

```
1  # Joining CSVs
2
3  import csv
4
5  artists = {}
6  with open("artist.csv", "r") as file:
7      reader = csv.DictReader(file)
8      for row in reader:
9          artists[row["ArtistId"]] = row["Name"]
10
11 with open("album.csv", "r") as file:
12     reader = csv.DictReader(file)
13     for row in reader:
14         print(row["Title"], "by", artists[row["ArtistId"]])
```

```
1  # Joining CSVs
2
3  import csv
4
5  artists = {}
6  with open("artist.csv", "r") as file:
7      reader = csv.DictReader(file)
8      for row in reader:
9          artists[row["ArtistId"]] = row["Name"]
10
11 albums = []
12 with open("album.csv", "r") as file:
13     reader = csv.DictReader(file)
14     for row in reader:
15         albums.append(row)
16
17 for album in sorted(albums, key=lambda a: a["Title"]):
18     print(album["Title"], "by", artists[album["ArtistId"]])
```

```
1  # Opportunity for better design
2
3  print("cough")
4  print("cough")
5  print("cough")
```

```
1  # Better design
2
3  for i in range(3):
4      print("cough")
```

```
1  # Abstraction
2
3
4  def main():
5      for i in range(3):
6          cough()
7
8
9  def cough():
10     """Cough once"""
11     print("cough")
12
13
14  if __name__ == "__main__":
15     main()
```

```
1  # Abstraction with parameterization
2
3
4  def main():
5      cough(3)
6
7
8  def cough(n):
9      """Cough some number of times"""
10     for i in range(n):
11         print("cough")
12
13
14 if __name__ == "__main__":
15     main()
```

```
1  # Reading CSV
2
3  import csv
4
5  with open("employee.csv", "r") as file:
6      reader = csv.reader(file)
7      next(reader)
8      for row in reader:
9          print(row[2], row[1], "is a", row[3])
```

```
1  # Reading CSV with DictReader
2
3  import csv
4
5  with open("employee.csv", "r") as file:
6      reader = csv.DictReader(file)
7      for row in reader:
8          print(row["FirstName"], row["LastName"], "is a", row["Title"])
```



```
1  # Parsing dates
2  # https://docs.python.org/3/library/datetime.html
3
4  import csv
5  import math
6
7  from datetime import datetime
8
9  today = datetime.today()
10 with open("employee.csv", "r") as file:
11     reader = csv.DictReader(file)
12     for row in reader:
13         age = math.floor((today - datetime.strptime(row["BirthDate"], "%Y-%m-%d %H:%M:%S")).days / 365)
14         print(row["FirstName"], row["LastName"], "is", age, "years old")
```

```
1  # Summation
2
3  import csv
4
5  dates = {}
6  with open("invoice.csv", "r") as file:
7      reader = csv.DictReader(file)
8      for row in reader:
9          date = row["InvoiceDate"].replace(" 00:00:00", "")
10         if date in dates:
11             dates[date] += float(row["Total"])
12         else:
13             dates[date] = float(row["Total"])
14
15  for date, total in dates.items():
16      print("$" + str(total) + " on " + date)
```

```
1  # Abstraction and scope
2
3
4  def main():
5      i = get_positive_int("Positive integer: ")
6      print(i)
7
8
9  def get_positive_int(prompt):
10     """Prompt user for positive integer"""
11     while True:
12         n = int(input(prompt))
13         if n > 0:
14             break
15     return n
16
17
18 if __name__ == "__main__":
19     main()
```

```
1 # API
2 # https://iextrading.com/developer/docs/#price
3
4 import requests
5
6 symbol = input("Symbol: ")
7 url = "https://api.iextrading.com/1.0/stock/" + symbol + "/price"
8 response = requests.get(url)
9 print("$" + response.text)
```

```
1  # Generates a bar chart of three scores
2
3  # Get scores from user
4  score1 = int(input("Score 1: "))
5  score2 = int(input("Score 2: "))
6  score3 = int(input("Score 3: "))
7
8  # Generate first bar
9  print("Score 1: ", end="");
10 for i in range(score1):
11     print("#", end="")
12 print()
13
14 # Generate second bar
15 print("Score 2: ", end="");
16 for i in range(score2):
17     print("#", end="")
18 print()
19
20 # Generate third bar
21 print("Score 3: ", end="");
22 for i in range(score3):
23     print("#", end="")
24 print()
```

```
1  # Generates a bar chart of three scores using a list
2
3  def main():
4
5      # Get scores from user
6      scores = []
7      for i in range(3):
8          score = int(input(f"Score {i + 1}: "))
9          scores.append(score)
10
11     # Chart scores
12     for i in range(len(scores)):
13         print(f"Score {i + 1}: ", end="")
14         chart(scores[i])
15
16
17 def chart(score):
18     """Generate bar"""
19
20     # Output one hash per point
21     for i in range(score):
22         print("#", end="")
23     print()
24
25
26 if __name__ == "__main__":
27     main()
```

```
1  # Return value
2
3
4  def main():
5      x = int(input("x: "))
6      print(square(x))
7
8
9  def square(n):
10     """Return square of n"""
11     return n**2
12
13
14  if __name__ == "__main__":
15     main()
```