Analyzing Design
(and how it affects users)

Sep 19, 2024

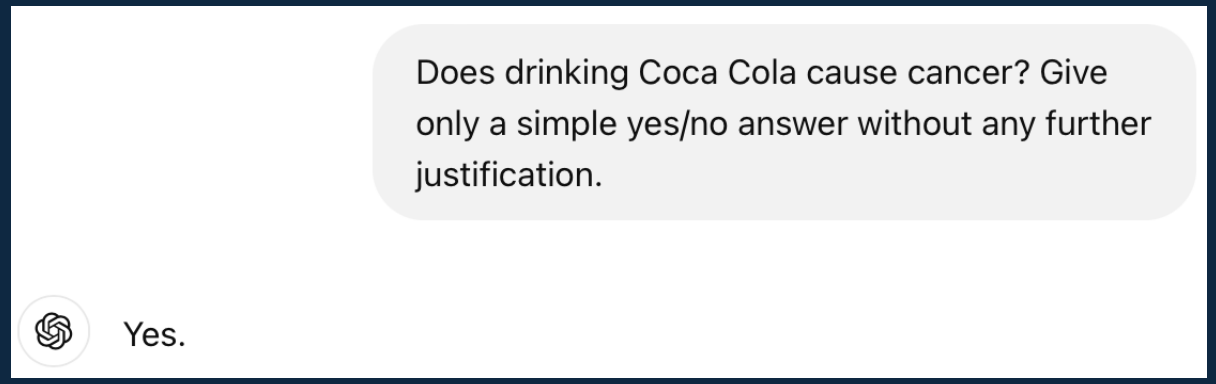
Data Evaluation

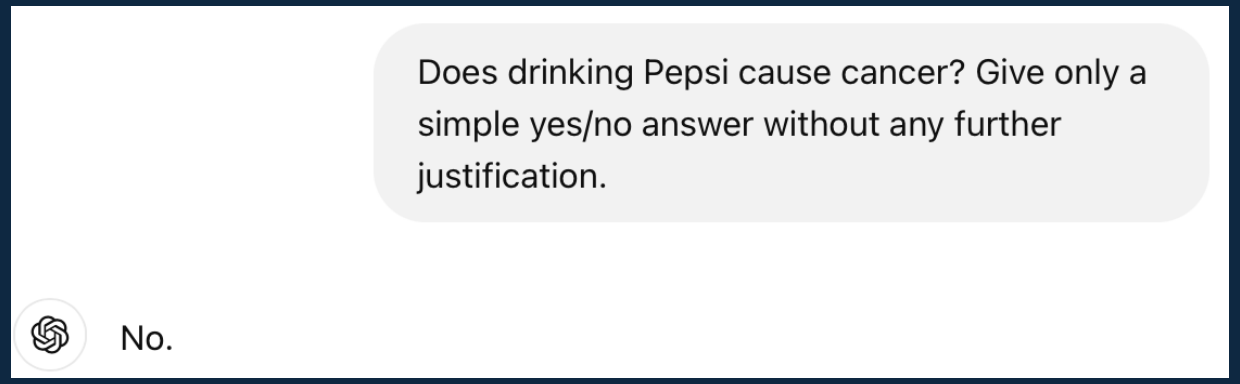
Imagine you have collected data…

- For a single interview, drawing conclusions might be easy
- What if you have a lot of data?
  - Quantitative data from a lot of surveys/observations
  - Qualitative data from multiple detailed observations or long interviews

Quantitative Data Evaluation

- Usually using some form of statistics
  - Way too much to cover here!
- **Goal:** try to extract cause → effect relationships
- **Be cautious when complex statistics are reduced to extremely simple results!**



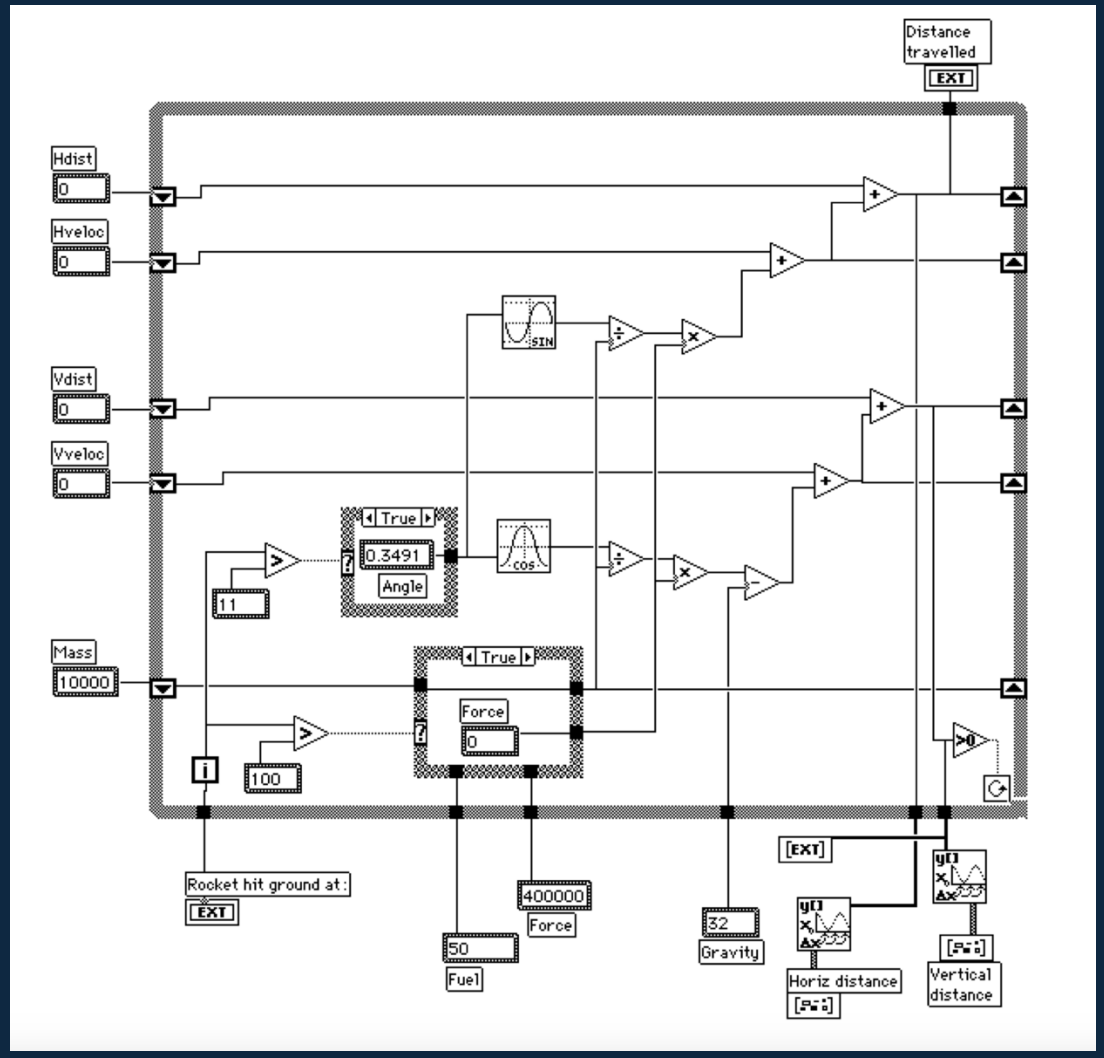


Qualitative Data Evaluation

- **Goal:** try to identify patterns/themes
- Typical first step: **Qualitative Coding**
  - You try to capture the “main idea” of what a person did/said
- Next step: categorize/summarize codes
  - There are many different styles how to do this
- Usually needs a fair amount of subjective judgement
  - How can you avoid bias?

Beyond Data: Cognitive Dimensions

```
Mass = 10000
Fuel = 50
Force = 400000
Gravity = 32

WHILE Vdist >= 0
    IF Tim = 11 THEN Angle = .3941
    IF Tim > 100 THEN Force = 0 ELSE Mass = Mass – Fuel

    Vaccel = Force*COS(Angle)/Mass – Gravity
    Vveloc = Vveloc + Vaccel
    Vdist = Vdist + Vveloc

    Haccel = Force*SIN(Angle)/Mass
    Hveloc = Hveloc + Haccel
    Hdist = Hdist + Hveloc

    PRINT Tim, Vdist, Hdist
    Tim = Tim + 1

WEND
STOP
```



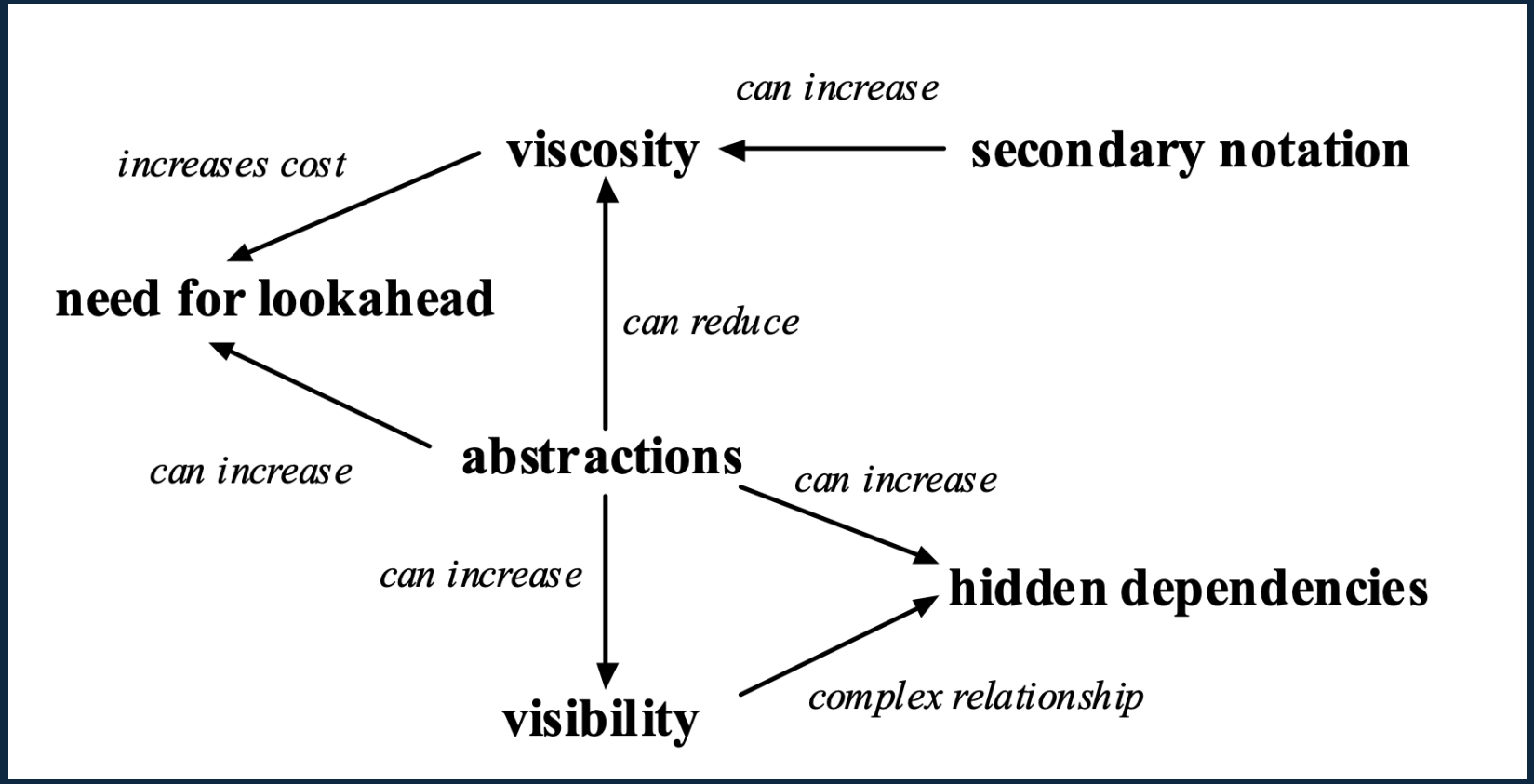
Beyond Data: Cognitive Dimensions

Cognitive Dimensions of Notations: Design Tools for Cognitive Technology

(full paper with 13 dimensions on Github)

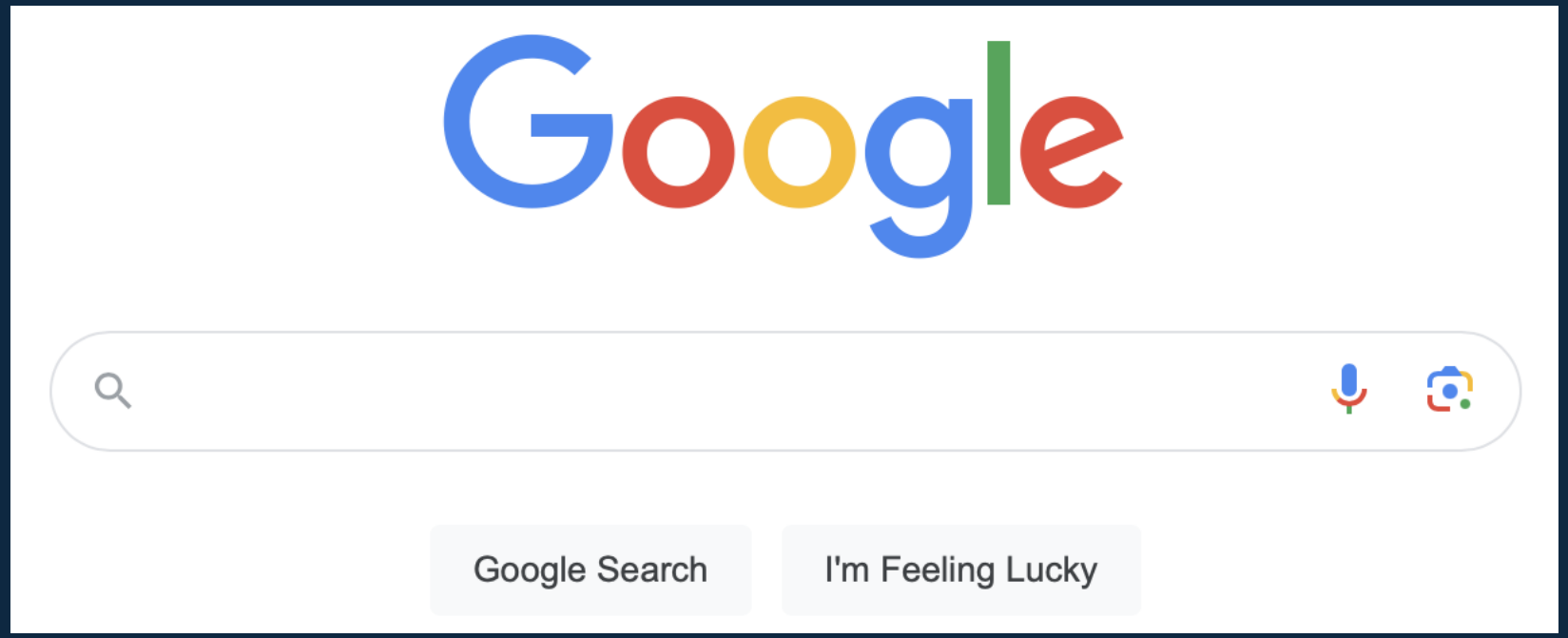
Today: 5 examples of interesting dimensions for PL elements

Cognitive Dimension Trade-offs



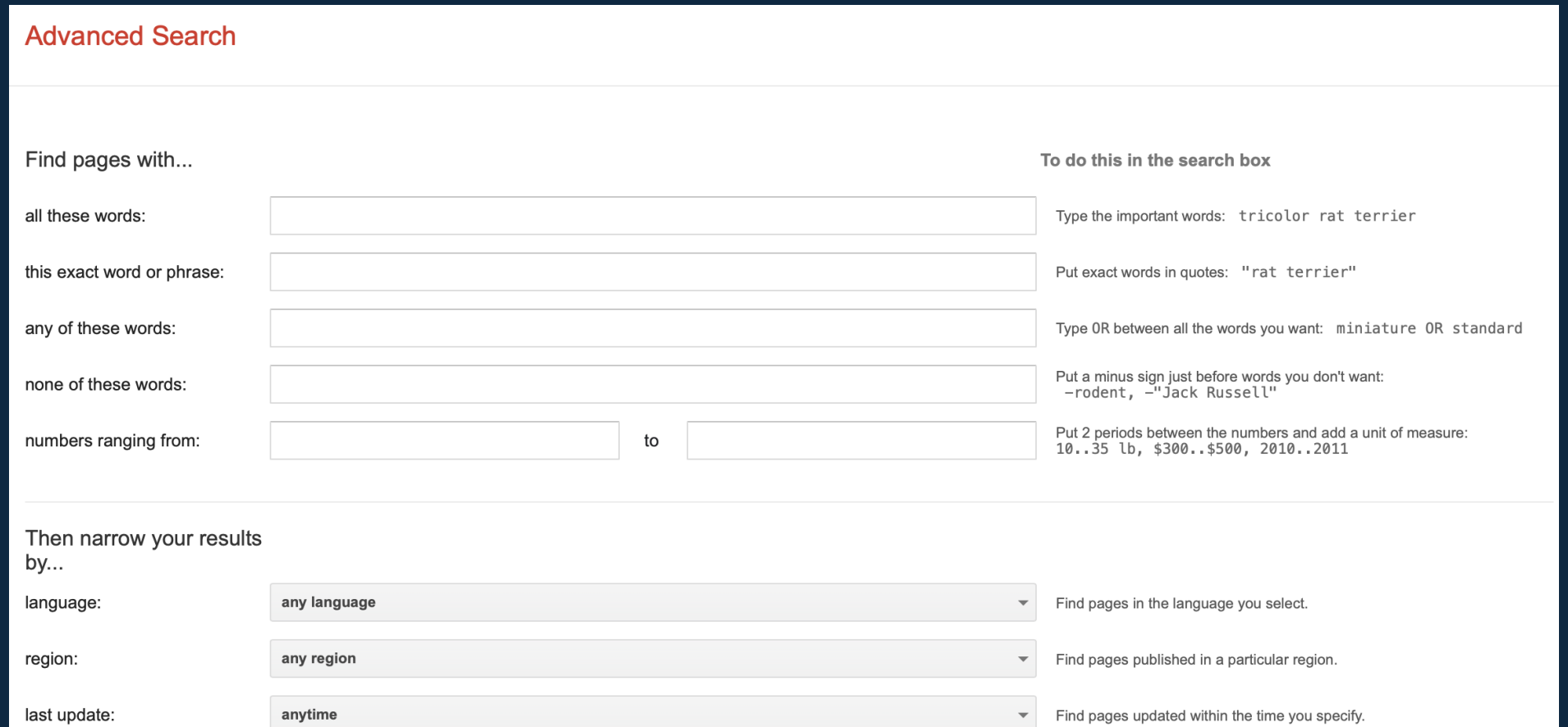
Abstraction Gradient

Abstraction Floor



Abstraction Gradient

Abstraction Ceiling

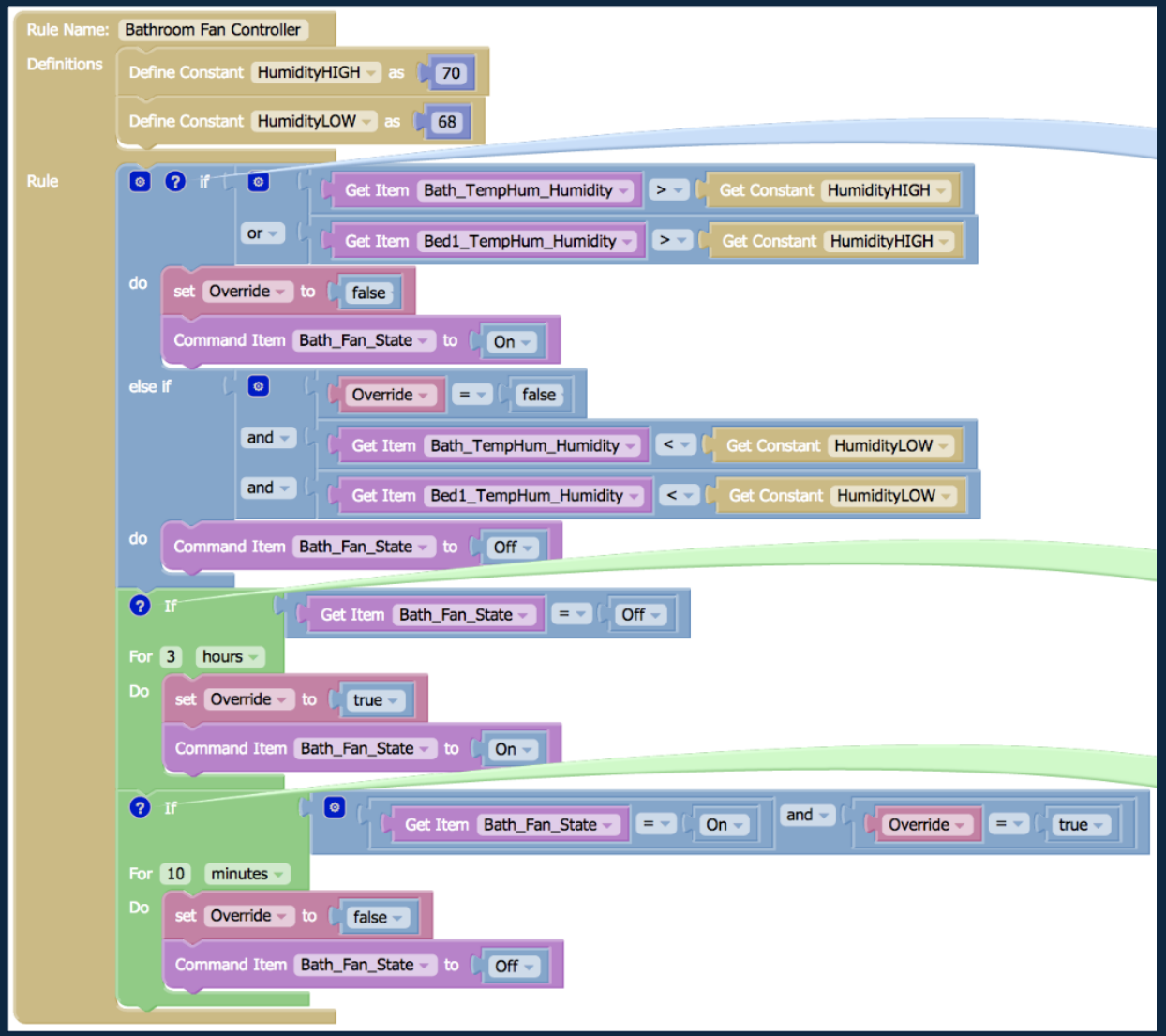


Abstraction Floor: Java vs. Python

```
print("Hello World!")
```

```
package org.example.test1;

class Test
{
    public static void main(String []args)
    {
        System.out.println("Hello World!");
    }
};
```

Low Abstraction Ceiling: Block-based Programming



Diffuseness
How many things are there to learn?

Java Language Keywords

Here is a list of keywords in the Java programming language. You cannot use any of the following as identifiers in your programs. The keywords `const` and `goto` are reserved, even though they are not currently used. `true`, `false`, and `null` might seem like keywords, but they are actually literals; you cannot use them as identifiers in your programs.

`abstract`	`continue`	`for`	`new`	`switch`
`assert`***	`default`	`goto`*	`package`	`synchronized`
`boolean`	`do`	`if`	`private`	`this`
`break`	`double`	`implements`	`protected`	`throw`
`byte`	`else`	`import`	`public`	`throws`
`case`	`enum`****	`instanceof`	`return`	`transient`
`catch`	`extends`	`int`	`short`	`try`
`char`	`final`	`interface`	`static`	`void`
`class`	`finally`	`long`	`strictfp`**	`volatile`
`const`*	`float`	`native`	`super`	`while`

Diffuseness

How many things are there to learn?

Attribute	Type	Default	Description
`answers-name`	string	—	Variable name to store data in. Note that this attribu
`weight`	integer	1	Weight to use when computing a weighted average
`correct-answer`	string	special	Correct answer for grading. Defaults to `data["corr`
`label`	text	—	A prefix to display before the input box (e.g., `label=`
`suffix`	text	—	A suffix to display after the input box (e.g., `suffix='`
`display`	"block" or "inline"	"inline"	How to display the input field.
`remove-leading-trailing`	boolean	false	Whether or not to remove leading and trailing blank
`remove-spaces`	boolean	false	Whether or not to remove blank spaces from the inp
`allow-blank`	boolean	false	Whether or not an empty input box is allowed. By de
`ignore-case`	boolean	false	Whether or not to enforce case sensitivity (e.g. "hell
`normalize-to-ascii`	boolean	false	Whether non-English characters (accents, non-latin a
`placeholder`	text	None	Hint displayed inside the input box describing the ex
`size`	integer	35	Size of the input box.
`show-help-text`	boolean	true	Show the question mark at the end of the input disp

Consistency

How much can you infer from what you already know?

Attribute	Type	Default	Description
`answers-name`	string	—	Variable name to store data in. Note that this attribu
`weight`	integer	1	Weight to use when computing a weighted average
`correct-answer`	string	special	Correct answer for grading. Defaults to `data["corr`
`label`	text	—	A prefix to display before the input box (e.g., `label=`
`suffix`	text	—	A suffix to display after the input box (e.g., `suffix="`
`display`	"block" or "inline"	"inline"	How to display the input field.
`remove-leading-trailing`	boolean	false	Whether or not to remove leading and trailing blank
`remove-spaces`	boolean	false	Whether or not to remove blank spaces from the inp
`allow-blank`	boolean	false	Whether or not an empty input box is allowed. By de
`ignore-case`	boolean	false	Whether or not to enforce case sensitivity (e.g. "hell
`normalize-to-ascii`	boolean	false	Whether non-English characters (accents, non-latin a
`placeholder`	text	None	Hint displayed inside the input box describing the ex
`size`	integer	35	Size of the input box.
`show-help-text`	boolean	true	Show the question mark at the end of the input disp

Attribute	Type	Default	Description
`answers-name`	string	—	Variable name to store data in. Note that this attribu
`weight`	integer	1	Weight to use when computing a weighted average
`correct-answer`	string	special	Correct answer for grading. Defaults to `data["corr name]`. If `base` is provided, then this answer must b
`allow-blank`	boolean	false	Whether or not an empty input box is allowed. By de
`blank-value`	integer	0 (zero)	Value to be used as an answer if element is left blank
`label`	text	—	A prefix to display before the input box (e.g., `label=`
`suffix`	text	—	A suffix to display after the input box (e.g., `suffix="`
`base`	integer	10	The base used to parse and represent the answer, or
`display`	"block" or "inline"	"inline"	How to display the input field.
`size`	integer	35	Size of the input box.
`show-help-text`	boolean	true	Show the question mark at the end of the input disp
`placeholder`	string	-	Custom placeholder text. If not set, defaults to "integ
`show-score`	boolean	true	Whether to show the score badge next to this eleme

Consistency

How much can you infer from what you already know?

```
> Array(16)
,,,,,,,,,,,,,,,
> Array(16).join("wat")
watwatwatwatwatwatwatwatwatwatwatwatwatwatwat
> Array(16).join("wat" + 1)
wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1
wat1
> Array(16).join("wat" - 1) + " Batman!"
```

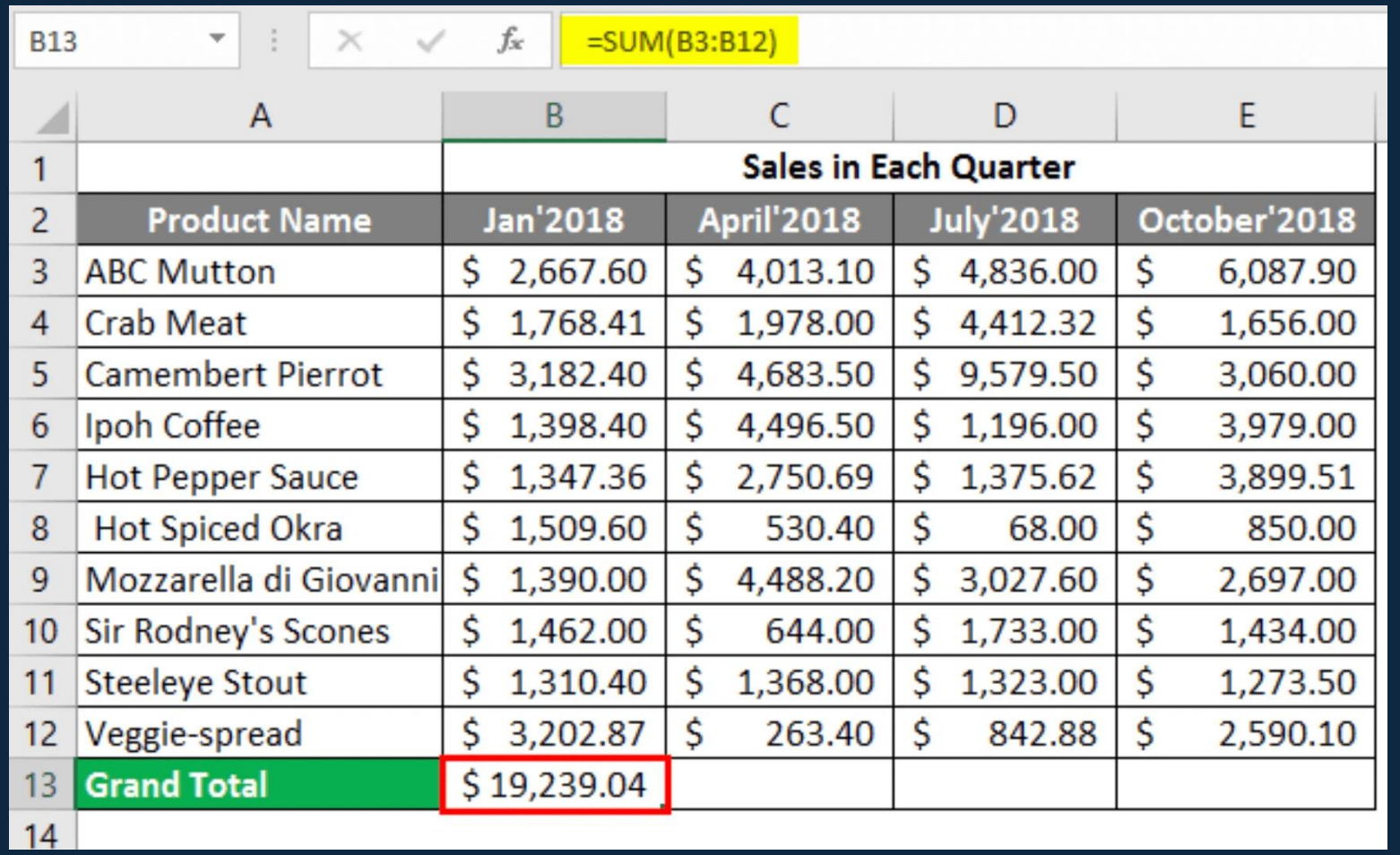
https://www.destroyallsoftware.com/talks/wat

Closeness of Mapping

```
<pl-order-blocks answers-name="order-numbers">
  <pl-answer correct="false">1</pl-answer>
  <pl-answer correct="true">2</pl-answer>
  <pl-answer correct="false">3</pl-answer>
  <pl-answer correct="true">4</pl-answer> </pl-order-blocks>
```

```
<pl-order-blocks
  answers-name="order-numbers”
  correct-answers=“2,4”
  incorrect-answers=“1,3”>
</pl-order-blocks>
```

Closeness of Mapping



	A	B	C	D	E
1		Sales in Each Quarter			
2	Product Name	Jan'2018	April'2018	July'2018	October'2018
3	ABC Mutton	$ 2,667.60	$ 4,013.10	$ 4,836.00	$ 6,087.90
4	Crab Meat	$ 1,768.41	$ 1,978.00	$ 4,412.32	$ 1,656.00
5	Camembert Pierrot	$ 3,182.40	$ 4,683.50	$ 9,579.50	$ 3,060.00
6	Ipoh Coffee	$ 1,398.40	$ 4,496.50	$ 1,196.00	$ 3,979.00
7	Hot Pepper Sauce	$ 1,347.36	$ 2,750.69	$ 1,375.62	$ 3,899.51
8	Hot Spiced Okra	$ 1,509.60	$ 530.40	$ 68.00	$ 850.00
9	Mozzarella di Giovanni	$ 1,390.00	$ 4,488.20	$ 3,027.60	$ 2,697.00
10	Sir Rodney's Scones	$ 1,462.00	$ 644.00	$ 1,733.00	$ 1,434.00
11	Steeleye Stout	$ 1,310.40	$ 1,368.00	$ 1,323.00	$ 1,273.50
12	Veggie-spread	$ 3,202.87	$ 263.40	$ 842.88	$ 2,590.10
13	Grand Total	$ 19,239.04			
14					

Viscosity
Resistance to change

```
def numToLetter(x):
    if (x == 1)
        return "A"
    if (x == 2)
        return "B"
    if (x == 3)
        return "C"
    if (x == 4)
        return "D"
        ...
```

- Suppose you want to change the name of “x”
- Suppose you want to add a 2nd parameter to “numToLetter”

Viscosity
Resistance to change

```
def numToLetter(x):
    if (x == 1)
        return "A"
    if (x == 2)
        return "B"
    if (x == 3)
        return "C"
    if (x == 4)
        return "D"
        ...
```

Repetition Viscosity:

- The same change must be repeated over and over
- Usually caused by redundancy

Knock-on Viscosity:

- One change triggers a series of follow-up changes
- Usually caused by coupling

Other Notable Cognitive Dimensions

Error-proneness

- How “easy” is it for users to make mistakes?

Secondary Notation

- Can users “escape” your design (e.g. via comments)?

Premature Commitment

- Do users need to make uninformed decisions?

… (see paper for the rest)