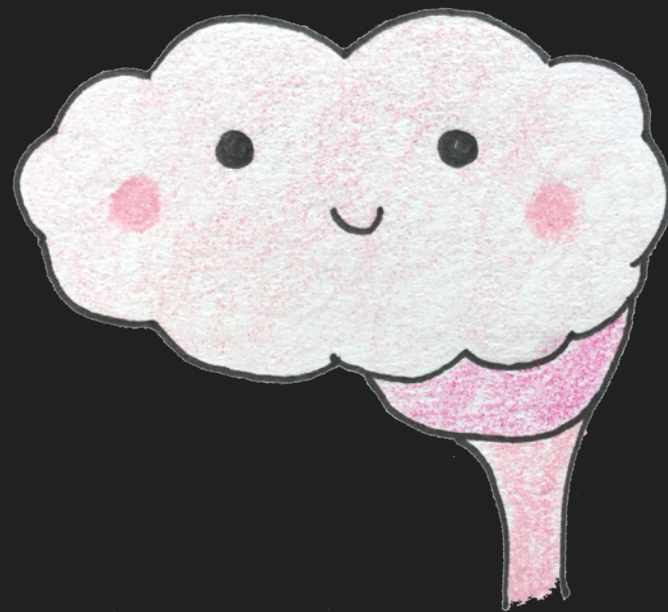
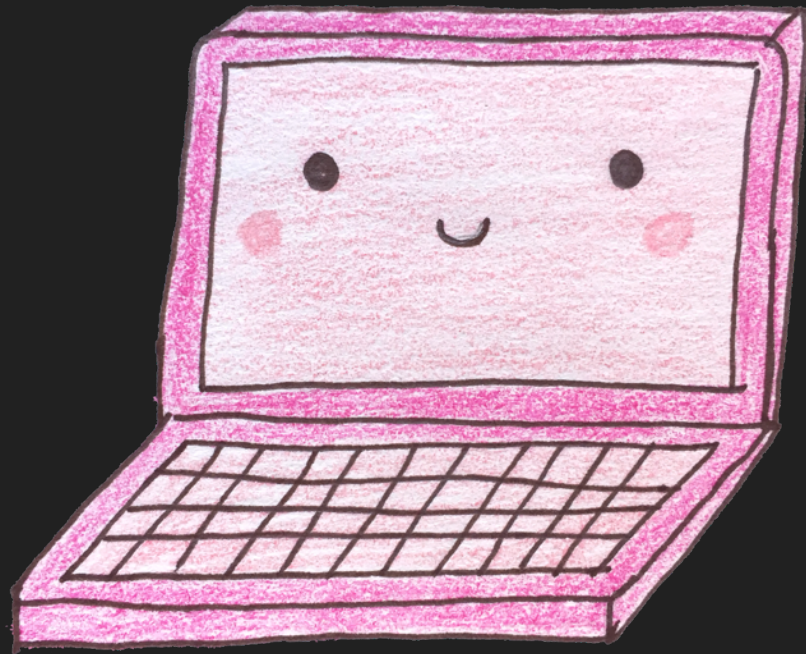


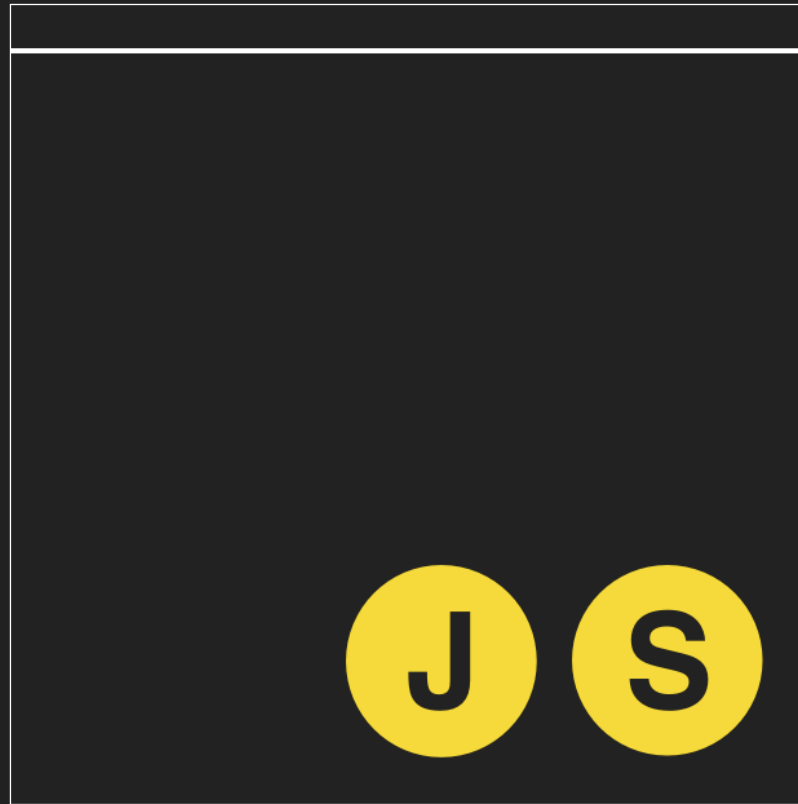
# The Browser and the Brain



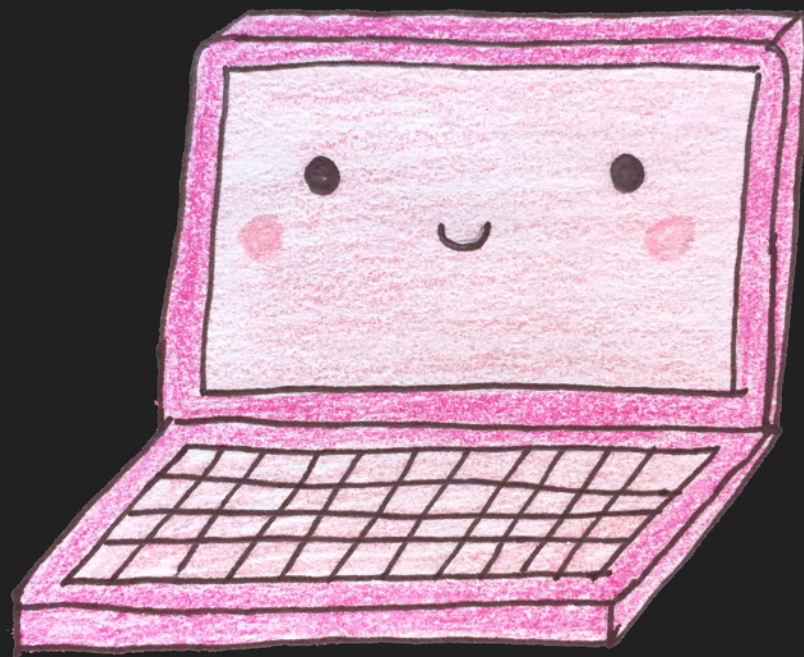
Jenna Zeigen • NationJS • December 1, 2017



jenna.is/at-nationjs



Gee, Brain. What is Jenna going to do in her talk today?

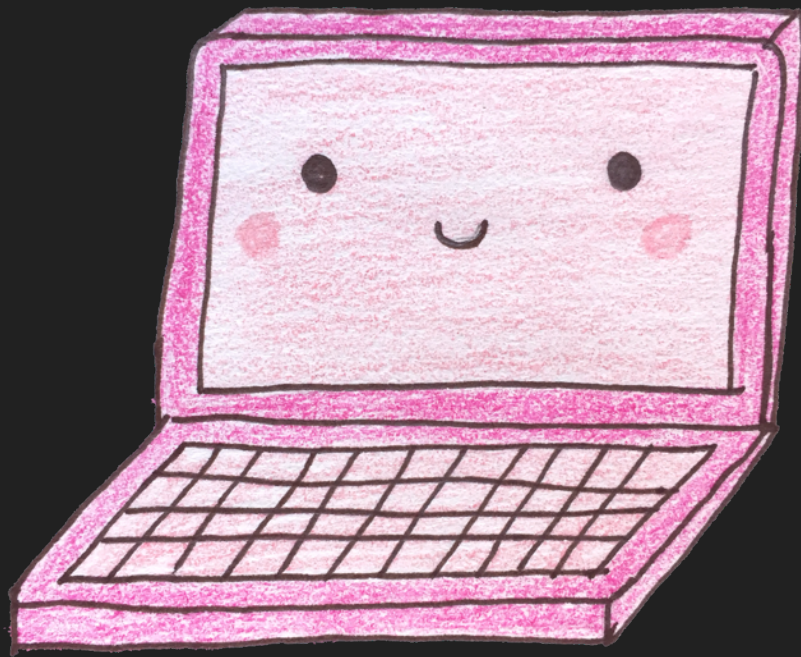


The same thing she does every time, Browser! Talk about cognition and computers!

1. Understanding
2. Visualizing
3. Task Management



Browser, are you processing  
the same way I'm processing?



I don't think so, Brain,  
but let's dig into it!

We made computers so we know all the answers. We do science on humans to find out the answers.

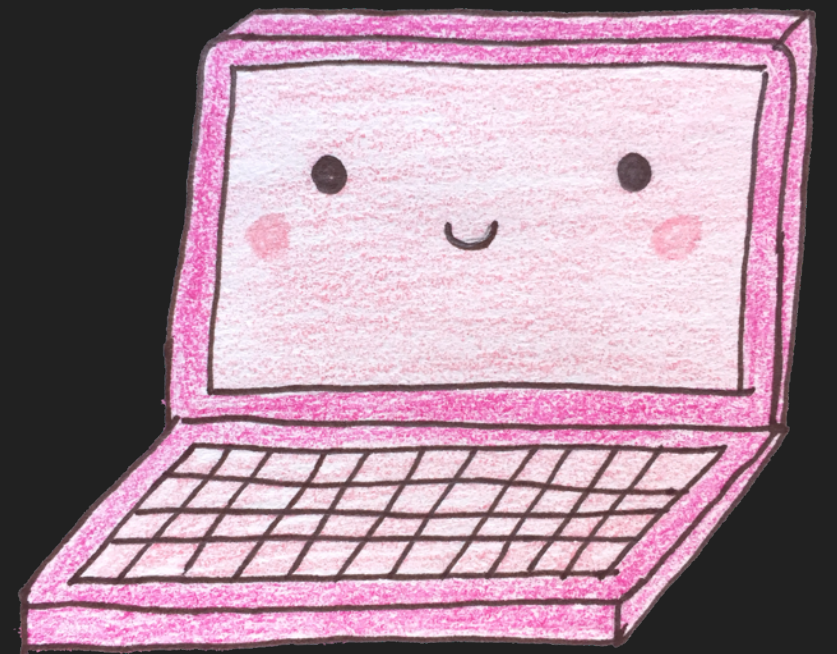
# Understanding



How does the browser process HTML, CSS, and JavaScript, and how does the human mind process natural languages?

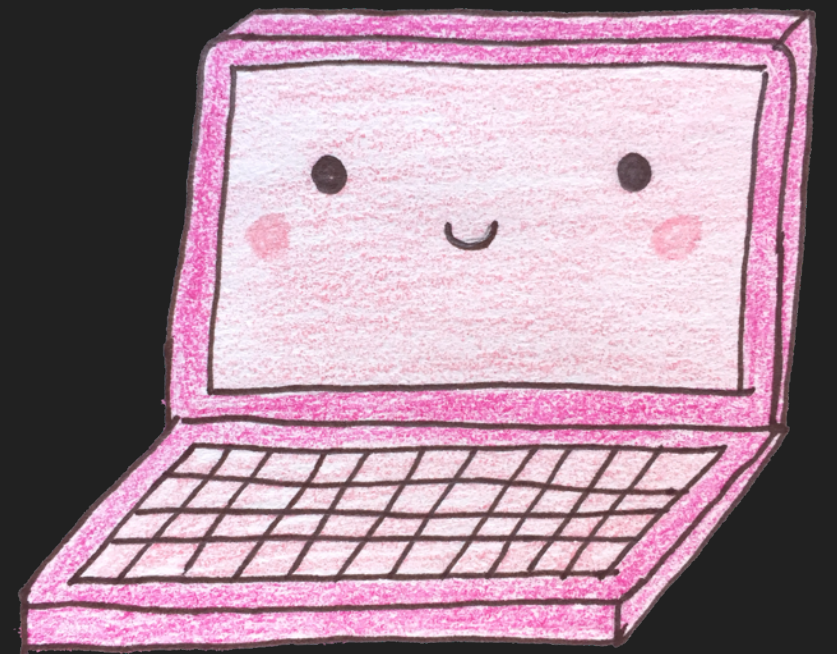
# HTML, CSS, & JS

Most programming languages have a vocabulary described using regular expressions and a syntax described by a context-free grammar



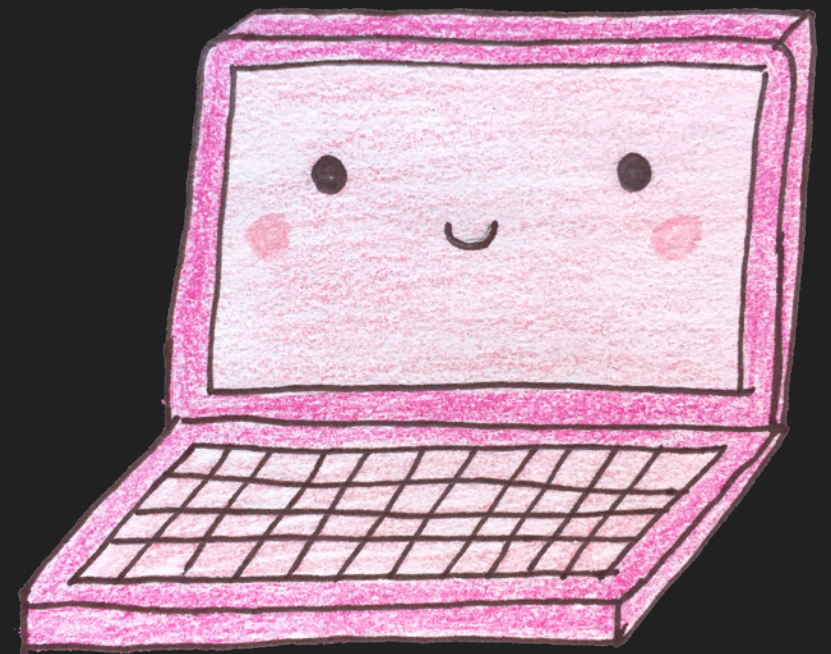
# HTML, CSS, & JS

A parser's job is to take a document and break it into a structure the code can use



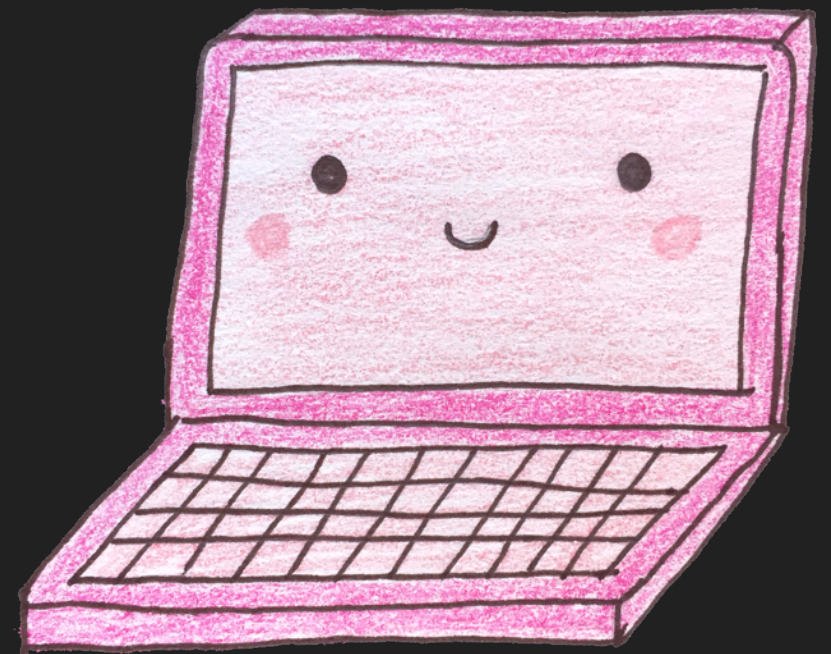
# HTML, CSS, & JS

Parsing can be separated into two parts — lexical and syntactic analysis— which are performed by a lexer and parser, respectively



# HTML, CSS, & JS

HTML isn't a context-free language and therefore can't be parsed by a regular parser





# HTML, CSS, & JS

CSS is a context-free language and therefore easier to parse.



# HTML, CSS, & JS

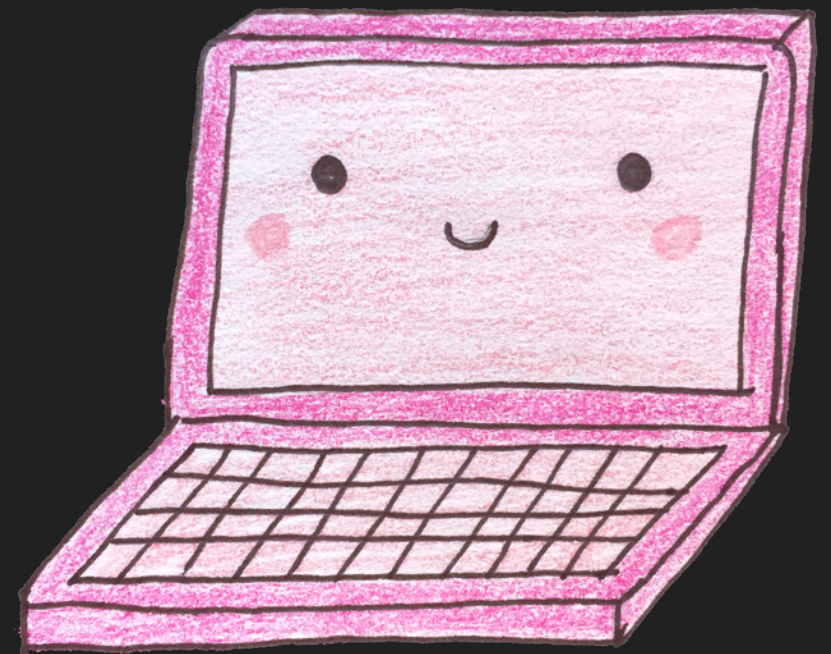
Both HTML and CSS parsers end up creating a tree representing the language it parsed, the DOM and CSSOM trees





# HTML, CSS, & JS

JavaScript is also context-free and can use a regular parser, but browsers complicate things in order to optimize



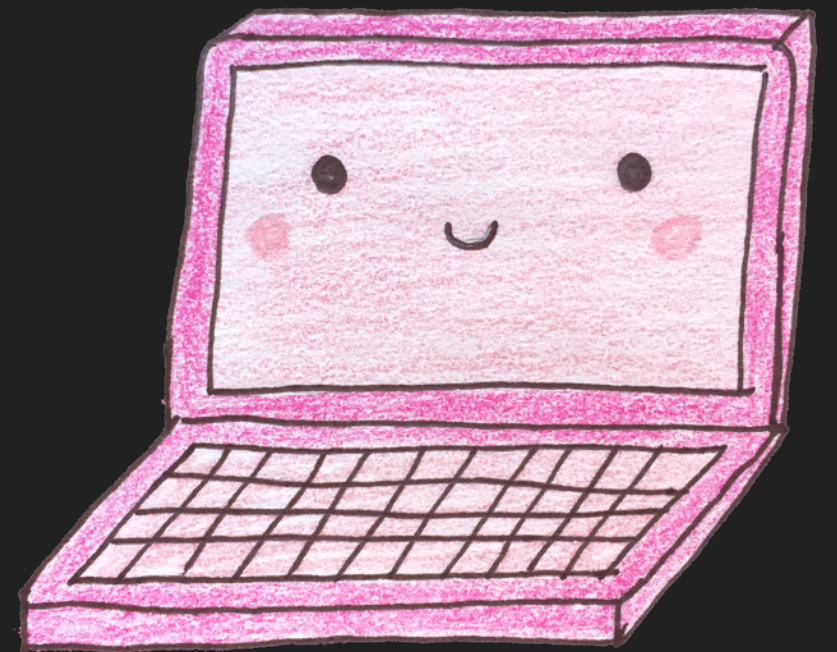
# HTML, CSS, & JS

V8 uses two parsers—eager and lazy —  
to eventually create an abstract syntax  
tree and scope structure



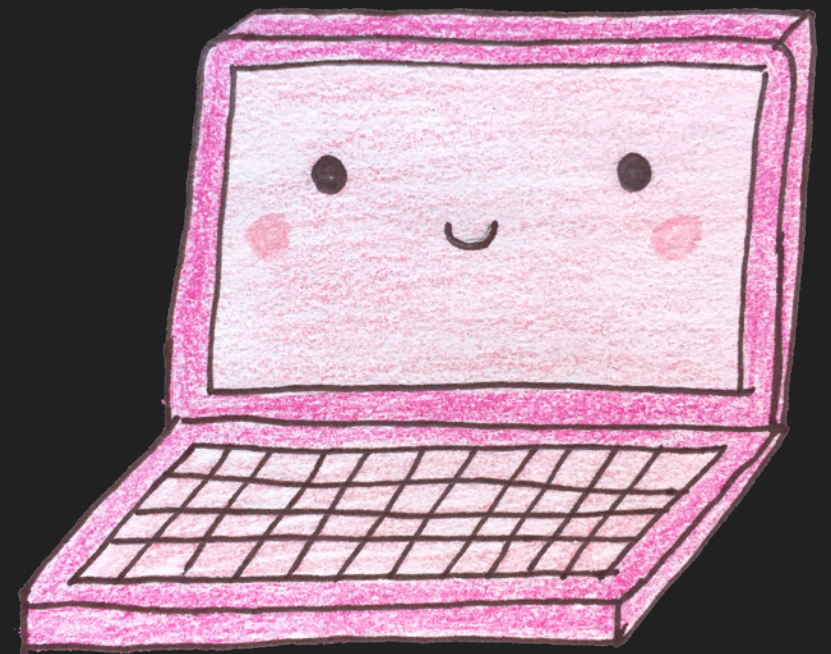
# HTML, CSS, & JS

The AST and scope structures get turned into low-level code, which then gets executed



# HTML, CSS, & JS

The bytecode also gets fed to the optimizing compiler which spits out machine code that then gets executed





# Natural Language

Human languages have a lexicon and syntax that cannot be described by a context-free grammar



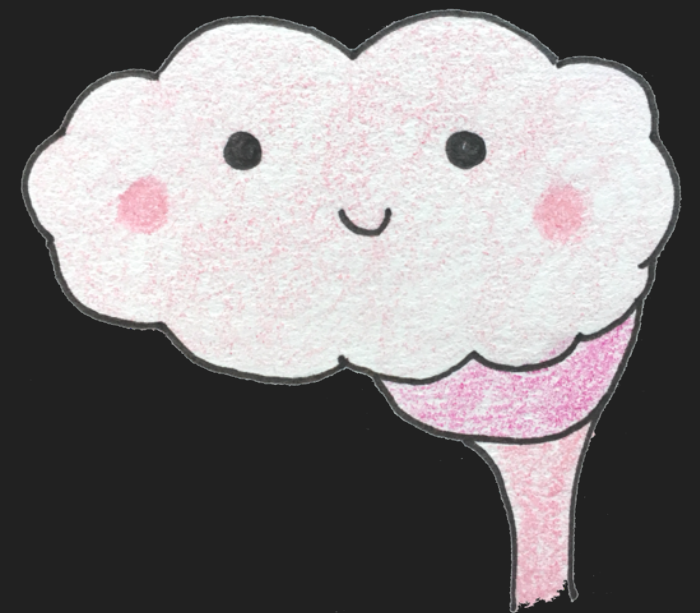
# Natural Language

Humans language contains a ton of ambiguity



# Natural Language

Step 1: To understand speech, humans break the unbroken speech stream into words





# Natural Language

Step 2: Our minds match the sounds to words in the mental lexicon



# Natural Language

Step 3: Once we access a word, we have access to its meaning and its syntactic and thematic roles



# Natural Language

Step 4: We then parse the sentence.  
But we're not 100% sure how...



# Natural Language

Modular View: the phases involved  
occur separately in different modules



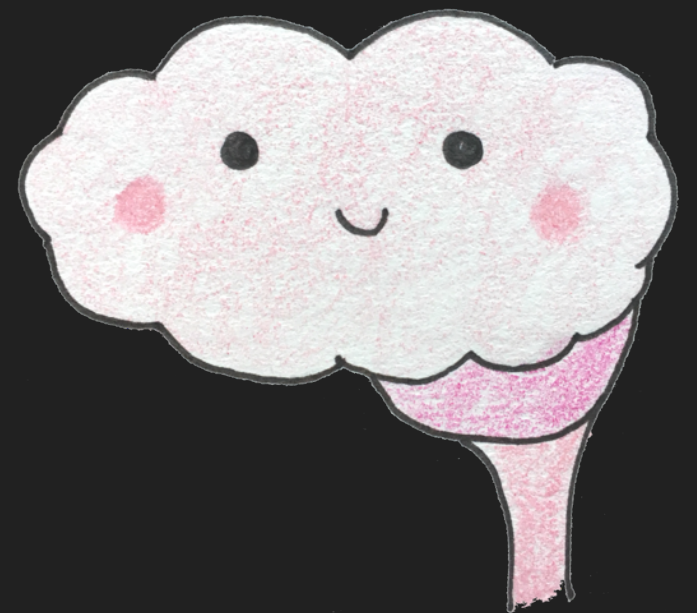
# Natural Language

Interactive View: all available information can be used at once in parsing the sentence



# Natural Language

Does this happen in serial or in parallel?



# Natural Language

Humans are forgiving of syntax errors





# Natural Language

Receptive language processing occurs in the dominant hemisphere of the brain, in Wernicke's area.



# Visualizing

How does the human visual system paint a picture of our world, and how does the browser render pixels to the screen?

# Human Vision

Step 1: Light goes into the eye via the cornea and lens



# Human Vision

Step 2: The retina turns the light into neural signals using rods and cones



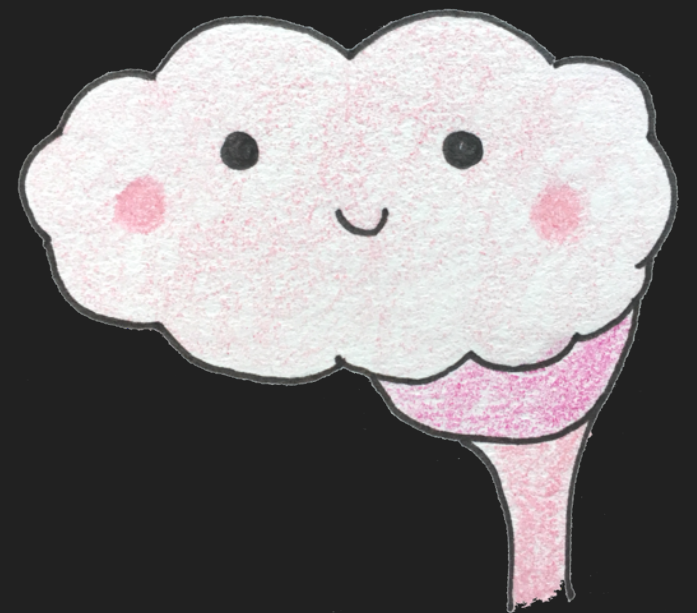
# Human Vision

Step 3: The neural signals get sent via the optic nerve to the brain



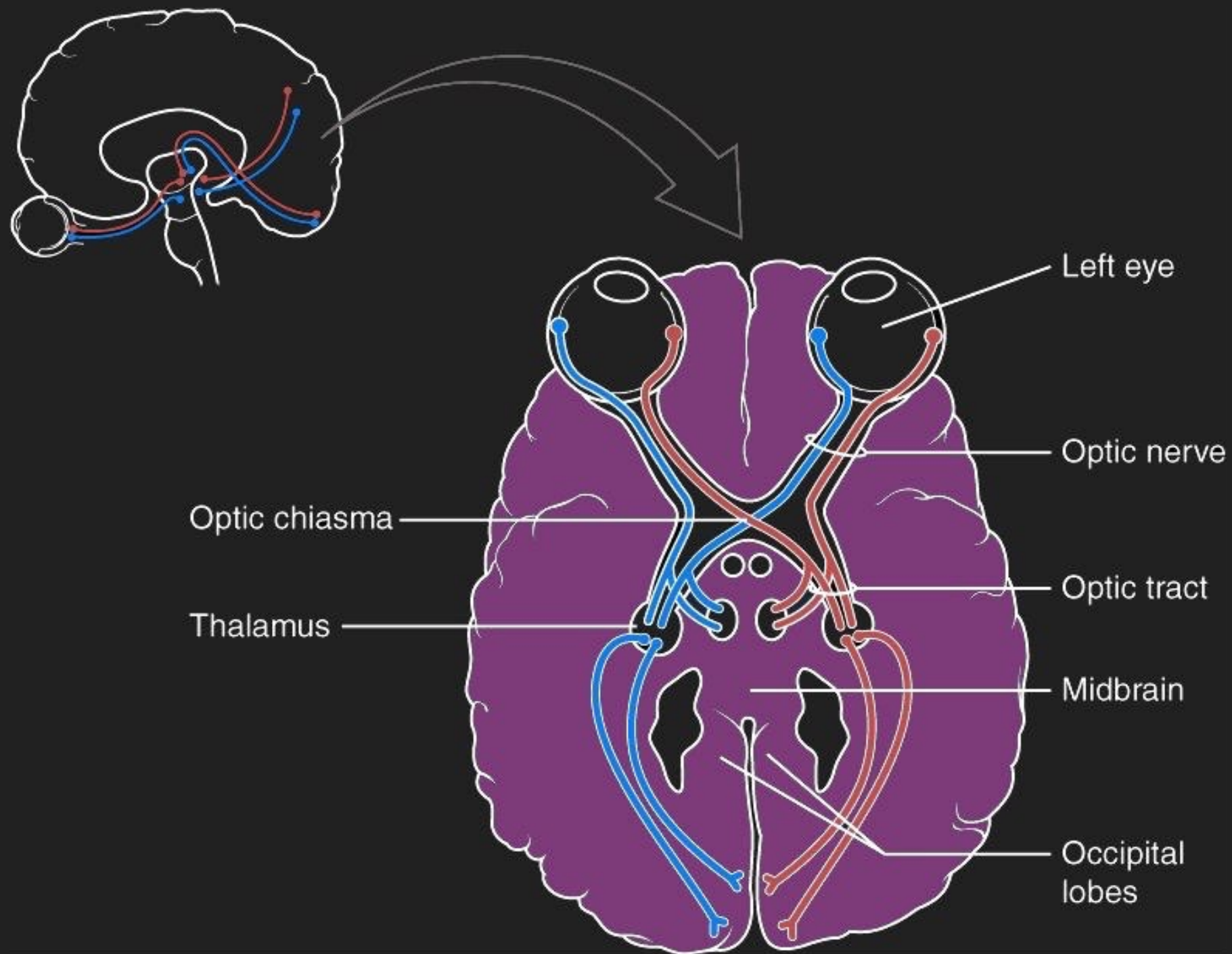
# Human Vision

Step 4: Signals from both eyes reach the optic chiasm, are combined, split by visual field, and sent to the opposite side of the brain





# Human Vision



# Human Vision

Step 5: Most signals get sent to the lateral geniculate nuclei



# Human Vision

Step 6: Signals then get sent to the primary visual cortex



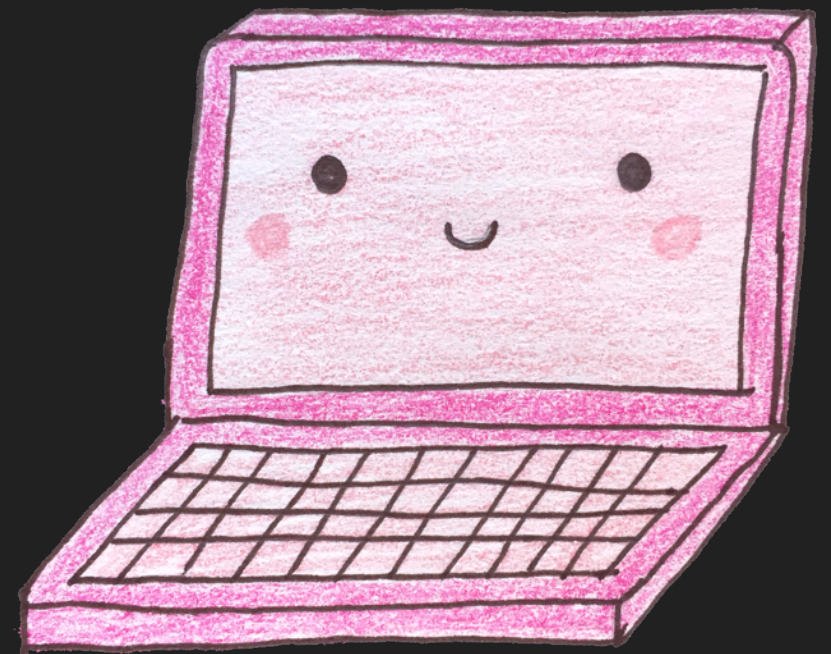
# Human Vision

Step 7: Signals gets sent to higher visual processing centers that help us actually perceive what we are seeing



# Rendering

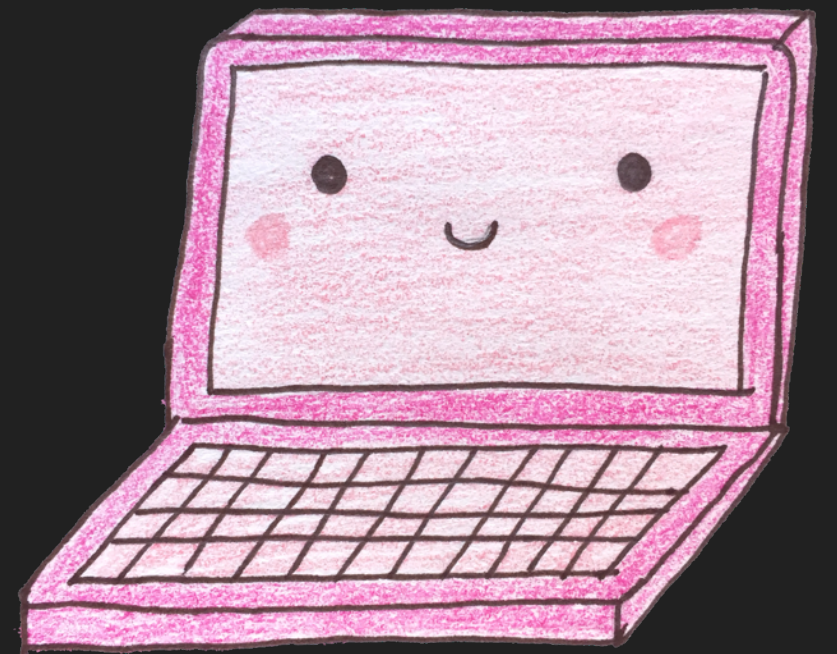
Step 1: HTML and CSS are parsed into DOM and CSSOM trees, respectively





# Rendering

Step 2: The DOM and CSSOM trees are combined to form the render tree





# Rendering

Step 3: The browser traverses the render tree, calculating the location and size of all elements



# Rendering

Step 4: The browser again traverses the render tree, creating bitmaps for each layer



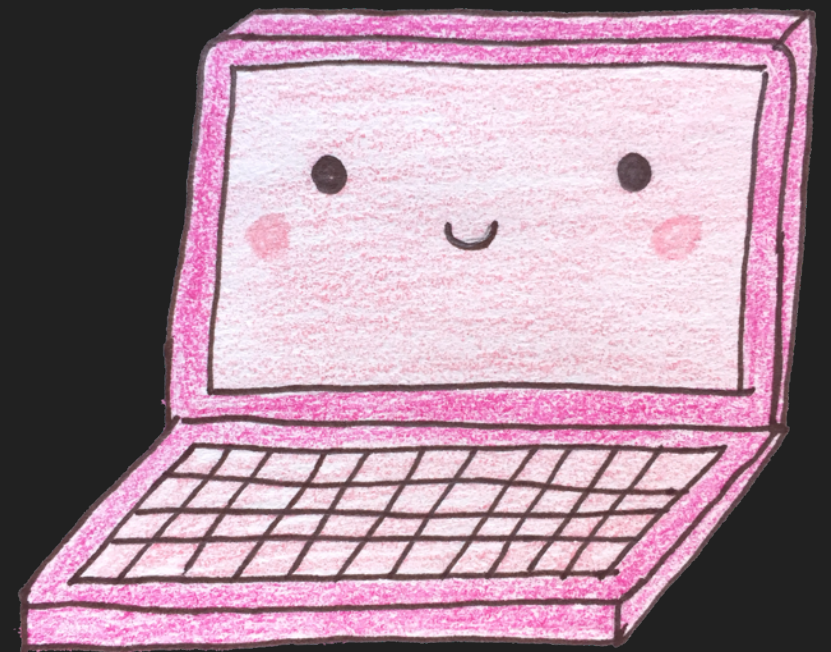
# Rendering

Step 5: Bitmaps are sent to the GPU for compositing



# Rendering

Step 6: Do it all again, maybe 60 times a second



# Task Management

Can the human mind multitask? Can the browser?



# Human Attention

- attention as a filter
- attention as a spotlight
- attention as control



# Human Attention

BLUE PURPLE RED

GREEN PURPLE

GREEN

# Human Attention



# Human Attention

BLUE PURPLE RED

GREEN PURPLE

GREEN

# Human Attention

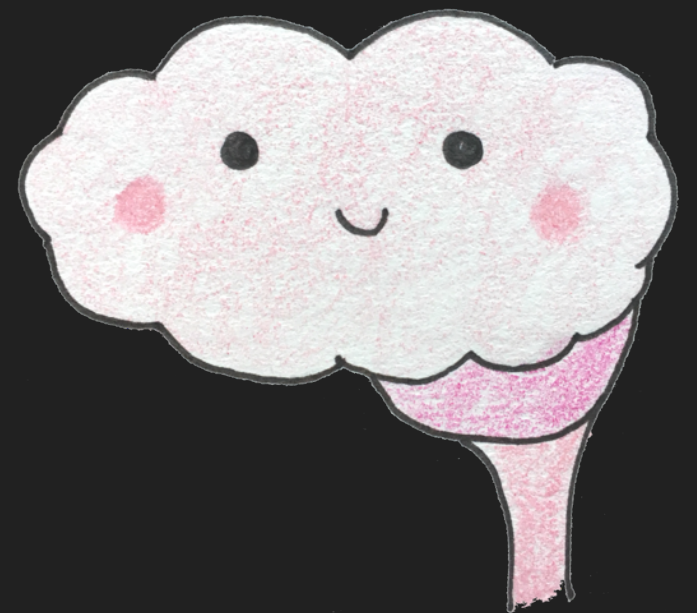
BLUE PURPLE RED

GREEN PURPLE

GREEN

# Human Attention

- attention as a filter
- attention as a spotlight
- attention as control





# Human Attention

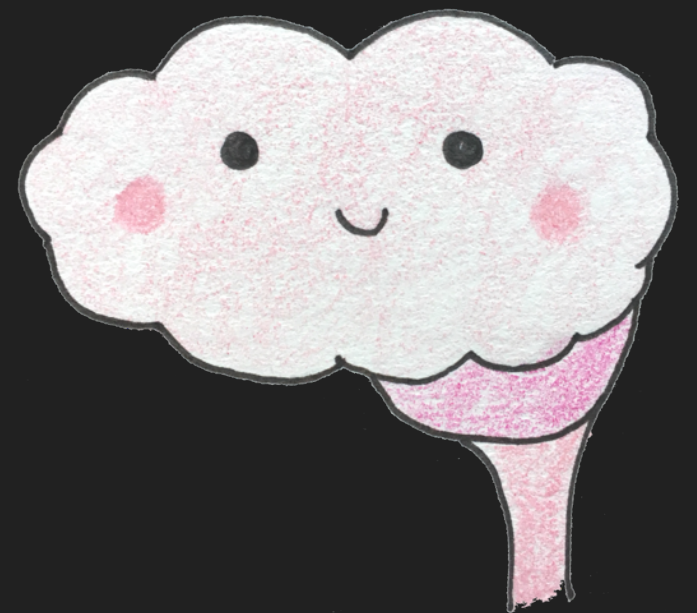
- attention as a filter
- attention as a spotlight
- attention as control
- attention as threads!



# Human Attention

Humans are pretty bad at multitasking:

- inattentional blindness
- dichotic listening task
- shadowing



# Human Attention

"These are the words you need to repeat back."

"These are the words you aren't supposed to be listening to."

"These are the words you need to repeat back."



# Human Attention

"These are the words you need to repeat back."

"Words these are the aren't supposed to you be to listening."

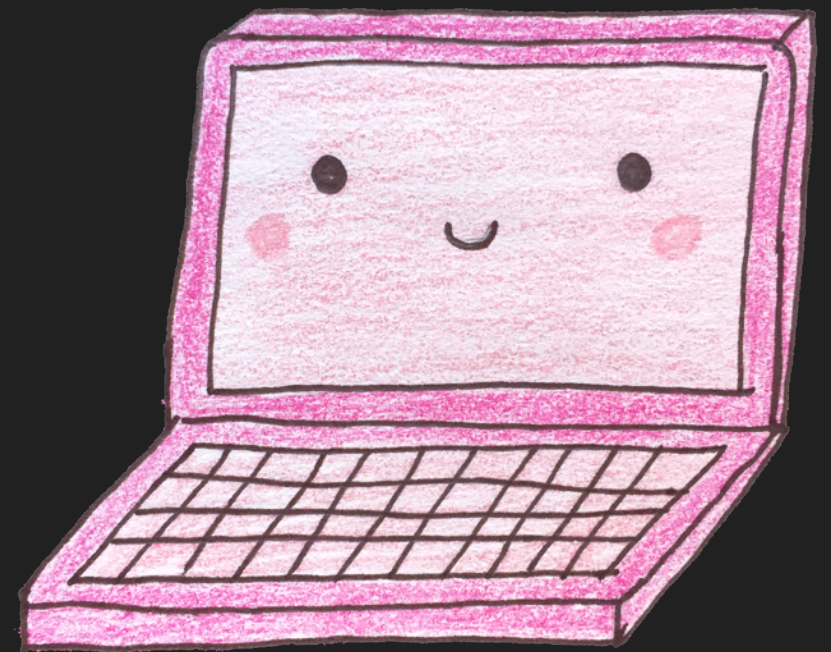
"These are the words you need to repeat back."



# Browser "Attention"

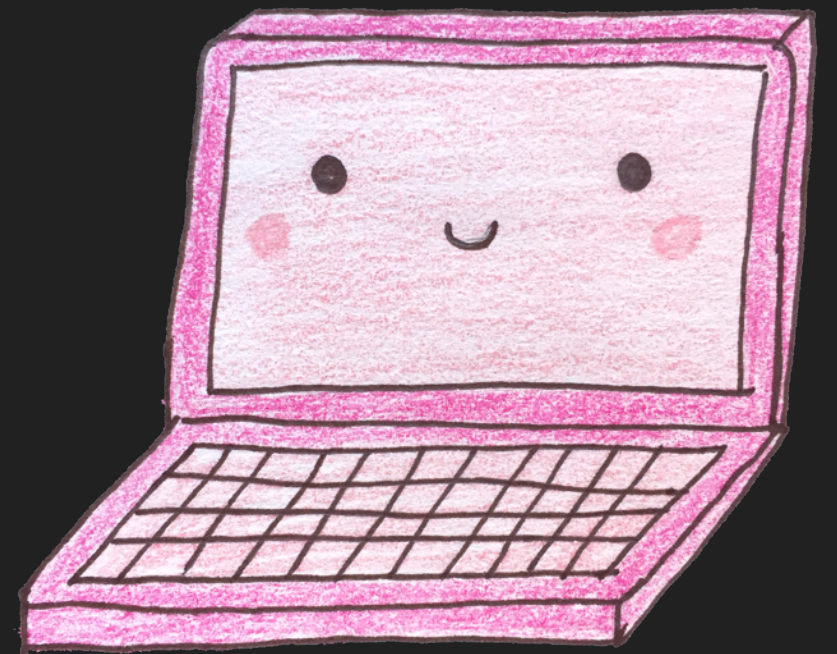
JavaScript itself doesn't multitask.

- single-threaded
- non-blocking
- asynchronous



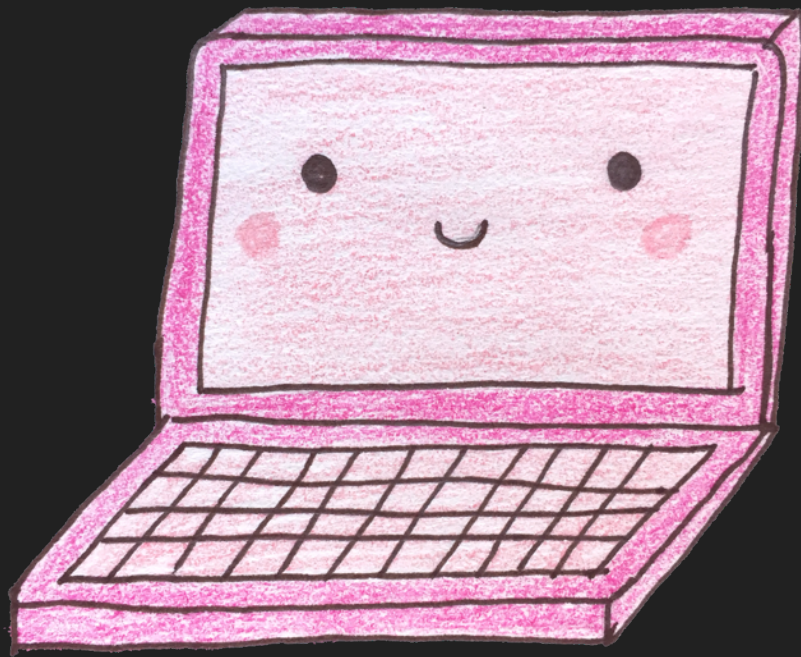
# Browser “Attention”

Within the browser, JavaScript can call out to APIs, making the browser itself able to effectively multitask!





Browser, are you processing  
the same way I'm processing?



Guess not, Brain! But I hope  
you learned something!

# Resources

## Books

- Blake, R., & Sekuler, R. & (2006). *Perception (5th ed.)*. Boston: McGraw-Hill.
- Harley, T. A. (2008). *Psychology of Language: From Data to Theory (3rd ed.)*. New York: Psychology Press.
- Kellogg, R. T. (2007). *Fundamentals of cognitive psychology*. Thousand Oaks, CA: SAGE.

## Websites

- <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork>
- <https://www.html5rocks.com/en/tutorials/speed/layers/>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/render-tree-construction>
- Marja Hölttä: Parsing JavaScript - better lazy than eager? (Video)
- Franziska Hinkelmann: JavaScript engines - how do they even? (Video)
- Chelsea Derrick: True Grit: Debugging CSS & Render Performance (Video)
- [www.ecma-international.org/ecma-262/](http://www.ecma-international.org/ecma-262/)
- [https://en.wikipedia.org/wiki/Visual\\_system](https://en.wikipedia.org/wiki/Visual_system)
- [https://en.wikipedia.org/wiki/Lateral\\_geniculate\\_nucleus](https://en.wikipedia.org/wiki/Lateral_geniculate_nucleus)
- [https://en.wikipedia.org/wiki/Language\\_processing\\_in\\_the\\_brain](https://en.wikipedia.org/wiki/Language_processing_in_the_brain)
- [https://en.wikipedia.org/wiki/Sentence\\_processing](https://en.wikipedia.org/wiki/Sentence_processing)
- <http://www.imdb.com/title/tt0112123/quotes>

# Thanks!

[jenna.is/at-nationjs](http://jenna.is/at-nationjs)



@zeigenvector