Assignment

- Implement a function that receives a list of URLs of HTML pages. It processes the URLs **in parallel** and for each URL it:
  - Downloads the HTML page
  - Parses the HTML and extracts URLs of images
    - To do this step, you can use the function given on the next slide
  - Downloads the images **in parallel** and stores them on disk

Extracting Image URLs from HTML

```
def get_img_urls(doc_url, doc_content):
    from html.parser import HTMLParser
    from urllib.parse import urljoin

    class ImgUrlParser(HTMLParser):
        def __init__(self):
            HTMLParser.__init__(self)
            self.output_list = []

        def handle_starttag(self, tag, attrs):
            nonlocal doc_url

            if tag == 'img':
                src_attr = dict(attrs).get('src')
                url = urljoin(doc_url, src_attr)
                self.output_list.append(url)

    parser = ImgUrlParser()
    parser.feed(doc_content)
    return parser.output_list


html_doc = '''
<!DOCTYPE html>
<html>
    <body>
        <img src="/modules/MatFyzWeb/img/logo_mff_cz.svg">
    </body>
</html>
'''

print(get_img_urls('https://www.mff.cuni.cz/en', html_doc))
```

Assignment

- Extend the previous assignment to download the same image only once
- This means that if a particular image URL has been downloaded when processing one HTML page, it won't be downloaded again when processing another HTML page
- Note that this requires to share state between simultaneously running co-routines

Assignment

- Extend the previous assignment to limit parallel downloads to maximum 3 simultaneously running downloads