

pow함수에 대해 조금 자세히 알아보자.

pow(x, y[, z])
Return x to the power y; if z is present, return **x to the power y**, modulo z (computed more efficiently than pow(x, y) % z). The two-argument form pow(x, y) is equivalent to using the power operator: x**y.

실제 값을 입력해보면서 알아보자.

| | |
|-----------------------------------|-----------------------------------|
| <code>2 ** 3</code> | # 2의 3승 |
| 8 | |
| <code>pow(2, 3)</code> | # 2의 3승 |
| 8 | |
| <code>pow(100, 2)</code> | # 100의 2승 |
| 10000 | |
| <code>pow(100, -2)</code> | # 100의 2승분의 1 |
| 0.0001 | |
| <code>pow(100, (1.0/2.0))</code> | # 100의 제곱근 |
| 10.0 | |
| <code>pow(100, -(1.0/2.0))</code> | # 100의 제곱근분의 1 |
| 0.1 | |
| <code>pow(100, (1.0/3.0))</code> | # 100의 세제곱근 |
| 4.641588833612778 | |
| <code>pow(100, -(1.0/3.0))</code> | # 100의 세제곱근분의 1 |
| 0.2154434690031984 | |
| <code>pow(2, 3) % 3</code> | # 2의 3승을 3으로 나눈 나머지 |
| 2 | |
| <code>pow(2, 3, 3)</code> | # 2의 3승을 3으로 나눈 나머지(위보다 더 효율적 처리) |
| 2 | |
| <code>pow(2, 4) % 3</code> | # 2의 4승을 3으로 나눈 나머지 |
| 1 | |
| <code>pow(2, 4, 3)</code> | # 2의 4승을 3으로 나눈 나머지(위보다 더 효율적 처리) |

. x^y 를 구해주는 함수임.