How the Browser Works:

The Browser has 3 main programs:

DOM interpreter -> takes HTML and converts to display in browser

CSS interpreter ->

JS Engine

All three at JIT

The source code is downloaded to the client and then is interpreted on the client into bytecode

The source code is compiled in real time on the user’s computer.

**Web Workers:**

* A means to run scripts in background threads, the worker thread can perform tasks without interfering with the user interface. In addition they can perform I/O using XMLHTTPRequest
* Once created, a worker can send messages to the JS code that created by posting messages to an event handler specified by that code.

*\*works run in another global context that is different from the current window.*

**Limitations**:

* you can’t directly manipulate the DOM from inside the worker,
* or use default methods and props of the window object.

**Advantages**:

* You can use websockets,
* data storage mechanisms like IndexedDB,
* as well asother items.
* Workers can spawn new workers

**Data is sent between workers and the main thread via a message system:**

* Both sides send messages through the *postMessage* method
* Both sides respond to messages through the *onmessage* eventHandler

Example code:

If (window.Worker) {

Var myWorker = new Worker(‘worker.js’);

myWorker.postMessage({a: true});

myWorker.onmessage = function(e) {

console.log(e.data);

};

myWorker.terminate() // worker can also terminate itself by calling close()

}