

World bank data

- Data on world economies for over half a century
- Indicators
 - Population
 - Electricity consumption
 - CO2 emissions
 - Literacy rates
 - Unemployment
 - Mortality rates

Using zip()

```
avengers = ['hawkeye', 'iron man', 'thor', 'quicksilver']  
names = ['barton', 'stark', 'odinson', 'maximoff']  
z = zip(avengers, names)  
print(type(z))
```

```
<class 'zip'>
```

```
print(list(z))
```

```
[('hawkeye', 'barton'), ('iron man', 'stark'),  
( 'thor', 'odinson'), ('quicksilver', 'maximoff')]
```

Defining a function

- raise.py

```
def raise_both(value1, value2):  
    """Raise value1 to the power of value2  
    and vice versa."""  
    new_value1 = value1 ** value2  
    new_value2 = value2 ** value1  
    new_tuple = (new_value1, new_value2)  
    return new_tuple
```

Re-cap: list comprehensions

Basic

```
[output expression for iterator variable in iterable]
```

Advanced

```
[output expression +  
conditional on output for iterator variable in iterable +  
conditional on iterable]
```

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 2)

Generators for the large data limit

- Use a generator to load a file line by line
- Works on streaming data!
- Read and process the file until all lines are exhausted

Build a generator function

- sequence.py

```
def num_sequence(n):  
    """Generate values from 0 to n."""  
    i = 0  
    while i < n:  
        yield i  
        i += 1
```

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 2)

Reading files in chunks

- Up next:
 - `read_csv()` function and `chunk_size` argument
 - Look at specific indicators in specific countries
 - Write a function to generalize tasks

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 2)

You've applied your skills in:

- User-defined functions
- Iterators
- List comprehensions
- Generators

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 2)