

forking.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2: #
3: #-----
4: # forking.py
5: # Author: Bob Dondero
6: #-----
7:
8: import multiprocessing
9:
10: def print_color(color):
11:     for _ in range(5):
12:         print(color)
13:     print(color + ' process terminated')
14:
15: def main():
16:
17:     blue_process = multiprocessing.Process(
18:         target=print_color, args=['blue'])
19:     red_process = multiprocessing.Process(
20:         target=print_color, args=['red'])
21:
22:     blue_process.start()
23:     red_process.start()
24:
25:     print('parent process terminated')
26:
27: if __name__ == '__main__':
28:     main()

```

waiting.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2: #
3: #-----
4: # waiting.py
5: # Author: Bob Dondero
6: #-----
7:
8: import multiprocessing
9:
10: def print_color(color):
11:     for _ in range(5):
12:         print(color)
13:     print(color + ' process terminated')
14:
15: def main():
16:
17:     blue_process = multiprocessing.Process(
18:         target=print_color, args=['blue'])
19:     red_process = multiprocessing.Process(
20:         target=print_color, args=['red'])
21:
22:     blue_process.start()
23:     red_process.start()
24:
25:     blue_process.join()
26:     red_process.join()
27:
28:     print('parent process terminated')
29:
30: if __name__ == '__main__':
31:     main()

```

spawning.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # spawning.py
5: # Author: Bob Dondero
6: #-----
7:
8: import threading
9:
10: #-----
11:
12: class PrinterThread (threading.Thread):
13:
14:     def __init__(self, color):
15:         threading.Thread.__init__(self)
16:         self._color = color
17:
18:     def run(self):
19:         for _ in range(5):
20:             print(self._color)
21:             print(self._color + ' thread terminated')
22:
23: #-----
24:
25: def main():
26:
27:     blue_thread = PrinterThread('blue')
28:     red_thread = PrinterThread('red')
29:
30:     blue_thread.start()
31:     red_thread.start()
32:
33:     print('main thread terminated')
34:
35: #-----
36:
37: if __name__ == '__main__':
38:     main()

```

joining.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # joining.py
5: # Author: Bob Dondero
6: #-----
7:
8: import threading
9:
10: #-----
11:
12: class PrinterThread (threading.Thread):
13:
14:     def __init__(self, color):
15:         threading.Thread.__init__(self)
16:         self._color = color
17:
18:     def run(self):
19:         for _ in range(5):
20:             print(self._color)
21:             print(self._color + ' thread terminated')
22:
23: #-----
24:
25: def main():
26:
27:     blue_thread = PrinterThread('blue')
28:     red_thread = PrinterThread('red')
29:
30:     blue_thread.start()
31:     red_thread.start()
32:
33:     blue_thread.join()
34:     red_thread.join()
35:
36:     print('main thread terminated')
37:
38: #-----
39:
40: if __name__ == '__main__':
41:     main()

```