Round Robin: This method is quite same as the FCFS but the difference is the in this case the processor will not process the whole job (process) at a time. Instead, it will complete an amount of job (quantum) at a turn and then will go to the next process and so on. When all job has got a turn, it will again start from the first job and work for a quantum of time/cycle on each job and proceed. Now consider a CPU and also consider a list in which the processes are listed as follows,

Arrival	Process	Burst Time
0	1	3
1	2	2
2	3	1

Quantum = 2 Second

Here, Arrival is the time when the process has arrived the list, Process Number is used instead of the process name, and Burst Time is the amount of time required by the process from CPU. Well, as the unit of time you can take anything like nano-second, second, minute etc whatever. We consider it as second.

Now for an instance, consider the above list as the ready queue for the CPU, that is the CPU will take processes from this list and will process it.

Here in Round Robin, what will happen is as follows,

AT 0s:

There is only 1 job, that is Process-1 with burst time 3. So CPU will do 1 quantum amount (that is 2 second of job) job of Process-1. Thus Process-1 has 1s more job to be done. And now the time is 2s as CPU worked on process-1 at 0s and 1s (for 2 second, right?)

AT 2s:

Now there are 3 jobs,

Process-1 that arrived at 0s and has 1s job to be done.
Process-2 that arrived at 1s and has 2s job to be done.
Process-3 that arrived at 2s and has 1s job to be done.

At the previous turn, CPU worked on process-2, so this time it will work on process-2. It will for a quantum time (that is, 2 second). Thus process-2 has 0s more job to be done and now time is 4s as CPU worked on process-2 at 2s and 3s.

AT 4s:

Now there are 3 jobs,

Process-1 that arrived at 0s and has 1s job to be done.
Process-3 that arrived at 2s and has 1s job to be done.

Now it is the turn of process-3. Process-3 has less job than a quantum. So CPU will work for the amount of job that process-3 has (CPU can work for less or equal amount of time than quantum). So process-3 has 0s job more to be done and now the time is 5s as CPU worked on process-3 only at 4s.

AT 5s:

There are no more jobs after process-3. So now process-3 will start from the beginning. So this is the turn of process-1. Process-1 has 1s job to be done. CPU will complete it as it is less than the quantum.

All Job Done.

We can show the above thing as the following time-line

Process-1	Process-1	Process-2	Process-2	Process-3	Process-1
0s	1s	2s	3s	4s	5s

A shortened view of the above time-line is as follows,

Process-1	Process-2	Process-3	Process-1	
0	2	4	5	6

So now came the main thing, Waiting Time. Ok, Look carefully,

See the following time-line for Process-1. Here, BROWN marked seconds symbolizes the seconds while Process-1 was waiting and GREEN marked second symbolizes the seconds while Process-1 was working.

🟩 Process-1	🟩 Process-1	🟫 Process-2	🟫 Process-2	🟫 Process-3	🟩 Process-1
🟩 0s	🟩 1s	🟫 2s	🟫 3s	🟫 4s	🟩 5s

Thus Process-1 waited for 3 seconds.

See the following time-line for Process-2. Here, BROWN marked seconds symbolizes the seconds while Process-2 was waiting and GREEN marked second symbolizes the seconds while Process-2 was working.

Process-1	🟫 Process-1	🟩 Process-2	🟩 Process-2	Process-3	Process-1
0s	🟫 1s	🟩 2s	🟩 3s	4s	5s

Thus Process-2 waited for 1 seconds.

See the following time-line for Process-3. Here, BROWN marked seconds symbolizes the seconds while Process-3 was waiting and GREEN marked second symbolizes the seconds while Process-3 was working.

Process-1	Process-1	Process-2	Process-2	Process-3	Process-1
0s	1s	2s	3s	4s	5s

Thus Process-3 waited for 2 seconds.

So total waiting time = (Waiting Time of Process-1)+ (Waiting Time of Process-2)
+ (Waiting Time of Process-3)
= (3+1+2)s
= 6s

So the average waiting time is = (Total waiting time / Number of Processes)s
= (6/3)s
= 2s