example.js

```js
import "./style.css";
import "./style2.css";
import { main } from "./style.module.css";
import("./lazy-style.css");

document.getElementsByTagName("main")[0].className = main;
```

style.css

```css
@import "style-imported.css";
@import "https://fonts.googleapis.com/css?family=Open+Sans";

body {
    background: green;
    font-family: "Open Sans";
}
```

dist/output.js

```js
/******/ (() => { // webpackBootstrap
/******/    "use strict";
/******/    var __webpack_modules__ = ({

/***/ 3:
/*!*************************!*\
  !*** ./images/file.png ***!
  \*************************/
/*! default exports */
/*! exports [not provided] [no usage info] */
/*! runtime requirements: module, __webpack_require__.p, __webpack_require__.* */
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

module.exports = __webpack_require__.p + "89a353e9c515885abd8e.png";

/***/ })

/******/    });
/* webpack runtime code */

/************************************************************************/
/******/    // The module cache
/******/    var __webpack_module_cache__ = {};
/******/
```

```
/******/     // The require function
/******/     function __webpack_require__(moduleId) {
/******/         // Check if module is in cache
/******/         var cachedModule = __webpack_module_cache__[moduleId];
/******/         if (cachedModule !== undefined) {
/******/             return cachedModule.exports;
/******/         }
/******/         // Create a new module (and put it into the cache)
/******/         var module = __webpack_module_cache__[moduleId] = {
/******/             // no module.id needed
/******/             // no module.loaded needed
/******/             exports: {}
/******/         };
/******/
/******/         // Execute the module function
/******/         __webpack_modules__[moduleId](module, module.exports, __webpack_require__);
/******/
/******/         // Return the exports of the module
/******/         return module.exports;
/******/     }
/******/
/******/     // expose the modules object (__webpack_modules__)
/******/     __webpack_require__.m = __webpack_modules__;
/******/
/************************************************************************/
/******/     /* webpack/runtime/ensure chunk */
/******/     (() => {
/******/         __webpack_require__.f = {};
/******/         // This file contains only the entry chunk.
/******/         // The chunk loading function for additional chunks
/******/         __webpack_require__.e = (chunkId) => {
/******/             return Promise.all(Object.keys(__webpack_require__.f).reduce((promises,
/******/                 __webpack_require__.f[key](chunkId, promises);
/******/                 return promises;
/******/             }, []));
/******/         };
/******/     })();
/******/
/******/     /* webpack/runtime/get css chunk filename */
/******/     (() => {
/******/         // This function allow to reference async chunks
/******/         __webpack_require__.k = (chunkId) => {
/******/             // return url for filenames based on template
/******/             return "" + chunkId + ".output.css";
/******/         };
/******/     })();
```

2

```
/******/
/******/        /* webpack/runtime/get javascript chunk filename */
/******/        (() => {
/******/                // This function allow to reference async chunks
/******/                __webpack_require__.u = (chunkId) => {
/******/                        // return url for filenames based on template
/******/                        return "" + chunkId + ".output.js";
/******/                };
/******/        })();
/******/
/******/        /* webpack/runtime/hasOwnProperty shorthand */
/******/        (() => {
/******/                __webpack_require__.o = (obj, prop) => (Object.prototype.hasOwnProperty.cal
/******/        })();
/******/
/******/        /* webpack/runtime/load script */
/******/        (() => {
/******/                var inProgress = {};
/******/                var dataWebpackPrefix = "app:";
/******/                // loadScript function to load a script via script tag
/******/                __webpack_require__.l = (url, done, key, chunkId) => {
/******/                        if(inProgress[url]) { inProgress[url].push(done); return; }
/******/                        var script, needAttach;
/******/                        if(key !== undefined) {
/******/                                var scripts = document.getElementsByTagName("script");
/******/                                for(var i = 0; i < scripts.length; i++) {
/******/                                        var s = scripts[i];
/******/                                        if(s.getAttribute("src") == url || s.getAttribute("data-webpack
/******/                                }
/******/                        }
/******/                        if(!script) {
/******/                                needAttach = true;
/******/                                script = document.createElement('script');
/******/
/******/                                script.charset = 'utf-8';
/******/                                script.timeout = 120;
/******/                                if (__webpack_require__.nc) {
/******/                                        script.setAttribute("nonce", __webpack_require__.nc);
/******/                                }
/******/                                script.setAttribute("data-webpack", dataWebpackPrefix + key);
/******/                                script.src = url;
/******/                        }
/******/                        inProgress[url] = [done];
/******/                        var onScriptComplete = (prev, event) => {
/******/                                // avoid mem leaks in IE.
/******/                                script.onerror = script.onload = null;
```

3

```
/******/                    clearTimeout(timeout);
/******/                    var doneFns = inProgress[url];
/******/                    delete inProgress[url];
/******/                    script.parentNode && script.parentNode.removeChild(script);
/******/                    doneFns && doneFns.forEach((fn) => (fn(event)));
/******/                    if(prev) return prev(event);
/******/                }
/******/                ;
/******/                var timeout = setTimeout(onScriptComplete.bind(null, undefined, { type:
/******/                script.onerror = onScriptComplete.bind(null, script.onerror);
/******/                script.onload = onScriptComplete.bind(null, script.onload);
/******/                needAttach && document.head.appendChild(script);
/******/            };
/******/        })();
/******/
/******/        /* webpack/runtime/make namespace object */
/******/        (() => {
/******/            // define __esModule on exports
/******/            __webpack_require__.r = (exports) => {
/******/                if(typeof Symbol !== 'undefined' && Symbol.toStringTag) {
/******/                    Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module
/******/                }
/******/                Object.defineProperty(exports, '__esModule', { value: true });
/******/            };
/******/        })();
/******/
/******/        /* webpack/runtime/publicPath */
/******/        (() => {
/******/            __webpack_require__.p = "dist/";
/******/        })();
/******/
/******/        /* webpack/runtime/css loading */
/******/        (() => {
/******/            // object to store loaded and loading chunks
/******/            // undefined = chunk not loaded, null = chunk preloaded/prefetched
/******/            // [resolve, reject, Promise] = chunk loading, 0 = chunk loaded
/******/            var installedChunks = {};
/******/
/******/            var uniqueName = "app";
/******/            var loadCssChunkData = (target, link, chunkId) => {
/******/                var data, token = "", token2, exports = {}, exportsWithId = [], exports
/******/                try { if(!link) link = loadStylesheet(chunkId); data = link.sheet.cssRu
/******/                data = data.getPropertyValue("--webpack-" + uniqueName + "-" + chunkId)
/******/                if(!data) return [];
/******/                for(; cc; i++) {
/******/                    cc = data.charCodeAt(i);
```

4

```
/******/                    if(cc == 40) { token2 = token; token = ""; }
/******/                    else if(cc == 41) { exports[token2.replace(/^_/, "")] = token.repla
/******/                    else if(cc == 47 || cc == 37) { token = token.replace(/^_/, ""); ex
/******/                    else if(!cc || cc == 44) { token = token.replace(/^_/, ""); exports
/******/                        module.exports = exports;
/******/                    }).bind(null, exports); token = ""; exports = {}; exportsWithId.len
/******/                    else if(cc == 92) { token += data[++i] }
/******/                    else { token += data[i]; }
/******/                }
/******/                installedChunks[chunkId] = 0;
/******/
/******/            }
/******/            var loadingAttribute = "data-webpack-loading";
/******/            var loadStylesheet = (chunkId, url, done) => {
/******/                var link, needAttach, key = "chunk-" + chunkId;
/******/
/******/                var links = document.getElementsByTagName("link");
/******/                for(var i = 0; i < links.length; i++) {
/******/                    var l = links[i];
/******/                    if(l.rel == "stylesheet" && (l.href == url || l.getAttribute("href"
/******/                }
/******/                if(!done) return link;
/******/
/******/                if(!link) {
/******/                    needAttach = true;
/******/                    link = document.createElement('link');
/******/                    link.setAttribute("data-webpack", uniqueName + ":" + key);
/******/                    link.setAttribute(loadingAttribute, 1);
/******/                    link.rel = "stylesheet";
/******/                    link.href = url;
/******/                }
/******/                var onLinkComplete = (prev, event) => {
/******/                    link.onerror = link.onload = null;
/******/                    link.removeAttribute(loadingAttribute);
/******/                    clearTimeout(timeout);
/******/                    if(event && event.type != "load") link.parentNode.removeChild(link)
/******/                    done(event);
/******/                    if(prev) return prev(event);
/******/                };
/******/                if(link.getAttribute(loadingAttribute)) {
/******/                    var timeout = setTimeout(onLinkComplete.bind(null, undefined, { typ
/******/                    link.onerror = onLinkComplete.bind(null, link.onerror);
/******/                    link.onload = onLinkComplete.bind(null, link.onload);
/******/                } else onLinkComplete(undefined, { type: 'load', target: link });
/******/
/******/                needAttach && document.head.appendChild(link);
```

```
/******/                return link;
/******/            };
/******/            loadCssChunkData(__webpack_require__.m, 0, 0);
/******/
/******/            __webpack_require__.f.css = (chunkId, promises) => {
/******/                // css chunk loading
/******/                var installedChunkData = __webpack_require__.o(installedChunks, chunkId
/******/                if(installedChunkData !== 0) { // 0 means "already installed".
/******/
/******/                    // a Promise means "currently loading".
/******/                    if(installedChunkData) {
/******/                        promises.push(installedChunkData[2]);
/******/                    } else {
/******/                        if(true) { // all chunks have CSS
/******/                            // setup Promise in chunk cache
/******/                            var promise = new Promise((resolve, reject) => (installedCh
/******/                            promises.push(installedChunkData[2] = promise);
/******/
/******/                            // start chunk loading
/******/                            var url = __webpack_require__.p + __webpack_require__.k(chu
/******/                            // create error before stack unwound to get useful stacktra
/******/                            var error = new Error();
/******/                            var loadingEnded = (event) => {
/******/                                if(__webpack_require__.o(installedChunks, chunkId)) {
/******/                                    installedChunkData = installedChunks[chunkId];
/******/                                    if(installedChunkData !== 0) installedChunks[chunkI
/******/                                    if(installedChunkData) {
/******/                                        if(event.type !== "load") {
/******/                                            var errorType = event && event.type;
/******/                                            var realSrc = event && event.target && even
/******/                                            error.message = 'Loading css chunk ' + chun
/******/                                            error.name = 'ChunkLoadError';
/******/                                            error.type = errorType;
/******/                                            error.request = realSrc;
/******/                                            installedChunkData[1](error);
/******/                                        } else {
/******/                                            loadCssChunkData(__webpack_require__.m, lin
/******/                                            installedChunkData[0]();
/******/                                        }
/******/                                    }
/******/                                }
/******/                            };
/******/                            var link = loadStylesheet(chunkId, url, loadingEnded);
/******/                        } else installedChunks[chunkId] = 0;
/******/                    }
/******/                }
```

```
/******/          };
/******/
/******/          // no hmr
/******/      })();
/******/
/******/      /* webpack/runtime/jsonp chunk loading */
/******/      (() => {
/******/          // no baseURI
/******/
/******/          // object to store loaded and loading chunks
/******/          // undefined = chunk not loaded, null = chunk preloaded/prefetched
/******/          // [resolve, reject, Promise] = chunk loading, 0 = chunk loaded
/******/          var installedChunks = {
/******/              0: 0
/******/          };
/******/
/******/          __webpack_require__.f.j = (chunkId, promises) => {
/******/                  // JSONP chunk loading for javascript
/******/                  var installedChunkData = __webpack_require__.o(installedChunks, chu
/******/                  if(installedChunkData !== 0) { // 0 means "already installed".
/******/
/******/                      // a Promise means "currently loading".
/******/                      if(installedChunkData) {
/******/                          promises.push(installedChunkData[2]);
/******/                      } else {
/******/                          if(0 == chunkId) {
/******/                              // setup Promise in chunk cache
/******/                              var promise = new Promise((resolve, reject) => (install
/******/                              promises.push(installedChunkData[2] = promise);
/******/
/******/                              // start chunk loading
/******/                              var url = __webpack_require__.p + __webpack_require__.u
/******/                              // create error before stack unwound to get useful stac
/******/                              var error = new Error();
/******/                              var loadingEnded = (event) => {
/******/                                  if(__webpack_require__.o(installedChunks, chunkId))
/******/                                      installedChunkData = installedChunks[chunkId];
/******/                                      if(installedChunkData !== 0) installedChunks[ch
/******/                                      if(installedChunkData) {
/******/                                          var errorType = event && (event.type === 'l
/******/                                          var realSrc = event && event.target && even
/******/                                          error.message = 'Loading chunk ' + chunkId
/******/                                          error.name = 'ChunkLoadError';
/******/                                          error.type = errorType;
/******/                                          error.request = realSrc;
/******/                                          installedChunkData[1](error);
```

7

```
/******/                                                    }
/******/                                                }
/******/                                            };
/******/                                            __webpack_require__.l(url, loadingEnded, "chunk-" + chu
/******/                                    } else installedChunks[chunkId] = 0;
/******/                                }
/******/                            }
/******/                    };
/******/
/******/                    // no prefetching
/******/
/******/                    // no preloaded
/******/
/******/                    // no HMR
/******/
/******/                    // no HMR manifest
/******/
/******/                    // no on chunks loaded
/******/
/******/                    // install a JSONP callback for chunk loading
/******/                    var webpackJsonpCallback = (parentChunkLoadingFunction, data) => {
/******/                        var [chunkIds, moreModules, runtime] = data;
/******/                        // add "moreModules" to the modules object,
/******/                        // then flag all "chunkIds" as loaded and fire callback
/******/                        var moduleId, chunkId, i = 0;
/******/                        if(chunkIds.some((id) => (installedChunks[id] !== 0))) {
/******/                            for(moduleId in moreModules) {
/******/                                if(__webpack_require__.o(moreModules, moduleId)) {
/******/                                    __webpack_require__.m[moduleId] = moreModules[moduleId];
/******/                                }
/******/                            }
/******/                            if(runtime) var result = runtime(__webpack_require__);
/******/                        }
/******/                        if(parentChunkLoadingFunction) parentChunkLoadingFunction(data);
/******/                        for(;i < chunkIds.length; i++) {
/******/                            chunkId = chunkIds[i];
/******/                            if(__webpack_require__.o(installedChunks, chunkId) && installedChun
/******/                                installedChunks[chunkId][0]();
/******/                            }
/******/                            installedChunks[chunkId] = 0;
/******/                        }
/******/
/******/                    }
/******/
/******/                    var chunkLoadingGlobal = self["webpackChunkapp"] = self["webpackChunkapp"]
/******/                    chunkLoadingGlobal.forEach(webpackJsonpCallback.bind(null, 0));
```

```
/******/            chunkLoadingGlobal.push = webpackJsonpCallback.bind(null, chunkLoadingGloba
/******/    })();
/******/
/************************************************************************/
var __webpack_exports__ = {};
// This entry need to be wrapped in an IIFE because it need to be isolated against other mod
(() => {
/*!********************!*\
  !*** ./example.js ***!
  \********************/
/*! namespace exports */
/*! exports [not provided] [no usage info] */
/*! runtime requirements: __webpack_require__, __webpack_require__.r, __webpack_exports__, _
__webpack_require__.r(__webpack_exports__);
/* harmony import */ var _style_css__WEBPACK_IMPORTED_MODULE_0__ = __webpack_require__(/*!
/* harmony import */ var _style2_css__WEBPACK_IMPORTED_MODULE_1__ = __webpack_require__(/*!
/* harmony import */ var _style_module_css__WEBPACK_IMPORTED_MODULE_2__ = __webpack_require



__webpack_require__.e(/*! import() */ 1).then(__webpack_require__.bind(__webpack_require__,

document.getElementsByTagName("main")[0].className = _style_module_css__WEBPACK_IMPORTED_MOI

})();

/******/ })()
;
```

dist/output.css

```css
@import url("https://fonts.googleapis.com/css?family=Open+Sans");
.img {
    width: 150px;
    height: 150px;
    background: url(89a353e9c515885abd8e.png);
}


body {
    background: green;
    font-family: "Open Sans";
}
```

```css
body {
    background: red;
}

:root {
    --app-6-large: 72px;
}

.app-6-main {
    font-size: var(--app-6-large);
    color: darkblue;
}

head{--webpack-app-0:_4,_2,_1,_5,large%main/_6;}
```

production

```css
@import url("https://fonts.googleapis.com/css?family=Open+Sans");
.img {
    width: 150px;
    height: 150px;
    background: url(89a353e9c515885abd8e.png);
}


body {
    background: green;
    font-family: "Open Sans";
}

body {
    background: red;
}

:root {
    --app-491-b: 72px;
}

.app-491-D {
    font-size: var(--app-491-b);
    color: darkblue;
}

head{--webpack-app-179:_548,_431,_258,_268,b%D/_491;}
```

dist/1.output.css

```css
body {
    color: blue;
}

head{--webpack-app-1:_7;}
```

Info

Unoptimized

```
assets by chunk 16.9 KiB (name: main)
  asset output.js 16.5 KiB [emitted] (name: main)
  asset output.css 385 bytes [emitted] (name: main)
asset 89a353e9c515885abd8e.png 14.6 KiB [emitted] [immutable] [from: images/file.png] (auxil
asset 1.output.css 49 bytes [emitted]
Entrypoint main 16.9 KiB (14.6 KiB) = output.js 16.5 KiB output.css 385 bytes 1 auxiliary as
chunk (runtime: main) output.js, output.css (main) 218 bytes (javascript) 335 bytes (css) 14
  > ./example.js main
  runtime modules 10 KiB 9 modules
  dependent modules 42 bytes (javascript) 14.6 KiB (asset) 335 bytes (css) 42 bytes (css-imp
  ./example.js 176 bytes [built] [code generated]
    [no exports]
    [used exports unknown]
    entry ./example.js main
chunk (runtime: main) 1.output.css 23 bytes
  > ./lazy-style.css ./example.js 4:0-26
  ./lazy-style.css 23 bytes [built] [code generated]
    [no exports]
    [used exports unknown]
    import() ./lazy-style.css ./example.js 4:0-26
webpack 5.66.0 compiled successfully
```

Production mode

```
assets by chunk 4.25 KiB (name: main)
  asset output.js 3.87 KiB [emitted] [minimized] (name: main)
  asset output.css 385 bytes [emitted] (name: main)
asset 89a353e9c515885abd8e.png 14.6 KiB [emitted] [immutable] [from: images/file.png] (auxil
asset 159.output.css 53 bytes [emitted]
Entrypoint main 4.25 KiB (14.6 KiB) = output.js 3.87 KiB output.css 385 bytes 1 auxiliary as
chunk (runtime: main) 159.output.css 23 bytes
  > ./lazy-style.css ./example.js 4:0-26
  ./lazy-style.css 23 bytes [built] [code generated]
    [no exports]
```

```
    import() ./lazy-style.css ./example.js 4:0-26
chunk (runtime: main) output.js, output.css (main) 218 bytes (javascript) 335 bytes (css) 14
  > ./example.js main
  runtime modules 10 KiB 9 modules
  dependent modules 42 bytes (javascript) 14.6 KiB (asset) 335 bytes (css) 42 bytes (css-imp
  ./example.js 176 bytes [built] [code generated]
    [no exports]
    [no exports used]
    entry ./example.js main
webpack 5.66.0 compiled successfully
```