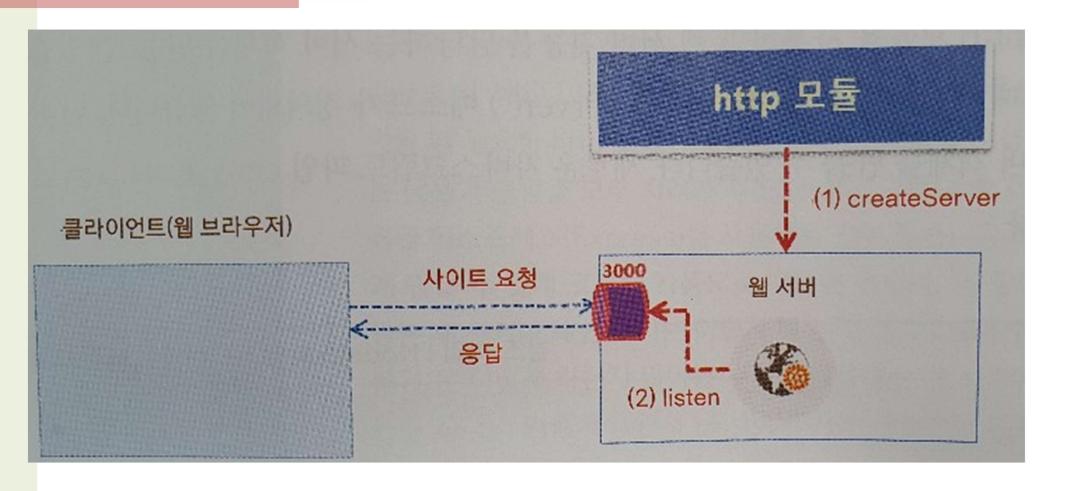
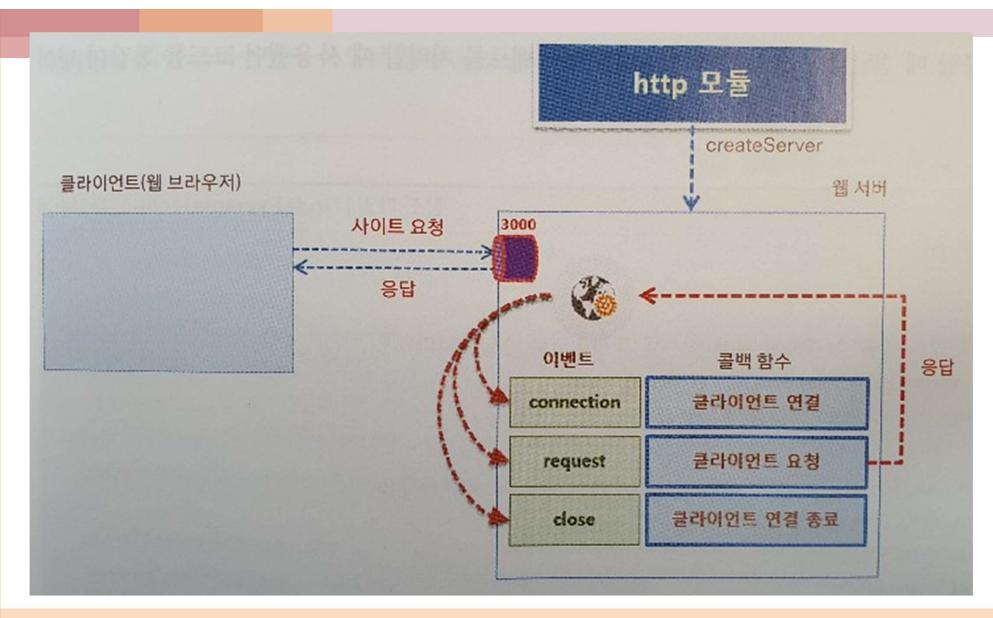
웹서버 만들기 기초

웹서버 만들기 기초

- server = http.createServer()
- server.listen(), server.close()
- ▶ client가 웹 서버에 요청할 때 발생하는 이벤트 처리
 - ▶ client가 데이터 요청 → event 발생 → callback 함수 등록하여 처리
 - event = connection, request, close
- ▶ res 객체: writeHead, write(), end()





```
const http = require('http');
const server = http.createServer();
const port = 3000;
server.listen(port, function() {
   console.log('web server starts... at port %d', port);
});
server.on('connection', function(socket) {
   let addr = socket.address();
   console.log('client connets: %s, %d', addr.address, addr.port);
});
server.on('request', function(req, res) {
   console.log('client requests...');
});
server.on('close', function() {
   console.log('server terminates...');
   server.close();
});
```

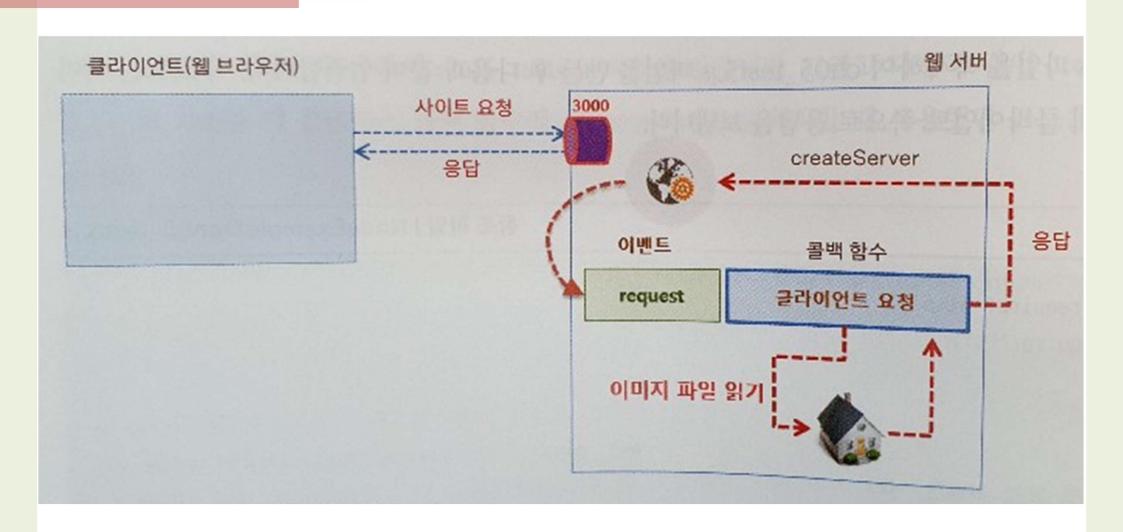
Client로 응답 보내기

Client로 응답 보내기

```
let http = require('http');
const server = http.createServer(function(err, res) {
  res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});
  res.write('<!DOCTYPE html>');
  res.write('<html>');
  res.write(' <head>');
  res.write(' <title>응답 페이지</title>');
  res.write(' </head>');
  res.write(' <body>');
  res.write(' <h1>Node.js server respond</h1>');
  res.write(' </body>');
  res.write('</html>');
  res.end();
}).listen(3000);
```

파일 읽어 응답하기

- image Content-Type : img/png
- html Content-Type : text/html
- text/plain, text/css, text/xml
- image/jpeg, video/mpeg, audio/mp3, application/zip



```
const fs = require("fs");
server.on('request', function(req, res) {
   const filename = 'index.html';
   fs.readFile(filename, function(err, data) {
      res.writeHead(200, {
         'Content-Type': 'text/html'
      });
      res.write(data);
      res.end();
```