

Typical Python App (Object Oriented)

```
1  #!/usr/bin/env python3
2  from urllib.request import urlretrieve
3  import sys
4
5  class Downloader():
6      def download(self, urls, target_dir):
7          for url in urls:
8              filename=url.split("/")[-1]
9              print(f"Downloading '#{url}'...", end="")
10             try:
11                 urlretrieve(url, f"{target_dir}/{filename}")
12                 print("OK")
13             except Exception as e:
14                 print(f"ERR ({e})")
15
16  class ImageDownloader(Downloader):
17      def __init__(self, target_dir="images"):
18          self.target_dir = target_dir
19      def __str__(self):
20          return f"The Image Downloader. We save to '{self.target_dir}'."
21      def down(self, urls):
22          super().download(urls, self.target_dir,)
23
24  if __name__ == "__main__":
25      urls = sys.argv[1:]
26      if len(urls)<1:
27          exit("Usage: download.py <url> [<url2> ...]")
28      dl = ImageDownloader()
29      print(dl)
30      dl.down(urls)
```

Typical Python App (Object Oriented)

```
1  #!/usr/bin/env python3
2  from urllib.request import urlretrieve
3  import sys
4
5  class Downloader():
6      def download(self, urls, target_dir):
7          for url in urls:
8              filename=url.split("/")[-1]
9              print(f"Downloading '#{url}'...", end="")
10             try:
11                 urlretrieve(url, f"{target_dir}/{filename}")
12                 print("OK")
13             except Exception as e:
14                 print(f"ERR ({e})")
15
16  class ImageDownloader(Downloader):
17      def __init__(self, target_dir="images"):
18          self.target_dir = target_dir
19      def __str__(self):
20          return f"The Image Downloader. We save to '{self.target_dir}'."
21      def down(self, urls):
22          super().download(urls, self.target_dir,)
23
24  if __name__ == "__main__":
25      urls = sys.argv[1:]
26      if len(urls)<1:
27          exit("Usage: download.py <url> [<url2> ...]")
28      dl = ImageDownloader()
29      print(dl)
30      dl.down(urls)
```

Explanations:

5. `class Downloader()`:
↖ Object-oriented programming (OOP) with 'class' keyword.
6. `def download(self, urls, target_dir)`:
↖ First parameter is always 'self' (variable name 'self' is just a convention). With 'self' we point to the instance. E.g. to modify internal state.
16. `class ImageDownloader(Downloader)`:
↖ Inheritance. Base class is 'Downloader'.
17. `def __init__(self, target_dir="images")`:
↖ Constructor with 'dunder' (double underscore) method .
18. ...
19. `def __str__(self)`:
↖ The toString method for pretty print.
20. ...
22. `super().download(urls, self.target_dir,)`:
↖ Use 'super()' to access base class .