Exercise 6 solution

Write a program called “exercise_6.1.py”.

1. Add a shebang line and a document string “Exercise #6.1 in Python3”.

```
#! /usr/bin/env python3

"""  Exercise #6.1 in Python3  """
```

2. Create a function, my_function() that accepts the following:
   - A “required” variable
   - Any number of additional variables
   - Any number of key/value sets.

```
def my_function(required, *args, **kwargs):
```

3. Add a function help statement.

```
    """ My Function """
```

4. Print the required variable type and value received.

```
    print(type(required), f"{required}\n")
```

5. Print any additional variable types and values received.

```
    if (args):
        print(type(args), args, "\n")
```

6. Print the key/values type and data received.

```
    if (kwargs):
        print(type(kwargs), kwargs, "\n")
```

7. Return a list including all the data received.

```
    return([required, args, kwargs])
```

8. Assign the returned data to a list, list_.

```
list_ = my_function("1",2,3,4,a=1,b=2,c=3,d=4)
```

9. Loop through the list, list_ and print out each element.

```
for x in list_:
    print(f"{x}\n")
```

10. List the full program.

```
$ cat ./exercise_6.1.py
#! /usr/bin/env python3

"""  Exercise #6.1 in Python3  """

def my_function(required, *args, **kwargs):
    """ My Function """

    print(type(required), f"{required}\n")

    if (args):
        print(type(args), args, "\n")

    if (kwargs):
        print(type(kwargs), kwargs, "\n")

    return([required, args, kwargs])

    # End function my_function()

list_ = my_function(1,2,3,4,a=1,b=2,c=3,d=4)

for x in list_:
    print(f"{x}\n")
```

11. Make the program executable.

```
$ chmod +x exercise_6.1.py
```

12. Run the program.

```
$ ./exercise_6.1.py
<class 'str'> 1

<class 'tuple'> (2, 3, 4)

<class 'dict'> {'a': 1, 'b': 2, 'c': 3, 'd': 4}

1

(2, 3, 4)

{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```