

```

_ci\n', ' id CODIGO_CI\n', '0 11001 4126450003\n', '1 11002 4126450005\n', '2
11003 4126450004\n', '3 11004 4126450083\n', '4 11005 4126450259\n', '5 11006
4126450092'] § Code bikes.merge(ciudades_ci, left_on='start_city_id', right_on='id')
§ Output > [' bike_id bike_type start_date start_hour ... dow_name id CODIGO_CI
\\n', '0 5 Electric 2019-09-01 06:30:36 AM ... domingo 11 412645 \n', '1 5 Electric
2019-09-01 06:30:36 AM ... domingo 11 412645 \n'...65003 \n', '2 50003 \n', '3
50003 \n', '4 50003 \n', '... ... \n', '41739 51001 \n', '41740 51001 \n', '41741 51001
\n', '41742 51001 \n', '41743 51001 \n', '\n', '[41744 rows x 15 columns]'] § Markdown
### Operaciones select Considerar el siguiente dataset § Code ciudad_ci =
pd.DataFrame({'id':[11001,11002,11003,11004,11005,11006],
'CODIGO_CI':['4126450003', '4126450005', '4126450004', '4126450083',
'4126450259', '4126450092']}) ciudad_ci § Output > [' id CODIGO_CI\n', '0 11001
4126450003\n', '1 11002 4126450005\n', '2 11003 4126450004\n', '3 11004
4126450083\n', '4 11005 4126450259\n', '5 11006 4126450092'] § Code ##
Seleccionar el id igual a 11003 ciudad_ci[ciudad_ci['id']==11003] § Output > [' id
CODIGO_CI\n', '2 11003 4126450004'] § Code ## Seleccionar los id igual o menor a
11004 ciudad_ci[ciudad_ci['id']<=11004] § Output > [' id CODIGO_CI\n', '0 11001
4126450003\n', '1 11002 4126450005\n', '2 11003 4126450004\n', '3 11004
4126450083'] § Code ## Seleccionar los ids que sean par mask_par =
ciudad_ci['id']%2==0 ciudad_ci[mask_par] § Output > [' id CODIGO_CI\n', '0 11001
4126450003\n', '2 11003 4126450004\n', '4 11005 4126450259\n', '5 11006
4126450092'] § Code ## Seleccionar los ids y codigos ci pares mask_par_medellin
= ciudad_ci['id']%2 ==0 ciudad_ci[mask_par_medellin][['id','CODIGO_CI']] § Output >
[' id CODIGO_CI\n', '0 11001 4126450003\n', '2 11003 4126450004\

```