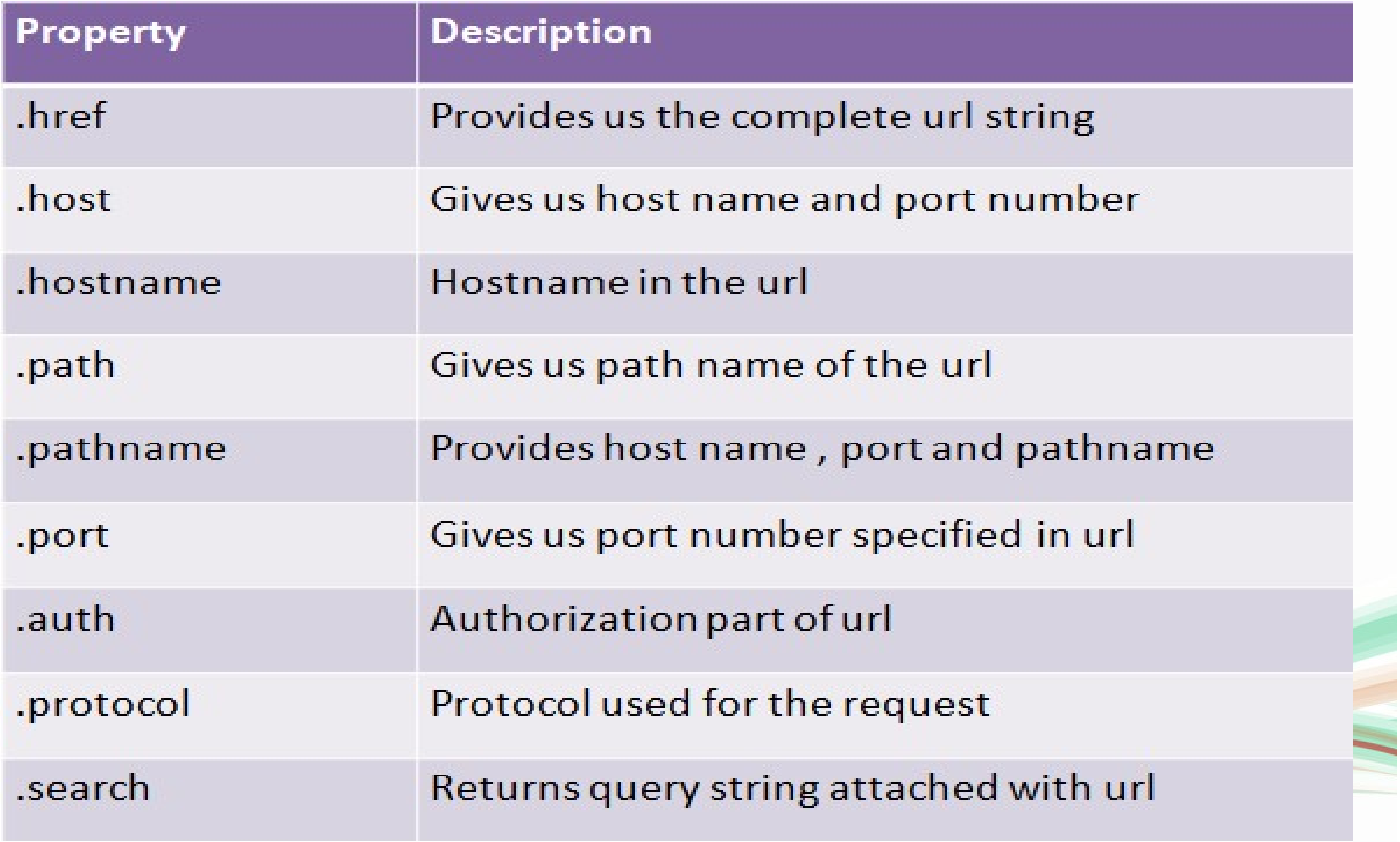
Unit – 3 Node JS

URL Module

- The URL module breaks a web address into readable parts.
- To use the URL module, you have to use require() method:

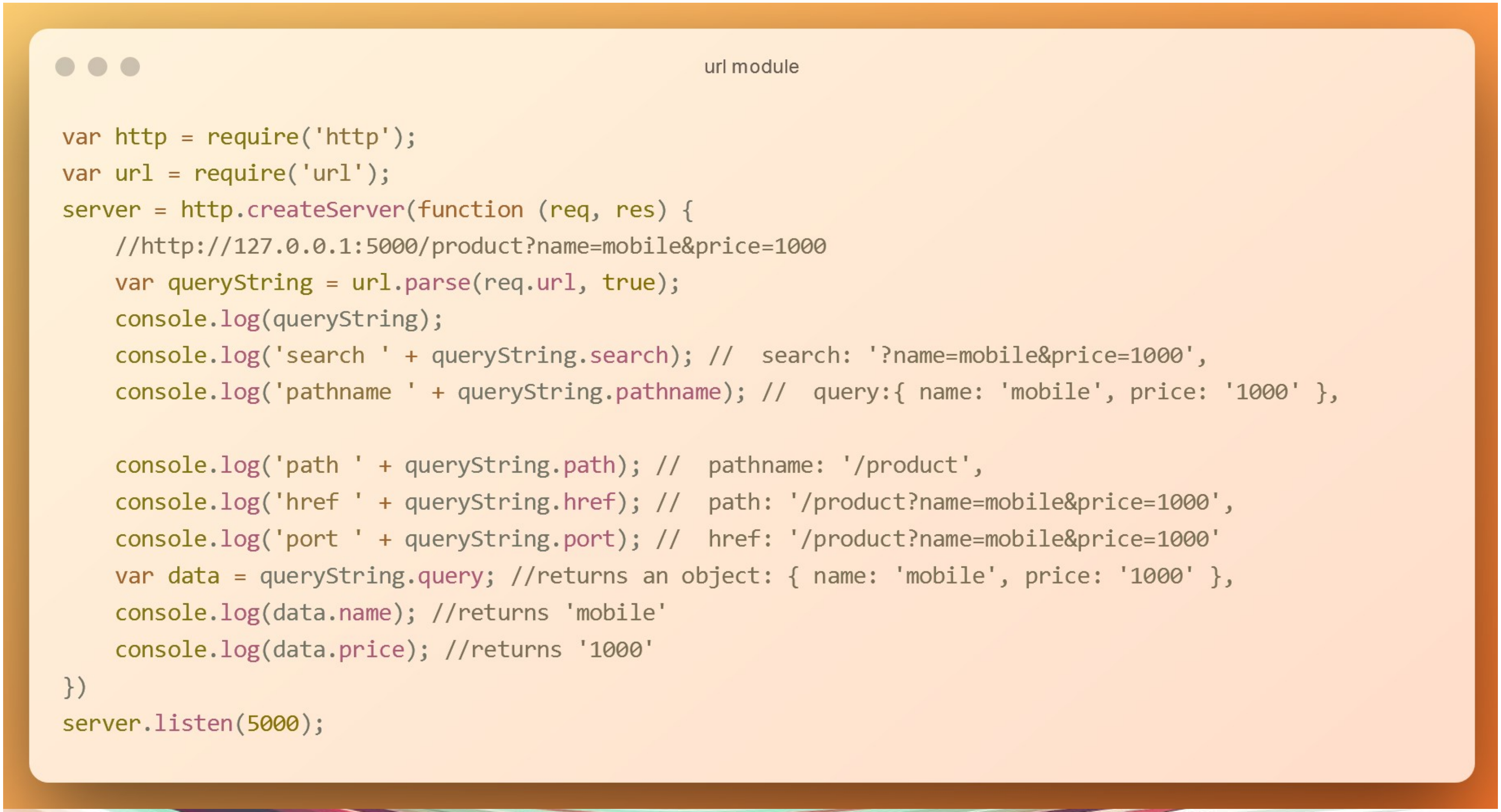
  var url = require('url');
- You can parse an address with the url.parse() method, and it will return a URL object with some properties

Important properties



Property	Description
.href	Provides us the complete url string
.host	Gives us host name and port number
.hostname	Hostname in the url
.path	Gives us path name of the url
.pathname	Provides host name , port and pathname
.port	Gives us port number specified in url
.auth	Authorization part of url
.protocol	Protocol used for the request
.search	Returns query string attached with url

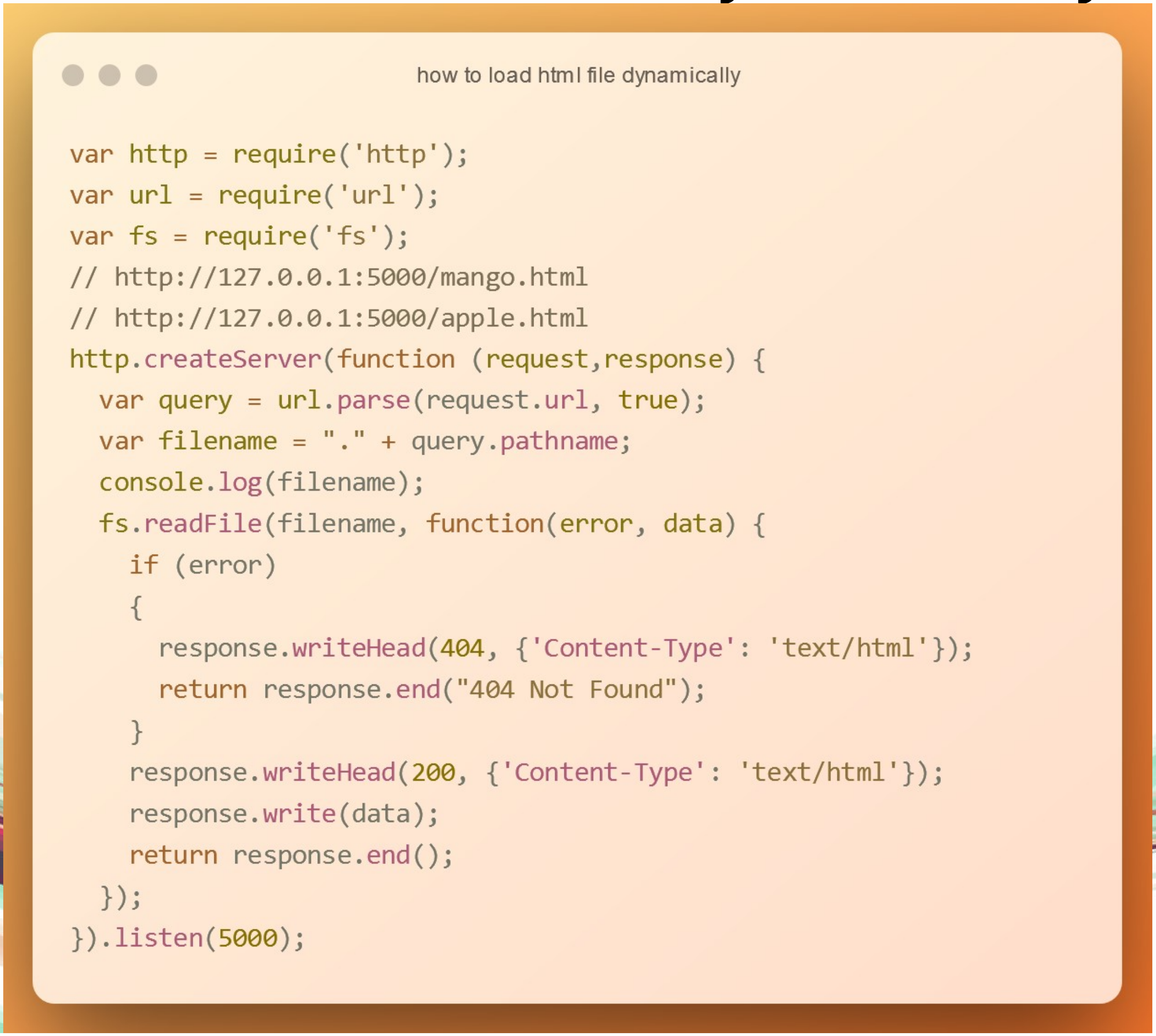
Example of url module



```
var http = require('http');
var url = require('url');
server = http.createServer(function (req, res) {
    //http://127.0.0.1:5000/product?name=mobile&price=1000
    var queryString = url.parse(req.url, true);
    console.log(queryString);
    console.log('search ' + queryString.search); //  search: '?name=mobile&price=1000',
    console.log('pathname ' + queryString.pathname); //  query:{ name: 'mobile', price: '1000' },

    console.log('path ' + queryString.path); //  pathname: '/product',
    console.log('href ' + queryString.href); //  path: '/product?name=mobile&price=1000',
    console.log('port ' + queryString.port); //  href: '/product?name=mobile&price=1000'
    var data = queryString.query; //returns an object: { name: 'mobile', price: '1000' },
    console.log(data.name); //returns 'mobile'
    console.log(data.price); //returns '1000'
})
server.listen(5000);
```

How to load html file dynamically



```
var http = require('http');
var url = require('url');
var fs = require('fs');
// http://127.0.0.1:5000/mango.html
// http://127.0.0.1:5000/apple.html
http.createServer(function (request,response) {
  var query = url.parse(request.url, true);
  var filename = "." + query.pathname;
  console.log(filename);
  fs.readFile(filename, function(error, data) {
    if (error)
    {
      response.writeHead(404, {'Content-Type': 'text/html'});
      return response.end("404 Not Found");
    }
    response.writeHead(200, {'Content-Type': 'text/html'});
    response.write(data);
    return response.end();
  });
}).listen(5000);
```