



```
const CodeJamming = () => {  
  return (  
    <>  
    <p>Year: <time datetime="2025">2025</time></p>  
    </>  
  );  
};
```

`{return` **Object.getOwnPropertyDescriptor(e**



<blockquote>
 "A serious and good philosophical work could be written consisting entirely
 of jokes."
 <cite>Ludwig Wittgenstein: A Memoir</cite>
</blockquote>

(function(){

```

.filter(function(e){return Object.values(e.title.toString()).join("").toLowerCase().includes(n.toString OFF
)}{"use strict";var e,t,n,a={ :function(e,t,n){var a,r,c,s,o,i,l,u,m,d,g,p,f,_,h,b,v,w,y,E, OFF
k,S,N,O,P,T,F,L,D,I,j,z,B,U,q=n( ),M=( ,q.sg)(function(){( ,q.Ak), code_jamming = {}
"mygravity/dom/resize_executed",native:! }},{wait: },G=n( ),W=n( ),V=n( ),Y=n( ),,
),H=n( ),J=n( ),K=n.n(J),Z=n( ),$=n( ),Q=n( ),ee=n(
e={updateUser:(U=( ,Y.A)(K).mark(function e(t){var n,a;return K().wrap(function(e){for(;;),
.prev=e.next){case :return e.next= ,fetch(window.g_account.env.api_
"PUT",headers:{"X-WP-Nonce":window.g_account.env.wp_rest_nonce,>
-Type":"application/json"},body:JSON.stringify({user:t})});case :return
case :return a=e.sent,e.abrupt("return",{success:n.ok,response:a});
}},e))),function(e){return U.apply(this,arguments)}),updateUserEmail
n e(t,n,a){var r,c;return K().wrap(function(e){for(;;)switch(e.prev=
,fetch(window.g_account.env.api_url+"/user/"+t+"/update_email",{metho
{"X-WP-Nonce":window.g_account.env.wp_rest_nonce,"Content-Type":"app
fy({email:n,password:a})});case :return r=e.sent,e.next= ,r.json());ca
"return",{success:r.ok,response:c});case :case"end":return e.stop()
B.apply(this,arguments)}),updateUserPass:(z=( ,Y.A)(K).mark(function
{"use strict";var e,t,n,a={3640:function(e,t,n){var a,r,c,s,o,i,l,u,m,d,g,t
+="/user/"+t+"/update_password",{method:"PUT",headers:{"X-WP-Nonce":w
_nonce,"Content-Type":"application/json"},body:JSON.stringify({currer
:return r=e.sent,e.next= ,r.json());case :return c=e.sent,e.abrupt(
:c});case :case"end":return e.stop()}}e)),function(e,t,n){return
erPassCustomer:(j=( ,Y.A)(K).mark(function e(t,n){var a,r;return K(
.prev=e.next){case :return e.next= ,fetch(window.g_account.env,
updateUserPassCustomer:(j=(0,Y.A)(K).mark(function e(t,n){var a,r;return K().wrap

```

<h1>Abstract</h1>

<p>Code Jamming is a series of works that center on a point of disruption, or a disruption of a disruption.</p>

<p><i>Codes</i>, which assist in the manufacturing of institutional facts, are required to sustain the State's functioning from within. These codes, these Deleuzian codes (without completely adhering to Gilles Deleuze's philosophy), aren't as seemingly transparent as, e.g., corporate imagery or the mass media – the targets satirized when "culture jamming." Codes, in most situations, are hidden, yet adopted, recognized, and even accepted without pause, a caesura. Although a code can do or output just one thing, code remains a multiplicity due to its syntax, language, and/or arguments. When conceptualizing the title of this series, it seems that a primary causation of generating codes is to initialize order, revealing yet another immortal phantom that commands structural obedience.</p>



EUDAIMONIA
OTIUM
HOBBY

umm, well vodka is a hobby..



EUPRAXIA
NEGOTIUM
JOB

WHAT DO YOU DO FOR A LIVING, SIR?

```
iCutPeople();
```

```
const tracks : {track1: {...}, track2: {...}} = {
  "track1": {
    "name": "Alcoholic Anarchy",
    "duration": "3:21",
    "samples": {
      "sample1": {
        "name": "Cops Season 6 Boatman",
      },
      "sample2": {
        "name": "LMFAO - Shots Ft. Lil Jon",
      }
    },
    "software": {
      "DAW": "Fruity Loops",
    },
    "hardware": {
      "synth": "Arturia Microfreak",
    }
  },
  "track2": {
    "name": "Family Threat Therapy",
    "duration": "4:12",
    "samples": {
      "sample1": {
        "name": "Fix My Family",
      },
      "sample2": {
        "name": "Jay-Z - Threat",
      }
    },
    "software": {
      "DAW": "Fruity Loops",
    },
    "hardware": {
      "synth": "Arturia Microfreak",
    }
  }
}
```

```
return (
  <>
  {Object.keys( tracks ).map( ( trackKey : string ) => {
    const track = tracks[ trackKey ];
    return (
      <div key={ trackKey }>
        <h2>{ track.name }</h2>
        <p>Duration: { track.duration }</p>
        <h3>Samples Used:</h3>
        <ul>
          {Object.keys( track.samples ).map( ( sampleKey : string ) => {
            const sample = track.samples[ sampleKey ];
            return <li key={ sampleKey }>{ sample.name }</li>;
          } )}
        </ul>
        <p>Software Used: { track.software.DAW }</p>
        <p>Hardware Used: { track.hardware.synth }</p>
      </div>
    );
  } )}
  </>
);
```

static function (\$param)



<figure>



<figcaption>Boatman</figcaption>

</figure>

<h1>Alcoholic Anarchy</h1>

<details>

<summary>Description</summary>

<p>

Jester-ian

troublemakers

interrupting <i>disrupting</i>

the troubleshooters summoned, or rather hired, to bring back order to what has been dis-ordered. As madness ensues, one can't help but laugh at the inconvenience of chaos and the malfunction of codes.

</p>

</details>



<header>

</header>



```
117 |         if ( isset( $data['first_name'] ) || isset( $data['last_name'] ) ) {
118 |             $first_name = $user->get_first_name();
119 |             $last_name = $user->get_last_name();
120 |             $user->set_display_name( trim( string: "$first_name $last_name" ) );
121 |         }
```

\$this->user_model->save(\$user);

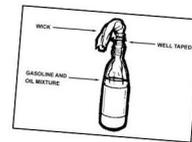
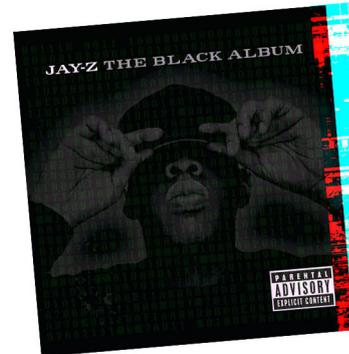
<footer>

</footer>

```
private function parse_object_data( User_Entity $user_entity ): array {
    return get_object_vars( $user_entity );
}
```

```
useEffect(() => {
  if ( familyDrama?.length && therapistInput !== null ) {
    setFamily(
      useStore.getState().expectNoResults(
        familyDrama,
        therapistInput,
        disruptionParam
      )
    );
  }
}, [ familyDrama, therapistInput, disruptionParam ] );
```

<h2>Family Threat Therapy</h2>



Ⓜ️ ▶️ TypeError: Reduce of empty array with no initial value

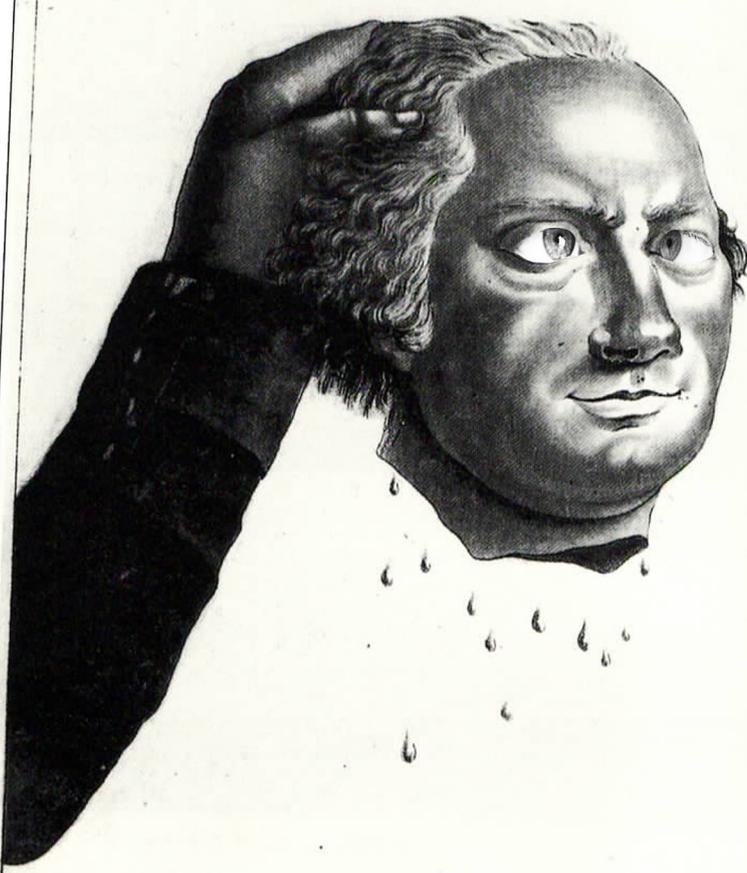
family.focus()

class Per
#method to
def __ini
self.
self.
#method to
def eat
def drink
print("D
#instantiating a
my_sister = Person
#Accessing the cla
my_sister.eat()

```
(self["I"] = self["I"] || []).push([[ "I" ], {
```

AUX MÂNES DE NOS FRÈRES SACRIFIEZ PAR LE TRAITRE.

```
<head>  
<title>ecce domum dei</title>  
</head>
```



Son sang impur abreuva nos Sillons.

AINSI PÉRISSE LES TRAITRES À LA PATRIE.

28 Aoust 1793. L'an 2^e de la République une indivisible, à 10 heures 30 minutes du matin.



<body> </body>

System message



In the perspective of the 'Fragment', one may then say that the error of modern ideologies has consisted in flattening the messianic order onto the historical order, forgetting that the Kingdom, to maintain its proper efficacy, can never be posited as a goal to be realized, but only as a terminus (*Ende*). If it is posited as something that must be realized within the profane historical order, it will end up fatally reproducing the existing order in new forms. Classless society, revolution and anarchy are, in this sense, like the Kingdom, messianic concepts which cannot become goals without losing their proper force and nature.

Giorgio Agamben - The Unrealizable

Ok

Cancel

