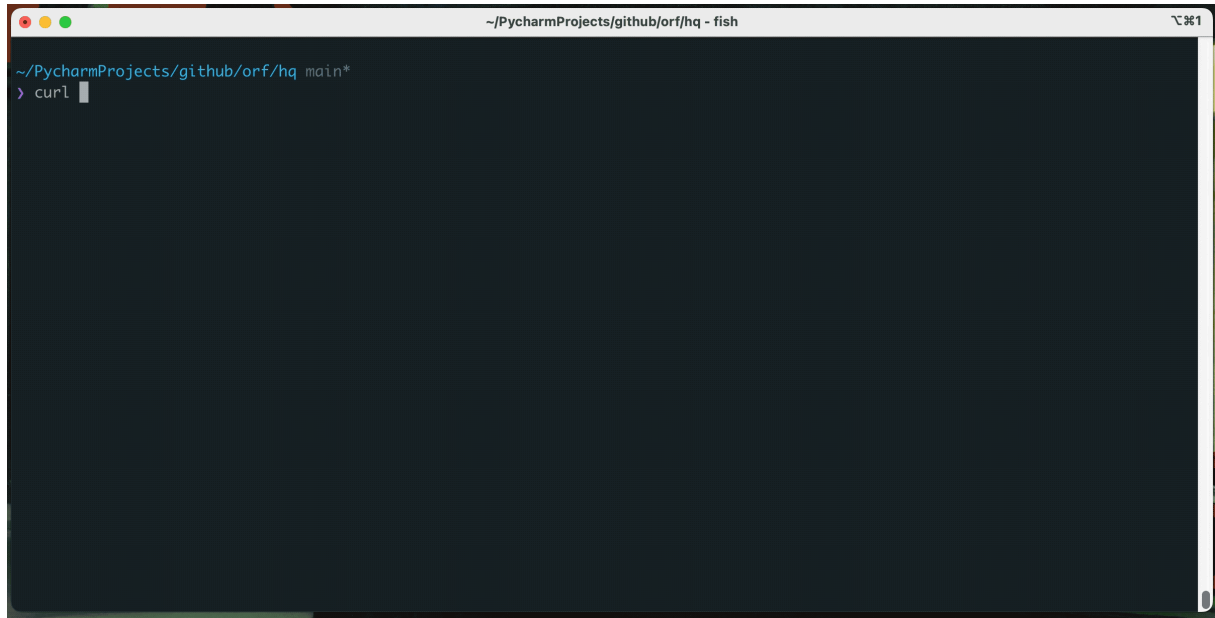


---

## hq

crates.io v1.2.2

jq, but for HTML. Try it in your browser here



hq reads HTML and converts it into a JSON object based on a series of CSS selectors. The selectors are expressed in a similar way to JSON, but where the values are CSS selectors. For example:

```
1 {posts: .athing | [ {title: .titleline > a, url: .titleline > a | @(href)} ] }
```

This will select all `.athing` elements, and it will create an array (`| [ {...} ]`) of objects for each element selected. Then for each element it will select the text of the `titleline > a` element, and the `href` attribute (`| @(href)`).

The end result is the following structure:

```
1 {
2   "posts": [
3     {
4       "title": "...",
5       "url": "..."
6     }
7   ]
8 }
```

---

## Install

`brew install jq, or cargo install html-query`

## Special query syntax

### Text

`.foo | @text`

This will select the text content from the first element matching `.foo`.

### Selecting attributes

`.foo | @(href)`

This will select the `href` attribute from the first element matching `.foo`.

### Parents

`.foo | @parent`

This will return the parent element from the first element matching `.foo`.

### Siblings

`.foo | @sibling(1)`

This will return the sibling element from the first element matching `.foo`.

## Examples

### Full hacker news story extraction

```
1 {posts: .athing | [{href: .titleline > a | @(href), title: .titleline >
  a, meta: @sibling(1) | {user: .hnuser, posted: .age | @(title) }]]}
```

This selects each `.athing` element, extracts the URL from the `href` attribute as well as the title. It then selects the *sibling* `.athing` element, and extracts the user and post time from that:

---

```
1 {
2   "posts": [
3     {
4       "title": "...",
5       "url": "...",
6       "meta": {
7         "posted": "...",
8         "user": "..."
9       }
10    }
11  ]
12 }
```