My cheat sheet

Basic Types:

int
float
str
bool
list
range
NoneType (this is the ‘type’ of expressions that don’t have a value, such as print(x) or y.append(3))

Operations on these types: Ordinary arithmetic for float and int, additionally // and % for ints. Concatenation (+) and repetition (* with an int) for str and list. and, or and not.

Relational operations, applicable to int, float, str, bool, list, yielding bool result: <, <=, >, >=, ==, !=

Variables and assignment

Built-in functions: print, input, float, int, str

Calling functions and writing user-defined functions

Most important: Understand how the function communicates with the code that calls it. Typically functions print nothing and do not read user input, unless that is part of their job—values are passed to the function through its parameters and back from the function through the return statement.

Branching:

if, else, elif

Looping:

while (same syntax as if)

for (always used in form for v in w: where v is a variable and w is an expression whose type is a sequence type—str, range or list).
Common to all sequence types (str, range, list)

- **Indexing:** `s[j]`
- **Slices:** `s[j:k]`
- **Len function**
- **Sorted function (returns a list)**
- **In**
- **Count method**
- **Index method**

**Methods specific to strings:**

- `upper`
- `lower`

**Methods for lists**

These change the list but do not return a value

- **Append**
- **Extend**
- **Sort**

**List comprehension**

`y = [E for x in u]` where `E` is an expression, `x` is a variable and `u` is an expression of a sequence type, is equivalent to

```python
y = []
for x in u:
    y.append(E)
```

`y = [E for x in u if v]` where `v` is an expression of type bool is equivalent to:

```python
y = []
for x in u:
    if v:
        y.append(E)
```