1. Difference between r, r+, w, w+, a and a+

1.1 Read below table for the difference between open modes `r`, `r+`, `w`, `w+`, `a` and `a+` in `open()` function.

	r	r+	w	w+	a	a+
read	*	*		*		*
write		*	*	*	*	*
create			*	*	*	*
truncate			*	*		
position at start	*	*	*	*		
position at end					*	*

PS In this context, truncate means delete the content of the file.

1.2 Definition of open modes `r`, `r+`, `w`, `w+`, `a`, `a+`:

- The `r` throws an error if the file does not exist or opens an existing file `without truncating` it for `reading`; the file pointer position at the beginning of the file.
- The `r+` throws an error if the file does not exist or opens an existing file `without truncating` it for `reading and writing`; the file pointer position at the beginning of the file.
- The `w` creates a new file or `truncates an existing file`, then opens it for `writing`; the file pointer position at the beginning of the file.
- The `w+` creates a new file or opens an existing file for `reading and writing`; the file pointer position at the `end of the file`.
- The `a` creates a new file or opens an existing file for `writing`; the file pointer position at the `end of the file`.
- The `a+` creates a new file or opens an existing file for `reading and writing`, and the file pointer position at the `end of the file`.

2. What is + means in open()?

The `+` adds either reading or writing to an existing open mode, aka update mode.

- The `r` means `reading` file; `r+` means `reading and writing` the file.
- The `w` means `writing` file; `w+` means `reading and writing` the file.
- The `a` means `writing` file, append mode; `a+` means `reading and writing` file, append mode.

3. Difference between r and r+ in open()

The `r+` adds `writing file` to the existing `r` mode.

A text file for testing.

file.txt

```
welcome to python 1
welcome to python 2
welcome to python 3
welcome to python 4
```

3.1 Read a file with r

```
with open('file.txt') as f:  # default `r` mode
    print(f.read())
```

Output

Terminal

```
welcome to python 1
welcome to python 2
welcome to python 3
welcome to python 4
```

In `r` mode, if we write the file, Python throws `io.UnsupportedOperation: not writable`

```
with open('file.txt', 'r') as f:  # default `r` mode
    f.write("test \n")            # throws UnsupportedOperation
```

3.2 Read and write a file with r+

In `r+` mode, we can read and write the file, but the file pointer position is at the beginning of the file; if we write the file offset, it will overwrite the beginning content.

See the below example:

```
with open('file.txt', 'r+') as f:
    f.write("new line \n")
```

Output

Terminal

```
new line
python 1
welcome to python 2
welcome to python 3
welcome to python 4
```

The below example uses `f.read()` to move the file pointer to the end of the file, and append a new line.

```
# alternative, open with `a` mode
with open('file.txt', 'r+') as f:
    f.read()          # move file position to the end of the file.
    f.write("new line \n")
```

Output

Terminal

```
welcome to python 1
welcome to python 2
welcome to python 3
welcome to python 4
new line
```

Please comment below if you know other `r+` examples, thanks.

4. Difference between w and w+ in open()

The `w+` adds `reading file` to the existing `w` mode.

4.1 Write a file with w

```
with open('file.txt', 'w') as f:  # create a new file or truncates it
    f.write("test 1\n")
    f.write("test 2\n")
    f.write("test 3\n")
```

Output

file.txt

```
test 1
test 2
test 3
```

4.2 Read and write a file with w+

```
with open('file.txt', 'w+') as f:  # create a new file or truncates it
    f.write("test 1\n")
    f.write("test 2\n")
    f.write("test 3\n")            # now the file pointer is at the end
    f.seek(0)                      # move the file pointer to the beginning
    lines = f.read()               # read it, now we can read
    print(lines)                   # print it
```

Output

Terminal

```
test 1
test 2
test 3
```

5. Difference between a and a+ in open()

The `a+` adds `reading file` to the existing `a` mode.

A text file for testing.

file.txt

```
welcome to python 1
welcome to python 2
welcome to python 3
welcome to python 4
```

5.1 Append a file with a

```
with open('file.txt', 'a') as f:
    f.write("4")
```

Output

file.txt

```
welcome to python 1
welcome to python 2
welcome to python 3
welcome to python 4
4
```

5.2 Append a file with a+

1. Open a file for reading and writing with `a+`.
2. Counts the number of the lines.
3. Append the result to the file.

```
with open('file.txt', 'a+') as f:
    f.seek(0)                  # file pointer at end, move to beginning
    lines = f.readlines()      # read all and file pointer at end again
    f.write("\n" + str(len(lines))) # append number of lines to a file
```

Output

file.txt

```
welcome to python 1
welcome to python 2
welcome to python 3
welcome to python 4
4
```

6. Difference between r+ and w+ in open()

Below is the difference between `r+` and `w+`:

1. If the file does not exist, `r+` throws `FileNotFoundError`; the `w+` creates the file.
2. If the file exists, `r+` opens it without truncating; the `w+` truncates the file and opens it.

7. Difference between r+ and a+ in open()

Below is the difference between `r+` and `a+`:

1. If the file does not exist, `r+` throws `FileNotFoundError`; the `a+` creates the file.
2. For `r+` mode, the initial file pointer position at the `beginning of the file`; For `a+` mode, the initial file pointer position at the `end of the file`.

8. Difference between w+ and a+ in open()

Below is the difference between `w+` and `a+`:

1. If the file exists, `w+` truncates the file and opens it; `a+` opens it without truncating.
2. For `w+` mode, the initial file pointer position at the `beginning of the file`; For `a+` mode, the initial file pointer position at the `end of the file`.