- for i, v in enumerate(xs)
- for i, v in reversed(list(enumerate(xs)))
- import **itertools**
    - accumulate( ) *p[,func]*
    - groupby( ) *iterable[,key]*
        - *key=lambda x: x*
    - permutations( ) *iterable[,r]* and combinations ( ) *iterable[,r]*, product( ), combinations_with_replacement( )
- import **operator**
- import **collections**
    - Counter( )
    - defaultDict
    - OrderedDict
    - Deque
- import **heapq**
- lambda, map, filter, zip
- **List Comprehension** [ <***expression***> for <***element***> in <***iterable***> if <***condition***> ]
- **Generators**
    - Expression ( <***expression***> for <***element***> in <***iterable***> if <***condition***> )
    - Functions **yield** <**expression**> in place of return
    - **next**( ) - useful when working with infinite sequences and need next value
    - Generators save memory
- f-Strings - print(**f**"**{** ***Variable1*** **}** Text to print **{** ***Variable2*** **}**")
- **sorted**( ) and **.sort**( ) - sorted is out-of-place sort and .sort is a in-place sort for lists
    - Sorting and returning a dictionary by value:

      { k : v **for** k, v **in sorted**( d.**items**(), **key**=**lambda** x: x[1] ) }
- **sum**( ), **max**( ) and **min**( )
- **all**( ) and **any**( ) - returns true if all or any items are true
- **bin**(num), **oct**(num), and **hex**(num)
- q, r = **divmod**(num, k) - returns a tuple of quotient and remainder

- **ord**(‘ c ’) - retrieves the decimal value from ASCII table for a character