

COMPUTALA

LCB Depot's exhibition & events programme highlights work from different creative practices every month.

This month the focus is on digital art with [@newmediaartclub](#)'s computala exploring art with the internet, with work from established professionals and emerging creatives. Exhibitions and events run from **13th – 31st January 2025**

Find out more [@lcbdepot](#) or at lcbdepot.co.uk/event/computala/

```
# a poem which is a program which is an art gallery
# Elliot Smith 2022-09-24
from http.client import HTTPSConnection as Observe
from http.server import HTTPServer as TheIdeaOfAPlaceToShowArt
from http.server import BaseHTTPRequestHandler as ThosePrivilegedToPresentUsWithArt
from html.parser import HTMLParser as TheAudienceTakingTheRoleOfAGod
from random import choice

the_world = "www.wikihow.com"
why_must_we_lie_to_experience_art = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) + \
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36"

striving_for_aesthetic_fulfilment = Observe(the_world)
striving_for_aesthetic_fulfilment.request(
    "GET",
    "/Set-Up-a-Digital-Art-Gallery",
    headers={'User-Agent': why_must_we_lie_to_experience_art}
)
artistic_response = striving_for_aesthetic_fulfilment.getresponse()
all_the_art = artistic_response.read()

god = TheAudienceTakingTheRoleOfAGod()
things_we_have_found = []
god.handle_starttag = lambda tag, attrs: [
    things_we_have_found.append(value) for attr, value in attrs
    if tag == 'img' and attr == 'src'
]

god.feed(f"{all_the_art}")
the_item_selected_for_beatification = choice(things_we_have_found)
if not the_item_selected_for_beatification.startswith("http"):
    # this is merely an approximation, but even a broken image is an image
    the_item_selected_for_beatification = f"https://{the_world}" + \
    the_item_selected_for_beatification

class TheGallery(ThosePrivilegedToPresentUsWithArt):
    def do_GET(self):
        # prosaic but necessary, machinery of our willing submission
        self.send_response(200)
        self.send_header("Content-type", "text/html")
        self.end_headers()
        self.wfile.write(bytes(f"""
        <html><head><title>An art gallery</title></head>
        <body>
        
        <p>Title: Whether I made this with my own hands is of no importance.
        I CHOSE| it.</p>
        </body>
        """, "utf-8"))

if __name__ == "__main__":
    the_vessel_through_which_art_is_delivered = \
        TheIdeaOfAPlaceToShowArt(("localhost", 8080), TheGallery)

    try:
        the_vessel_through_which_art_is_delivered.serve_forever()
    except KeyboardInterrupt:
        pass

# all things must end
the_vessel_through_which_art_is_delivered.server_close()
```

a_poem_which_is_a_program_which_is_an_art_gallery

Elliot Smith

A poem which is a program which is an art gallery is a piece of code which is a poem and a functional program which acts as an art gallery. The code is available at

https://github.com/townxelliot/a_poem_which_is_a_program_which_is_an_art_gallery under an MIT licence.

Artist Bio

I am not at university. I've been producing computational works of art on and off (not in massive quantities) for several years. I work as a professional software engineer in government.

n/a