Lambda Functions

- A *lambda* function is a one-line mini-function defined on the fly
  - They can be used anywhere a function is required
- For example, here's a simple function *double* that doubles an argument *x* by 2

```
def double(x):
    return x * 2

double(4)
```

- We can also define a *lambda* function *double_l* that does the same thing

```
double_l = lambda x: x * 2 #given parameter x, return x * 2
double_l(4)
```

Lambda Functions – With Sorting

- *Lambda* functions are especially useful as arguments to other functions
- For example, in the *sorted* function:
  sorted(*list*, *key*) – sorts the *list* based on the optional *key* (*lambda* function)
  - *key* optionally specifies a function that tells *sorted* what to sort by
  - The default value (None) compares the elements directly

For reference: https://docs.python.org/3.5/library/functions.html#sorted

Lambda Functions – With Sorting

- What if we want to sort our albums by artist (“Artist”)?

```
albums_sorted = sorted(albums, key = lambda x: x["Artist"])
print(albums_sorted)
```

  - When sorting *albums*, use *key* to compare them
  - *key* is a lambda function -- given parameter x (row), use the “Artist” column to compare

For reference: https://docs.python.org/3.5/library/functions.html#sorted

Lambda Functions

- Some other functions that accept *lambda* functions as arguments:

  max(*list*, *key*) – returns the maximum value from the *list* based on the optional *key* (*lambda* function)
  - *key* optionally specifies a function that tells *max* what to sort by
  - The default value (None) compares the elements directly

  min(*list*, *key*) – returns the minimum value from the *list* based on the optional *key* (*lambda* function)
  - *key* optionally specifies a function that tells *min* what to sort by
  - The default value (None) compares the elements directly

For reference: http://www.diveintopython.net/power_of_introspection/lambda_functions.html