

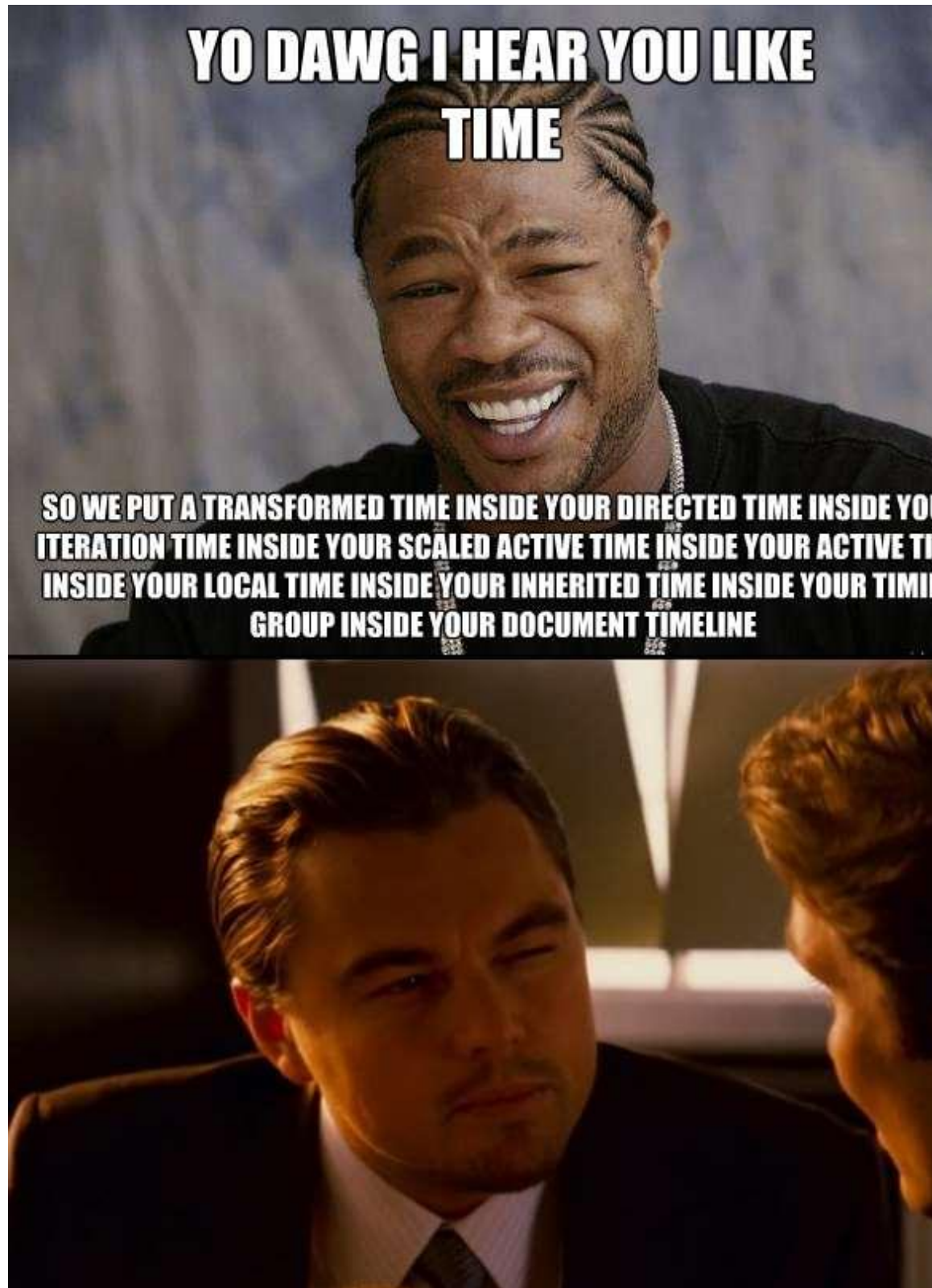
- 17. Overview of the animation model
 - LGTMx4
- 18. Animation effects
 - Effect model values should be updated immediately, not after a sample. The timing model and effect model values should never be inconsistent with each other. This doesn't mean the style itself needs to be updated (unless eg. computedStyle is read).
 - Strengthen the wording for animations without effect to indicate this applies to:
 - animations with no effect
 - animations with an effect with zero target properties
 - animations with an effect with zero animatable target properties
 - 18.1 Target properties
 - Switch sentences 2 & 3 and reword sentence 2 as a result:

“Each animation effect can have zero or more associated target properties. Properties that may be targeted by an animation effect are defined as animatable. The set of animatable properties is defined in section 22. Note that not all properties are animatable.”
 - Note that “properties that may be targeted by an animation effect are defined as animatable” conflicts with “An animation effect that targets a property that is not animatable...”. One of these needs rewording :-D
 - 18.2 Target property types
 - the bolded ‘property types’ does not match the paragraph heading ‘target property types’.
 - “each animatable property defines one or more property types to use when animating. Properties with different types (e.g. color and position) animate differently.” Add an example here too.
 - Every property is interpolable and additive. The default interpolation and addition is pretty boring :)
 - Remove “initial”.
 - 18.3 Animation values
 - We need to think about accumulation and compositing more, but the current approach doesn't seem right. For animate to 50px with an underlying value of 100px, for example, the simple animation value at the end of the first iteration is 50px, and the simple animation value at the beginning is 100px, so the animation value at the beginning of the second iteration is 150px.
 - note that accumulation is not defined for animate to in SVG.
 - maybe we should just not have accumulation and unroll animations from SVG that want to accumulate?
 - or, alternatively if we take the simple animation value at the end of the previous iteration and use `_it_` as the underlying value for the next iteration then things work better.

- Add something like “Animation values don’t represent the result of the animation but are instead intermediate values that are composed with each other and the underlying value of a property (as defined in Section 19) to produce a final result.”
 - 18.3.1 **Simple animation values**
 - A better name would be “iteration animation value” or “unaccumulated animation value”
 - 18.3.2 **Accumulating animation values**
 - Shane thinks paragraphs 1 and 2 are inconsistent with each other because 1 implies that some animation effects have an accumulation behaviour but 2 implies that all animation effects have an accumulation behaviour. Maybe just remove paragraph 1 or make it non-normative.
 - A better name would be “accumulated animation value”
 - “Accumulation behaviour is only applied...” should be removed. Every property is additive.
 - We think the algorithmic definition is clearer. But s/last value/final value/.
- 19. **Combining animations**
 - Consider renaming ‘compositing’ to ‘combining’ (‘reducing’?) so we don’t mix up this operation with something the compositor thread would do. Otherwise switch every occurrence of ‘combining’ to ‘compositing’.
 - CSS Transitions may need to insert somewhere differently to CSS Animations. I don’t think this has been resolved by the CSSWG yet.
 - The caption for figure 26 should indicate whether priority is higher at the top or the bottom of the stack.
 - 19.1 **The animation stack**
 - 1. Reword this, it’s hard to understand at the moment.
 - 3. Should add the zero time, not subtract it. Also, ‘effect’ start time is misleading as this is the player start time.
 - Move step 1, 2, 3 to be setup rather than steps. The procedure actually starts in step 4.
 - With SVG sort ordering, shouldn’t being in the same document as the target make an animation **higher** priority when compared to an animation in a different document?
 - If [yada yada] Animations [yada yada] script: play isn’t called on players, it’s called on the timeline and provided with the Animation. Remove the second paragraph as it’s an implementation detail. Delete the issue, this order is not arbitrary. It’s essential to do it this way to meet author expectations (i.e. stuff earlier in the list applies first).
 - 19.2 **Calculating the result of an animation stack**
 - Don’t use ‘final value’ as it is already used in 18.
 - The last sentence seems to call out the highest priority value as a special case but it isn’t really.

- 19.3 Animation composition
 - The description of 'replace' could be simpler.
 - It's fine to just describe the order of operands without specifically calling out commutativity.
 - The last sentence of 19.3 is false. For example, 45px and 15% are not of the same property type (they are <length> and <percent> respectively).
- 19.4 Combining with other styles: the override stylesheet
 - do we want used values? Used values for transforms are always matrices, aren't they? I think we want specified values when numeric and computed values otherwise. We probably never want to do used values (e.g. can't animate to auto so why would we let you animate from auto?).
 - The override stylesheet needs to be regenerated on every style recalc, not just animation samples.
 - point 4 - 'in such a way that it applies only to the targeted element and property'?
- 20. Keyframe frame animation effects
 - fractional offset isn't a good phrase
 - extrapolating might need a clear definition
 - We should fix the last sentence now that we've decided on a new approach
 - 'overlap' is the wrong word - does this mean when two keyframes have the same offset?
 - 20.1 The simple animation value of a keyframe animation effect
 - Don't need step 3? Or 4?
 - Move step 5 to just before 9? Also make it clearer.
 - Steps 6 and 7 could be merged together.
 - Maybe we don't want to restrict keyframe offsets to [0, 1], and only create a keyframe at 0 if there is no keyframe ≤ 0 . Likewise only create a keyframe at 1 if there is no keyframe ≥ 1
 - As an extension of this, do we even need to create keyframes if we've got a pair? They define a gradient we can use.
 - We should do keyframe creation after selecting before and after keyframes. In fact, all of step 11 should wait until we've discovered before and after keyframes.
 - There's a better way of generating the end keyframe than forcing the start keyframe to have a value less than one in step 14.
 - 16 is wrong.
 - 20.2 Procedure for evenly distributing keyframes
 - The first paragraph is not normative
 - Also (27.14.4) maybe it's better to always distribute keyframes rather than returning an empty list when the keyframes aren't loosely sorted.
- 21. Path animation effects
 - 21.1 The simple animation value of a path animation effect

- 21.2 Calculating the normal to a path
- 26. Timing events
 - Why do you say “Timing events are a property of the Web Animations timing model”. Does it mean that they are generated by the timing model, not the rest?
 - Too many italics
 - Should either move timing events after the IDL section, or add a section discussing IDL-specific issues regarding timing events (e.g. how CSS and SVG events interact with Web Animation timing events).
 - 26.1 Types of timing events
 - instead of ‘this element’, use ‘its child’ for clarity
 - why is there a timingcancel but not a timingplay? Steve wants to call them timingassociate and timingdissociate. Doug wants to call them timingattach and timingdetach.
 - timingnewanimation should never be an event. If you want this, just hook the constructor.
 - timingpause, timingseek, timingdirectionchange should fire on the player, not on the TimedItem (i.e. playerpause, playerseek, playerdirectionchange). Should add playerstart and playerend to that list.
 - Is bubbling up the DOM, and capturing down? If so then these propagate but they don’t bubble. Or do you want them to bubble too?
 - Having a timingpause that fires on the actual TimedItems that were active at the point of pause might be useful, but this sounds hard to do well.
 - If these events are used to generate CSS and SVG events, maybe they should be cancellable?
 - 26.2 Uneased timing
 - Add a comma before ‘called’, otherwise only the timing functions called uneased timing aren’t applied.
 - Lose the analogue :)
 - uneased is not a good name. Natural timing?
 - Why are timing functions a problem when iterations aren’t a problem?
 - We think that it’s more appropriate to control where you get events by choosing to register on the parent rather than the children, and to generate events every time a child starts and stops.



- 26.3 Inverse timing calculations
- 26.4 Event parameters

- providing uneased timing for event local time will be very confusing.
 - is event timeline time just global time?
- 26.5 **Propagation path**
 - As specced propagation goes from root TimedItem to target TimedItem, then back to root TimedItem (because all the events bubble). Is this what you intended? If so then describe the bubbling phase here too.
- 26.6 **Sequence of events**
 - What does it mean for the events to be synchronous? Do you just want the events to be delivered in a defined order or is there more to it than that? The wording in DOM 3 events indicates that the DOM can't change between generation and reception of events - this is probably a Bad Thing. For example, as specified is **impossible to run any animations with events registered on them in another thread**.
 - We probably should do asynchronous event logs like DOM mutation events have become. This will mean that we definitely can't cancel propagation or prevent default actions. We can still guarantee internally consistent orderings.
- 26.7 **Event dispatch**
 - 26.7.1 **Seeked event dispatch**
 - 26.7.2 **Event dispatch and seeking a player**
 - 26.7.3 **Event dispatch and time adjustments**
 - 26.7.4 **Event dispatch and unattached timed items**
 - 26.7.5 **Event dispatch and extended delays**