What if your brain were

~*literally*~

JavaScript?

by Jenna Zeigen
@zeigenvector

User Controls Team Lead
@ DigitalOcean

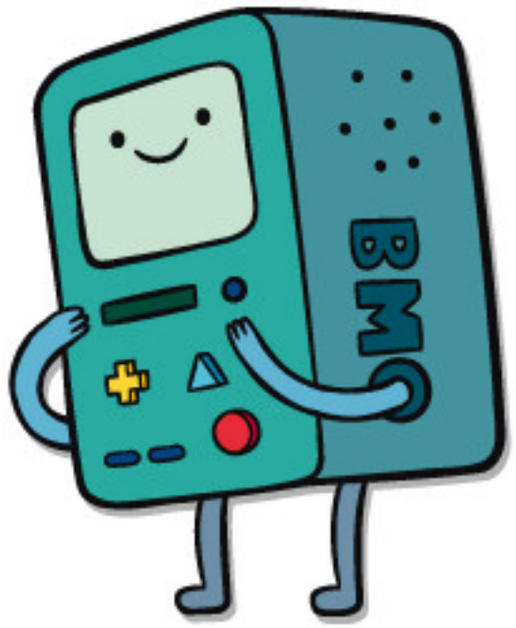
zeigenvector

jennazee

jenna.is/txjs15

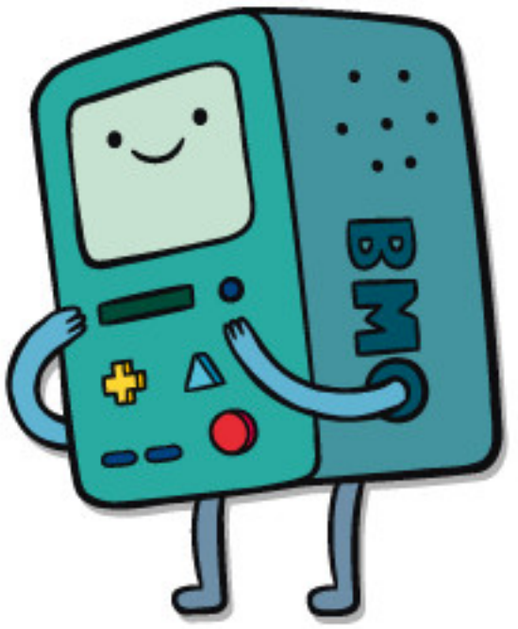
Human

JavaScript



¯_(ツ)_/¯

!!!! ‿ !



lol, cool.

Caveat: stretches.

Attention&
Imagery&
Perception&
Thinking&
Concepts&
Categories&
Memory&
Language&
Judgement&
Reasoning&
Decision Making&
Consciousness…

Attention&
Imagery&
Perception&
Thinking&
Concepts&
Categories&
Memory&
Language&
Judgement&
Reasoning&
Decision Making&
Consciousness…

1. Language Processing
2. Concepts + Categories // Objects, Prototypes, + Primitives
3. Attention // Event loop

Language Processing.

Language Processing

natural language vs. programming language

- regulation
- evolution
- learning

Language Processing

Programming languages

create and **manipulate** the environment, rather than just describe it.

Language Processing

- syntax
- semantics
- morphology
- phonology
- pragmatics

Language Processing

context.

http://www.ecma-international.org/ecma-262/6.0/index.html#sec-syntactic-and-lexical-grammars

Language Processing

"I saw the unicorn with the binoculars."

Language Processing

context.

Language Processing

Reference: anaphora + cataphora

https://en.wikipedia.org/wiki/Anaphora_(linguistics)
https://en.wikipedia.org/wiki/Binding_(linguistics)
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Function/bind

Language Processing

Anaphora: "Jenna gave a talk on the cognitive science of JavaScript, and she totally rocked it."

Language Processing

Cataphora: “Though **she** had never given **it** before, **Jenna** knew **her talk** was going to be a hit.”

Language Processing

Reference: anaphora + cataphora

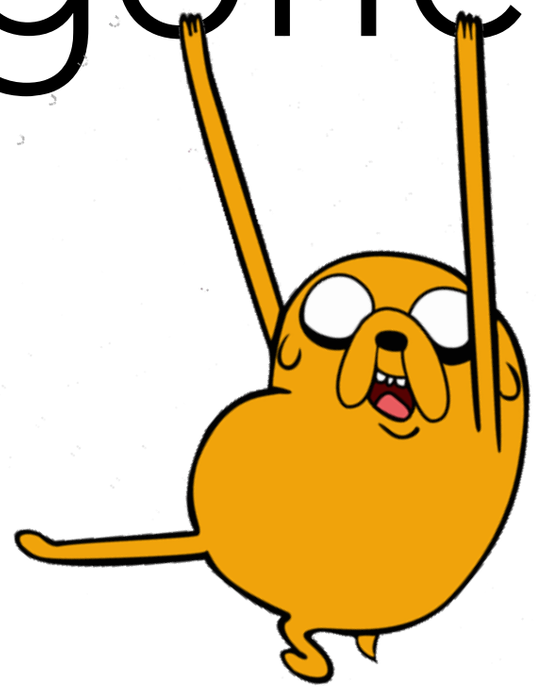
Language Processing

syntax errors, reference errors

http://www.ecma-international.org/ecma-262/6.0/index.html#sec-error-handling-and-language-extensions

Concepts +

Categories.



Concepts + Categories

“knowledge representation”

Concepts + Categories

http://www.photogalaxy.com/pic/eventhorizon-20/yorkie_puppy.jpg

Concepts + Categories

http://www.photogalaxy.com/pic/eventhorizon-20/yorkie_puppy.jpg

Concepts + Categories

classical

vs.

prototype

(categorization theories)

classical

vs.

prototypal

(inheritance)

Concepts + Categories

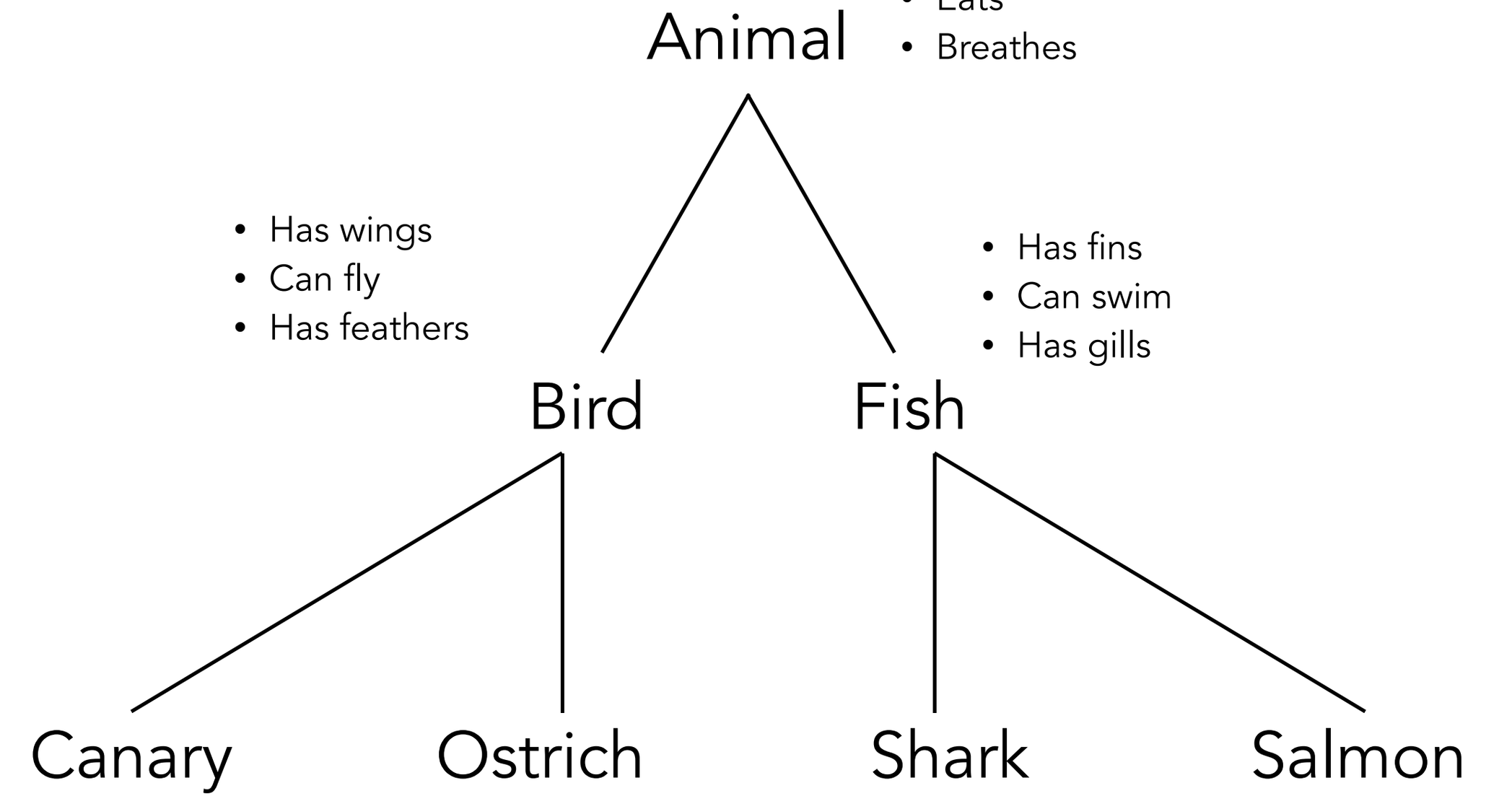
classical vs. prototype (categorization theories)	classical vs. prototypal (inheritance)

coincidence? ¯_(ツ)_/¯

Concepts + Categories

"In a computer system designed for the storage of semantic information, it is **more economical to store generalized information with superset nodes**, rather than with all the individual nodes to which such a generalization might apply. But such a storage system incurs the cost of additional processing time in retrieving the information. When the implications of such a model were tested for human [subjects] using **well-ordered hierarchies** that are part of the common culture, there was a substantial agreement between the predictions and the data."

(Collins & Quillian, 1969)



Collins & Quillian, 1969

Is a canary a bird?

- Has skin
- Can move around
- Eats
- Breathes

Animal

- Has wings
- Can fly
- Has feathers

Bird

- Has fins
- Can swim
- Has gills

Fish

Canary

- Can sing
- Is yellow

Ostrich

- Has long, thin legs
- Is tall
- Can’t fly

Shark

- Can bite
- Is dangerous

Salmon

- Is pink
- Is edible
- Swims upstream to lay eggs

Collins & Quillian, 1969

Is a canary an animal?

- Has skin
- Can move around
- Eats
- Breathes

Animal

- Has wings
- Can fly
- Has feathers

Bird

- Has fins
- Can swim
- Has gills

Fish

Canary

- Can sing
- Is yellow

Ostrich

- Has long, thin legs
- Is tall
- Can’t fly

Shark

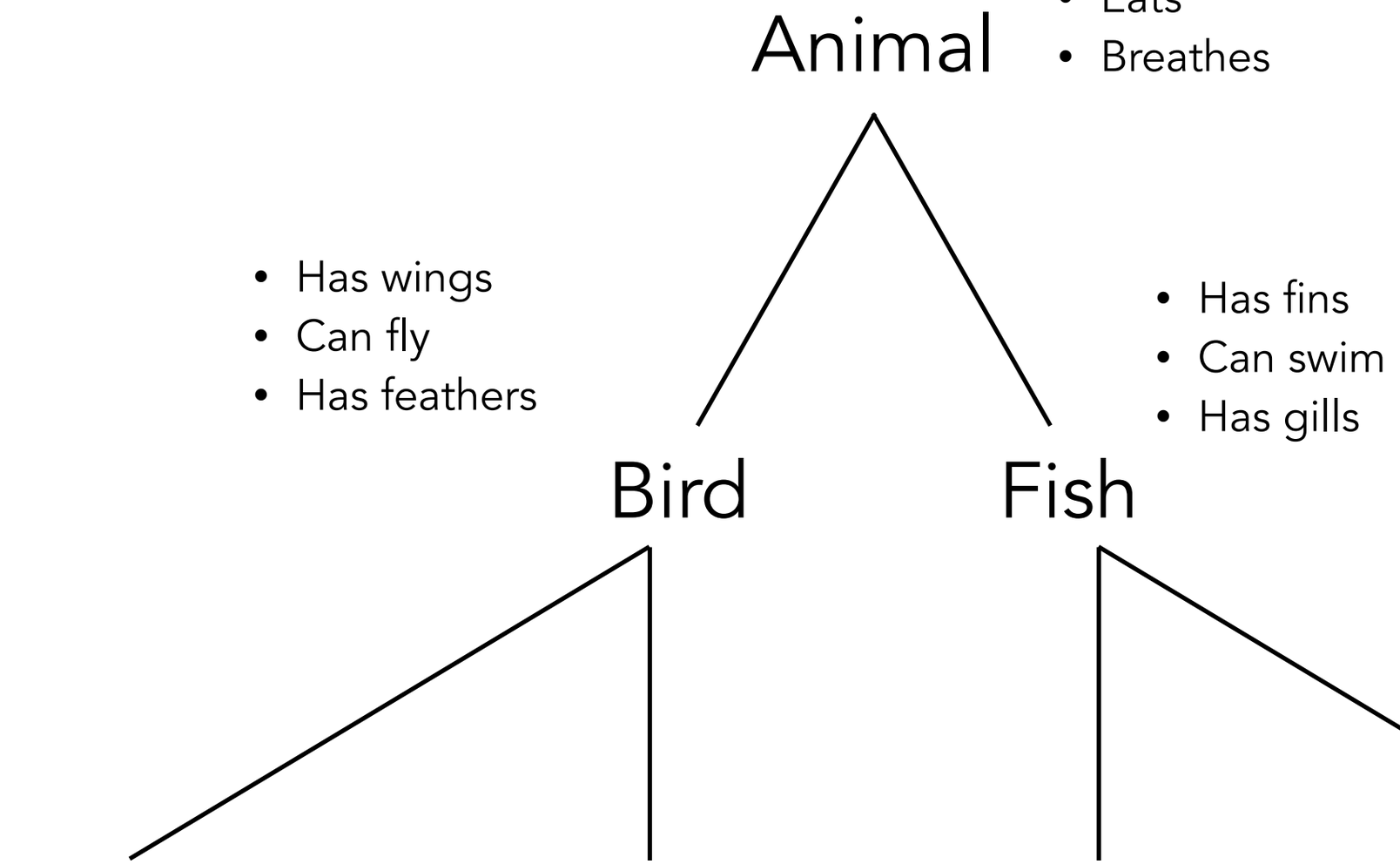
- Can bite
- Is dangerous

Salmon

- Is pink
- Is edible
- Swims upstream to lay eggs

Collins & Quillian, 1969

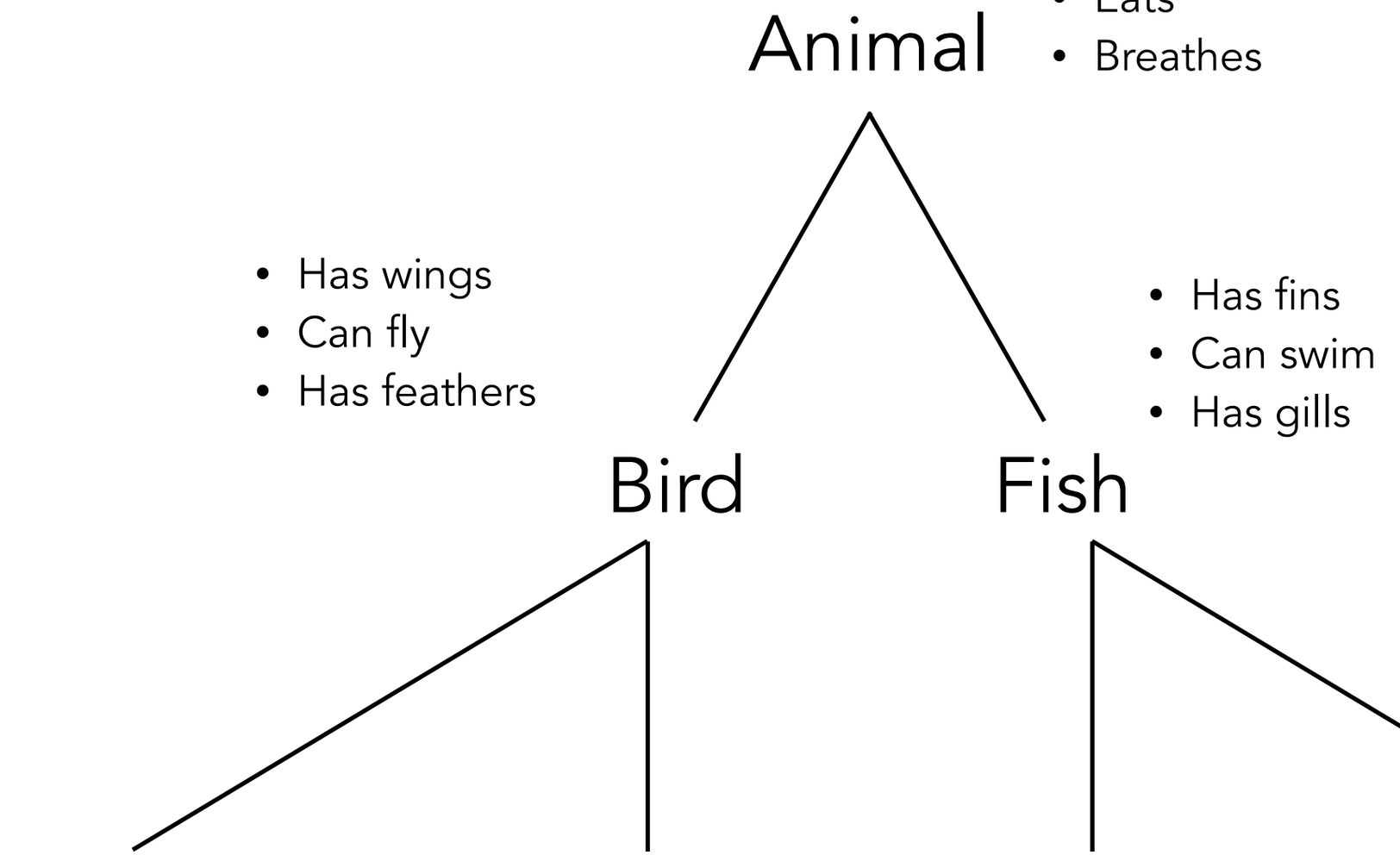
Is a canary yellow?



Collins & Quillian, 1969

Does a canary breathe?

- Has skin
- Can move around
- Eats
- Breathes



- Has wings
- Can fly
- Has feathers

- Has fins
- Can swim
- Has gills

- Can sing
- Is yellow

- Has long, thin legs
- Is tall
- Can’t fly

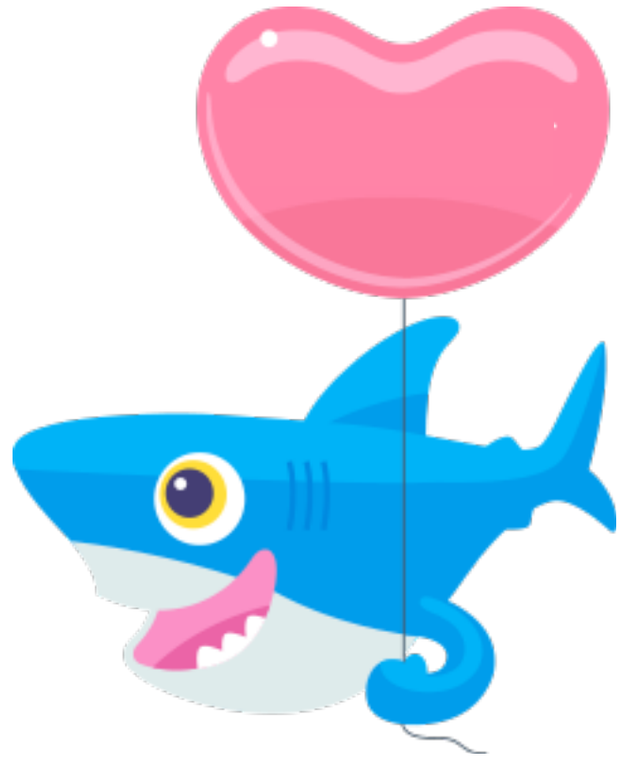
- Can bite
- Is dangerous

- Is pink
- Is edible
- Swims upstream to lay eggs

Collins & Quillian, 1969

Concepts + Categories

Is a shark a fish????



Concepts + Categories

Prototype theory (Rosch, 1973):

- we store an average ideal representation of a category

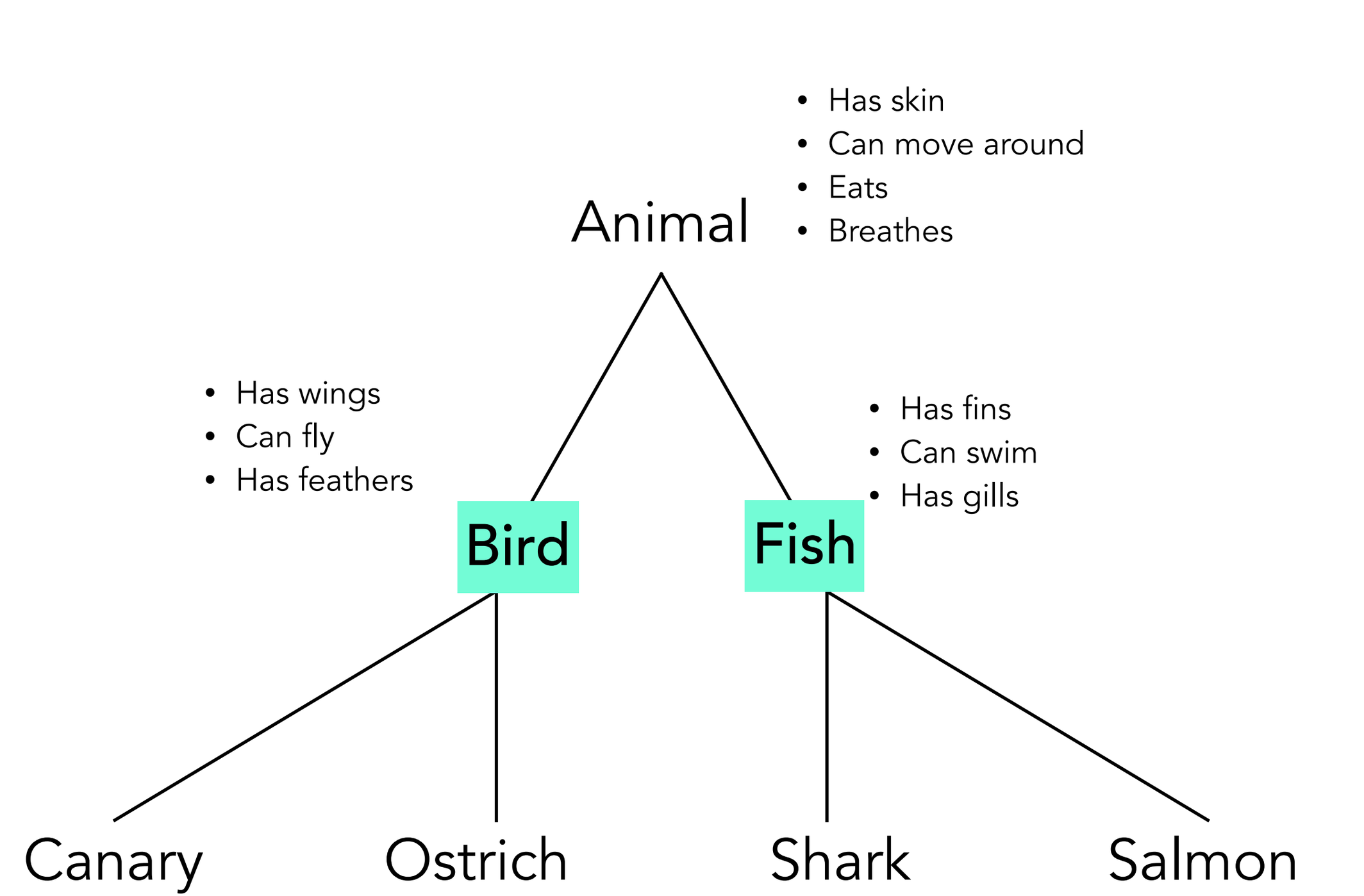
Exemplar theory

- we store an **instance** of a category that is a combination of all experienced exemplars

Concepts + Categories

Basic Level Categories:

A “natural” level of categorization



Collins & Quillian, 1969

Concepts + Categories

Basic Level Categories:

ECMAScript types?
(Boolean, Null, Undefined, Number, String, Symbol)

```
> String.prototype.__proto__
< ▶ Object {}
> Boolean.prototype.__proto__
< ▶ Object {}
> Number.prototype.__proto__
< ▶ Object {}
> Symbol.prototype.__proto__
< ▶ Object {}
```

http://www.ecma-international.org/ecma-262/6.0/index.html#sec-ecmascript-data-types-and-values

Concepts + Categories

Basic Level Categories:

ECMAScript types?
(Boolean, Null, Undefined, Number, String, Symbol)

```
> undefined.prototype
▶ Uncaught TypeError: Cannot read property 'prototype' of undefined
    at <anonymous>:2:10
    at Object.InjectedScript._evaluateOn (<anonymous>:905:140)
    at Object.InjectedScript._evaluateAndWrap (<anonymous>:838:34)
    at Object.InjectedScript.evaluate (<anonymous>:694:21)
> null.prototype
▶ Uncaught TypeError: Cannot read property 'prototype' of null
    at <anonymous>:2:5
    at Object.InjectedScript._evaluateOn (<anonymous>:905:140)
    at Object.InjectedScript._evaluateAndWrap (<anonymous>:838:34)
    at Object.InjectedScript.evaluate (<anonymous>:694:21)
```

http://www.ecma-international.org/ecma-262/6.0/index.html#sec-ecmascript-data-types-and-values

Concepts + Categories

Basic Level Categories:

But what about Arrays? Functions?? Dates? Promises?

“Well-Known Intrinsic Objects”

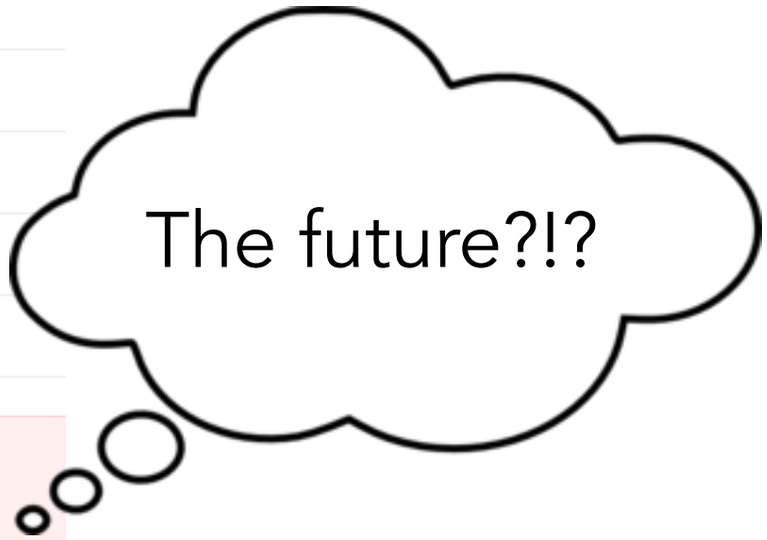
http://www.ecma-international.org/ecma-262/6.0/index.html#sec-ecmascript-data-types-and-values

```
> Array.prototype.__proto__
◂ ▶ Object {}
> ArrayBuffer.prototype.__proto__
◂ ▶ Object {}
> Boolean.prototype.__proto__
◂ ▶ Object {}
> DataView.prototype.__proto__
◂ ▶ Object {}
> Date.prototype.__proto__
◂ ▶ Object {}
> Error.prototype.__proto__
◂ ▶ Object {}
> EvalError.prototype.__proto__
◂ ▶ d {name: "Error", message: ""}
> Float32Array.prototype.__proto__
◂ ▶ Object {}
> Float64Array.prototype.__proto__
◂ ▶ Object {}
> Function.prototype.__proto__
◂ ▶ Object {}
> Int8Array.prototype.__proto__
◂ ▶ Object {}
> Map.prototype.__proto__
◂ ▶ Object {}
> Map.prototype.__proto__
◂ ▶ Object {}
> Number.prototype.__proto__
◂ ▶ Object {}
> Object.prototype.__proto__
◂ null
> Proxy.prototype.__proto__
✕ ▶ Uncaught ReferenceError: Proxy is not defined
      at <anonymous>:2:1
      at Object.InjectedScript._evaluateOn (<anonymous>:895:140)
      at Object.InjectedScript._evaluateAndWrap (<anonymous>:828:34)
      at Object.InjectedScript.evaluate (<anonymous>:694:21)
> Promise.prototype.__proto__
◂ ▶ Object {}
> RangeError.prototype.__proto__
◂ ▶ d {name: "Error", message: ""}
> ReferenceError.prototype.__proto__
◂ ▶ d {name: "Error", message: ""}
> RegExp.prototype.__proto__
◂ ▶ Object {}
> Set.prototype.__proto__
◂ ▶ Object {}
> String.prototype.__proto__
◂ ▶ Object {}
> WeakMap.prototype.__proto__
◂ ▶ Object {}
```

```
EvalError.prototype.__proto__
▶ d {name: "Error", message: ""}
```

```
> Array.prototype.__proto__
▶ Object {}
> ArrayBuffer.prototype.__proto__
▶ Object {}
> Boolean.prototype.__proto__
▶ Object {}
> DataView.prototype.__proto__
▶ Object {}
> Date.prototype.__proto__
▶ Object {}
> Error.prototype.__proto__
▶ Object {}
> EvalError.prototype.__proto__
▶ d {name: "Error", message: ""}
> Float32Array.prototype.__proto__
▶ Object {}
> Float64Array.prototype.__proto__
▶ Object {}
> Function.prototype.__proto__
▶ Object {}
> Int8Array.prototype.__proto__
▶ Object {}
> Map.prototype.__proto__
▶ Object {}
> Number.prototype.__proto__
▶ Object {}
> Object.prototype.__proto__
null
> Proxy.prototype.__proto__
▶ Uncaught ReferenceError: Proxy is not defined
    at <anonymous>:2:1
    at Object.InjectedScript._evaluateOn (<anonymous>:895:140)
    at Object.InjectedScript._evaluateAndWrap (<anonymous>:828:34)
    at Object.InjectedScript.evaluate (<anonymous>:694:21)
> Promise.prototype.__proto__
▶ Object {}
> RangeError.prototype.__proto__
▶ d {name: "Error", message: ""}
> ReferenceError.prototype.__proto__
▶ d {name: "Error", message: ""}
> RegExp.prototype.__proto__
▶ Object {}
> Set.prototype.__proto__
▶ Object {}
> String.prototype.__proto__
▶ Object {}
> WeakMap.prototype.__proto__
▶ Object {}
```

```
EvalError.prototype.__proto__
▶ d {name: "Error", message: ""}
```



```
> Array.prototype.__proto__
◂ ▶ Object {}
> ArrayBuffer.prototype.__proto__
◂ ▶ Object {}
> Boolean.prototype.__proto__
◂ ▶ Object {}
> DataView.prototype.__proto__
◂ ▶ Object {}
> Date.prototype.__proto__
◂ ▶ Object {}
> Error.prototype.__proto__
◂ ▶ Object {}
> EvalError.prototype.__proto__
◂ ▶ d {name: "Error", message: ""}
> Float32Array.prototype.__proto__
◂ ▶ Object {}
> Float64Array.prototype.__proto__
◂ ▶ Object {}
> Function.prototype.__proto__
◂ ▶ Object {}
> Int8Array.prototype.__proto__
◂ ▶ Object {}
> Map.prototype.__proto__
◂ ▶ Object {}
> Number.prototype.__proto__
◂ ▶ Object {}
> Object.prototype.__proto__
◂ null
> Proxy.prototype.__proto__
✕ ▶ Uncaught ReferenceError: Proxy is not defined
      at <anonymous>:2:1
      at Object.InjectedScript._evaluateOn (<anonymous>:895:140)
      at Object.InjectedScript._evaluateAndWrap (<anonymous>:828:34)
      at Object.InjectedScript.evaluate (<anonymous>:694:21)
> Promise.prototype.__proto__
◂ ▶ Object {}
> RangeError.prototype.__proto__
◂ ▶ d {name: "Error", message: ""}
> ReferenceError.prototype.__proto__
◂ ▶ d {name: "Error", message: ""}
> RegExp.prototype.__proto__
◂ ▶ Object {}
> Set.prototype.__proto__
◂ ▶ Object {}
> String.prototype.__proto__
◂ ▶ Object {}
> WeakMap.prototype.__proto__
◂ ▶ Object {}
```

```
EvalError.prototype.__proto__
▶ d {name: "Error", message: ""}
```

```
zeigenspace:~ jenna$ node --harmony
> Proxy
{ create: [Function],
  createFunction: [Function] }
> Proxy.prototype
undefined
> _
```

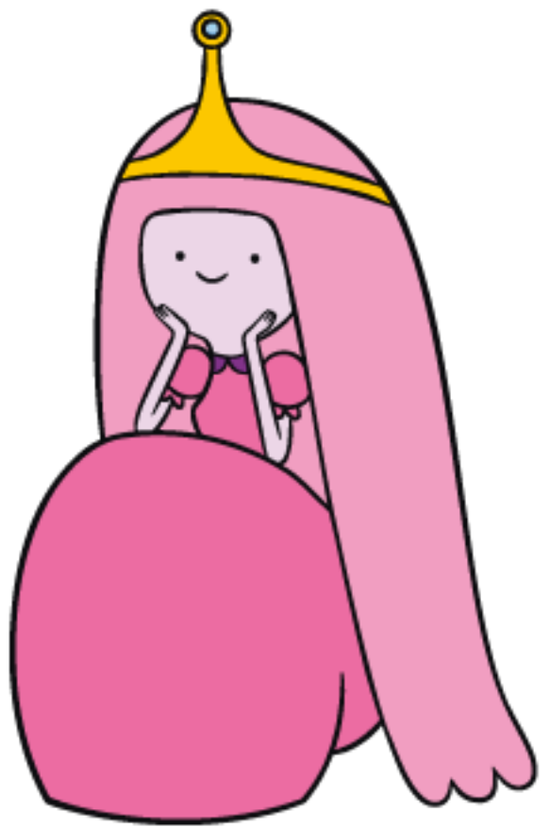
```
> Array.prototype.__proto__
▶ Object {}
> ArrayBuffer.prototype.__proto__
▶ Object {}
> Boolean.prototype.__proto__
▶ Object {}
> DataView.prototype.__proto__
▶ Object {}
> Date.prototype.__proto__
▶ Object {}
> Error.prototype.__proto__
▶ Object {}
> EvalError.prototype.__proto__
▶ d {name: "Error", message: ""}
> Float32Array.prototype.__proto__
▶ Object {}
> Float64Array.prototype.__proto__
▶ Object {}
> Function.prototype.__proto__
▶ Object {}
> Int8Array.prototype.__proto__
▶ Object {}
> Map.prototype.__proto__
▶ Object {}
> Number.prototype.__proto__
▶ Object {}
> Object.prototype.__proto__
null
> Proxy.prototype.__proto__
▶ Uncaught ReferenceError: Proxy is not defined
    at <anonymous>:2:1
    at Object.InjectedScript._evaluateOn (<anonymous>:895:140)
    at Object.InjectedScript._evaluateAndWrap (<anonymous>:828:34)
    at Object.InjectedScript.evaluate (<anonymous>:694:21)
> Promise.prototype.__proto__
▶ Object {}
> RangeError.prototype.__proto__
▶ d {name: "Error", message: ""}
> ReferenceError.prototype.__proto__
▶ d {name: "Error", message: ""}
> RegExp.prototype.__proto__
▶ Object {}
> Set.prototype.__proto__
▶ Object {}
> String.prototype.__proto__
▶ Object {}
> WeakMap.prototype.__proto__
▶ Object {}
```

```
EvalError.prototype.__proto__
▶ d {name: "Error", message: ""}
```

```
zeigenspace:~ jenna$ node --harmony
> Proxy
{ create: [Function],
  createFunction: [Function] }
> Proxy.prototype
undefined
> _
```

¯_(ツ)_/¯

Attention.



Attention

- attention as a filter
- attention as a spotlight
- attention as glue

Attention

- attention as a filter
- attention as a spotlight
- attention as glue

Attention as threads!

Attention

Humans are pretty bad at multitasking:

- inattentional blindness
- dichotic listening task
- shadowing

(Simons, 1999; Cherry, 1953; Triesman, 1964; Allport et al., 1972)

Attention

Humans are pretty bad at multitasking:

- inattentional blindness
- dichotic listening task
- shadowing

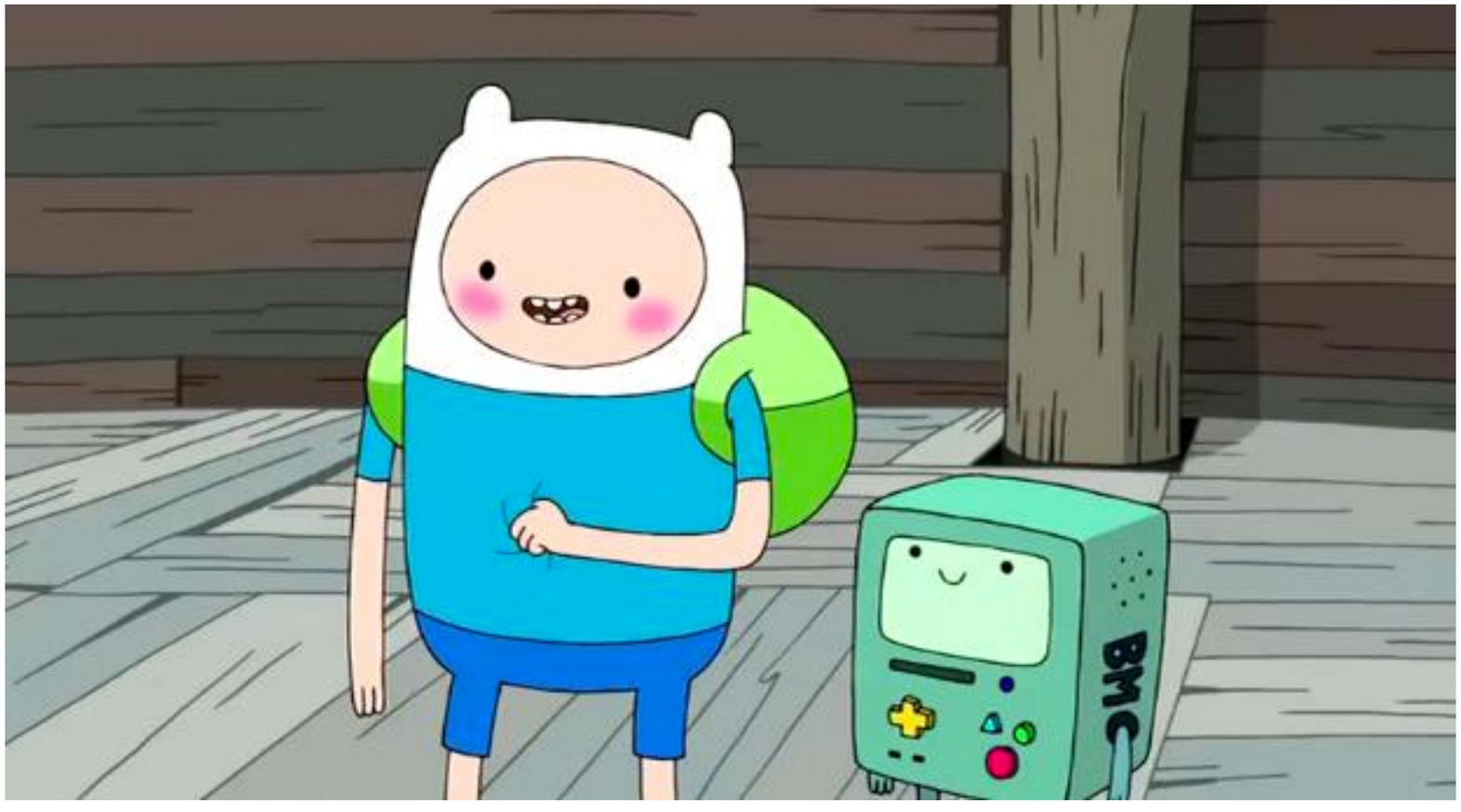
task-specific resources

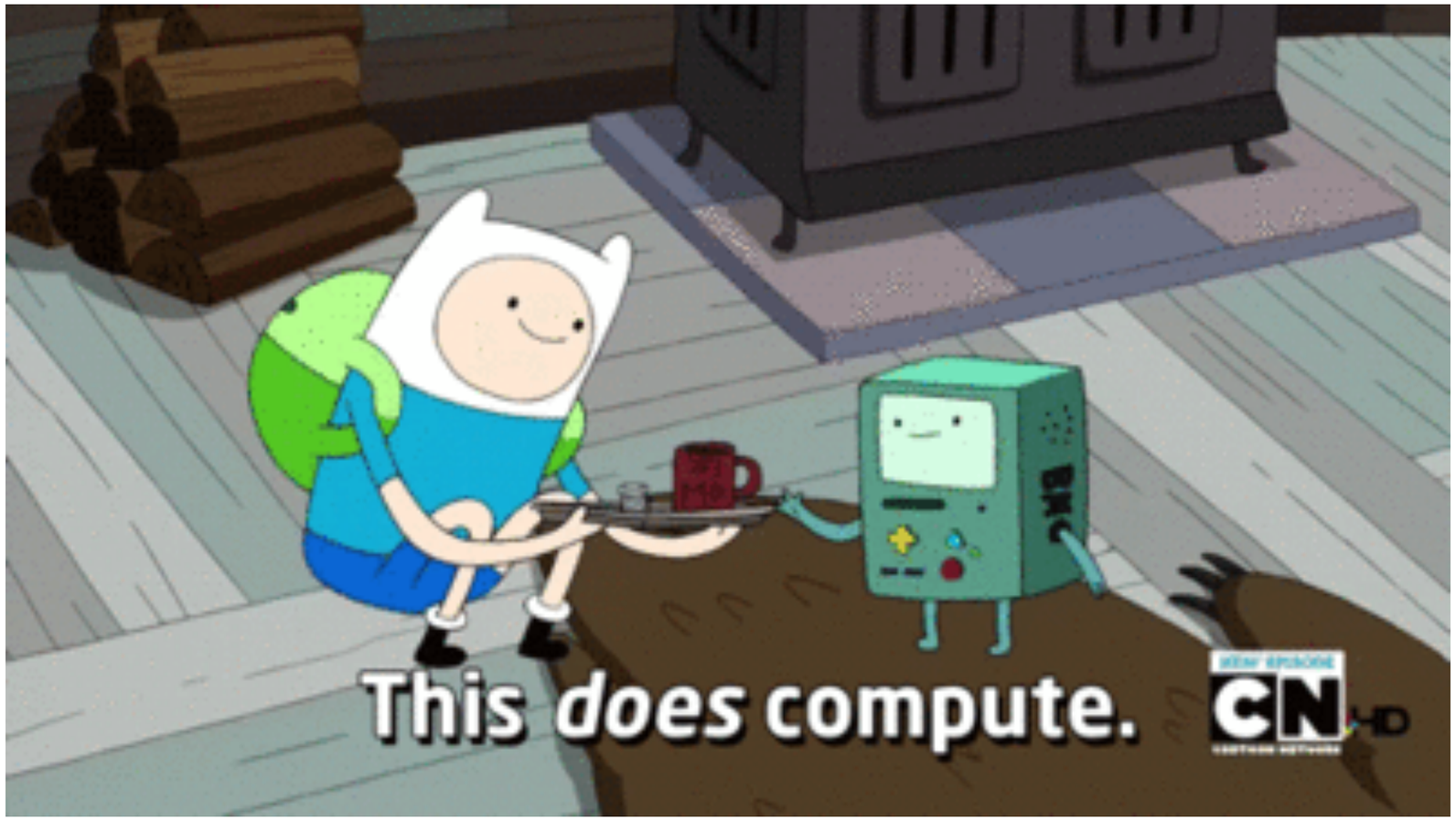
(Simons, 1999; Cherry, 1953; Triesman, 1964; Allport et al., 1972)

Attention

JavaScript does not multitask.

- single-threaded
- non-blocking
- asynchronous


BMO



http://itsadventuretimepeasants.tumblr.com/

Thanks!

↑

Me, @zeigenvector