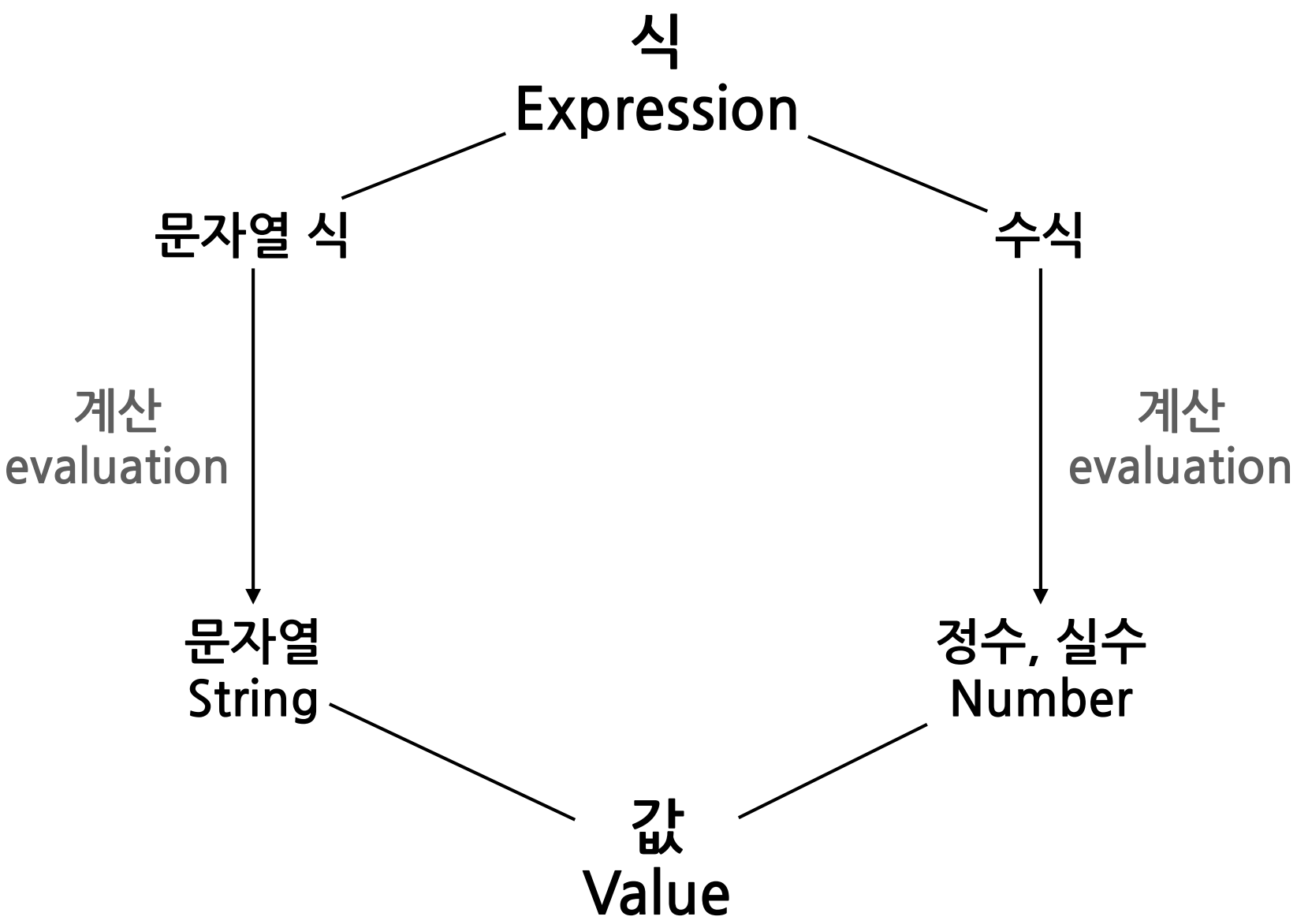
한양대학교 ERICA 소프트웨어학부

2024년도 1학기

python™

목차

식


식
Expression
문자열 식
수식
계산
evaluation
계산
evaluation
문자열
String
정수, 실수
Number
값
Value

문자열

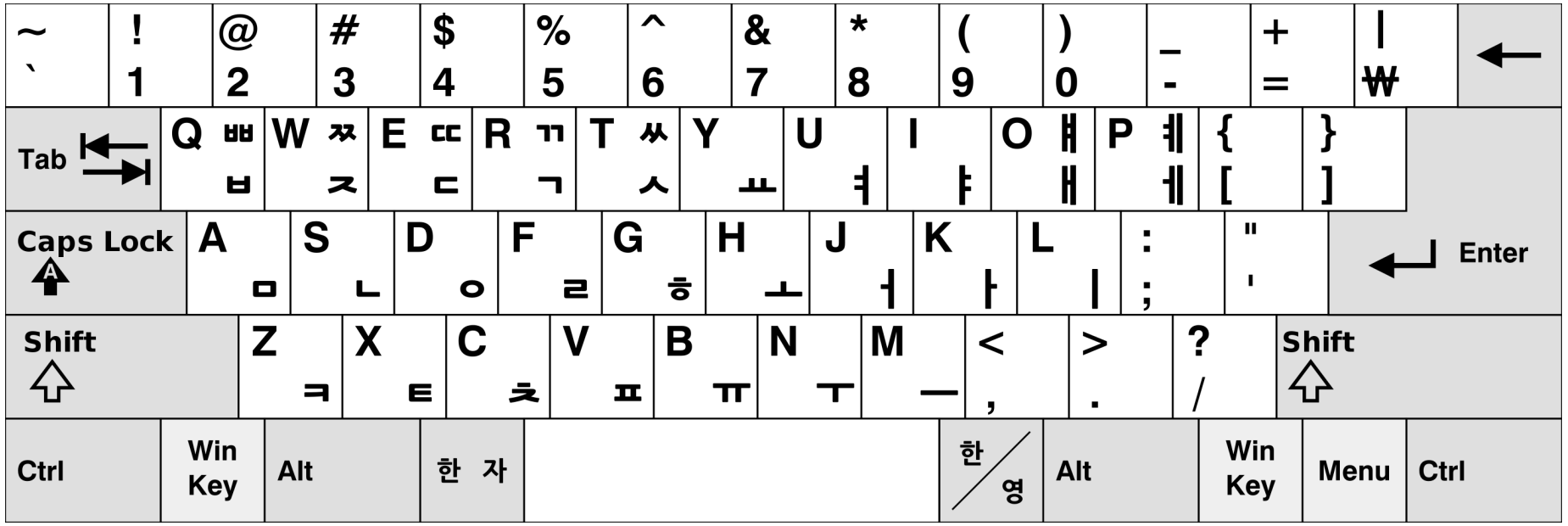
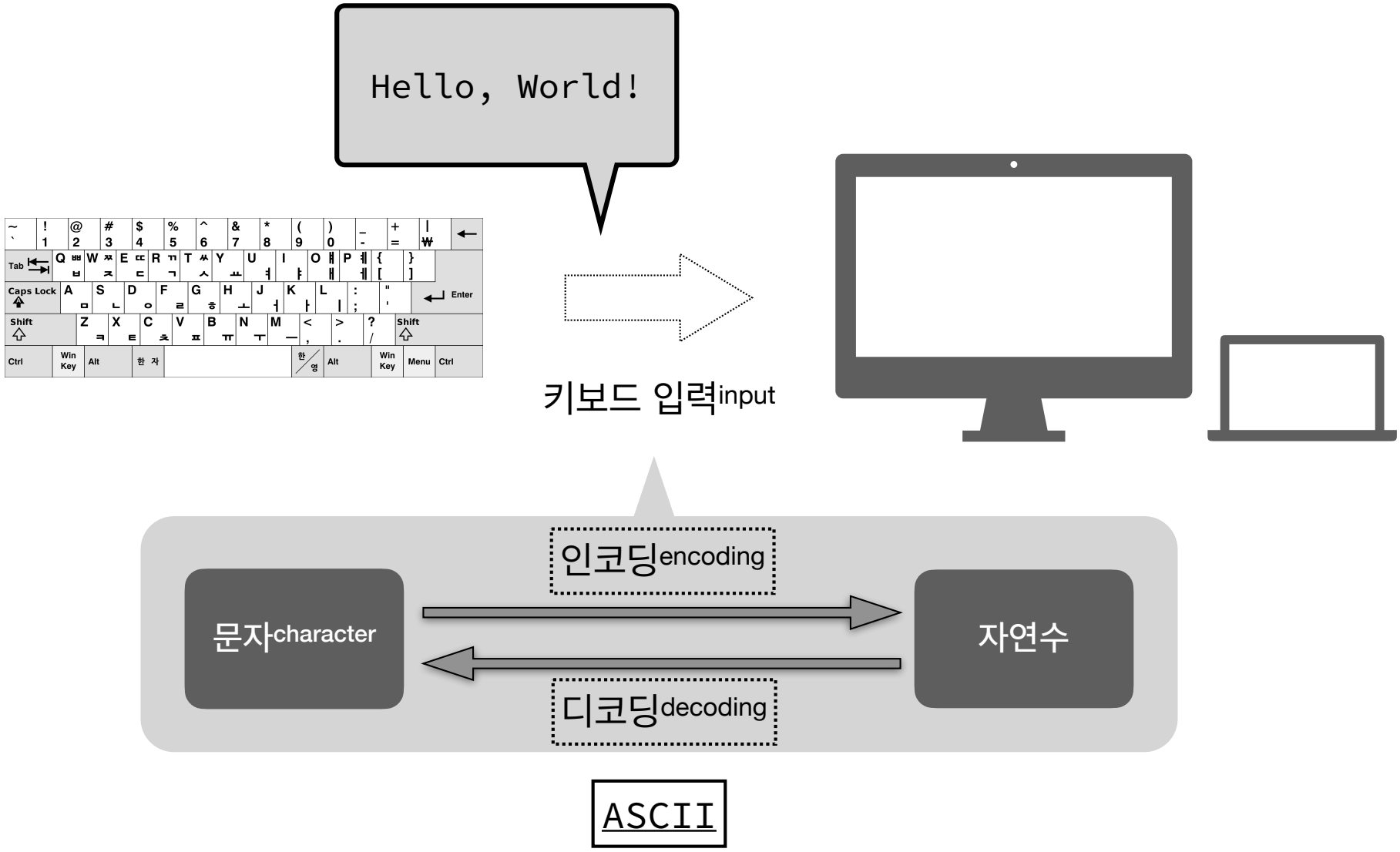


그림 출처 : https://ko.wikipedia.org/wiki/자판_배열



American Standard Code for Information Interchange

그림 출처 : https://ko.wikipedia.org/wiki/자판_배열

문자character

- 일반적으로 (작은)따옴표(')로 둘러쌓아 표현, 길이는 1
- Python 에서는 큰따옴표(")로 둘러쌓아도 됨

	기능	사용 예시	사용 예시 결과
ord	입력받은 문자의 아스키 코드 값을 리턴	`ord('a')`	`97`
chr	입력받은 아스키 코드 값에 해당하는 문자를 리턴	`chr(97)`	`'a'`

문자열string

- 문자를 차례로 이어 붙여 놓은 것
- Python에서는 큰따옴표(")나 작은따옴표(')로 둘러쌓아 표현
  (보통 다른 언어에서는 작은따옴표: 문자 / 큰따옴표: 문자열)
- 문자열의 시작과 끝을 나타내는 큰따옴표(")와 작은따옴표(')를 문자열 구분문자string delimiter 라고 함

	기능	사용 예시	사용 예시 결과
+	붙이기 연산	`'a' + 'b'`	`'ab'`
''	작은 따옴표를 사용한 빈 문자열empty string	`''`	`''`
""	큰 따옴표를 사용한 빈 문자열empty string	`""`	`''`
*	곱하기 연산	`'a' * 3`	`'aaa'`

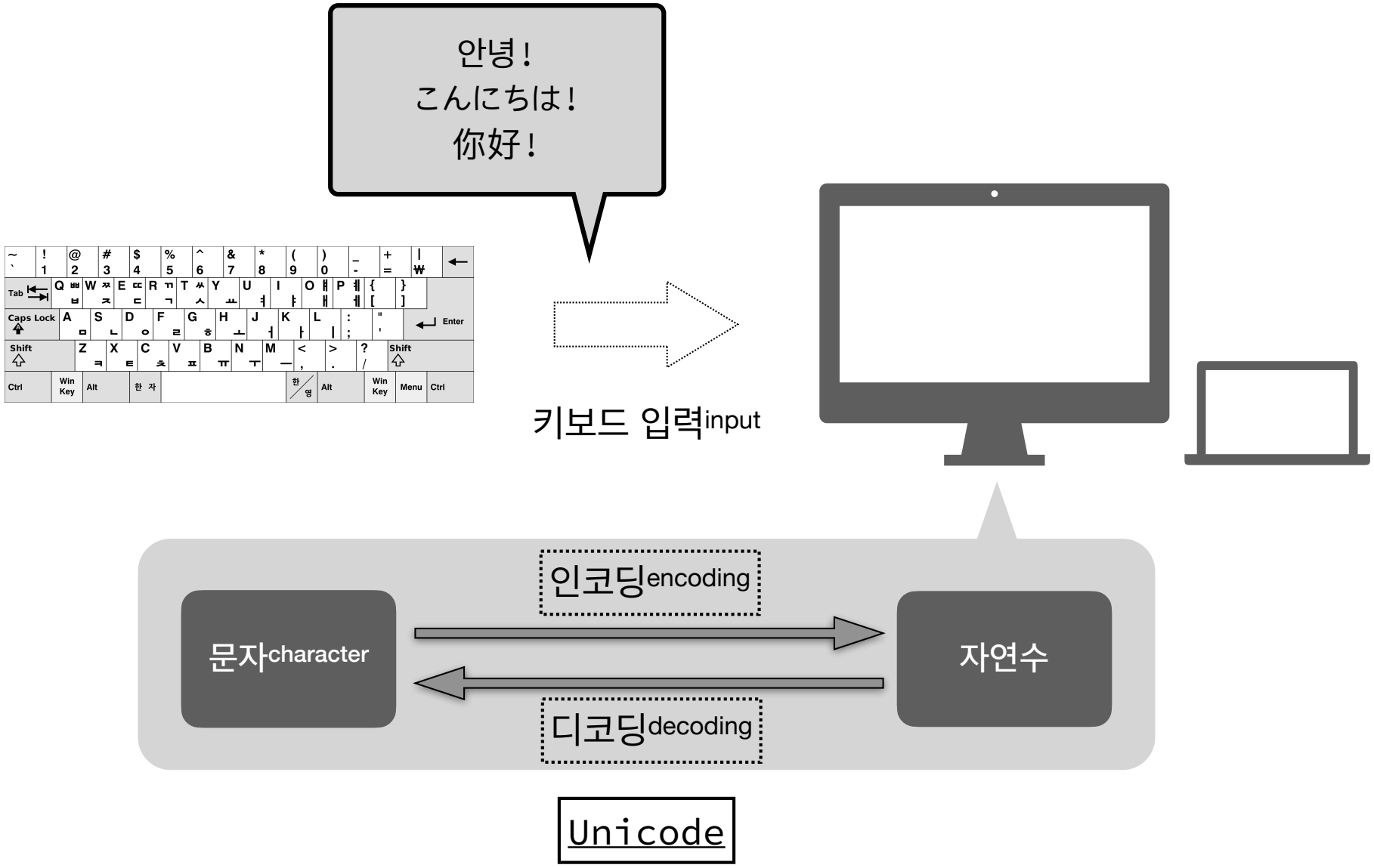


그림 출처 : https://ko.wikipedia.org/wiki/자판_배열

문자character

- Python 의 문자는 기본적으로 유니코드를 사용
- 한글을 포함한 세계 각국의 문자를 혼용해서 사용 가능

	기능	사용 예시	사용 예시 결과
ord	입력받은 문자의 **유니코드** 값을 리턴	`ord('한')`	`54620`
chr	입력받은 **유니코드** 값에 해당하는 문자를 리턴	`chr(54620)`	`'한'`

문자열 프린트print하기

- 계산 결과를 모니터 창에 보여줄 때 `print()` 명령을 사용
- 표준 출력standard output : 모니터 창에 프린트 하는 것

	기능
`\n`	새줄newline 문자
`\t`	tab 문자
`\`	문자열 내부 줄넘기기
`"""`	문자열을 입력한 모양대로 구성하기
`'''`	

`sep, end`	`print()` 명령의 기본 옵션

새줄문자 \n

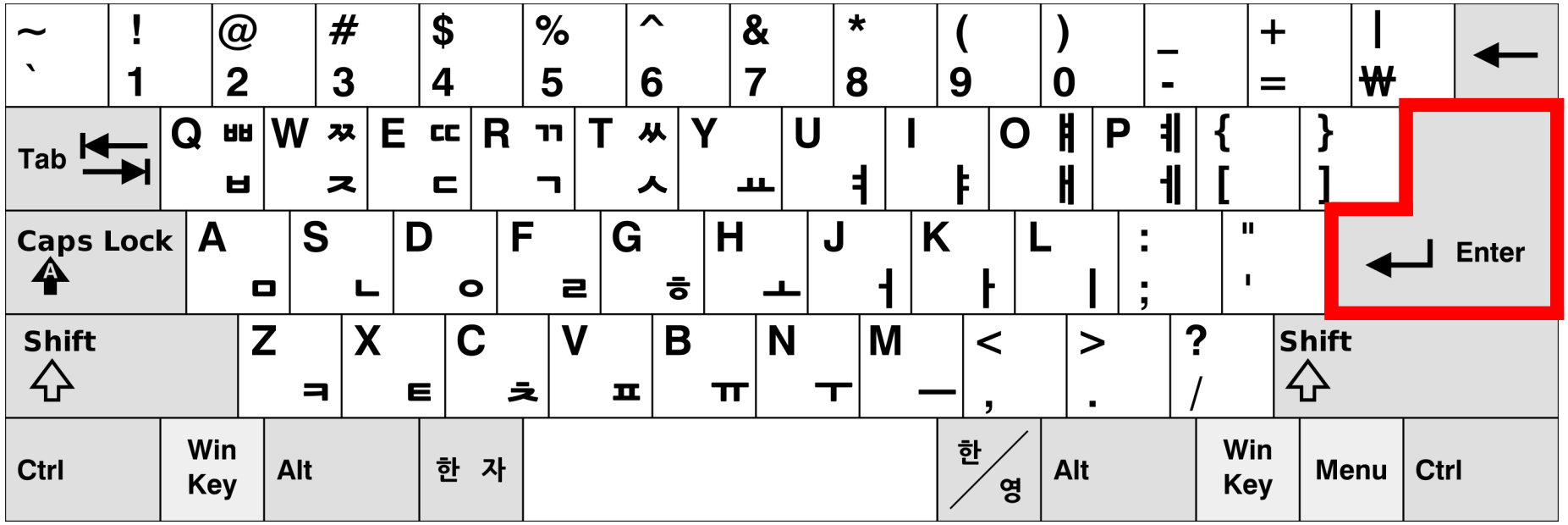


그림 출처 : https://ko.wikipedia.org/wiki/자판_배열

tab 문자 \t

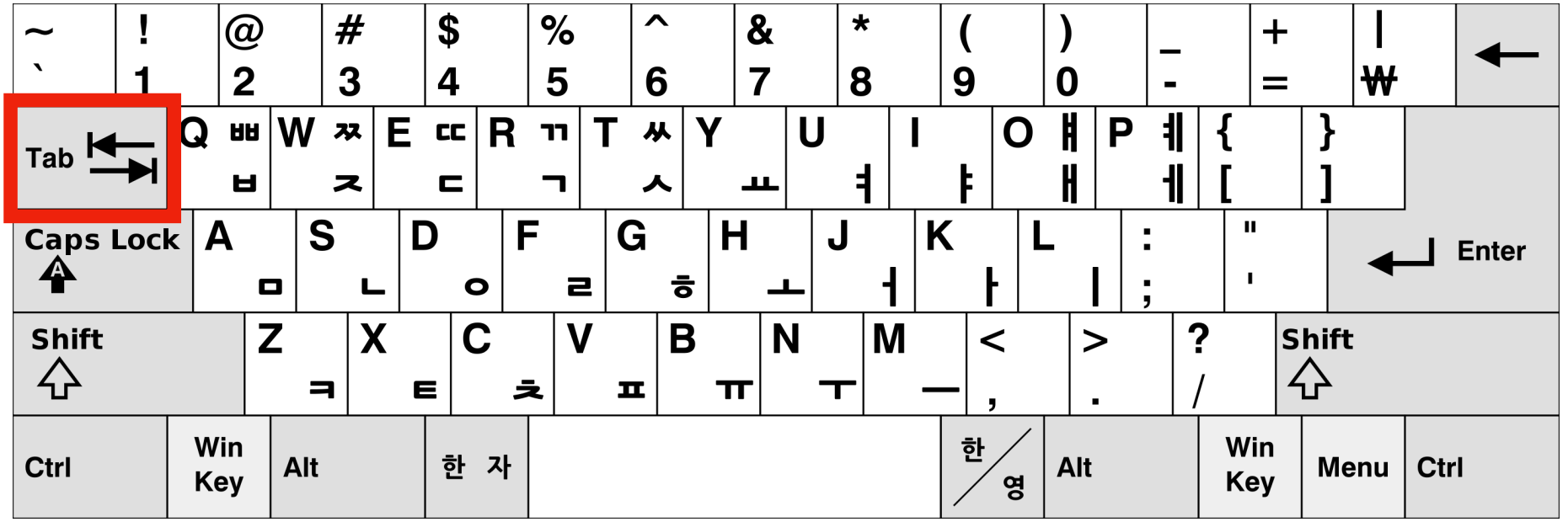


그림 출처 : https://ko.wikipedia.org/wiki/자판_배열

역슬래시|back slash

그림 출처 : https://www.apple.com/kr/shop/product/MLA22KC/A/magic-keyboard-한국어

역슬래시|back slash

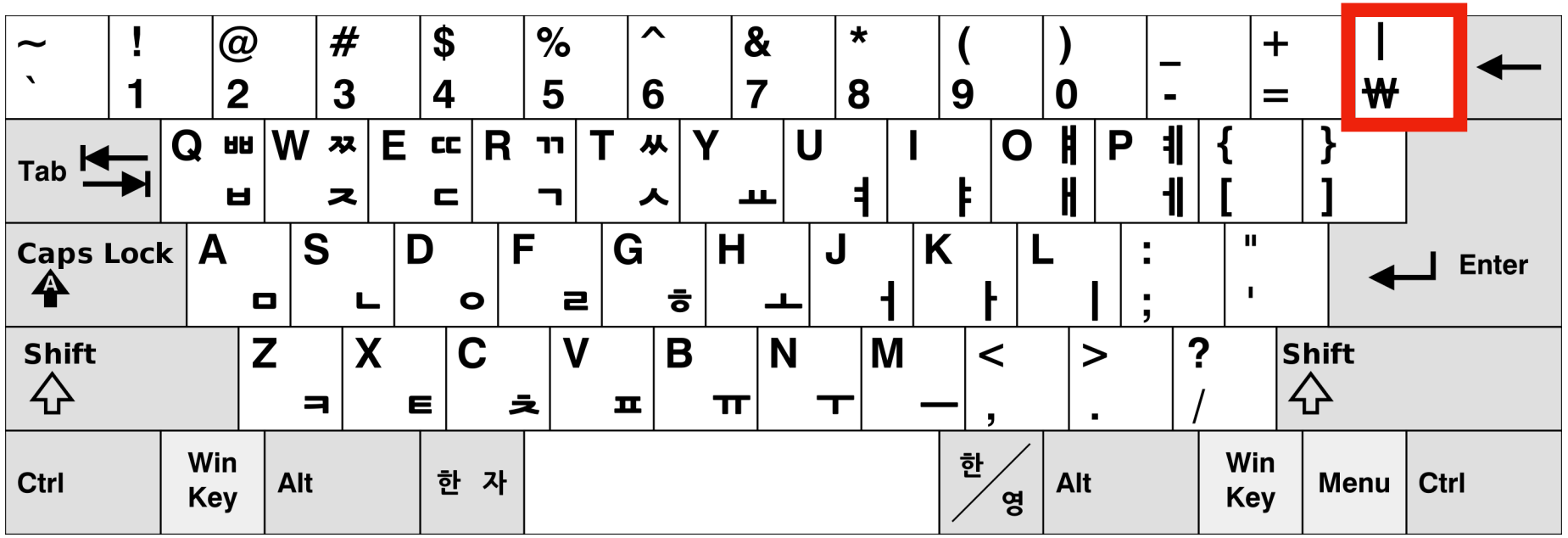


그림 출처 : https://ko.wikipedia.org/wiki/자판_배열

꽃(김춘수)

내가 그의 이름을 불러주기 전에는
그는 다만
하나의 몸짓에 지나지 않았다.

```
>>> print("내가 그의 이름을 불러주기 전에는\n그는 다만\n하나의 몸짓에 지나지 않았다.")

>>> print("내가 그의 이름을 불러주기 전에는\n\t그는 다만\n하나의 몸짓에 지나지 않았다.")

>>> print("내가 그의 이름을 불러주기 전에는\
\t그는 다만\
하나의 몸짓에 지나지 않았다.")

>>> print("내가 그의 이름을 불러주기 전에는\n\
\t그는 다만\n\
하나의 몸짓에 지나지 않았다.")

>>> print("""내가 그의 이름을 불러주기 전에는
\t그는 다만
하나의 몸짓에 지나지 않았다.""")

>>> print("내가 그의 이름을 불러주기 전에는",
          "\t그는 다만",
          "하나의 몸짓에 지나지 않았다.", sep="\n")
```

	기능
\n	새줄newline 문자
\t	tab 문자
\	문자열 내부 줄넘기기
"""	문자열을 입력한 모양대로 구성하기
'''	
sep	print()의 값 사이 출력
end	print() 맨 뒤 출력

특수문자 \t

```
print("1\t1234567890\t123")
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1				**\t**				**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**			**\t**				**1**	**2**	**3**					

그림 출처 : https://ko.wikipedia.org/wiki/자판_배열

문자의 탈바꿈escape

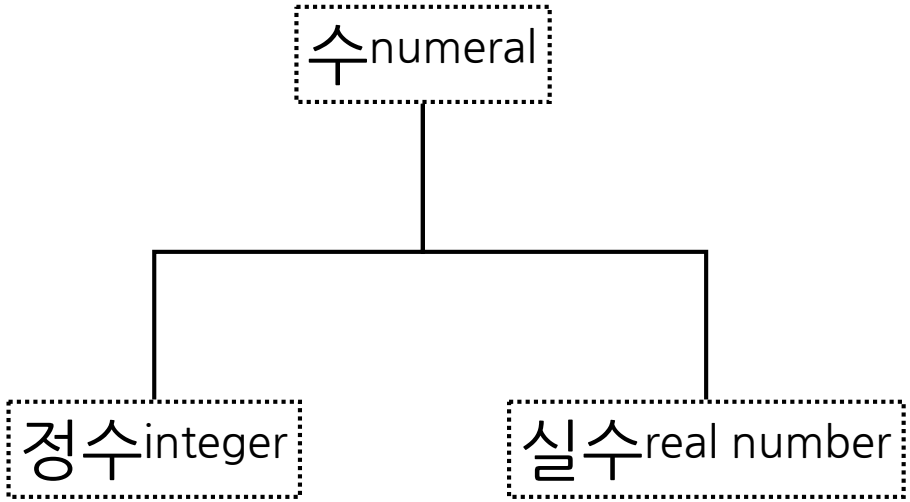
- (문자열) 구분문자를 문자열 내용의 일부로 포함할 때 사용

	기능	사용 예시	사용 예시 결과
\	탈바꿈 문자	'I\'m Good.'	'I\'m Good.'

```
But I still haven't found what I'm looking for
 print("But I still haven't found what I'm looking for")
 print('But I still haven\'t found what I\'m looking for')
He said, "Run!"
 print('He said, "Run!"')
 print("He said, \"Run!\"")
```

수식arithmetic expression

수의 표현



정수

- [Python 인터프리터]
  - 55
  - +3
  - 0
  - -13

실수

- 고정소수점fixed point 방식
- [Python 인터프리터]
  - 3.141592
  - +1.414
  - -324.8

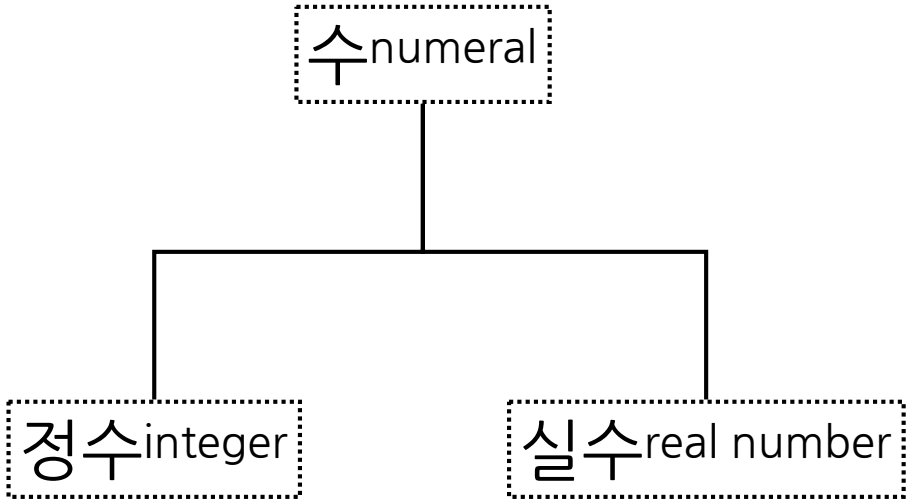
- 부동소수점floating point 방식

$$\underbrace{2.5}_{\text{significand}} \times \underbrace{10}_{\text{base}}^{\overbrace{-9}^{\text{exponent}}}$$

가수 significand base 기저 exponent 지수

- [Python 인터프리터]
  - 2.5e-9
  - 0.25e-8
  - 0.0000000025

수의 표현



산술 연산자 기호

이항연산자binary operator, 중위표기infix notation

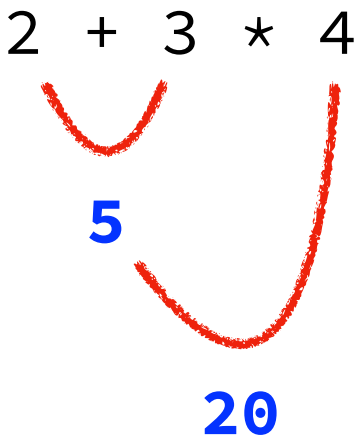
연산	더하기	빼기	곱하기	나누기	몫	나머지	거듭제곱
기호	+	-	*	/	//	%	**

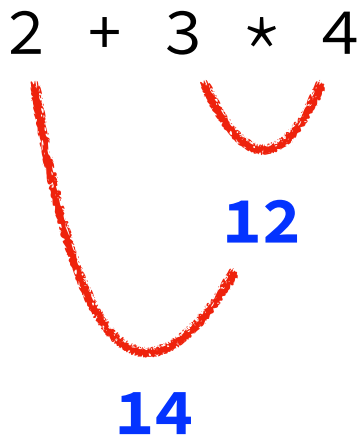
단항연산자unary operator, 전위표기prefix notation

연산	부호바꾸기
기호	-

우선순위precedence

- 연산자가 두 개 이상 섞여 있는 수식에서 연산자의 계산 순서





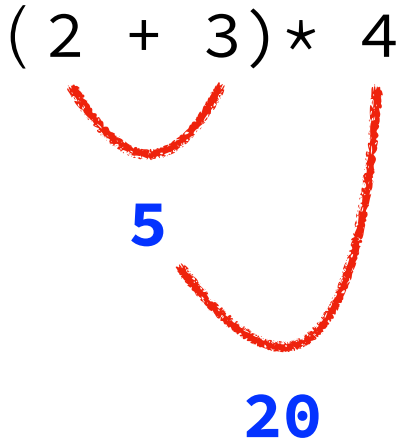
우선순위precedence

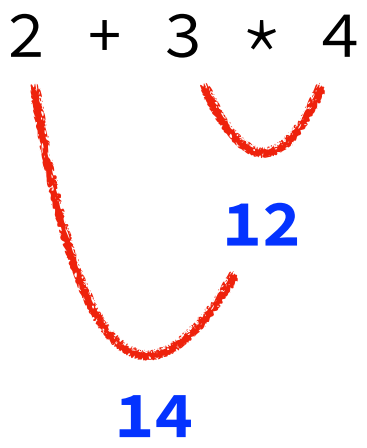
- 연산자가 두 개 이상 섞여 있는 수식에서 연산자의 계산 순서
- [Python 인터프리터]
  - 2 + 3 * 4

우선순위	연산자	설명
가장 높음	**	거듭제곱
높음	-	부호바꾸기
낮음	*, /, //, %	곱하기, 나누기, 몫, 나머지
가장 낮음	+, -	더하기, 빼기

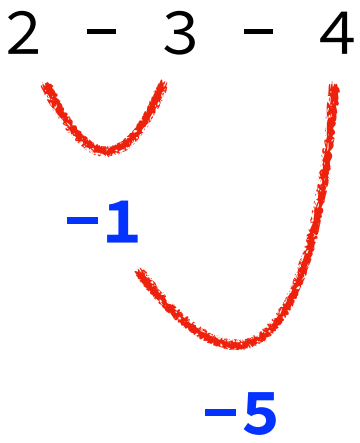
우선순위precedence

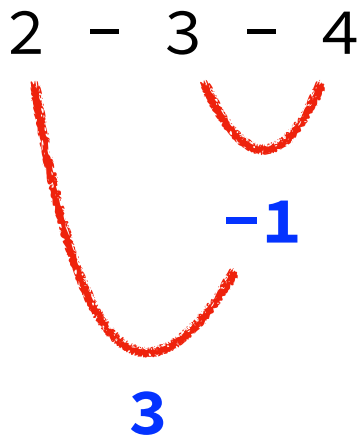
- 연산자가 두 개 이상 섞여 있는 수식에서 연산자의 계산 순서





결합순서associativity



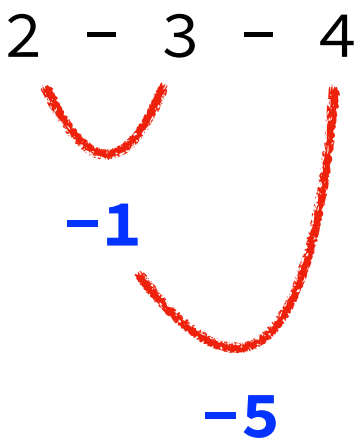


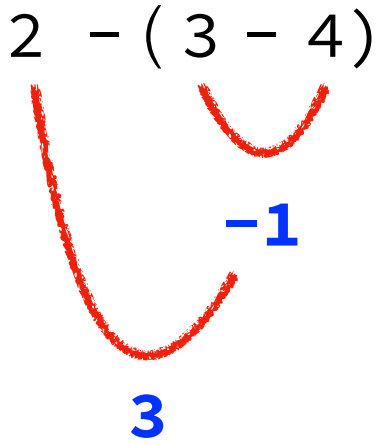
결합순서associativity

- $e_1\ op_1\ e_2\ op_2\ e_3$
  - 좌결합left associative : 왼쪽 연산자 먼저 계산
    $(e_1\ op_1\ e_2)\ op_2\ e_3$
  - 우결합right associative : 오른쪽 연산자 먼저 계산
    $e_1\ op_1\ (e_2\ op_2\ e_3)$

우선순위	연산자
우결합	**
좌결합	*, /, //, %, +, -

결합순서associativity





실수오차

정수 int	실수 float
무한히 많으나 셀 수 있음	셀 수 없이 무한히 많음
가용 메모리 한도 안에서 아무리 큰 수라도 파이썬 프로그램으로 모두 처리 가능	모두 처리가 불가능해서 불가피하게 근사치로 처리

정수

- 정수는 (가용 메모리가 허용하는 한도 안에서) 모두 처리 가능
- [Python 인터프리터]
  - `2 ** 100`
  - `2 ** 1000`
  - `2 ** 2000`

실수오차

- 컴퓨터는 이진수로만 계산이 가능
- 곱셈을 하기 전에 실수를 이진수로 바꾸면서 오차 발생
- [Python 인터프리터]
  - `0.1 * 0.1`

실수오차

이진수	십진수
0_2	0
1_2	1
10_2	2
11_2	3
100_2	4
101_2	5
110_2	6
111_2	7
1000_2	8
1001_2	9
1010_2	10
1011_2	11
1100_2	12
1101_2	13
1110_2	14

이진수	십진수
0.1_2	0.5
0.01_2	0.25
0.11_2	0.75
0.001_2	0.125
0.011_2	0.375
0.101_2	0.625
0.111_2	0.875
0.0001_2	0.0625
0.0011_2	0.1875
0.0101_2	0.3125
0.0111_2	0.4375
0.1001_2	0.5625
0.1011_2	0.6875
0.1101_2	0.8125
0.1111_2	0.9375

십진수 0.1?

이진수	십진수
0.0001_2	0.0625
0.00011_2	0.09375
0.000111_2	0.109375
0.0001101_2	0.1015625
0.00011001_2	0.09765625
0.000110011_2	0.099609375
0.0001100111_2	0.1005859375
0.00011001101_2	0.10009765625
0.000110011001_2	0.099853515625
…	…

타입type

- 타입 : 식expression을 계산한 값을 식별하는 분류
  - 문자열 타입 : **`str`**
  - 정수 타입 : **`int`**
  - 실수 타입 : **`float`**

연산자 중복사용overloading

- + 연산자는 정수 더하기, 실수 더하기, 문자열 붙이기에 모두 사용 가능
- [Python 인터프리터]
  - `20 + 23`
  - `20 + 23.0`
  - `"20" + "23"`
  - `"Year " + 2023`

타입type 변환함수

타입 변환 함수	기능
str(x)	정수 또는 부동소수점수 x를 문자열로 변환
int(x)	정수문자열 또는 부동소수점수 x를 정수로 변환 (부동소수점수를 정수로 변환할 때 소수점 이하는 버림)
float(x)	수문자열 또는 정수 x를 부동소수점수로 변환

- [Python 인터프리터]
  - `str(20) + str(23.3)`
  - `int("2000") + int(23.3)`
  - `float("2000") + float("23.3")`
  - `float(2023)`
  - `int("2023.3")`

오류

오류Error (=버그Bug)

구문 오류 Syntax Error	실행 오류 Run-time Error
문법 오류	타입 오류 Type Error
	값 오류 Value Error
	나누기0 오류 Zero Division Error
	…

- 디버깅Debugging : 오류를 찾고 원인을 밝혀서 수정하는 작업 과정

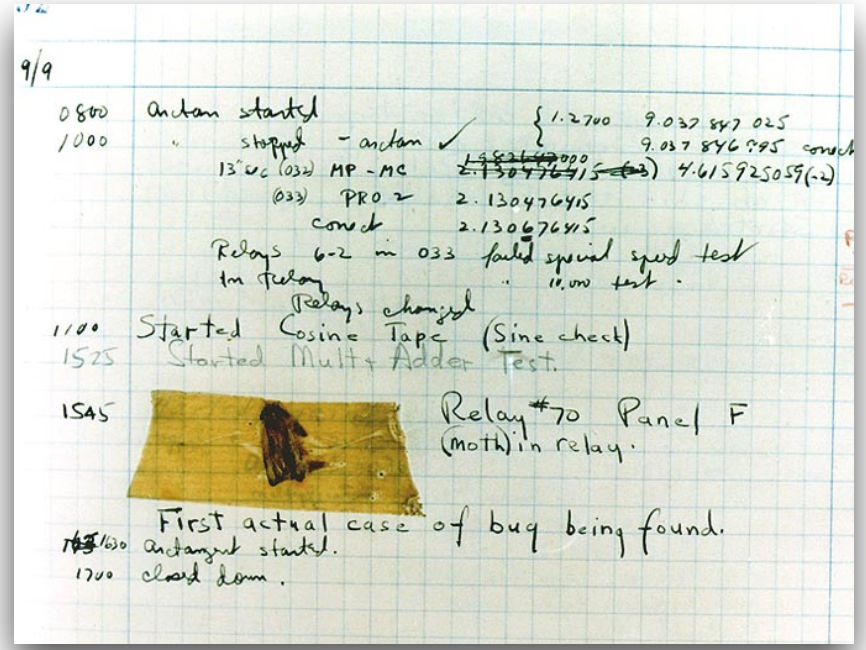


그림 출처 : https://en.wikipedia.org/wiki/Debugging

구문 오류syntax error

- 프로그램을 문법에 맞지 않게 작성하여 실행 전 발생하는 오류
- [Python 인터프리터]
  - `3 + 4 *`
  - `"Korea'`

실행 오류run-time error

- 문법 검사를 통과한 프로그램이 실행 도중 발생하는 오류

- [Python 인터프리터]
  - `"20" + 23`
  - `int("2023.3")`
  - `2023 / 0`