LISTS EXERCISE

TASK 1: ADD ALL DICTIONARY VALUES

WORK WITH

```
toys = {"robot": "$40.0", "car": "$25", "ironman": "$12"}
```

DESIRED OUTPUT

```
77
```

HINTS?

1. Use the **values()** method.
2. Use **list**
3. Use the built-in Python function, **eval()** to convert a string to a number.
4. Use list slicing with the appropriate index to grab the numbers. Add them together.
5. Use the **int** type.

In [56]:

```
toys = {"robot": "$40.0", "car": "$25", "ironman": "$12"}
```

TASK 2: USE COMPARISON OPERATORS IN A LIST

WORK WITH

```
questions = [10  4, 50  50, 90  10, "c"  ("a", "b", "c"), 100  100]

!=
==
in
```

DESIRED OUTPUT

[True, True, False, True, False]

HINTS?

1. Use only the three of the operators shown above.

In [151]:

```
questions = [10  4, 50 50, 90 10, "c"  ("a", "b", "c"), 100  100]
```

```
questions
```

Out[151]:

```
[True, True, False, True, False]
```

TASK 3: LEN KEY VALUES WITH COMPARISON OPERATORS

WORK WITH

```
films = {"k1": "blade runner 2049",  "k2": "matrix",  "k3": "terminat
or"}


<
>
len()
```

DESIRED OUTPUT

True

HINTS?

1. Use only the two operators shown above.
2. Use the built-in function, **len**().
3. Use the operators in the same order as the dictionary keys.

In [95]:

```
films = {"k1": "blade runner 2049",  "k2": "matrix",  "k3": "ninja scroll"}
```

TASK 4: UPDATE DICTIONARY

WORK WITH

```
life_stages = {0: "embryo", 1: "fetus", 2:"baby", 3:"infant",4: "teen
"}
```

DESIRED OUTPUT

{0: 'embryo', 1: 'fetus', 2: 'baby', 3: 'infant', 4: 'teen', 5: 'adult', 6: 'big kid!'}

HINTS?

1. Create a new dictionary called midlife, with keys 5 and 6, and key values "adult" and "big kid!".
2. Add midlife dictionary to life_stages using one of the dictionary methods.

In [94]:

```
life_stages = {0: "embryo", 1: "fetus", 2:"baby", 3:"infant",4: "teen"}
```

TASK 5: ADD ALL VALUES FROM LIST

WORK WITH

```
nest1 = [(1,2,3), {"k1": [8, 1, 300, 2, 77], "k2": [10,20,30]}, ["a",
"500", "c"]]

sorted()
eval()
```

DESIRED OUTPUT

```
833.0
```

HINTS?

1. Add 3, 300, 30 and 500 together from each of the nested tuples, dictionaries or lists.
2. Use only the two built-in python functions shown above.
3. Use the Python built-in function, **sorted** for **"k1"**.
4. For the nested dictionary,**"k1"**, index only by -1, not 2.
5. Use the **float**.

In [27]:

```
nest1 = [(1,2,3), {"k1": [30, 1, 300, 2, 77], "k2": [10,20,30]}, ["a", "500"
, "c"]]
nest1
```

Out[27]:

```
[(1, 2, 3), {'k1': [30, 1, 300, 2, 77], 'k2': [10, 20, 30]}, ['a', '500', '
c']]
```

TASK 6: ADD ALL LIST VALUES INTO A STRING

WORK WITH

```
prices = ["a", "b", "9", "c", "d", "FOUR", "e", "f", "2.5"]
sentence = """The bill for the  {}#!/,?? {}#!/ ??and drink came to {
}??"""
```

DESIRED OUTPUT

```
prices = ["9", "FOUR", "2.5"]


'The bill for the pizza, chips and drink came to $15.5'
```

HINTS?

1. Use the **format** function

1. Use the **format** function.
2. Use slicing and stride
3. Also use **eval**, **len** and **str**.
4. Add all the string numbers from the prices list.
5. Use the replace function twice.

In [181]:

```
prices = ["a", "b", "9", "c", "d", "FOUR", "e", "f", "2.5"]
cash = prices[2::3]
```

In [89]:

```
sentence = """The bill for the  {}#!/,?? {}#!/ ??and drink came to {}??"""
```