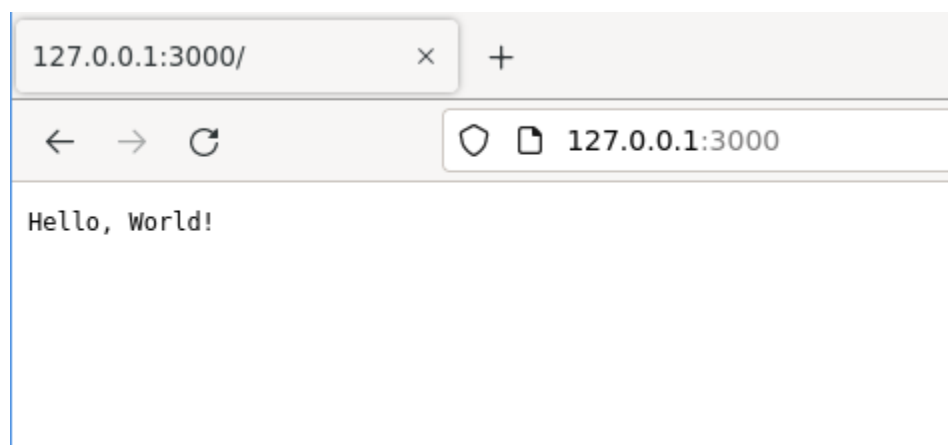
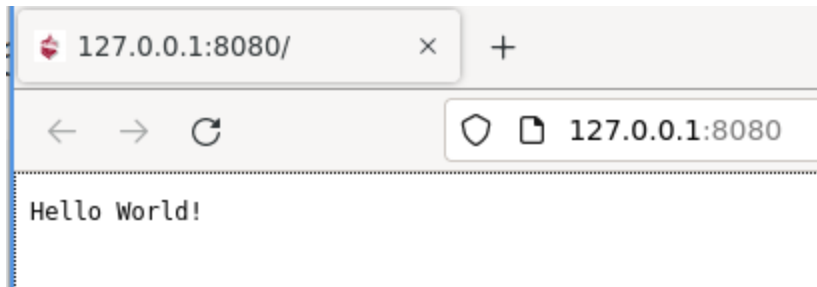


```
user@box:~$ node -v
v17.7.1
user@box:~$ npm -v
8.5.2
user@box:~$ node -h
Usage: node [options] [ script.js ] [arguments]
       node inspect [options] [ script.js | host:port ] [arguments]

Options:
  -                script read from stdin (default if no
                    file name is provided, interactive mode
                    if a tty)
  --               indicate the end of node options
  --abort-on-uncaught-exception
                    aborting instead of exiting causes a
                    core file to be generated for analysis
  -c, --check      syntax check script without executing
  --completion-bash
                    print source-able bash completion
                    script
  -C, --conditions=...
                    additional user conditions for
                    conditional exports and imports
  --cpu-prof       Start the V8 CPU profiler on start up,
                    and write the CPU profile to disk
```

```
user@box:~$ cd ~/iot/lesson6
user@box:~/iot/lesson6$ node hello-world.js
Server running at http://127.0.0.1:3000/
^C
user@box:~/iot/lesson6$ node hello.js
Server running at http://127.0.0.1:8080/
response end call done
request end event fired
```





```
user@box:~/iot/lesson6$ node http.js
0
```

```
user@box:~/iot/lesson6$ cat say_hello.mustache
Hello, {{to}}!
user@box:~/iot/lesson6$ cat say_hello.py
# https://github.com/defunkt/pystache
import pystache
print(pystache.render('Hi {{person}}!', {'person': 'Alexa'}))

# Create dedicated view classes to hold view logic
class SayHello(object):
    def to(self):
        return "World"
hello = SayHello()

# Use template in say_hello.mustache
renderer = pystache.Renderer()
print(renderer.render(hello))

# Pre-parse a template
parsed = pystache.parse('Hey {{#who}}{{.}}!{{/who}}')
print(parsed)
print(renderer.render(parsed, {'who': 'Google'}))
print(renderer.render(parsed, {'who': 'Siri'}))
```

```
user@box:~/iot/lesson6$ python3 say_hello.py
Hi Alexa!
Hello, World!

['Hey ', _SectionNode(key='who', index_begin=12, index_end=18, parsed=[_EscapeNode(key='.'), '!'])]
Hey Google!
Hey Siri!
```