All this reviews are my opinion about the subject they presented, not the authors.

All this reviews are my opinion about the subject they presented, not the authors.

Nevertheless many things are a simple copy of their content :P.

Use CSS next to transpile CSS4 code to CSS3.

Some features

There’s no CSS4, only level 4 modules.

Custom media queries:

```
@custom-media --narrow-window (max-width: 30em);
```

Custom media queries:

```
@custom-media --narrow-window (max-width: 30em);

<script>
CSS.customMedia.set('--foo', 5);
</script>
```

Custom Selectors

```
@custom-selector :--heading h1, h2, h3, h4, h5, h6;
```

Color manipulation

color() //applies zero or more “color adjusters

Custom Selectors

```
@custom-selector :--heading h1, h2, h3, h4, h5, h6;
```

Color manipulation

color() //applies zero or more “color adjusters

hwb() //Hue-Whiteness-Blackness

Custom Selectors

```
@custom-selector :--heading h1, h2, h3, h4, h5, h6;
```

Color manipulation

color() //applies zero or more “color adjusters

hwb() //Hue-Whiteness-Blackness

gray() //

Custom Selectors

```
@custom-selector :--heading h1, h2, h3, h4, h5, h6;
```

Color manipulation

color() //applies zero or more “color adjusters

hwb() //Hue-Whiteness-Blackness

gray() //

Filters

```
#image {
  filter: grayscale(100%);
}
```

Grunt CSSnext

Nice insights about how animations are built for real, in places like Disney.

Nice insights about how animations are built for real, in places like Disney.

Book recommendation: The illusion of the life.

Nice insights about how animations are built for real, in places like Disney.

Book recommendation: The illusion of the life.

WebAnimation specification.

Program imperatively animations know done by CSS.

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}
```

It’s on draft stage by now.

```
elem.getAnimations().filter(
  animation =>
    animation.effect instanceof
    KeyframeEffectReadOnly &&
    animation.effect.getFrames().some(
      frame => frame.hasOwnProperty('transform')
    )
).forEach(animation => {
  animation.currentTime = 0;
  animation.playbackRate = 0.5;
});
```

```
elem.getAnimations().filter(
  animation =>
    animation.effect instanceof
    KeyframeEffectReadOnly &&
    animation.effect.getFrames().some(
      frame => frame.hasOwnProperty('transform')
    )
).forEach(animation => {
  animation.currentTime = 0;
  animation.playbackRate = 0.5;
});
```

Is a specification since 2011, it’s based on OpenGL ES 2.0.

Shaders

It’s all about shaders, there’s many types of them. The main one’s are shaders for vertices and for fragments.

Shaders

It’s all about shaders, there’s many types of them. The main one’s are shaders for vertices and for fragments.

Vertices shaders unity specify where the vertices are located. Shaders don’t create vertices, they only manipulate them.

Shaders

It’s all about shaders, there’s many types of them. The main one’s are shaders for vertices and for fragments.

Vertices shaders unity specify where the vertices are located. Shaders don’t create vertices, they only manipulate them.

Fragments shaders calculate colors on each pixel of the image.

Putting together

Game engines uses this primitives and build abstract concepts that are useful to games.

Example: treejs

Optimizing

Beware: CPU vs GPU optimization.

Optimizing

Beware: CPU vs GPU optimization.

CPU? JavaScript, DOM, etc.

GPU?

- Texture compression
- Mipmaps (LOD)
- Draw calls

GPU?

- Texture compression
- Mipmaps (LOD)
- Draw calls

Use chrome debugger.

WebGL is specified by the Kronos working group, their site is a nice resource about it.

WebGL is specified by the Kronos working group, their site is a nice resource about it.

A good set of videos for introduction on 3D theory for applications development: OnlineMediaTutor : Maya modeling & animation tutorials.

WebGL is specified by the Kronos working group, their site is a nice resource about it.

A good set of videos for introduction on 3D theory for applications development: OnlineMediaTutor : Maya modeling & animation tutorials.

Advanced topics can be found at LearningWebgl as well.

Using WebGL is hard, but one can became a master by practising, maybe downloading a simple demo and tweaking it further, making abstractions, building it's own "framework".

Webvr

Is a JavaScript API for translation of movement information of the device into 3D scene movements.

Is build upon WebGL and 3D foundations.

To see more on MDN.

Mozvr

Tiny useful tips.

Use strict

- For you own sanity
- You can use it inside of functions

Use Call

You can write a method once and then inherit it in another object, without having to rewrite the method for the new object.

```
myfunc.call(anotherObject)
```

Maybe a good trait schema?

Avoid global scope

IIFE - Immediatly invoked function expression

```
(function() {
    'use strict';

    function bar() {
        return 'foo';
    }

    window.bar = bar;
})();

window.bar();
```

Use local scope

```
(function(win) {
    'use strict';

    function bar() {
        return win.url;
    }

    window.bar = bar;
})(window);

window.bar();
```

Use a namespace

```
(function(win) {
    'use strict';

    function bar() {
        return 'foo';
    }

    win.ThiIsACopiedSample =
    win.ThiIsACopiedSample || {};
    win.ThiIsACopiedSample.bar = bar;
})(window);

ThiIsACopiedSample.bar();
```

Array like into array

```
var $links = Array.prototye.slice.call(
    document.querySelectorAll('a')
);

console.log(Array.isArray($links)); //true
```

Real types

```
function test() {
    return typeof arguments;
}
test(); //== Object
```

```
function is(type, object) {
    type = type[0].toUpperCase + type.slice(1);
    return Object.prototype.toString.call(object)
    === '[object ]' +  type + ']';
}
```

Functional on arrays

```
array.forEach(callback, thisArgument);
callback(currentValue, index, array);
```

Functional on arrays

```
array.forEach(callback, thisArgument);
callback(currentValue, index, array);

array.forEach(callback, thisArgument);
callback(currentValue, index, array);
```

Functional on arrays

```
array.forEach(callback, thisArgument);
callback(currentValue, index, array);

array.forEach(callback, thisArgument);
callback(currentValue, index, array);

var isAllEven = array.every(checkEven);
function checkEven(number) {
    return number % 2 === 0;
}
```

```
var hasEven = array.some(checkEven);
function checkEven(number) {
    return number % 2 === 0;
}
```

```
var justEven = array.filter(checkEven);
function checkEven(number) {
    return number % 2 === 0;
}
```

```
var total = array.reduce(sum, 0);
function sum(previous,actual) {
    return previous + actual;
}


//reduceRight
```

When to avoid these functions?

When to avoid these functions?

When you need to break the loop.

Use events!

```
object.addEventListener('click', function(){});
```

Use events!

```
object.addEventListener('click', function(){});
```

is better than . . .

```
object.onclick = function() {

}
```

Nice resources

http://devdocs.io/

http://developer.mozilla.org/

JavaScript: The Definitive Guide

Clean Code: A Handbook of Agile Software Craftsmanship

You don’t need to know all the insane collections of technologies available for frontend nowadays. What matters is that you know about them in a way that, when you need some of them, you will now the tool you need exists, then you can learn and apply them.

What matters in this kind of events is the feeling, that inspiration one's get. You don't need really advanced talks to get it, and certainly will not get it from YouTube.

You may not learn awesome new things, but you will see people like you, struggling with similar problems. You will see their will to overcome these problems and make others life easier. And you see all of it and see the same on yourself, and this serves as an fuel for your technology aspirations.

Thanks.

A review about the event.