# Meta Refresh Conference : 16th - 17th April 2015

**Sharing my learning at the meta refresh conference, plus some of the key web technologies used by frontend developers:**

**How JavaScript flow works in browser**

* Compile the JavaScript code
* Generate binary code
* Interpreter interprets the binary code
* Initialize context variable
* Executes

**Why Use JavaScript** - it does below actions on browser

* Template (templating with less memory is the idea need to achieve)
* Animation
* Validation
* Execution

**Event Bubbling - capturing**

* Events are heavy
* Event should be captured on propagation stage
* All the handlers works on bubbling stage, means all the events will be available form targeted item to parent item which is not a good approach. But using addEventListener(type, method, true) we can avoid bubbling stage, which is the only way to capture the event in capturing phase.
* We can avoid bubbling phase using event.cancleBubble=true for IE and event.stopProppagtion() for rest browsers

**Event propagation on Canvas**

* When event fired on the canvas the browser engine initialize the event and observe the event.
* Then call the event
* Inline events handlers are lazy to complie
* Never bind scroll event in Mobile web and apps. Best way is use touch start and touch end functionality.

**PkQuery.js**

* Conceptual framework for data interactivity
* Best for organize complexity around data-driven interactive Single page web/apps

**Component approach for building web application**

* Building application based on component
* Web component consists of 4 technology [Custom element, html imports, templates and Shadow DOM ]
* One of the good library is Google polymer (which follows Atomic design pattern)

**Vulcanize:**

* This library helps in processing web components into one output file.
* Reduces load on html file
* its dependent html imports into a single file

**Sprite.js**

* This is a framework that let user create animation and games using sprites in effective ways.
* This is a common framework can be used across the platform
* Always binds and unbinds as event to the DOM element.

**Bracket:**

* Adobe introduces it as a free open source code editor for web
* Brackets open a live connection to the user's browser when live files preview is enabled.
* Browser shows real time changes to the css class & properties , html markup as user typed.

**Optimizing critical rendering path**

* Analyze web performance using Chrome extension
* Use Chrome web inspector => Time line => Paint to observe the site performance
* Add all relevant css on top inside the style tag
* Then load the rest of the css and javascript using loadCss() and loadJS() functions.