

	주요 문법	설명	예제코드
변수	변수명 = 데이터	숫자시작X, 특수기호X,예약어X, 밑줄O	a = 12345
리스트	list1 = [1,2,3]	대괄호[] 를 사용, 빈리스트는 []	score = [80, 90, 95, 100]
	list1 + list2	두 리스트를 연결	[1,2,3,4] + [5,6,7,8]
	list1*3	곱한 수만큼 반복	list1 * 3
	list[(start): end : (step)]	항목 일부 가져오기	list3[0: 9 : 3]
	del	항목 제거	del list3[4]
	in	값이 있는지 여부 확인	3 in list3
	append	항목을 맨 마지막에 추가	list4.append(10)
	insert	특정 위치에 항목을 추가	list4.insert(4, 'test')
	extend	항목 여러 개를 마지막에 추가	list4.extend([11, 12, '십삼'])
	remove	첫번째로 일치하는 항목 삭제	list4.remove('test')
	pop	마지막 항목을 제거한 후에 반환	list4.pop()
	index	리스트의 위치를 반환	list4.index('십삼')
	count	리스트 안에 포함된 항목의 개수를 반환	list4.count(4)
	sort	순방향 정렬	list6.sort()
	reverse	역순 정렬	list6.reverse()
튜플	tuple1 = (1,)	소괄호() 사용해서 생성	tuple1 = (1,2,3,4)
세트	set1 = {1,2,3}	중괄호{} 를 사용해서 생성	set1 = {1, 2, 3}
	intersection, &	교집합	set1.intersection(set2)
	union,	합집합	set1.union(set2)
	difference, -	차집합	set1.difference(set2)
딕셔너리	{키:값, 키:값}	생성	dict1 = {"대한민국":"서울", "일본":"도쿄"}
	추가	추가	dict1["중국"] = "베이징"
	변경	변경	dict1["일본"] = "오사카"
	del	삭제	del dict1["일본"]
	keys	키 전체를 리스트 형태로 반환	dictionary1.keys()
	values	값 전체를 리스트 형태로 반환	dictionary1.values()
	items	(키, 값)을 튜플 형태로 반환	dictionary1.items()
	update	딕셔너리에 딕셔너리 추가	dictionary1.update(dictionary2)
	clear	모든 항목 삭제	dictionary1.clear()
분기문	if	if <조건문 1>: <코드 블록 1>	if x >= 90: print("Pass")
	if ~ elif ~ else		if x >= 90: print(A) elif (x >= 80) and (x < 90): print("B") else: print("F")
반복문	for	for <반복 변수> in <반복 범위>: <코드 블록>	for a in [0, 1, 2, 3, 4, 5]: print(a)
	range(start, stop, step)	step이 1인 경우 생략 가능 start가 0인 경우 생략 가능	for a in range(0, 6, 1): print(a)
	while	while <조건문>: <코드 블록>	k=0 while True: k = k + 1 if(k > 3): break
	리스트 컴프리헨션	[<반복 실행문> for <반복 변수> in <반복 범위>]	square = [i**2 for i in numbers]
		[<반복 실행문> for <반복 변수> in <반복 범위> if <조건문>]	square = [i**2 for i in numbers if i>=3]
	print	문자열을 print() 함수 안에 삽입	print("Hello Python!!")
	print(a, b, c)	문자열 여러 개를 연결해서 출력	print("Best", "python", "book")
	sep	빈칸 대신 다른 문자열 삽입	print("Best", "python", "book", sep = "-.*:-")
	+	빈칸 없이 두 문자열을 연결	print("abcd" + "efg")
	Wn	문자열을 여러 줄로 출력	print("He is my friend.WnHe is Korean.")
	end=""	문자열을 한 줄로 연결해서 출력	print("Welcome to ", end="") print("python!")

	주요 문법	설명	예제코드
기본입출력	%s	%s로 문자열을 대입한 변수를 출력	name = "아이유" print("%s는 나의 친구입니다." % name)
	"{}".format()	형식 지정 문자열의 출력	print("동물: {0},{1}".format(ani0, ani1))
	{N:2d}	정수를 공백 포함해 두 자리로 표시	
	{N:05d}	정수를 다섯 자리로 표시. 앞의 공백은 0으로 채움	
	{N:>5d}	정수를 다섯 자리로 표시. 숫자는 오른쪽으로 정렬	
	{N:.3f}	실수를 소수점 셋째 자리까지 표시	
	{N:;}	통화 표시처럼 끝에서 셋째 자리마다 콤마(,)를 표시	
	{N:.1%}	소수를 퍼센트(%)로 표시	
	{N:.2e}	숫자를 지수로 표시	
	{N:#x}	숫자를 16진수로 표시. #기호가 없으면 0x 없이 출력됨	
	{N:#o}	숫자를 8진수로 표시. #기호가 없으면 0o 없이 출력됨	
	{N:#b}	숫자를 2진수로 표시. #기호가 없으면 0b 없이 출력됨	
	input	키보드로 데이터를 입력	yourName = input("당신의 이름은? ")
파일입출력	open	파일 열기	f = open('file_name', 'wr', encoding = '')
	write	문자열 한 줄씩 쓰기	f.write(str)
	read	파일 쓰기	text = f.read()
	readline()	문자열 한 줄을 읽음	f.readline()
	readlines()	파일 전체의 모든 줄을 읽어서 한 줄씩을 요소로 갖는 리스트 타입으로 반환	lines = f.readlines()
	close	파일 닫기	f.close()
	with 문		with open('C://myTextFile2.txt', 'w') as f: f.write('File read/write test2: line1\n')
함수	def	def 함수명([인자1, 인자2, ..., 인자n]): <코드 블록> [return <반환 값>]	def my_func(): print("My first function!")
	lamda	lambda <인자> : <인자 활용 수행 코드>	value = lambda x,y,z : 2*x + 3*y + z
		(lambda <인자> : <인자 활용 수행 코드>) (<인자>)	(lambda x : x**2) (3)
내장함수	int()	정수형으로 변환	int(0.123)
	float()	실수형으로 변환	float(0)
	str()	문자형으로 변환	str(123)
	list()	리스트로 변환	list((1,2,3)), list({1,2,3})
	tuple()	튜플로 변환	tuple([(1,2,3)], tuple({1,2,3}))
	set()	세트로 변환	set([1,2,3]), set({1,2,3})
	bool()	True 혹은 False의 결과를 반환	bool(0), bool('a')
	min()	최소값	tmp = (10, 5, 12, 0, 3.5, 99.5, 42) [min(tmp), max(tmp)]
	max()	최대값	
	abs()	절대값	abs(10)
	sum()	합계	sumList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] sum(sumList)
	len()	항목의 개수	len([1, 2, 3, 4, 5, 6, 7, 8])
객체지향 프로그래밍	클래스 선언		class 클래스명(): [변수1] # 클래스 변수 [변수2] ... def __init__(self, 인자1, 인자2, ..., 인자n): <코드 블록> ... def 함수1(self, 인자1, 인자2, ..., 인자n): <코드 블록>
	split	문자열 분리	str.split([sep])
	strip	문자열 삭제	str.strip([chars])
	join	문자열 연결	str.join(seq)
	find	문자열 찾기	str.find(search_str, start, end)
	count	문자열 일치 횟수 반환	str.count(search_str) str.count(search_str, start) str.count(search_str, start, end)
	startswith	문자열이 특정 문자열로 시작하는지 검사	str.startswith(prefix) str.startswith(prefix, start) str.startswith(prefix, start, end)

	주요 문법	설명	예제코드
문자열	endswith	문자열이 특정 문자열로 끝나는지 검사	str.endswith(suffix) str.endswith(suffix, start) str.endswith(suffix, start, end)
	replace	문자열 바꾸기	str.replace(old, new[, count])
	lower	소문자로 변경	str.lower()
	upper	대문자로 변경	str.upper()
	isalpha()	문자열이 숫자, 특수 문자, 공백이 아닌 문자로 구성돼 있을 때만 True, 그 밖에는 False 반환	str.isalpha()
	isdigit()	문자열이 모두 숫자로 구성돼 있을 때만 True, 그 밖에는 False 반환	str.isdigit()
	isalnum()	문자열이 특수 문자나 공백이 아닌 문자와 숫자로 구성돼 있을 때만 True, 그 밖에는 False 반환	str.isalnum()
	isspace()	문자열이 모두 공백 문자로 구성돼 있을 때만 True, 그 밖에는 False 반환	str.isspace()
	isupper()	문자열이 모두 로마자 대문자로 구성돼 있을 때만 True, 그 밖에는 False 반환	str.isupper()
	islower()	문자열이 모두 로마자 소문자로 구성돼 있을 때만 True, 그 밖에는 False 반환	str.islower()
모듈	random()	0.0 <= 실수 < 1.0 범위의 임의의 실수를 반환	random.random()
	randint(a,b)	a <= 정수 <= b의 범위의 임의의 정수 반환	random.randint(1,6)
	randrange([start,] stop [,step])	range([start,] stop [,step])에서 임의의 정수를 반환	random.randrange(0, 10, 2)
	choice(seq)	공백이 아닌 시퀀스(seq)에서 임의의 항목을 반환	random.choice(['1,2,3'])
	sample(population, k)	시퀀스로 이뤄진 모집단(population)에서 중복되지 않는 k개의 인자를 반환	random.sample(['1,2,3,4,5'], 2)
	calendar(year [,m=3])	지정된 연도(year)의 전체 달력을 문자열로 반환 (기본 형식은 3개의 열)	calendar.calendar(2017)
	month(year, month)	지정된 연도(year)와 월(month)의 달력을 문자열로 반환	calendar.month(2019,1)
	monthrange(year, month)	지정된 연도(year)와 월(month)의 시작 요일과 일수 반환. 요일의 경우 0(월요일) ~ 6(일요일) 사이의 숫자로 반환	calendar.monthrange(2020,1)
	firstweekday()	달력에 표시되는 주의 첫 번째 요일값을 반환. 기본값으로는 월요일(0)로 지정됨	calendar.firstweekday()
	setfirstweekday(weekday)	달력에 표시되는 주의 첫 번째 요일을 지정	calendar.setfirstweekday(6)
	weekday(year,month,day)	지정된 날짜[연도(year), 월(month), 일(day)]의 요일을 반환	calendar.weekday(year,month,day)
	isleap(year)	지정된 연도(year)가 윤년인지를 판단해 윤년이면 True를, 아니면 False를 반환	calendar.isleap(2020)
	date(연, 월, 일)	입력한 값을 날짜로 반환	datetime.date(2019, 3, 1)
	today()	오늘 날짜 반환	datetime.date.today()
	time(시, 분, 초)	입력한 값을 시간으로 반환	datetime.time(15, 30, 45)
	now()	현재 시간을 반환	datetime.datetime.now()
	year, month, day	날짜의 연, 월, 일 반환	set_day.year, set_day.month, set_day.day
	hour, minute, second	시간의 시, 분, 초 반환	set_time.hour, set_time.minute, set_time.second