

• BY JENN CREIGHTON •



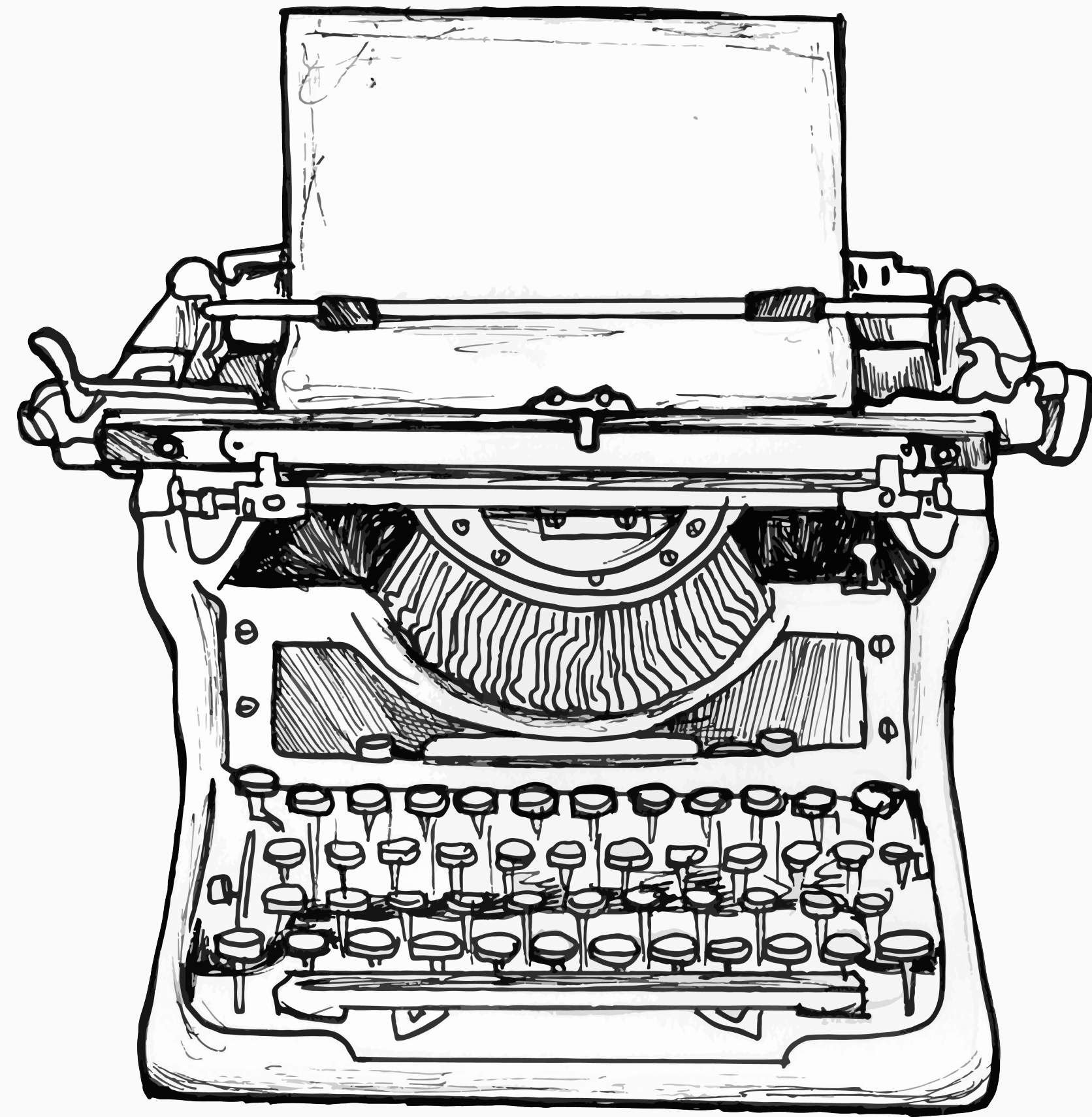
@gurlcode

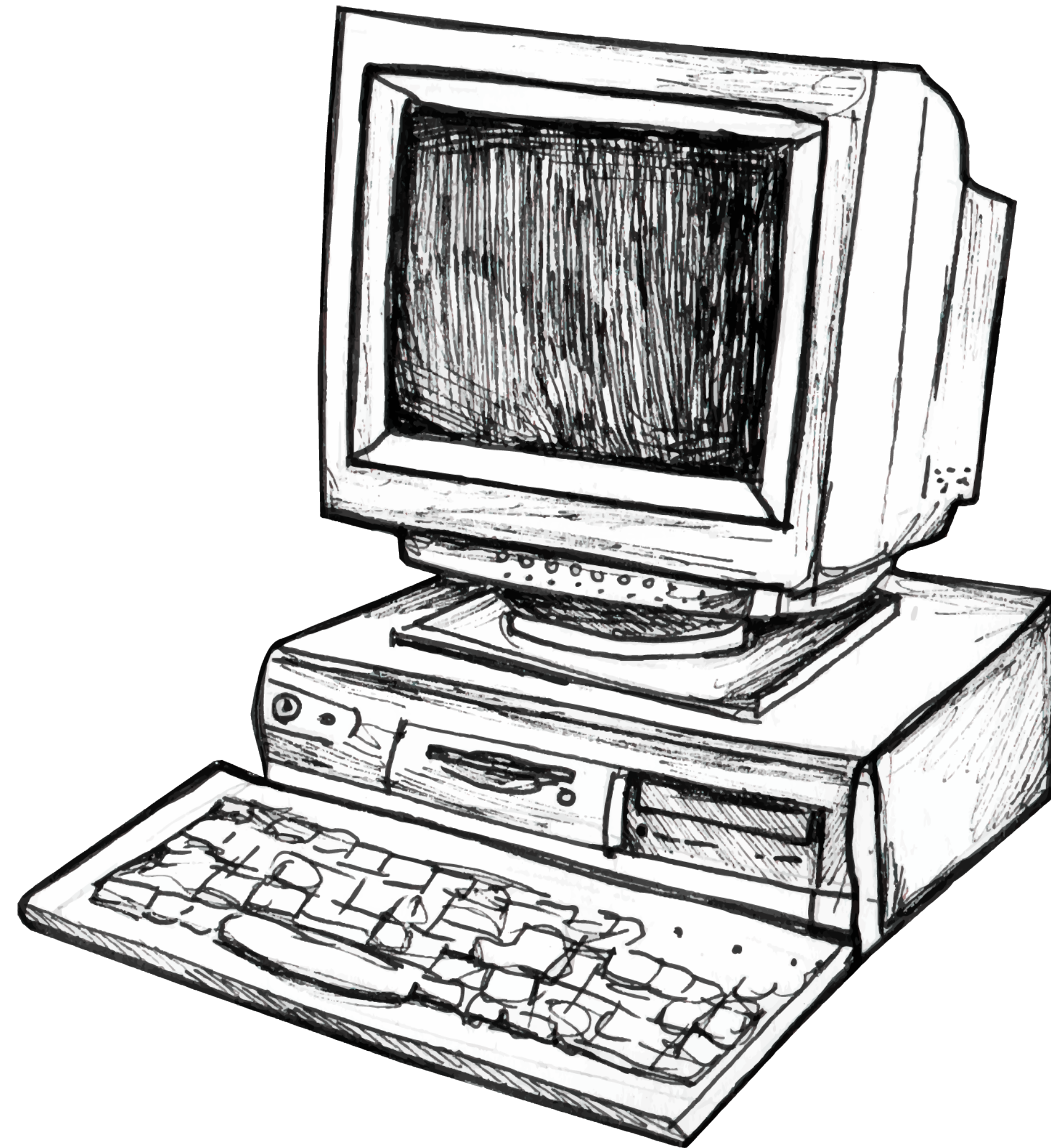


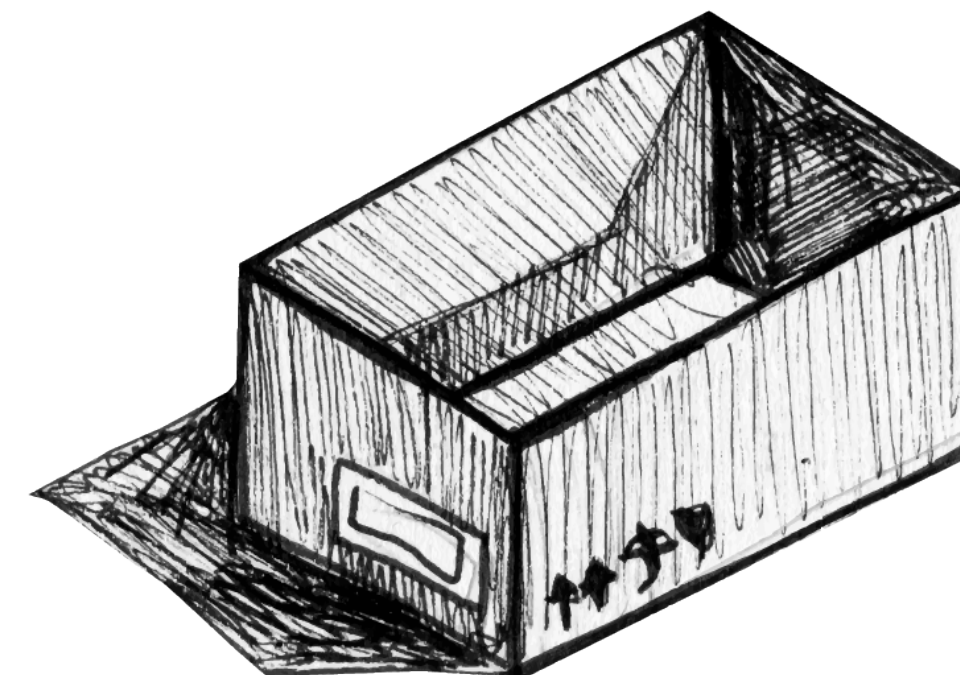
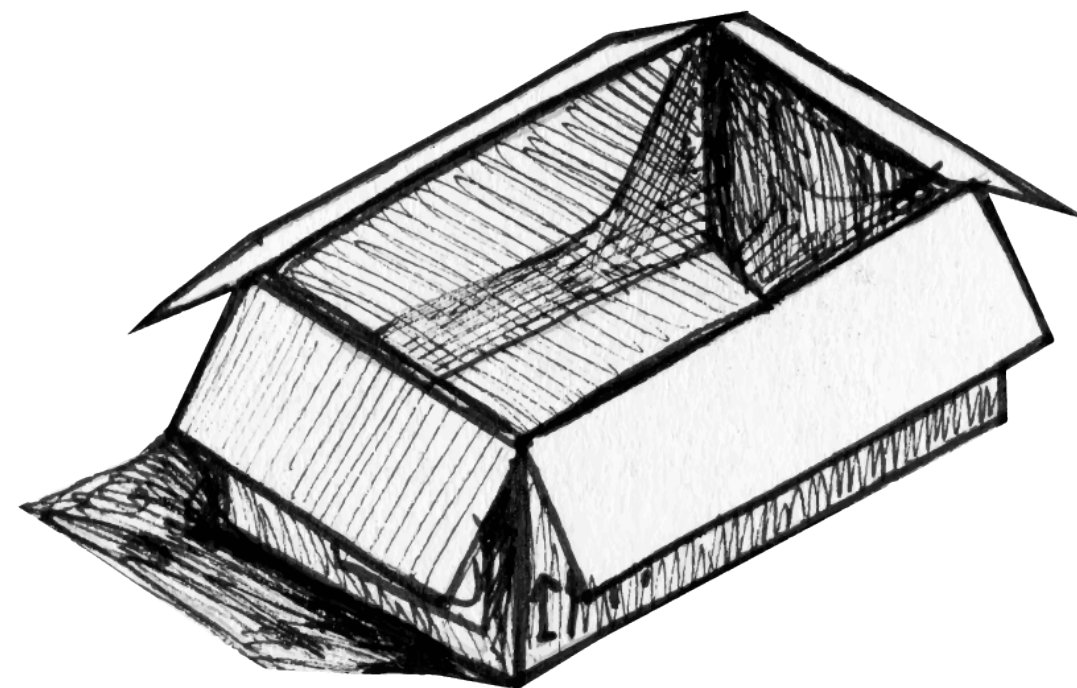
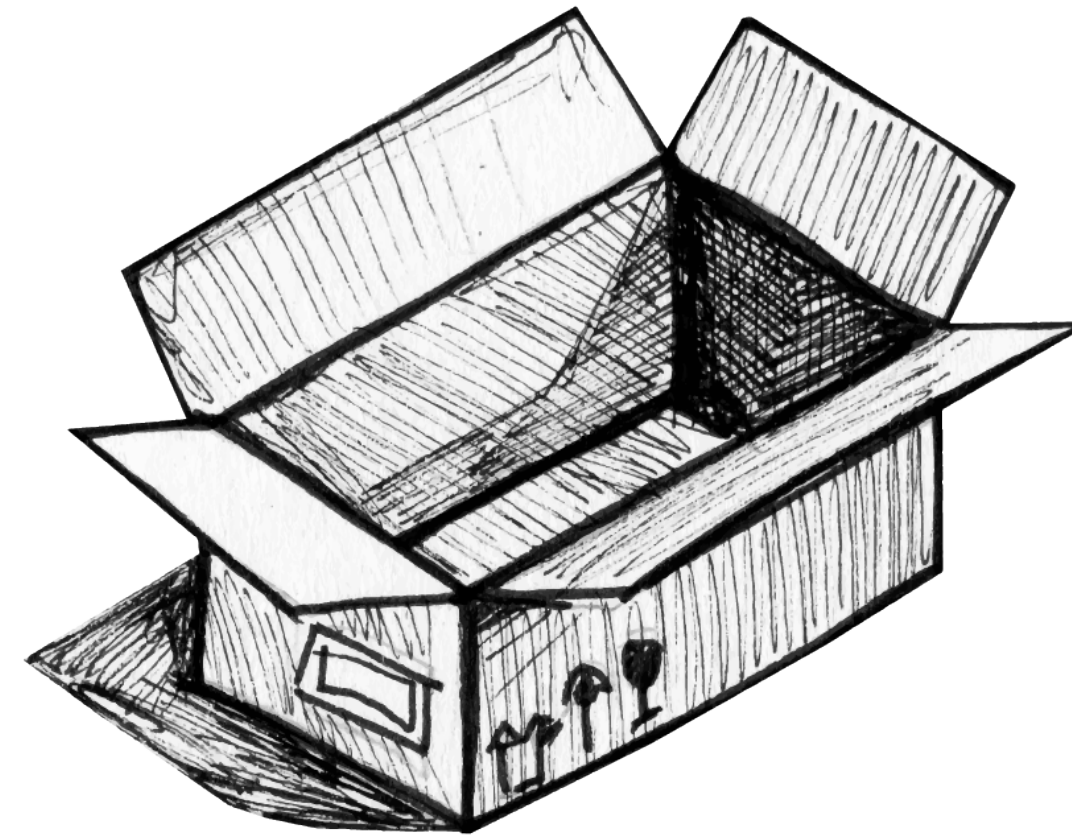
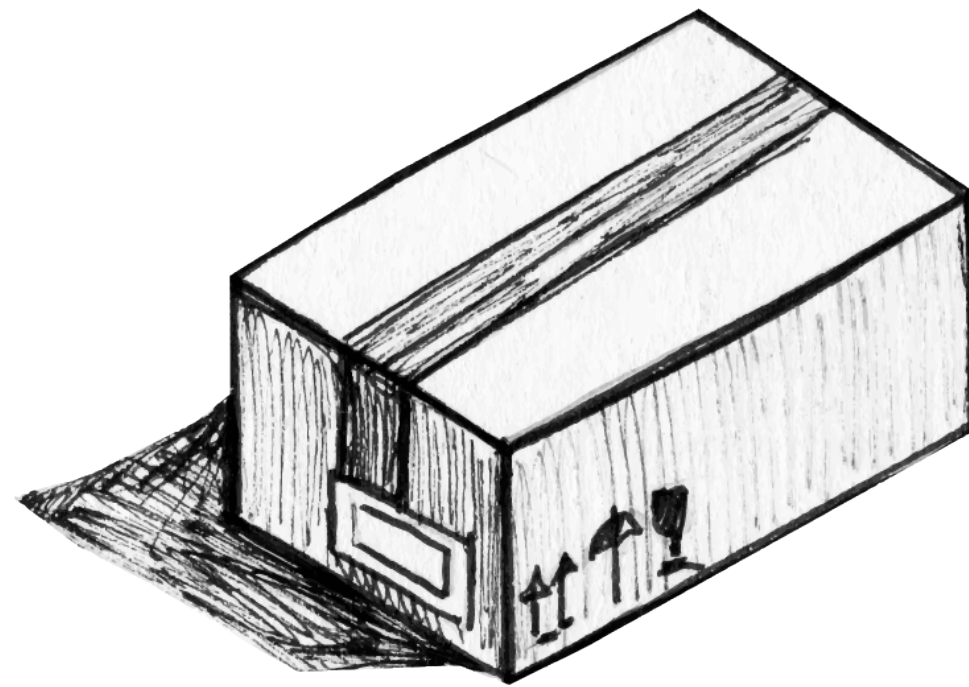
"Very cool at programming"
- my Director of Engineering

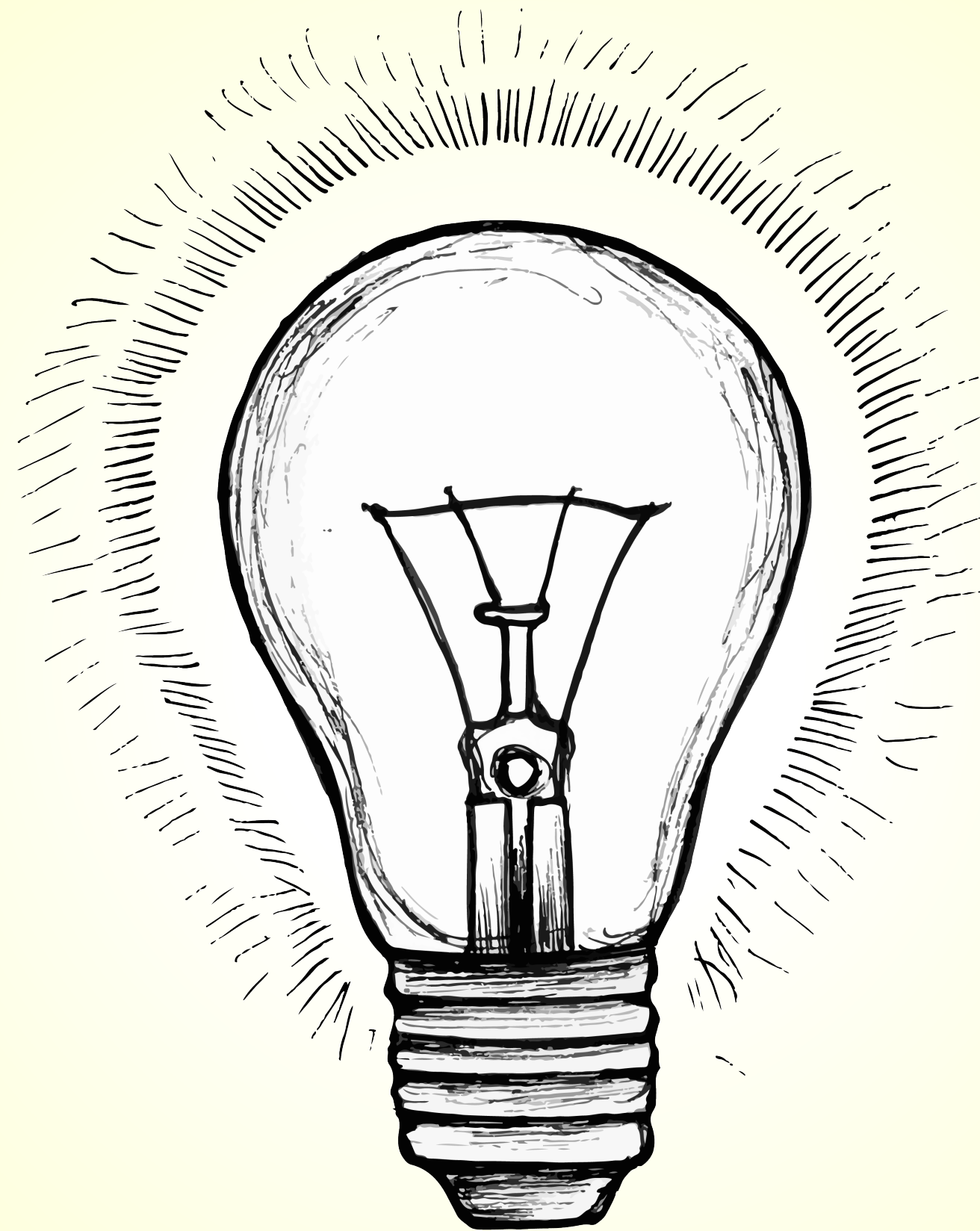
Frontend Architect @ The Wing
Co-organizer @useReactNYC

@gurlcode









Show, don't tell

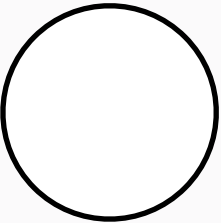
She is tired.

Her footsteps are heavier than before, the weight increasing as she trudges towards the bed, collapsing face-first onto the mattress.

Declarative, not imperative.



The Wing Spaces Happenings Login | Sign Up

The Wing Home Happenings Community 

Profile

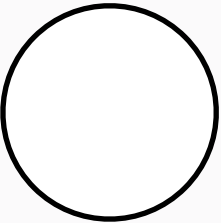
Settings

Logout

Be alert for possibilities of
paragraphing.

Joyce Carol Oates

The Wing Spaces Happenings Login | Sign Up

The Wing Home Happenings Community 

Profile

Settings

Logout

The Wing

Spaces

Happenings

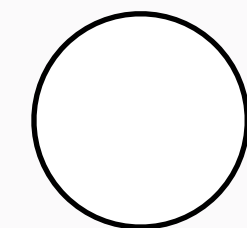
Login | Sign Up

The Wing

Home

Happenings

Community



Profile

Settings

Logout

The Wing

Spaces

Happenings

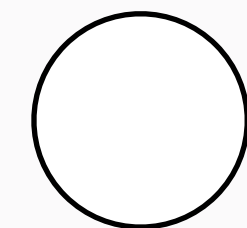
Login | Sign Up

The Wing

Home

Happenings

Community



Profile

Settings

Logout

[illegible]

```
const Navigation = () => (  
  <div>  
    // logo  
  
  </div>  
) ;
```



```
const Navigation = () => (  
  <div>  
    // logo  
    <nav>  
      // links  
    </nav>  
  
  </div>  
) ;
```

```
const Navigation = ({ isLoggedIn }) => (  
  <div>  
    // logo  
    <nav>  
      // links  
    </nav>  
  
  </div>  
) ;
```

```
const Navigation = ({ isLoggedIn }) => (  
  <div>  
    // logo  
    <nav>  
      // links  
    </nav>  
    {!isLoggedIn && // login + signup}  
    {isLoggedIn && // avatar}  
  </div>  
);
```


All good writers write [terrible first drafts].
This is how they end up with good second
drafts and terrific third drafts.

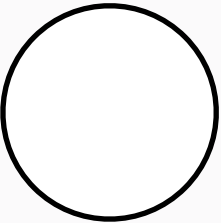
Anne Lamott

```
const Navigation = ({ isLoggedIn }) => (  
  <div>  
    // logo  
    <nav>  
      // links  
    </nav>  
    {!isLoggedIn && // login + signup}  
    {isLoggedIn && // avatar + dropdown}  
  </div>  
);
```

```
<Navigation isLoggedIn={isLoggedIn} />
```


Show, don't tell

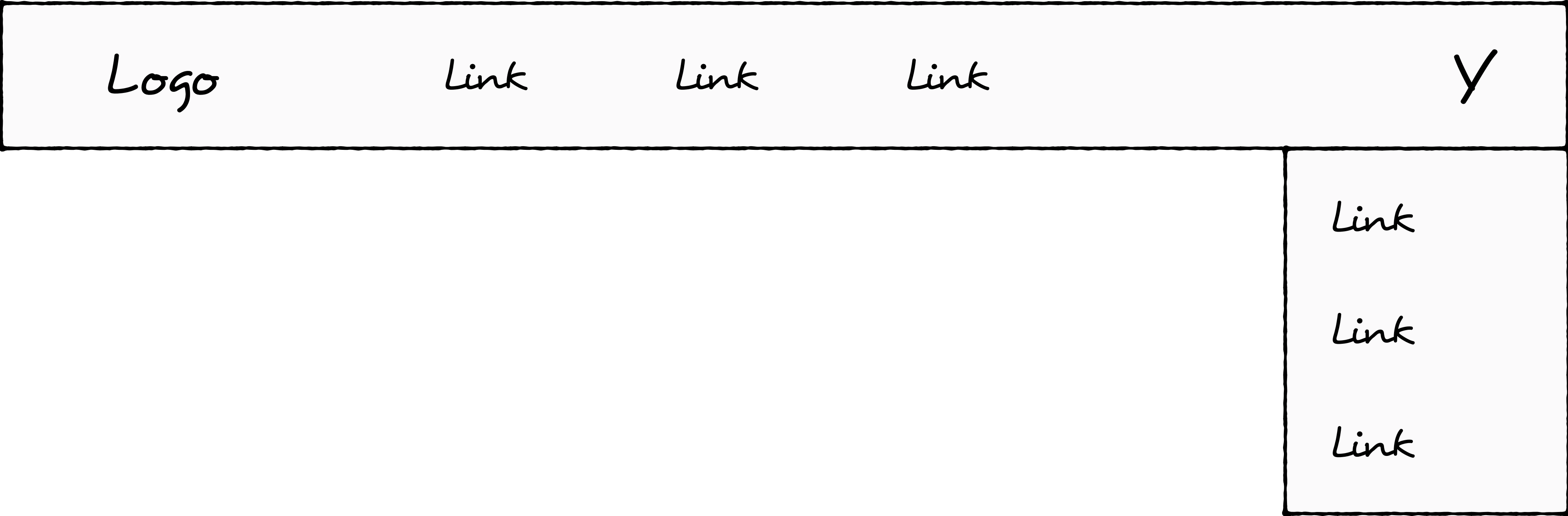
The Wing Spaces Happenings Login | Sign Up

The Wing Home Happenings Community 

Profile

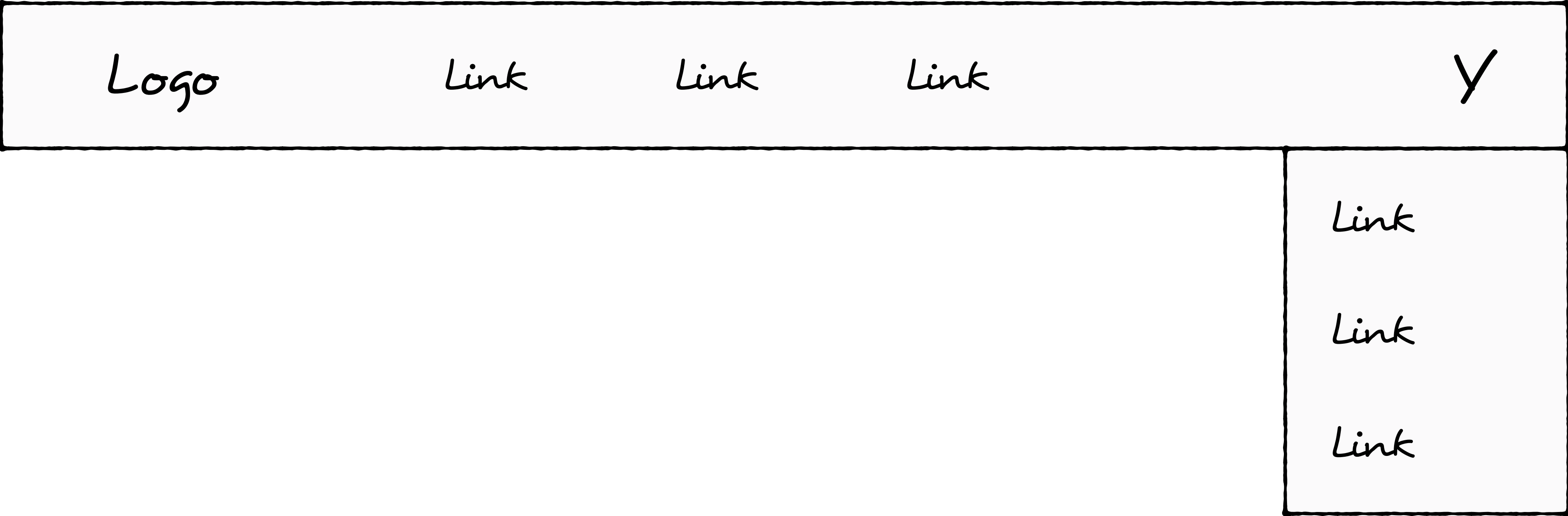
Settings

Logout



Every character should want something,
even if it is only a glass of water.

Kurt Vonnegut



The Wing

Happenings

The Wing

Happenings

Spaces

Login | Sign Up

HomeCommunity

Profile

Settings

Logout

```
<LoggedInNavigation />
```

```
<LoggedOutNavigation />
```



```
import { Logo } from './logo';

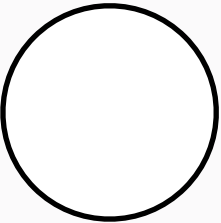
const LoggedInNavigation = () => (
  <div>
    <Logo />

  </div>
);
```

```
import { Logo } from './logo';

const LoggedInNavigation = () => (
  <div>
    <Logo />
    <nav>
      // links
    </nav>
    // avatar
  </div>
);
```

The Wing Spaces Happenings Login | Sign Up

The Wing Home Happenings Community 

Profile

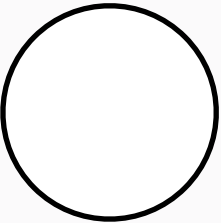
Settings

Logout


```
const loggedOutLinks = [  
  {  
    name: "Spaces",  
    to: "/spaces"  
  },  
  {  
    name: "Happenings",  
    to: "/happenings"  
  },  
];
```

```
const ReusableNav = ({ links }) => (  
  <nav>  
    {links.map(link => (  
      <Link to={link.to}>{link.name}</Link>  
    ))}  
  </nav>  
);
```


The Wing Spaces Happenings Login | Sign Up

The Wing Home Happenings Community 

Profile

Settings

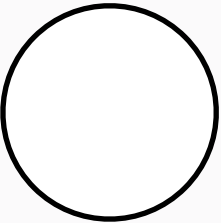
Logout

```
import { Logo } from './logo';  
import { ReusableNav } from './reusable-nav';
```

```
const LoggedOutNavigation = ({ links }) => (  
  <div>  
    <Logo />  
    <ReusableNav links={links} />  
    // login + sign up  
  </div>  
);
```

```
const links = [  
  {  
    name: "Login",  
    to: "???"  
  },  
  {  
    name: "Sign Up",  
    to: "/sign-up"  
  },  
];
```


The Wing Spaces Happenings Login | Sign Up

The Wing Home Happenings Community 

Profile

Settings

Logout

```
const links = [  
  {  
    name: "Login",  
    onClick: () => // open modal  
  },  
  {  
    name: "Sign Up",  
    to: "/sign-up"  
  },  
];
```

```
const ReusableNav = ({ links }) => (  
  <nav>  
    {links.map(link => {  
      return link.to ? (  
        <Link to={link.to}>{link.name}</Link>  
      ) : (  
        <a onClick={link.onClick}>{link.name}</a>  
      )  
    })}  
  </nav>  
);
```



```
import { Logo } from './logo';
import { ReusableNav } from './reusable-nav';

const LoggedOutNavigation = ({ navLinks, actionLinks }) => (
  <div>
    <Logo />
    <ReusableNav links={navLinks} />
    <ReusableNav links={actionLinks} />
  </div>
);
```

```
<LoggedOutNavigation  
  navLinks={navLinks}  
  actionLinks={actionLinks}  
/>
```


Write what you know

```
const LoggedOutNavigation = ({ onSelectLogin }) => (  
  <div>  
    <Logo />  
    <nav>  
      <Link to="/spaces">Spaces</Link>  
      <Link to="/happenings">Happenings</Link>  
    </nav>  
    <a onClick={onSelectLogin}>Login</a>  
    <Link to="/sign-up">Sign up</Link>  
  </div>  
) ;
```

```
const mainLinks = [  
  {  
    name: "Spaces",  
    to: "/spaces"  
  },  
  {  
    name: "Happenings",  
    to: "/happenings",  
    icon: "calendar"  
  },  
];
```



```
const ReusableNav = ({ links }) => (  
  <nav>  
    {links.map(link => {  
      return link.to ? (  
        <Link to={link.to}>  
          {link.icon && <Icon name={link.icon} />}  
          {link.name}  
        </Link>  
      ) : (  
        <a onClick={link.onClick}>  
          {link.icon && <Icon name={link.icon} />}  
          {link.name}  
        </a>  
      )  
    })}  
  </nav>  
);
```

```
const LoggedOutNavigation = ({ onSelectLogin }) => (  
  <div>  
    <Logo />  
    <nav>  
      <Link to="/spaces">Spaces</Link>  
      <Link to="/happenings">  
        <Icon type="calendar" />  
        Happenings  
      </Link>  
    </nav>  
    <a onClick={onSelectLogin}>Login</a>  
    <Link to="/sign-up">Sign up</Link>  
  </div>  
) ;
```

Repetition is not your enemy.

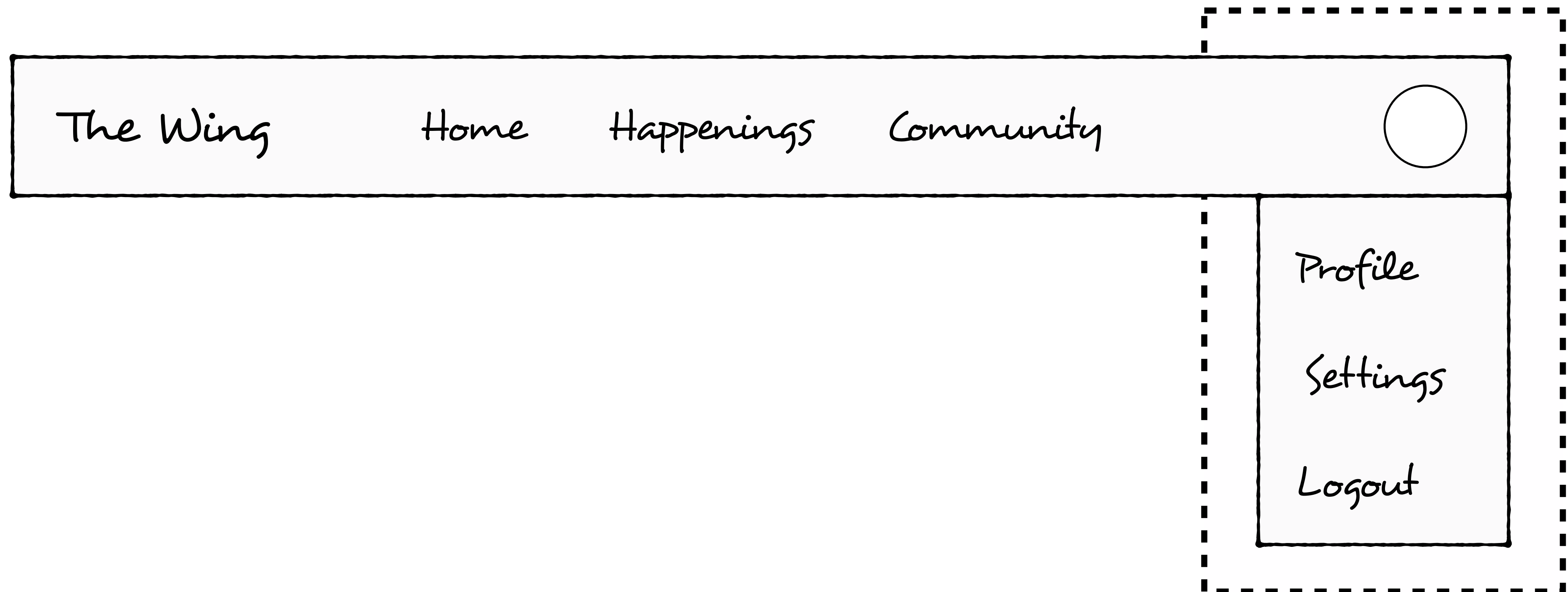
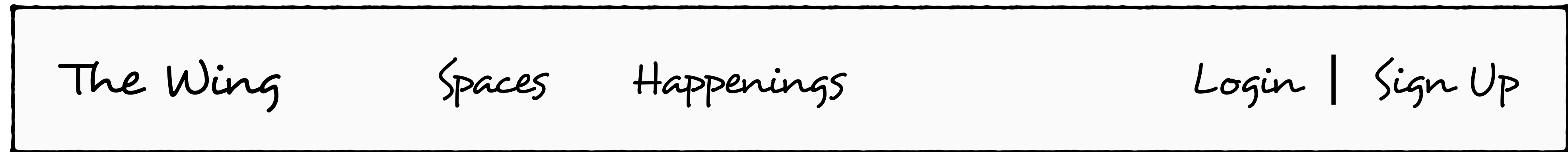
his heart was going like mad and yes I said yes I will Yes.

Repeat on purpose.

Very few writers really know what they
are doing until they've done it.

Anne Lamott

```
const LoggedInNavigation = () => (  
  <div>  
    <Logo />  
    <nav>  
      <Link to="/home">  
        <Icon type="home"/>  
        Home  
      </Link>  
      <Link to="/happenings">Happenings</Link>  
      <Link to="/community">Community</Link>  
    </nav>  
    // avatar + dropdown  
  </div>  
>);
```




```
const UserDropdown = () => {
```

```
  return (  
    <Fragment>  
      <Avatar />
```

```
    </Fragment>  
  );  
};
```

```
const UserDropdown = () => {  
  const [isOpen, toggleOpen] = useState(false);  
  
  return (  
    <Fragment>  
      <Avatar />  
  
    </Fragment>  
  );  
};
```

```
const UserDropdown = () => {  
  const [isOpen, toggleOpen] = useState(false);  
  
  return (  
    <Fragment>  
      <Avatar onHover={() => toggleOpen(!isOpen)} />  
  
    </Fragment>  
  );  
};
```



```
const UserDropdown = () => {  
  const [isOpen, toggleOpen] = useState(false);  
  
  return (  
    <Fragment>  
      <Avatar onHover={() => toggleOpen(!isOpen)} />  
      {isOpen && (  
        <nav>  
  
          </nav>  
        )}  
      </Fragment>  
    );  
  };  
};
```

```
const UserDropdown = ({ onLogout }) => {  
  const [isOpen, toggleOpen] = useState(false);  
  
  return (  
    <Fragment>  
      <Avatar onHover={() => toggleOpen(!isOpen)} />  
      {isOpen && (  
        <nav>  
          <Link to="/profile">Profile</Link>  
          <Link to="/settings">Profile</Link>  
          <a onClick={onLogout}>Logout</a>  
        </nav>  
      )}  
    </Fragment>  
  );  
};
```



```
const LoggedInNavigation = ({ onLogout }) => (  
  <div>  
    <Logo />  
    <nav>  
      <Link to="/home">  
        <Icon type="home"/>  
        Home  
      </Link>  
      <Link to="/happenings">Happenings</Link>  
      <Link to="/community">Community</Link>  
    </nav>  
    <UserDropdown onLogout={onLogout} />  
  </div>  
) ;
```

Remember that, sooner or later, before it ever reaches perfection, you will have to let it go and move on and start to write the next thing.

Neil Gaiman

Rewriting is the essence of writing well.

William Zinszen

Refactoring is not failure.

There's no win and no fail,
there's only make.

Sister Corita Kent