

숫자 및 연산		
함수	설명	예시 코드
<b>abs(x)</b>	숫자의 절댓값을 반환합니다.	<code>print(abs(-5)) → 5</code>
<b>divmod(a, b)</b>	a를 b로 나눈 몫과 나머지를 튜플로 반환합니다.	<code>print(divmod(7, 3)) → (2, 1)</code>
<b>max(iterable)</b>	최댓값을 반환합니다.	<code>print(max([1, 2, 5])) → 5</code>
<b>min(iterable)</b>	최솟값을 반환합니다.	<code>print(min([1, 2, 5])) → 1</code>
<b>pow(x, y)</b>	x의 y 거듭제곱 값을 반환합니다.	<code>print(pow(2, 3)) → 8</code>
<b>round(number)</b>	실수를 반올림한 값을 반환합니다.	<code>print(round(3.14159, 2)) → 3.14</code>
<b>sum(iterable)</b>	모든 요소의 합을 반환합니다.	<code>print(sum([1, 2, 3])) → 6</code>
자료형 변환		
함수	설명	예시 코드
<b>int(x)</b>	값을 정수형으로 변환합니다.	<code>print(int('100')) → 100</code>
<b>float(x)</b>	값을 실수형으로 변환합니다.	<code>print(float('3.14')) → 3.14</code>
<b>str(x)</b>	값을 문자열로 변환합니다.	<code>print(str(123)) → '123'</code>
<b>list(iterable)</b>	반복 가능한 객체를 리스트로 변환합니다.	<code>print(list('hello')) → ['h', 'e', 'l', 'l', 'o']</code>
<b>tuple(iterable)</b>	반복 가능한 객체를 튜플로 변환합니다.	<code>print(tuple([1, 2, 3])) → (1, 2, 3)</code>
<b>set(iterable)</b>	반복 가능한 객체를 세트로 변환합니다. (중복 제거)	<code>print(set([1, 2, 2, 3])) → {1, 2, 3}</code>
<b>bool(x)</b>	값을 <b>**불리언 (True/False)**</b> 으로 변환합니다.	<code>print(bool(0)) → False</code>
자료형 변환		
함수	설명	예시 코드
<b>int(x)</b>	값을 정수형으로 변환합니다.	<code>print(int('100')) → 100</code>
<b>float(x)</b>	값을 실수형으로 변환합니다.	<code>print(float('3.14')) → 3.14</code>
<b>str(x)</b>	값을 문자열로 변환합니다.	<code>print(str(123)) → '123'</code>
<b>list(iterable)</b>	반복 가능한 객체를 리스트로 변환합니다.	<code>print(list('hello')) → ['h', 'e', 'l', 'l', 'o']</code>
<b>tuple(iterable)</b>	반복 가능한 객체를 튜플로 변환합니다.	<code>print(tuple([1, 2, 3])) → (1, 2, 3)</code>
<b>set(iterable)</b>	반복 가능한 객체를 세트로 변환합니다. (중복 제거)	<code>print(set([1, 2, 2, 3])) → {1, 2, 3}</code>
<b>bool(x)</b>	값을 <b>**불리언 (True/False)**</b> 으로 변환합니다.	<code>print(bool(0)) → False</code>

시퀀스 및 반복 처리		
함수	설명	예시 코드
<b>len(s)</b>	객체의 **길이(요소 수)**를 반환합니다.	<code>print(len('python')) → 6</code>
<b>range(start, stop)</b>	지정된 범위의 숫자 시퀀스를 생성합니다.	<code>print(list(range(5))) → [0, 1, 2, 3, 4]</code>
<b>sorted(iterable)</b>	요소를 정렬한 새로운 리스트를 반환합니다.	<code>print(sorted([3, 1, 2])) → [1, 2, 3]</code>
<b>reversed(seq)</b>	시퀀스를 거꾸로 뒤집는 이터레이터를 반환함	<code>print(list(reversed('abc'))) → ['c', 'b', 'a']</code>
<b>enumerate(iter)</b>	인덱스와 값을 튜플 형태로 묶어서 반환합니다	<code>for i, v in enumerate(['a','b']): print(i,v)</code>
<b>zip(*iterables)</b>	여러 시퀀스를 동일한 인덱스끼리 묶어줍니다	<code>names=['A', 'B']&lt;br&gt;ages=[20, 30]&lt;br&gt;print(list(zip(names, ages)))</code>
<b>filter(func, iter)</b>	함수 결과가 True인 요소만 모아서 반환합니다	<code>def is_even(n): return n % 2 == 0&lt;br&gt;print(list(filter(is_even, [1,2,3,4])))</code>
<b>map(func, iter)</b>	각 요소에 함수를 적용한 결과를 반환합니다.	<code>def double(n): return n * 2&lt;br&gt;print(list(map(double, [1, 2, 3])))</code>
입출력 및 기타		
함수	설명	예시 코드
<b>print(*objects)</b>	주어진 값을 화면에 출력합니다.	<code>print('Hello', 'Python') → Hello Python</code>
<b>input([prompt])</b>	사용자로부터 키보드 입력을 받습니다.	<code>name = input('이름 입력: ')</code>
<b>type(object)</b>	객체의 자료형을 반환합니다.	<code>print(type(123)) → &lt;class 'int'&gt;</code>
<b>isinstance(obj, cls)</b>	객체가 특정 클래스의 인스턴스인지 확인합니	<code>print(isinstance(1, int)) → True</code>
<b>open(file, mode)</b>	파일을 열고 파일 객체를 반환합니다.	<code>f = open('test.txt', 'w')</code>