

CSCI 145 · Quiz 4 Reference Sheet (Lectures 7–8)

LOOPING OVER A STRING OR LIST

```
for item in sequence:
    statements
```

Each pass, the variable holds the next character or item

```
for i in range(len(seq)):
    statements using seq[i]
```

Use when you need the position (index) too

THE ACCUMULATOR PATTERN

```
result = starting value      # before the loop
for item in sequence:
    update result             # inside the loop
use result                   # after the loop
```

Building	Typical starting value
a count or sum	0
a product	1
a string	"" (the empty string)
a list	[] (the empty list)

- $x += y$ is shorthand for $x = x + y$
- **Best so far:** start with one candidate, and replace it whenever you find a better one
- `return` ends the function immediately, and the loop with it

NESTED LOOPS

A loop body can contain another loop. The inner loop runs all the way through for each pass of the outer loop.

INPUT FROM THE USER

input(prompt)	Prints the prompt, waits for Enter, and returns what the user typed, always as a string
int(s)	Converts a string of digits to an int, e.g. <code>int("19")</code> is 19

THE RANDOM MODULE (import random)

random.randint(a, b)	A random int from a to b, including both a and b
random.choice(seq)	One item of a sequence, chosen at random (a character, for a string)

NAMED CONSTANTS

```
MAX_TRIES = 5
VOWELS = "aeiou"
```

- A name in ALL_CAPS for a value that never changes, defined at the top of the file, outside any function
- Use one instead of a **magic number** (a value with no name)

LISTS

An ordered sequence of values of any type, in square brackets: `[4, "hi", 2.5]`. The empty list is `[]`.

Indexing, slicing, `len`, `+`, `*`, `in`, and `for` work the same way as for strings:

seq[i]	Item at index i; the first index is 0, and -1 is the last item
seq[a:b]	A new sequence from index a up to but not including b
len(seq)	Number of items
a + b	A new sequence: the items of a, then the items of b
seq * n	A new sequence: the items repeated n times
x in seq	True if x is in the sequence. Otherwise, False

MUTABLE VS. IMMUTABLE

- Lists are **mutable**: `items[i] = x` replaces the item at index i
- Strings are **immutable**: assigning to `s[i]` is an error
- String methods (`upper`, `replace`, ...) return a **new** string
- The list methods below change the list they are called on

LIST METHODS

Call	What it does	Returns
items.append(x)	Adds x to the end	None
items.insert(i, x)	Puts x at index i	None
items.remove(x)	Removes the first x	None
items.pop()	Removes the last item	removed item
items.pop(i)	Removes the item at index i	removed item

BUILT-IN FUNCTIONS FOR LISTS OF NUMBERS

sum(nums)	The total of the items
min(nums)	The smallest item
max(nums)	The largest item