crush.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
   <head>
      <meta charset="utf-8">
      <title>My App</title>
      <meta name="viewport" content="width=device-width,initial-scale=1">
      <link rel="stylesheet" href="https://bootswatch.com/4/litera/bootstrap.min.css" type="text/css">
      <link rel="stylesheet" href="assets/css/basic.css"/>
      <script src="assets/js/lib.js"></script>
      <script src="index.js"></script>
   </head>
   <body>
    <main>
      <h2> How Compatible Are You and Your Crush </h2>
      <br>
      <h5> Your Name </h5>
      <input type="text" class="form-control" id="name"></input>
      <br>
      <h5> Your Crush </h5>
      <input type="text" class="form-control" id="crush"></input>
      <br>
      <div style="width: 100%; text-align: center; margin-bottom: 3rem;">
        <button type="button" class="btn btn-success" style="width: 4rem; font-size: 1rem;" onclick="buttonClicke
      </div>

      <h5> Result </h5>
      <div id="output">
      </div>
    </main>
   </body>
</html>
```

game.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
   <head>
      <meta charset="utf-8">
      <title>My App</title>
      <meta name="viewport" content="width=device-width,initial-scale=1">
      <link rel="stylesheet" href="https://bootswatch.com/4/litera/bootstrap.min.css" type="text/css">
      <link rel="stylesheet" href="assets/css/basic.css"/>
      <script src="assets/js/lib.js"></script>
      <script src="game.js"></script>
   </head>
   <body>
    <main>
      <h2> Play a text based Adventure </h2>
      <br>
      <div id="output" style="height: 50vh">
        <h5> You awake in a dark room </h5>
      </div><br>
      <input type="text" class="form-control" id="text-input"></input>
      <br>
      <div style="width: 100%; text-align: center; margin-bottom: 3rem;">
        <button type="button" class="btn btn-success" style="width: 4rem; font-size: 1rem;" onclick="buttonClicke
      </div>
    </main>
   </body>
</html>
```

objects.js

```
function canDrinkFunction() {
  return this.age >= 18;
}

let zain = {
  name: "zain",
  age: 3,
  canDrink: ()=> {
    this.age >= 18;

  };
};

function Person(name) {
  this.name = name;
  this.canDrink = function () {
    return this.age >= 18;
  }
}

zain = new Person("zain");


class Person {

  constructor (name) {
    this.name = name;
  }
  canDrink() {

  }
}

zain = Person("zain");
```

crush.js

```
/* const let var nothing
 * loop types
 * concept of undefined with functions
 * try and detect when names not provided
 *        name == false
 */
function buttonClicked() {
  let name = Library.getInput("name");
  let crush = Library.getInput("crush");
  if (name == false) {
    Library.print("Enter in some names!");
    return;
  }

  let a = crush.split(",");
  let i = 0;
  for(name of a) {
    console.log(name);
  }
}

function getPercentageChance() {
  x = Math.random()*100;
  return x;
}

function printOutMessage(n,a) {
  if(a < 50) {
    Library.print(n+": Not gonna happen");
  } else {
    Library.print(n+": True Love!");
  }
}
```

game.js

```
// anon function this (get item + open bag)
class Player {
  constructor () {
    this.level = 100;
    this.position = {
      x: 0,
      y: 0
    }
    this.inventory = [
      {
        name: "match",
        minLevel: 10
      }
    ]
  }

}

const player = new Player();

function buttonClicked() {
    let input = Library.getInput("text-input");
    try {
      let requestItem = input.match(/use (\w+)/)[1];
      let validItems = player.inventory.filter((e)=>player.level >= e.minLevel);
      console.log(validItems);
    } catch(err) {

    }

}
```

lib.js

```
document.addEventListener('DOMContentLoaded', () => {
  const output = document.getElementById('output');

  Library = {  /* eslint-disable-line */
    getJSON(path) {
      return fetch(path).then(response => response.json());
    },

    print(msg) {
      const text = document.createElement('h5');
      text.innerText = msg;
      output.appendChild(text);
    },

    printLight(msg, top) {
      const text = document.createElement('p');
      text.className = 'lead';
      text.innerText = msg;

      /* is prepend standard? */
      if (top) {
        output.prepend(text);
      } else {
        output.appendChild(text);
      }
    },

    clearOutput() {
      document.getElementById('output').innerHTML = '';
    },
    getInput(i) {
      return document.getElementById(i).value;
    }
  };
});
```

this.js

```
a = "global";

let test = {
  a: "object",
  func: function() {
    return this.a
  }
};

function test2() {
    return this.a;
}

function test3() {
  let a = "nested";
  function test4() {
    a = this.a;
  }
  test4();
  return a;
}

console.log(test.func());
console.log(test2());
console.log(test2.call({a:"context"}));
console.log(test3());

// __proto__

// new f()  is
let o = f.prototype;
f().call(o)
return o;

// how prototype works
```