

Fragments allow you to reuse parts of GraphQL queries. It also allows you to split up complex queries into smaller, easier to understand components.

The building blocks of a fragment

Here is an example fragment:

```
fragment FragmentName on TypeName {  
  field1  
  field2  
}
```

A fragment consists of three components:

1. `FragmentName` : the name of the fragment that will be referenced later.
2. `TypeName` : the [GraphQL type](#) of the object the fragment will be used on. This is important because you can only query for fields that actually exist on a given object.
3. The body of the query. You can define any fields with any level of nesting in here, the same that you would elsewhere in a GraphQL query

Creating and using a fragment

A fragment can be created inside any GraphQL query, but it's good practice to create the query separately. More organization advice in the [Conceptual Guide](#).

```
import React from "react"  
import { graphql } from "gatsby"  
  
export default function IndexPost( props ) {  
  return (...)  
}  
  
export const query = graphql`  
  fragment SiteInformation on Site {  
    siteMetadata {  
      title  
      siteDescription  
    }  
  }  
`  
`
```

This defines a fragment named `SiteInformation` . Now it can be used from within the page's GraphQL query:

```
import React from "react"  
import { graphql } from "gatsby"  
import IndexPost from "../components/IndexPost"  
  
export default function Main({ data }) {  
  return (  
    <div>
```

```

    <h1>{data.site.siteMetadata.title}</h1>
    <p>{data.site.siteMetadata.siteDescription}</p>

    /**
     * Or you can pass all the data from the fragment
     * back to the component that defined it
     */
    <IndexPost siteInformation={data.site.siteMetadata} />
  </div>
)
}

export const query = graphql`
  query {
    site {
      ...SiteInformation
    }
  }
`

```

When compiling your site, Gatsby preprocesses all GraphQL queries it finds. Therefore, any file that gets included in your project can define a snippet. However, only Pages can define GraphQL queries that actually return data. This is why you can define the fragment in the component file - it doesn't actually return any data directly.

Further reading

- [Querying Data with GraphQL - Fragments](#)
- [GraphQL Docs - Fragments](#)