[05]

List Lacing

Processing data effectively

THE PRESENTER

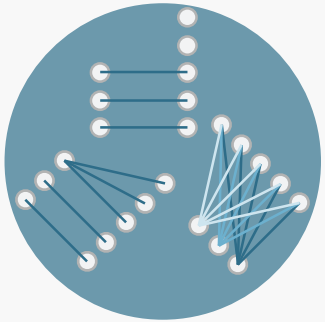
Jacob Small
Designated Support Specialist: Generative Design & BIM

@JacobWSmall

SAFE HARBOUR

During the course of this presentation, we may make statements regarding future events and/or statements regarding planned or future development efforts for our existing or new products and services. We wish to caution you that such statements reflect our current expectations, estimates and assumptions based on factors currently known to us and that actual events or results could differ materially. Also, these statements are not intended to be a promise or guarantee of future delivery of products, services or features but merely reflect our current plans, which may change. Purchasing decisions should not be made based upon reliance on these statements. The statements made in this presentation are being made as of the time and date of its live presentation. We do not assume any obligation to update any statements we make to reflect events that occur or circumstances that exist after the date of this presentation.

AGENDA



01.

15 Minute Presentation:

List Lacing

02.

Questions & Answers

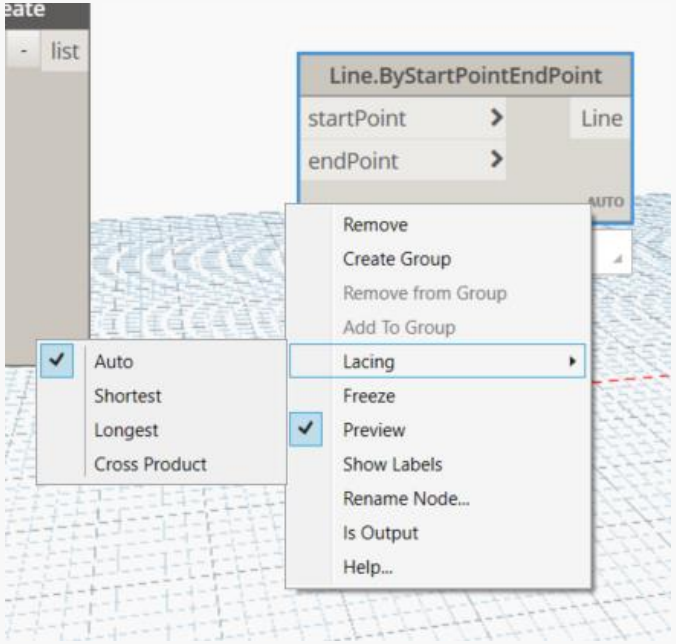
03.

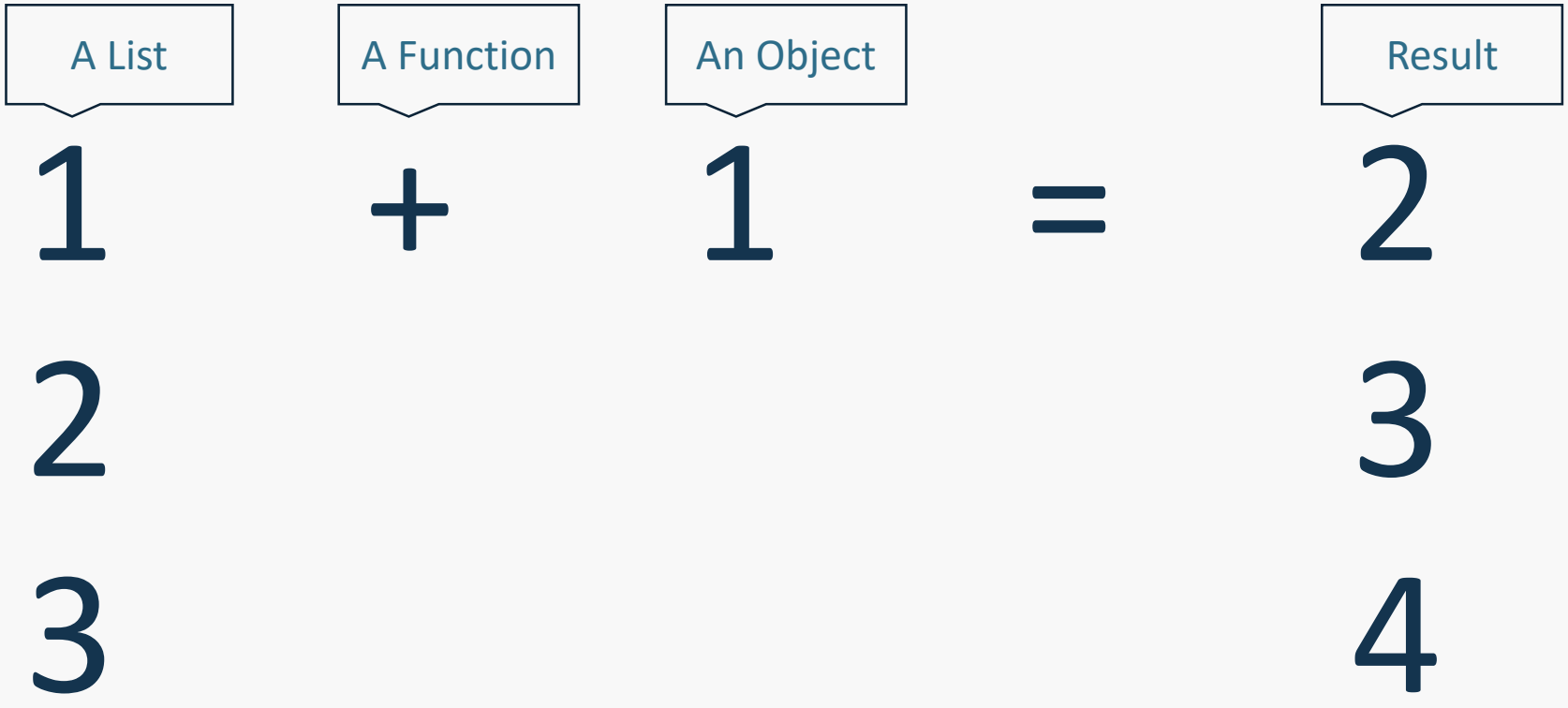
Bonus:

List Levels

Lacing

- Each node is a FUNCTION and can iterate over a list of input, this is how Visual Programming handles “for loops”
- If the function has multiple inputs, the lacing strategy on the node is used to combine the input in an understandable order
- The lacing affects the structure of the output



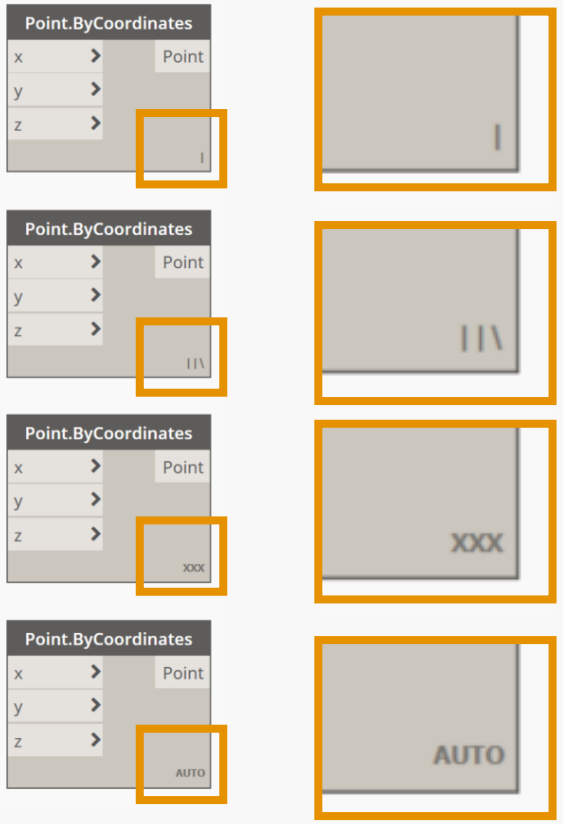


A List	A Function	A List		Result
1	+	1	=	2
2		2		4
3		3		6

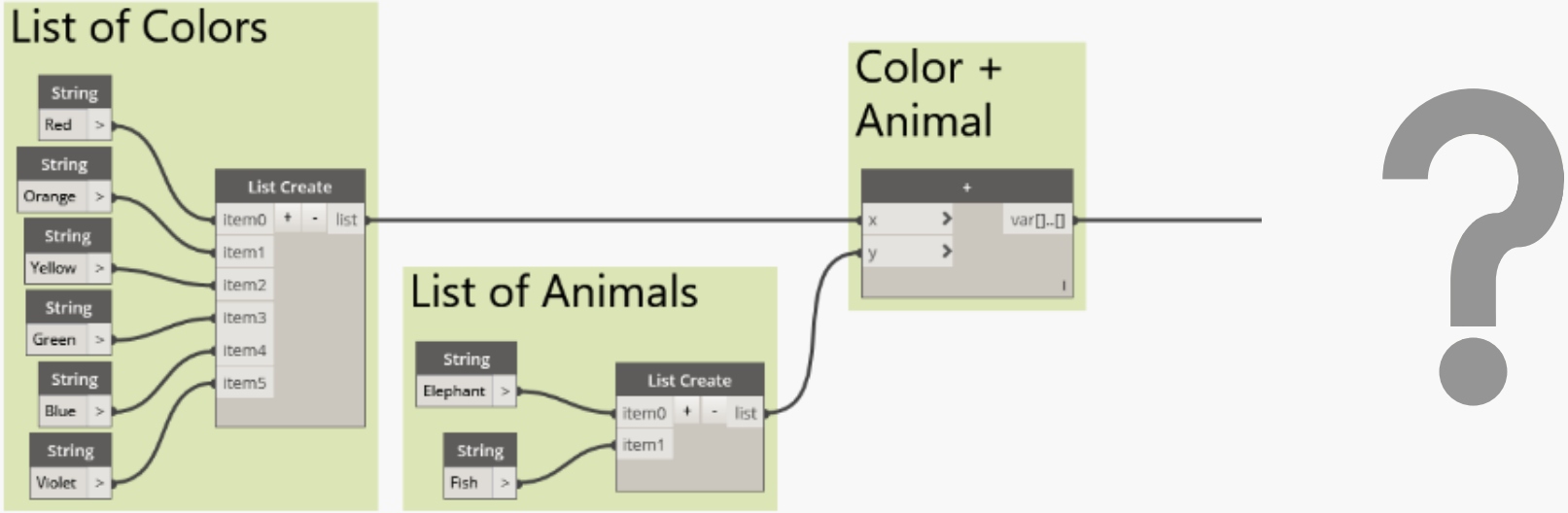
A List	A Function	A List		Result
1	+	1	=	2
2		2		4
3				???

Lacing

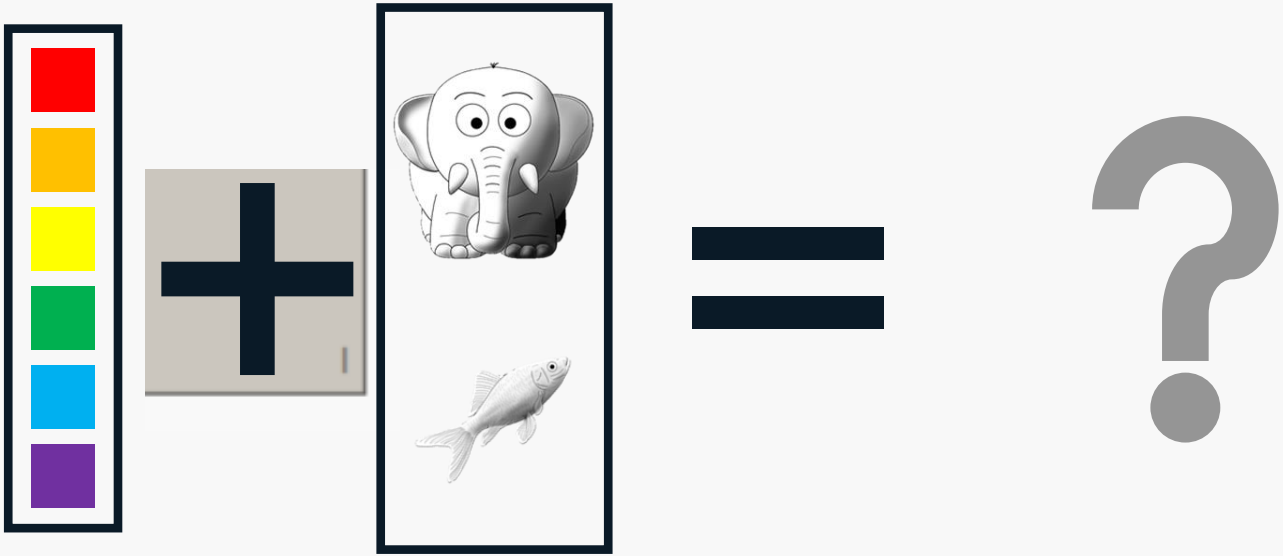
- Shortest
  - Matches index to index, discarding items from the longer list.
- Longest
  - Matches index to index and reuses the last item in the shorter list.
- Cross Product
  - Matches every item in list one to every item in list two.
- Automatic
  - Applies shortest lacing, but ‘pushes’ single items as if longest lacing were selected.
  - This is the default!



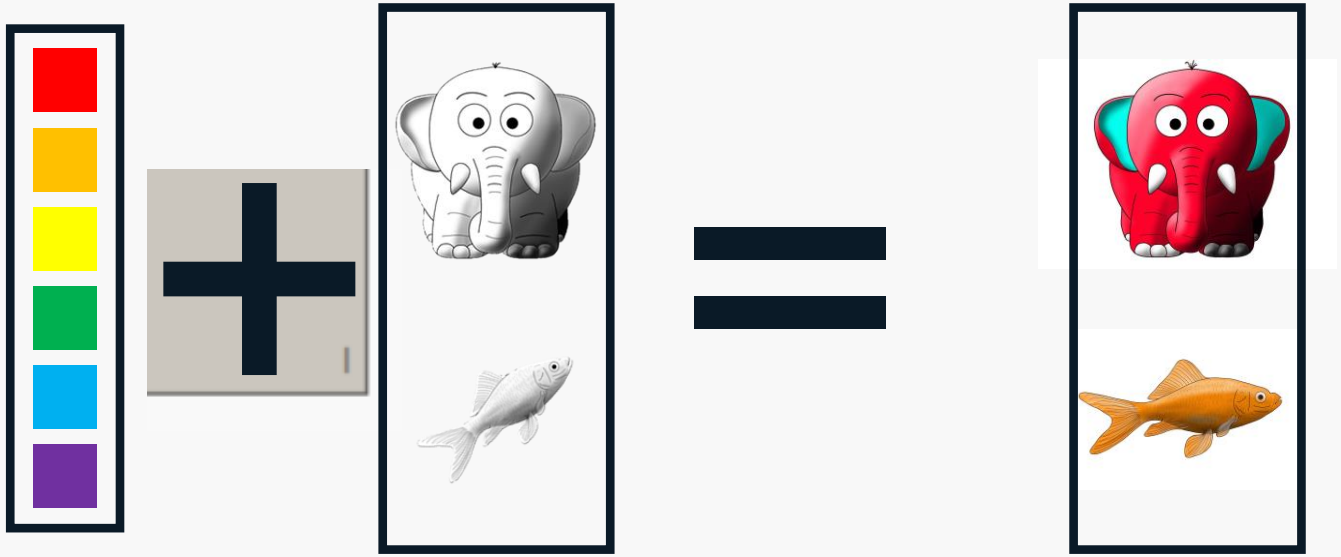
Lacing



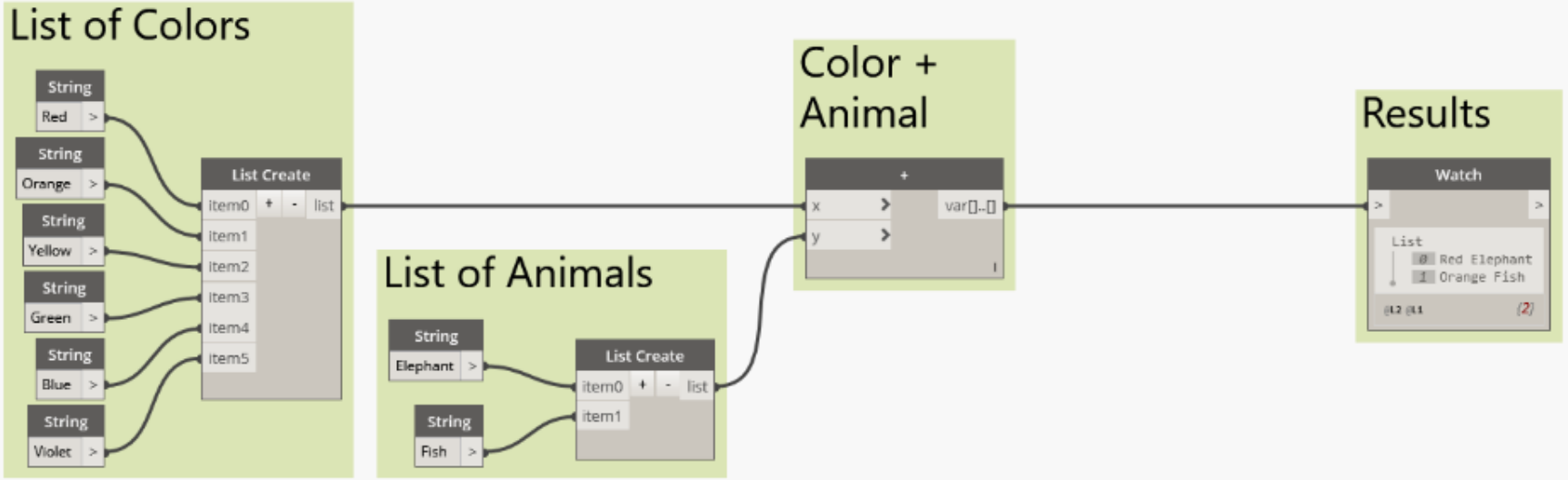
Lacing



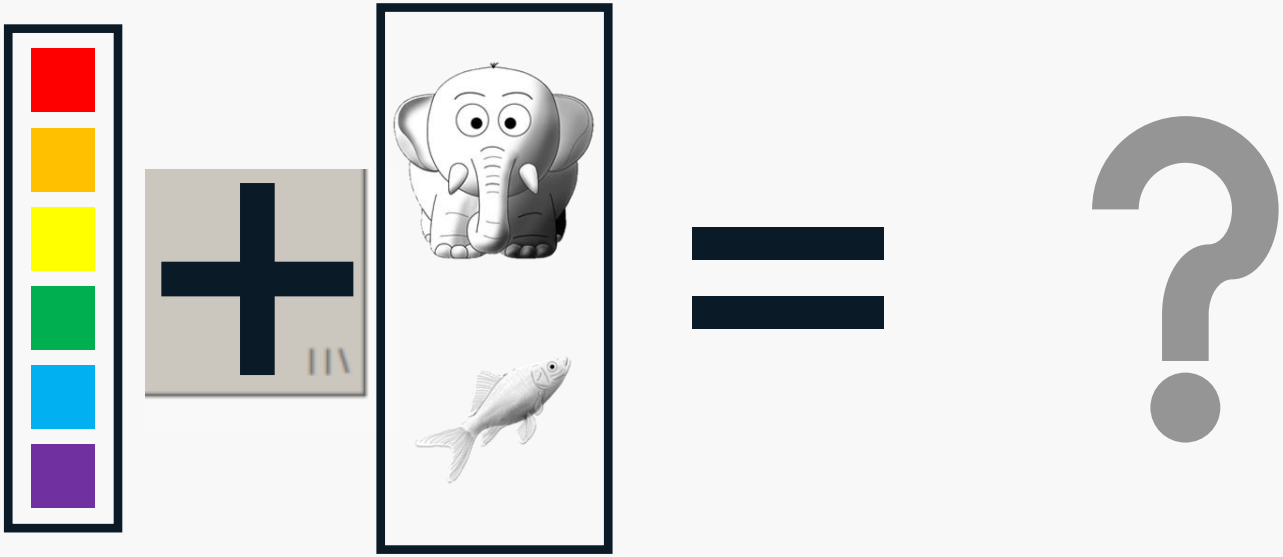
Lacing - Shortest



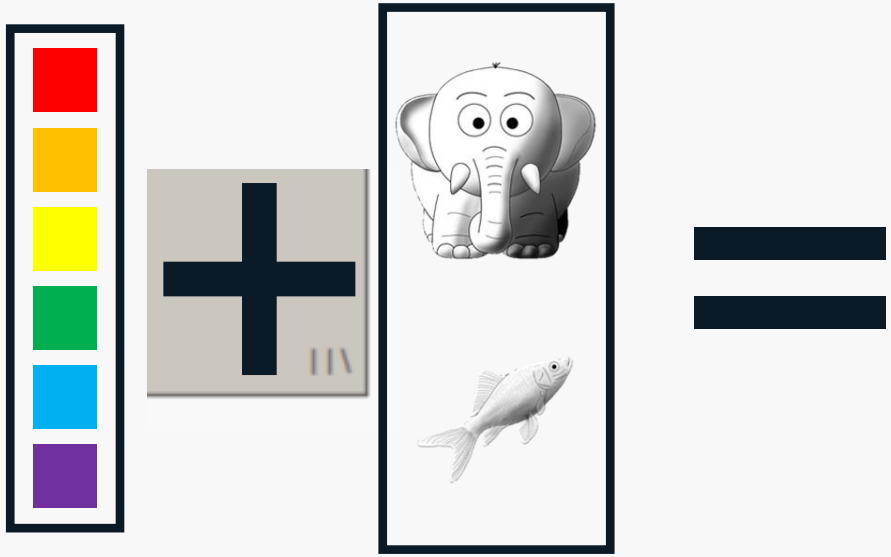
Lacing



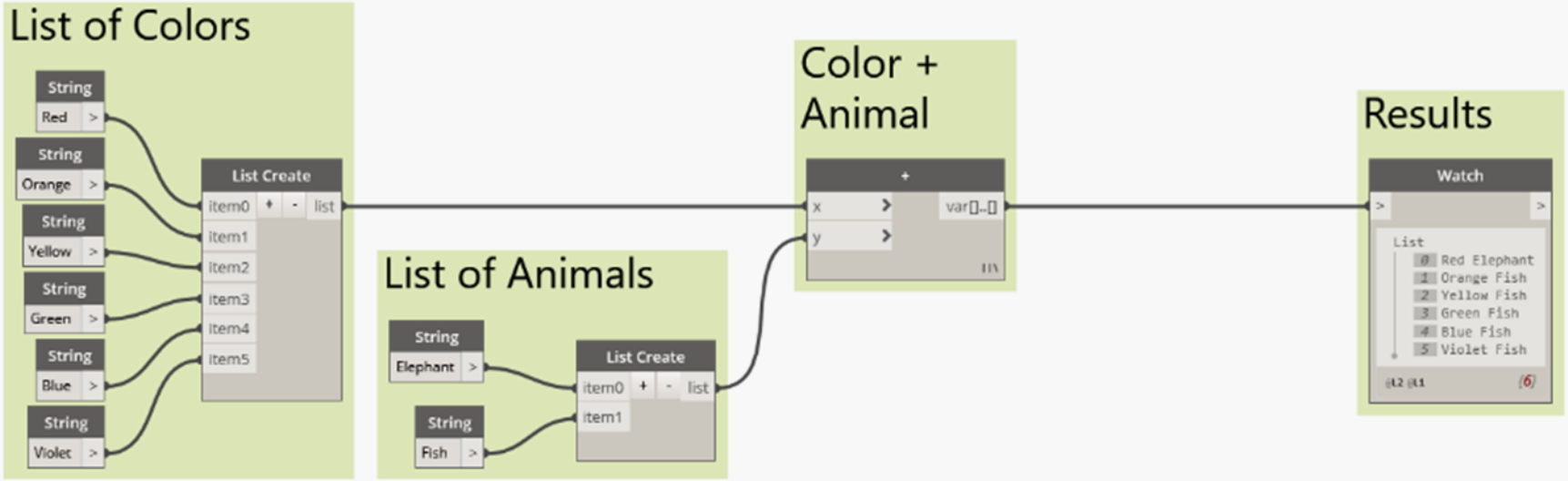
Lacing



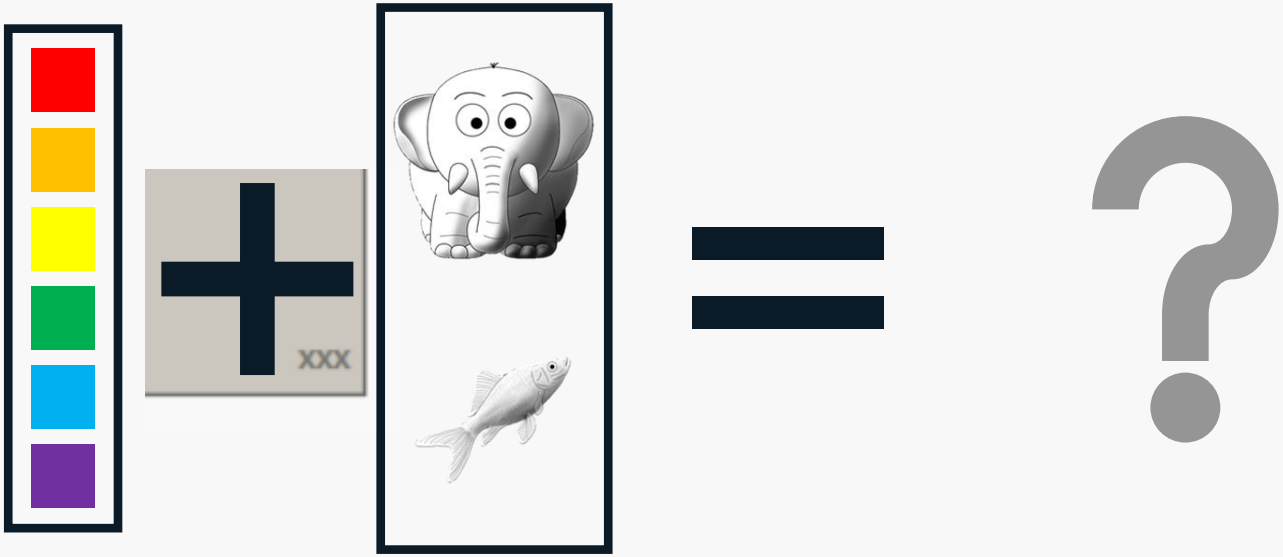
Lacing - Longest



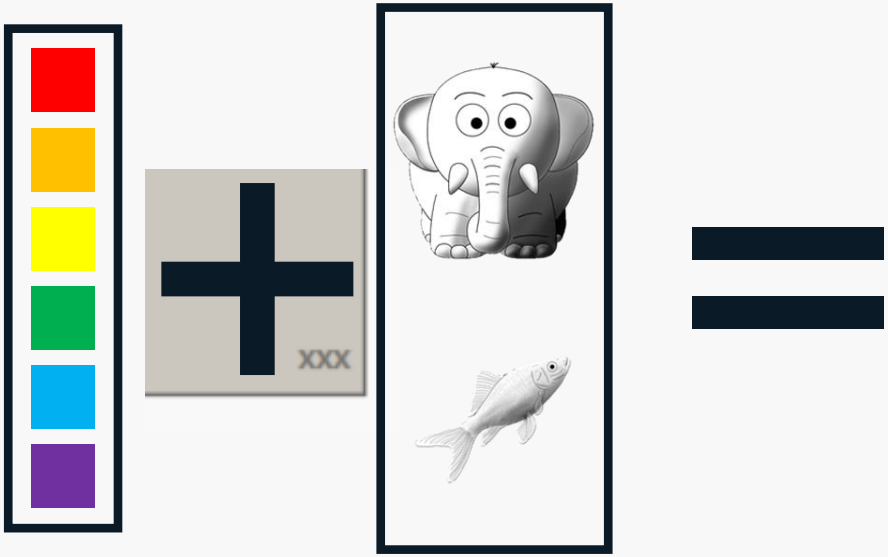
Lacing

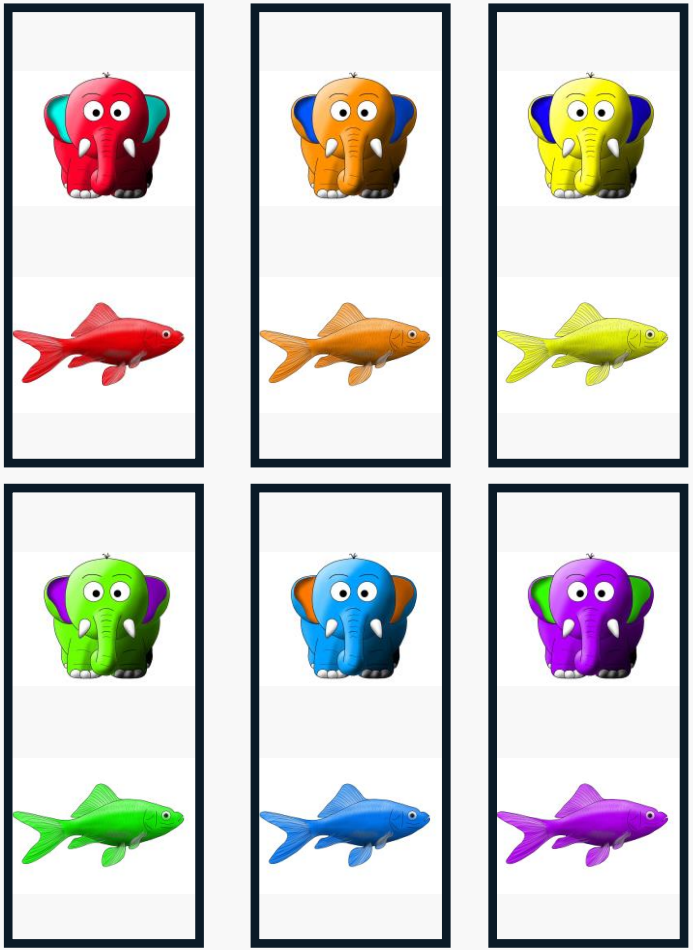


Lacing

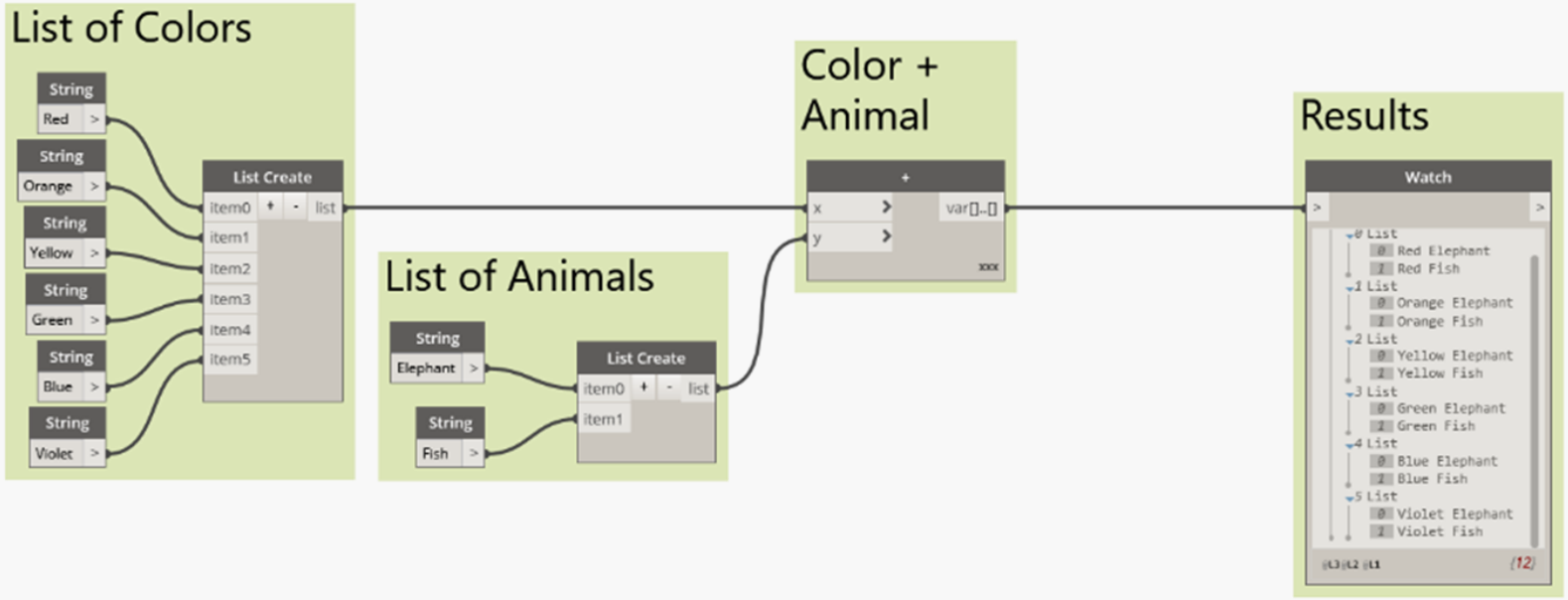


Lacing – Cross Product

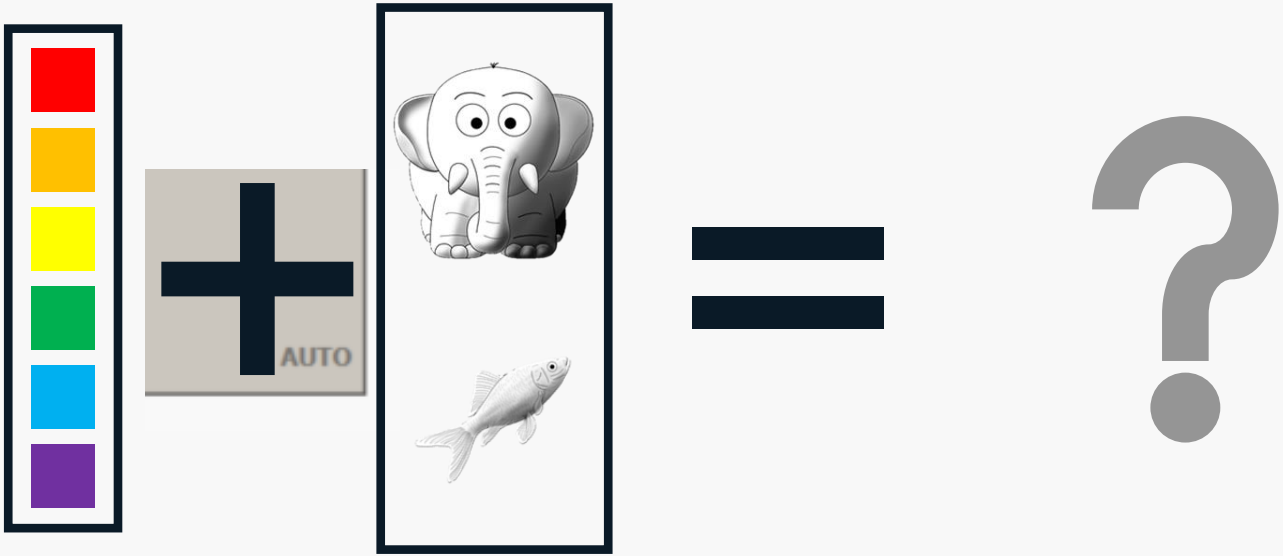




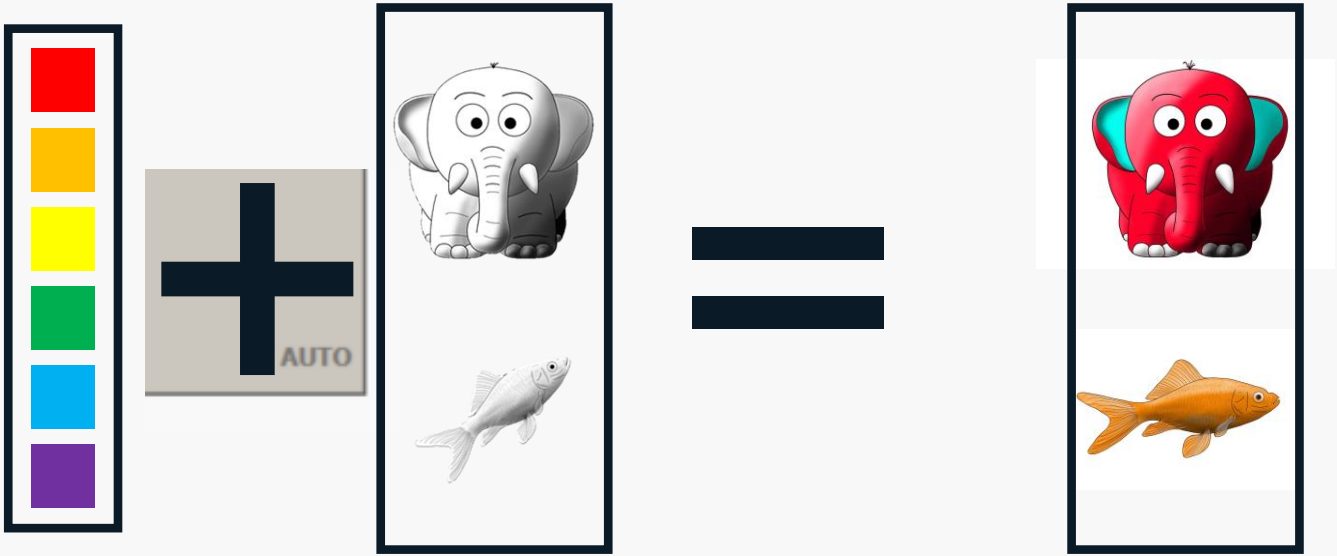
Lacing



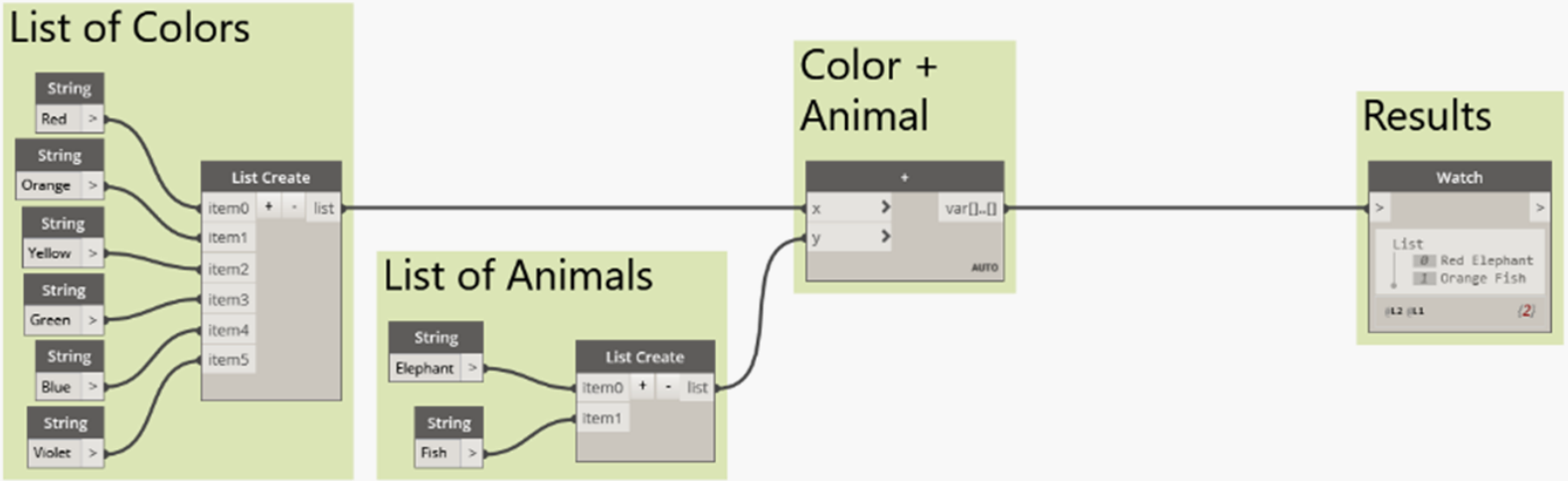
Lacing



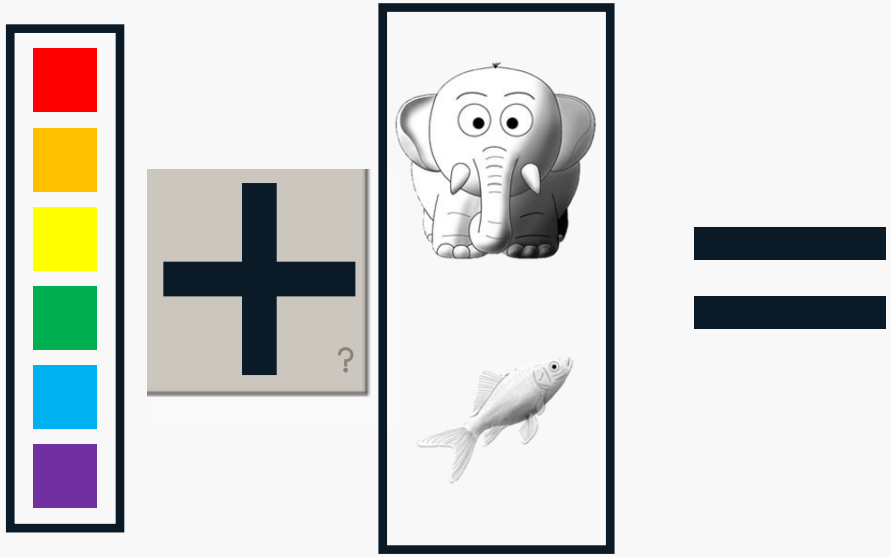
Lacing - Auto



Lacing

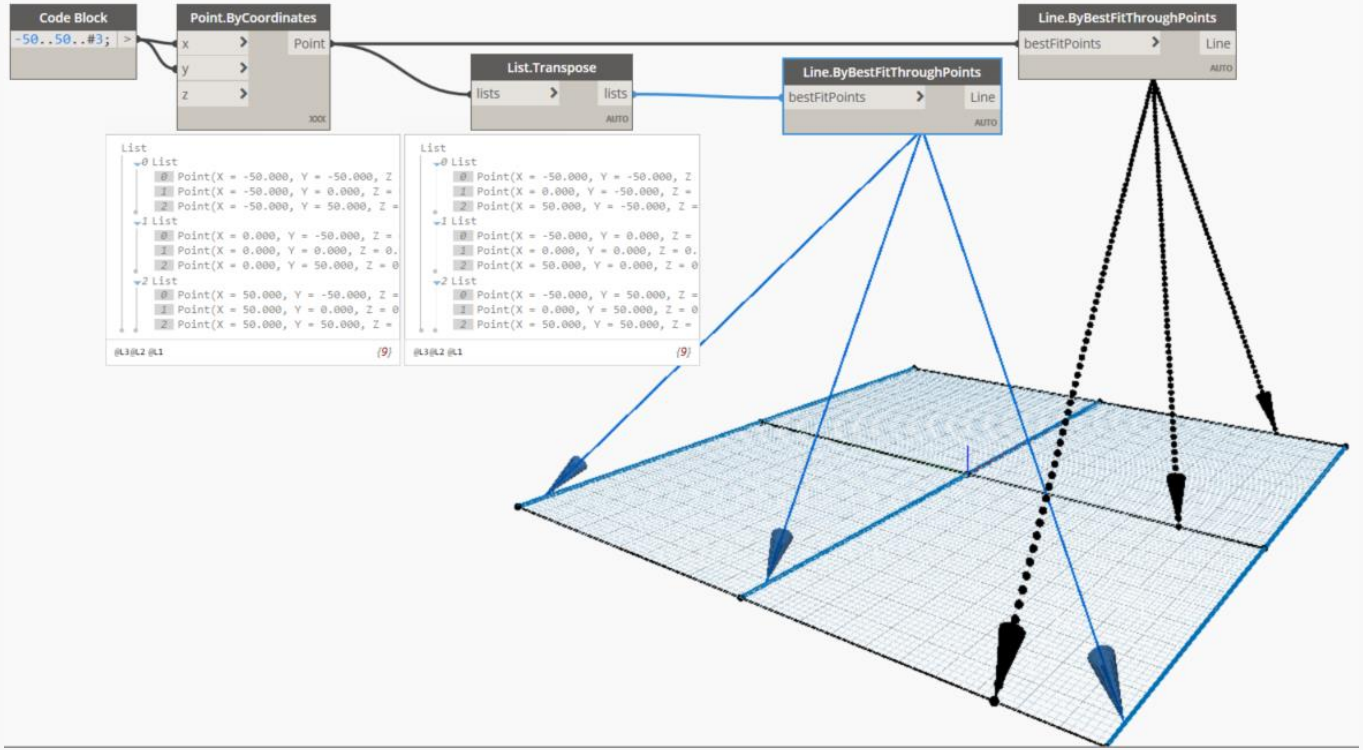


Lacing – ???

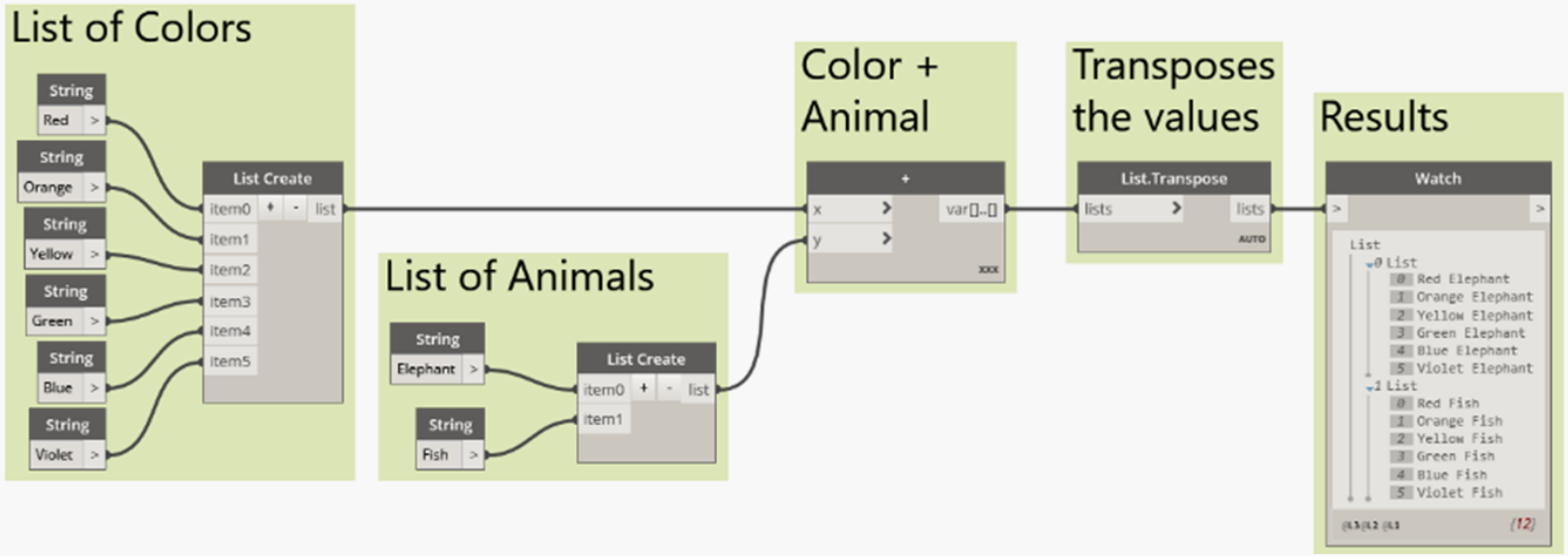


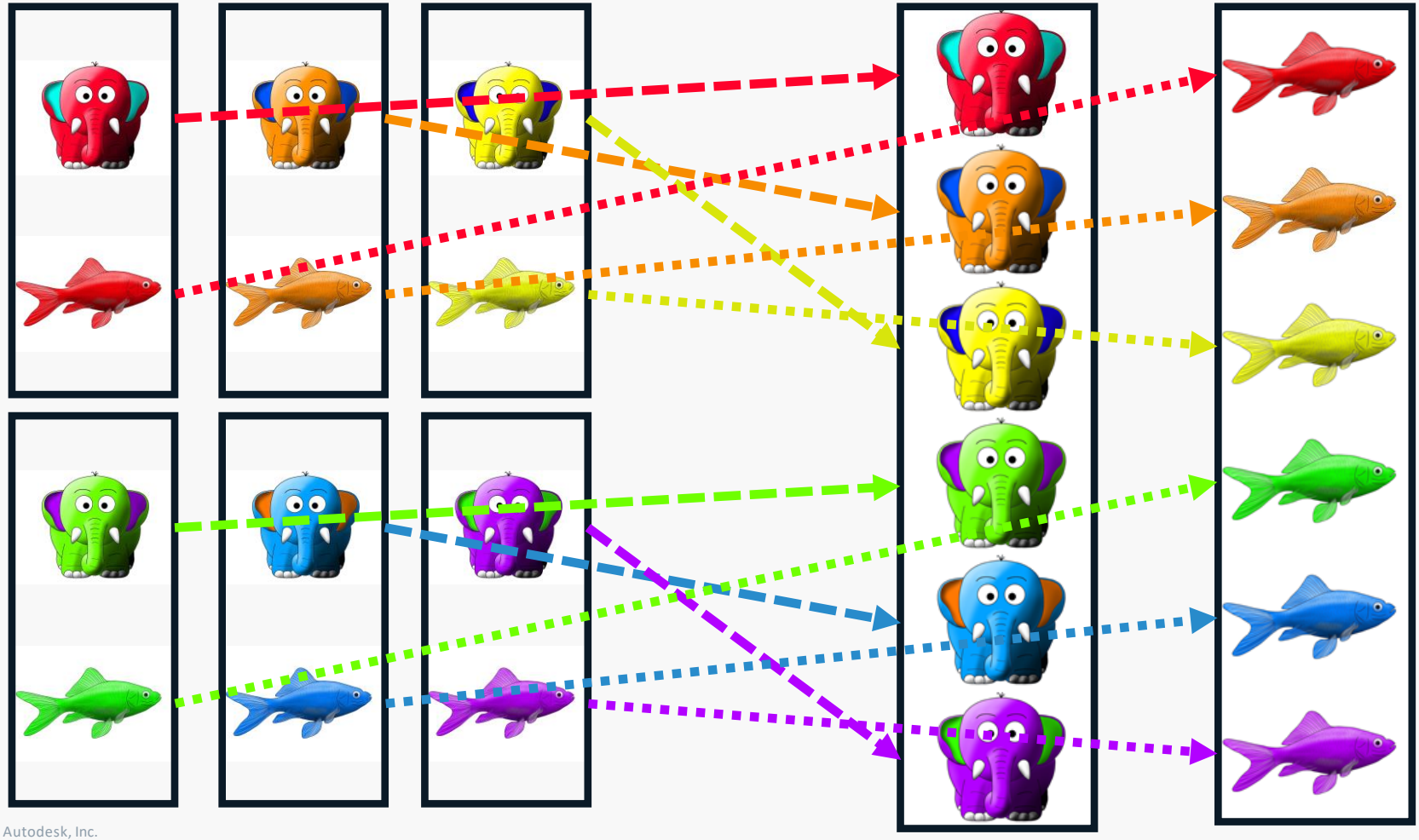
Working with Lists

Transpose



Lacing





LEARNING RESOURCES

Bookmark these in your preferred browser

Dynamo Info / News

- Main: http://dynamobim.org/
- Blog: https://dynamobim.org/blog/
- Dynamo Builds: http://dynamobuilds.com/
- Dynamo GitHub: https://github.com/DynamoDS/Dynamo

Dynamo Learning

- Dynamo Primer: http://primer.dynamobim.org
- Dynamo Forums: https://forum.dynamobim.com/
- Dynamo Dictionary: https://dictionary.dynamobim.com
- Dynamo Nodes: https://dynamonodes.com/

Design Script:

- Design Script Language Summary: http://designscript.io/DesignScript_user_manual_0.1.pdf
- Design Script Language Guide: https://dynamobim.org/wp-content/links/DesignScriptGuide.pdf
- Design Script Presentation: https://github.com/Amoursol/dynamoDesignScript

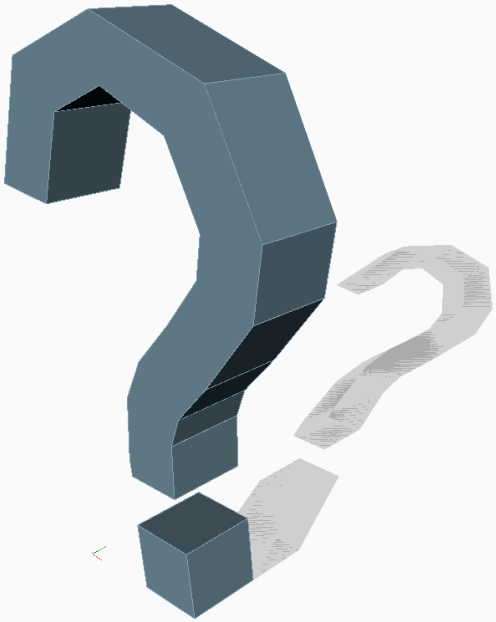
Dynamo Python:

- Python for Dynamo AU Lab Handout 1: https://docs.google.com/document/d/1_ms_ZyyKoaWbWbcio3CN5qTI2Ywq4OJep3SIsWOXXuI/edit#heading=h.76xq4kaz9plc
- Python for Dyanmo AU Lab Handout 2: https://github.com/Amoursol/dynamoPython/blob/master/images/DivingDeeper_ABeginnersLookAtPythonInDynamo_AU_London2018.pdf
- Python for Dynamo examples: https://github.com/Amoursol/dynamoPython

Generative Design:

- Info: https://www.autodesk.com/campaigns/refinery-beta
- Generative Design Primer: https://www.generativedesign.org/
- Beta Site: https://feedback.autodesk.com/key/RefineryLanding

Questions?



AUTODESK®

Make anything™