

Cheat Sheet

Flex

display: flex
justify-content: flex-start flex-end center space-between space-around
align-items: flex-start flex-end center
flex-direction: column

Grid

display: grid
grid-template-rows: **width**
grid-template-columns: **height**

Javascript

parseInt(**value**)
parseFloat(**value**)
Math.floor(**value**)
Math.random()
s.split(“ “)

Styling With JS

element.style.**propertyNameCamelCase** =
element.className =
window.getComputedStyle(**element**).**propertyNameCamelCase**
element.classList.add(**className**)

Timers

setTimeout(**function**, **delay**)
setInterval(**function**, **delay**)
clearTimeout(**timerID**)
clearInterval(**timerID**)

DOM manipulation

Document.createElement(**type**)
document.createTextNode(“**text**”)
element.appendChild(**node**)
element.insertBefore(**new**, **old**)
element.removeChild(**node**)
element.replaceChild(**new**, **old**)
childNodesjknndtey

Canvas

HTML: <canvas width=**width** height=**height**></canvas>
var canvas = document.getElementById(**id**);
var ctx = canvas.getContext(“2d”);
setting colors: ctx.fillStyle = **colorString**;
 ctx.strokeStyle = **colorString**;
rectangle: ctx.fillRect(**x**, **y**, **width**, **height**);
circle: ctx.beginPath();
 ctx.arc(**x**, **y**, **radius**, **startAngle**, **endAngle**);
 ctx.stroke();
line: ctx.moveTo(**startX**, **startY**);
 ctx.lineTo(**endX**, **endY**); moveTo(**x**, **y**);
 ctx.stroke();
animation: window.requestAnimationFrame(**function**);
gradient: context.createLinearGradient(**x**, **y**, **width**, **height**);
 gradient.addColorStop(**start**, **color**);
pixel data: context.getImageData(**x**, **y**, **width**, **height**);

Mouse Event Properties

clientX
clientY
screenX
screenY
button

Ajax request:

```
var url = <URL_HERE>;
fetch(url)
  .then(checkStatus)
  .then(function(responseText) {
    // deal with responseText
  })
  .catch(function(error) {
  });
```

Parse JSON:

```
JSON.parse(<JSON_STRING>)
```

Handling a GET request:

```
app.get('/', function (req, res) { })
```

Handling a POST request:

```
app.post('/', jsonParser, function (req, res) { })
```

Accessing GET parameters:

```
req.query.<PARAMETER_NAME>
```

Accessing POST parameters:

```
req.body.<PARAMETER_NAME>
```

Send response from server:

```
res.send(<TEXT>);
res.json(<JSON>);
```

Read file:

```
fs.readFileSync(<FILE_NAME>, 'utf8');
```

All Files in a directory:

```
fs.readdirSync(<DIRECTORY>);
```

Append to a File:

```
fs.appendFile(<FILE_NAME>, <TEXT>,
function(err) {
  if(err) {
    return console.log(err);
  }
});
```

Get matches using a regex:

```
<STRING>.match(</REGULAR_EXPRESSION>/)
```

POST parameters for Ajax request:

```
const fetchOptions = {
  method : 'POST',
  headers : {
    'Accept': 'application/json',
    'Content-Type' : 'application/json'
  },
  body : JSON.stringify(<POST_PARAMS>)
};
```

Query a database:

```
async function query(collection) {
  var doc = { }; // query
  var result = await collection.find(doc).toArray();
  // do something with result
}
```

Access cookie:

```
req.cookies.<COOKIE_NAME>
```

Set cookie:

```
res.cookie(<COOKIE_NAME>, <VALUE>, {maxAge : <AGE>});
```

Access session variable:

```
req.session.<VARIABLE_NAME>
```

Connect to Database:

```
var con = mysql.createConnection({
  host: host,
  database: database,
  user: username,
  password: password,
  debug: debug
});
con.connect(function(err) {
  if (err) throw err;
  con.query(query,
    function (err, result, fields) {
      if (err) throw err;
      // result[0][fieldname]
    });
});
```