

# Python Cheat Sheet

## LISTS

### Creating Lists

Create an empty list and use a for loop to append new values.

```
#add two to each value
my_list = []
for x in range(1,11):
    my_list.append(x+2)
```

We can also do this in one step using list comprehensions:

```
my_list = [x + 2 for x in range(1,11)]
```

### Creating Lists with Conditionals

As above, we will create a list, but now we will only add 2 to the value if it is even.

```
#add two, but only if x is even
my_list = []
for x in range(1,11):
    if x % 2 == 0:
        my_list.append(x+2)
    else:
        my_list.append(x)
```

Using a list comp:

```
my_list = [x+2 if x % 2 == 0 else x \
           for x in range(1,11)]
```

### Selecting Elements and Basic Stats

Select elements by index.

```
#get the first/last element
first_ele = my_list[0]
last_ele = my_list[-1]
```

Some basic stats on lists:

```
#get max/min/mean value
biggest_val = max(my_list)
smallest_val = min(my_list)
avg_val = sum(my_list) / len(my_list)
```

## DATAFRAMES

### Reading in Data to a DataFrame

We first need to import the pandas module.

```
import pandas as pd
```

Then we can read in data from csv or xlsx files:

```
df_from_csv = pd.read_csv('path/to/my_file.csv',
                          sep=',',
                          nrows=10)

xlsx = pd.ExcelFile('path/to/excel_file.xlsx')
df_from_xlsx = pd.read_excel(xlsx, 'Sheet1')
```

### Slicing DataFrames

We can slice our DataFrame using conditionals.

```
df_filter = df[df['population'] > 1000000]
df_france = df[df['country'] == 'France']
```

Sorting values by a column:

```
df.sort_values(by='population',
               ascending=False)
```

### Filling Missing Values

Let's fill in any missing values with that column's average value.

```
df['population'] = df['population'].fillna(
    value=df['population'].mean()
)
```

### Applying Functions to Columns

Apply a custom function to every value in one of the DataFrame's columns.

```
def fix_zipcode(x):
    '''
    make sure that zipcodes all have leading zeros
    '''
    return str(x).zfill(5)

df['clean_zip'] = df['zip code'].apply(fix_zipcode)
```