# gatsby-remark-copy-linked-files

Copies local files linked to/from Markdown (`.md|.markdown`) files to the root directory (i.e., `public` folder).

**A sample markdown file:**

```
---
title: My awesome blog post
---

Hey everyone, I just made a sweet PDF with lots of interesting stuff in it.

[Download it now](my-awesome-pdf.pdf)
```

**When you build your site:**

The `my-awesome-pdf.pdf` file will be copied to the root directory (i.e., `public/some-really-long-contenthash/my-awesome-pdf.pdf`) and the generated HTML page will be modified to point to it.

> **Note**: The `my-awesome-pdf.pdf` file should be in the same directory as the markdown file.

---

## Install plugin

```
npm install gatsby-remark-copy-linked-files
```

## Add plugin to Gatsby Config

**Default settings:**

Add `gatsby-remark-copy-linked-files` plugin as a plugin to `gatsby-transformer-remark`:

```
// In your gatsby-config.js

// add plugin by name only
plugins: [
  {
    resolve: `gatsby-transformer-remark`,
    options: {
      plugins: [`gatsby-remark-copy-linked-files`],
    },
  },
]
```

**Custom settings:**

```
// In your gatsby-config.js

// add plugin by name and options
plugins: [
  {
    resolve: `gatsby-transformer-remark`,
    options: {
      plugins: [
        {
          resolve: `gatsby-remark-copy-linked-files`,
          options: {
            destinationDir: `path/to/dir`,
            ignoreFileExtensions: [`png`, `jpg`, `jpeg`, `bmp`, `tiff`],
          },
        },
      ],
    },
  },
]
```

---

## Custom set where to copy the files using `destinationDir`

By default, all files will be copied to the root directory (i.e., `public` folder) in the following format: `contentHash/fileName.ext`.

> For example, `[Download it now](my-awesome-pdf.pdf)` will copy the file `my-awesome-pdf.pdf` to something like `public/2a0039f3a61f4510f41678438e4c863a/my-awesome-pdf.pdf`

### Simple usage

To change this, set `destinationDir` to a path of your own choosing (i.e., `path/to/dir`).

```
// In your gatsby-config.js
plugins: [
  {
    resolve: `gatsby-transformer-remark`,
    options: {
      plugins: [
        {
          resolve: "gatsby-remark-copy-linked-files",
          options: {
            destinationDir: "path/to/dir",
          },
        },
```

```
      ],
    },
  },
]
```

So now, `[Download it now](my-awesome-pdf.pdf)` will copy the file `my-awesome-pdf.pdf` to `public/path/to/dir/2a0039f3a61f4510f41678438e4c863a/my-awesome-p`

### Advanced usage

For more advanced control, set `destinationDir` to a function expression using properties `name` and/or `hash` to specify the path.

**Examples:**

```
# save `my-awesome-pdf.pdf` to `public/my-awesome-pdf.pdf`
destinationDir: f => `${f.name}`

# save `my-awesome-pdf.pdf` to `public/2a0039f3a61f4510f41678438e4c863a.pdf`
destinationDir: f => `${f.hash}`

# save `my-awesome-pdf.pdf` to `public/downloads/2a0039f3a61f4510f41678438e4c863a/my-awesome
destinationDir: f => `downloads/${f.hash}/${f.name}`

# save `my-awesome-pdf.pdf` to `public/downloads/2a0039f3a61f4510f41678438e4c863a-my-awesome
destinationDir: f => `downloads/${f.hash}-${f.name}`

# save `my-awesome-pdf.pdf` to `public/my-awesome-pdf/2a0039f3a61f4510f41678438e4c863a.pdf`
destinationDir: f => `${f.name}/${f.hash}`

# save `my-awesome-pdf.pdf` to `public/path/to/dir/hello-my-awesome-pdf+2a0039f3a61f4510f416
destinationDir: f => `path/to/dir/hello-${f.name}+${f.hash}_world`
```

> **Note:** Make sure you use either `name` or `hash` property in your function expression! If you don't include both `name` and `hash` properties in your function expression, `gatsby-remark-copy-linked-files` plugin will resolve the function expression to a string value and use default settings as a fallback mechanism to prevent your local files from getting copied with the same name (causing files to get overwritten).

```
# Note: `my-awesome-pdf.pdf` is saved to `public/hello/2a0039f3a61f4510f41678438e4c863a/my-a
# because `name` and `hash` properties are not referenced in the function expression.
# So these function expressions are treated the same way
destinationDir: _ => `hello`
destinationDir: `hello`
```

**Caveat: Error thrown if `destinationDir` points outside the root directory (i.e. `public` folder)**

> **Note:** An error will be thrown if the destination points outside the root directory (i.e. `public` folder).

**Correct:**

```
# saves to `public/path/to/dir/`
destinationDir: `path/to/dir`

# saves to `public/path/to/dir/`
destinationDir: _ => `path/to/dir`

# saves to `public/path/to/dir/fileName.ext`
destinationDir: f => `path/to/dir/${f.name}`

# saves to `public/contentHash.ext`
destinationDir: f => `${f.hash}`
```

**Error thrown:**

```
# cannot save outside root directory (i.e., outside `public` folder)
destinationDir: `../path/to/dir`
destinationDir: _ => `../path/to/dir`
destinationDir: f => `../path/to/dir/${f.name}`
destinationDir: f => `../${f.hash}`
```

---

**Custom set which file types to ignore using `ignoreFileExtensions`**

By default, the file types that this plugin ignores are: `png`, `jpg`, `jpeg`, `bmp`, `tiff`.

> For example, `[Download it now](image.png)` will be ignored and not copied to the root dir (i.e. `public` folder)

To change this, set `ignoreFileExtensions` to an array of extensions to ignore (i.e., an empty array `[]` to ignore nothing).

```
// In your gatsby-config.js
plugins: [
  {
    resolve: `gatsby-transformer-remark`,
    options: {
      plugins: [
        {
          resolve: "gatsby-remark-copy-linked-files",
          options: {
            // `ignoreFileExtensions` defaults to [`png`, `jpg`, `jpeg`, `bmp`, `tiff`]
```

```
            // as we assume you'll use gatsby-remark-images to handle
            // images in markdown as it automatically creates responsive
            // versions of images.
            //
            // If you'd like to not use gatsby-remark-images and just copy your
            // original images to the public directory, set
            // `ignoreFileExtensions` to an empty array.
            ignoreFileExtensions: [],
          },
        },
      ],
    },
  },
]
```

So now, `[Download it now](image.png)` will be copied to the root dir (i.e. `public` folder)

---

### Supported Markdown tags

- img - `![Image](my-img.png)`
- link - `[Link](myFile.txt)`

### Supported HTML tags

- `<img />`
- `<video />`
- `<audio />`
- `<a />`