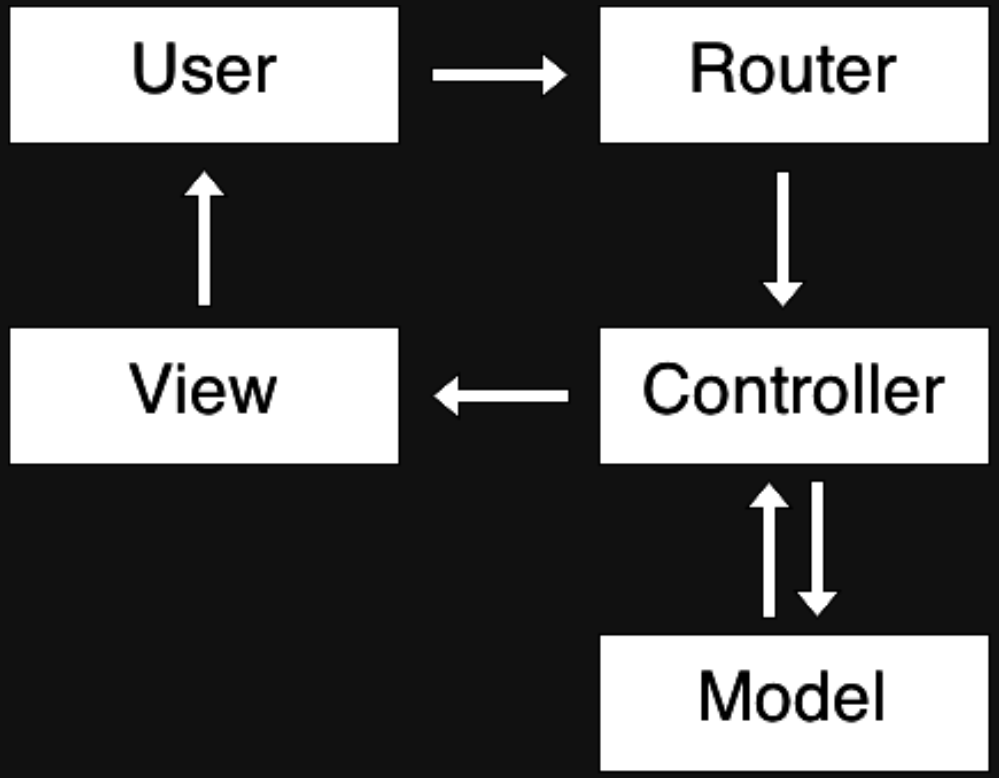
# MVC JavaScript

# MVC



# Let's build this:

http://euglazer.github.io/raindrops/

*and let's be organized and sensible about it*

# HTML

```
<!DOCTYPE html>

<head>
  <link rel="stylesheet" type="text/css" href="./public/stylesheets/index.css">
  <script type="text/javascript" src="./app/models/model.js"></script>
  <script type="text/javascript" src="./app/views/view.js"></script>
  <script type="text/javascript" src="./app/controllers/controller.js"></script
  <script type="text/javascript" src="./public/javascripts/router.js"></script>
</head>

<body>
  <div></div>
</body>
```

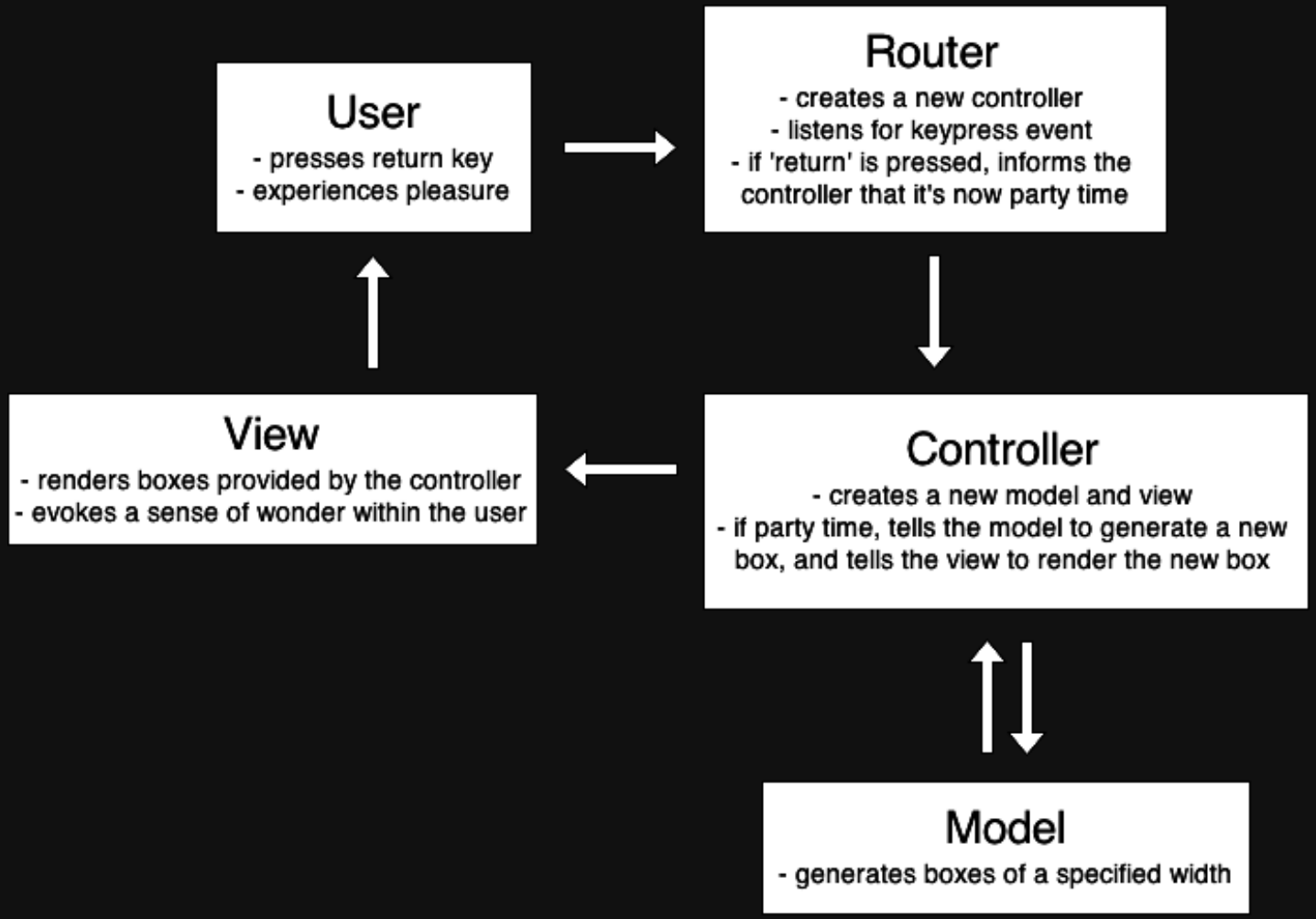
# CSS

```
div {
  height: 10px;
  width: 10px;
  border: 25px dashed black;
}
```

```
├── app
│   ├── controllers
│   │   └── controller.js
│   ├── models
│   │   └── model.js
│   └── views
│       └── view.js
├── index.html
└── public
    ├── javascripts
    │   └── router.js
    └── stylesheets
        └── index.css
```

# JS

```
¯\_(ツ)_/¯
```


User
- presses return key
- experiences pleasure
Router
- creates a new controller
- listens for keypress event
- if 'return' is pressed, informs the controller that it's now party time
View
- renders boxes provided by the controller
- evokes a sense of wonder within the user
Controller
- creates a new model and view
- if party time, tells the model to generate a new box, and tells the view to render the new box
Model
- generates boxes of a specified width

# Router

```
function bindListeners() {
  var controller = new Controller();
  document.body.onkeyup = function(e) {
    if(e.keyCode === 13) {
      controller.partyTime();
    }
  }
}

function ready(fn) {
  if (document.readyState != 'loading'){
    fn();
  } else {
    document.addEventListener('DOMContentLoaded', fn);
  }
};

ready(bindListeners);

/* THE JQUERY WAY
$(document).ready(function() {
  $('body').on('keyup', function(e) {
    var controller = new Controller();
    if(e.keyCode === 13) {
      controller.partyTime();
    }
  });
});
*/
```

# Controller

```
function Controller() {

  var view = new View();    // creates new view
  var model = new Model(); // creates new model
  var size = 10;           // size initialized at 10

  var initialize = function() {
    view.clearScreen();
  };

  var stopIfDone = function(interval) {
    if (size > 1500) {
      clearInterval(interval);
    }
  };

  this.partyTime = function() {
    initialize();                          // tells view to clear the screen
    var cycle = setInterval(function() {  // starts interval
      var newBox = model.generateBox(size);      // tells model to generate a box with specifi
      view.addBox(newBox);                   // tells view to render the box
      view.scrollDown();                     // tells the view to scroll downwards
      size += 2;                             // increments size
      stopIfDone(cycle);                     // if size too large, stops interval
    },15);                                  // interval is repeated every 15 ms
  };

}
```

# Controller

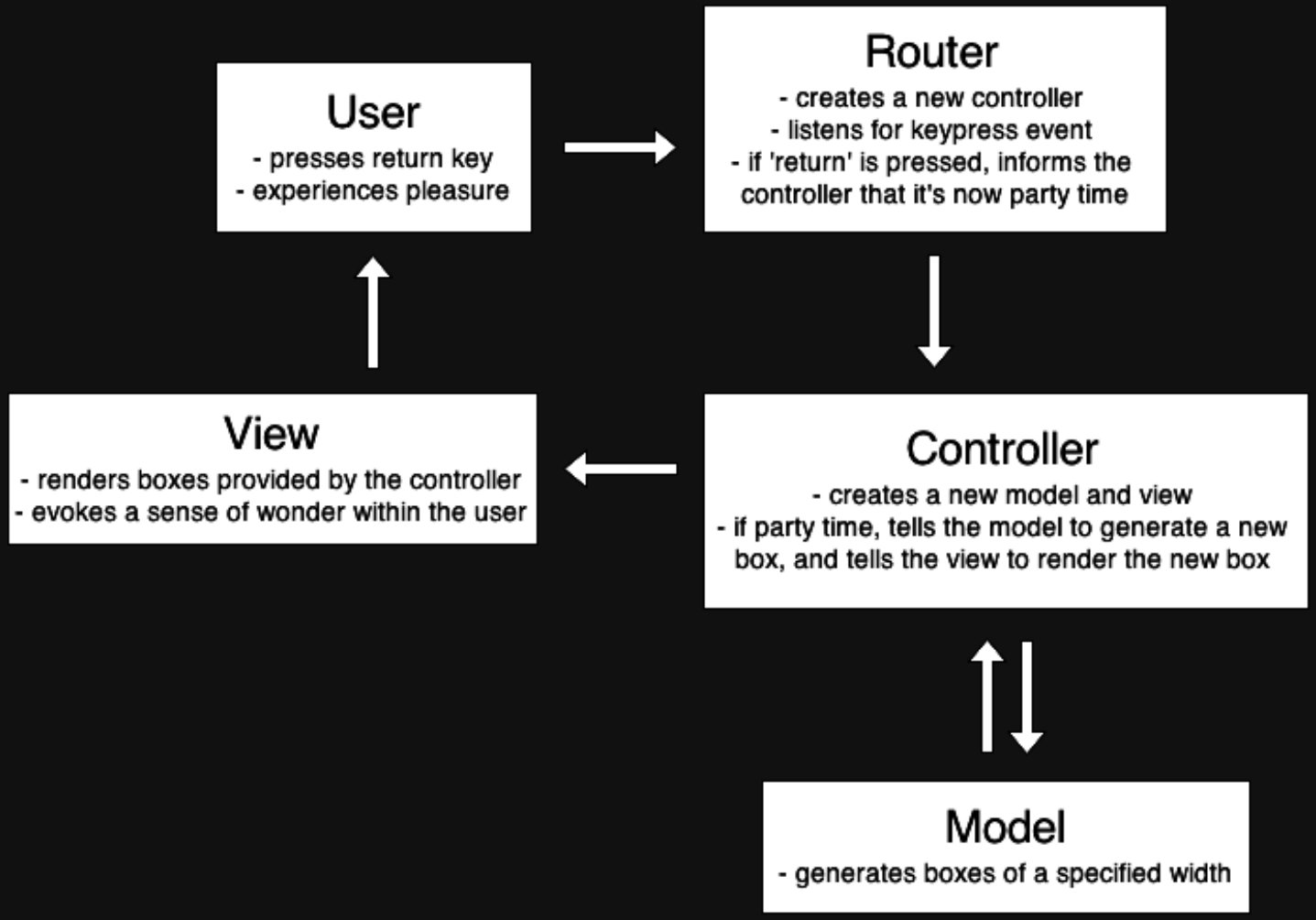
```
function Controller() {

  var view = new View();
  var model = new Model();
  var size = 10;

  var initialize = function() {
    view.clearScreen();
  };

  var stopIfDone = function(interval) {
    if (size > 1500) {
      clearInterval(interval);
    }
  };

  this.partyTime = function() {
    initialize();
    var cycle = setInterval(function() {
      var newBox = model.generateBox(size
      view.addBox(newBox);
      view.scrollDown();
      size += 2;
      stopIfDone(cycle);
    },15);
  };

}
```

# View

```
function View() {

  var seedBox = document.body.children[0];
  var largestBox = seedBox;

  this.clearScreen = function() {
    seedBox.innerHTML = '';
  };

  this.addBox = function(box) {
    largestBox.appendChild(box);
    largestBox = box;
  };

  this.scrollDown = function() {
    window.scrollTo(document.body.scrollHeight,docum
  };

}
```

# Model

```
function Model() {

  this.generateBox = function(size) {
    var div = document.createElement('DIV');
    div.style.height = size + 'px';
    div.style.width = size + 'px';
    return div;
  };

}
```




http://euglazer.github.io/raindrops/