Tony Veale presents

A Game Of Tropes

A Song of Novelty and Reuse


A trope is a **recurring pattern of use** that defines a **genre**.

When we refer to the "**genre**" or "**style**" or "**paradigm**" in Art, Science or Commercial Design, we are referring to this idea of **Recurring Patterns** ...

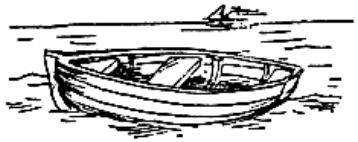
Tropes codify the **identifiable norms** of good design in a domain, and apply in many disciplines.

Even "Rule-Breakers" Follow Rules

LES MOTS ET LES IMAGES (1928)

Un objet ne tient pas tellement à son nom qu'on ne puisse lui en trouver un autre qui lui convienne mieux

Il y a des objets qui se passent de nom :



Un mot ne sert parfois qu'à se désigner soi-même :

Un objet rencontre son image, un objet rencontre son nom. Il arrive que l'image et le nom de cet objet se rencontrent ·

Parfois le nom d'un objet tient lieu d'une image

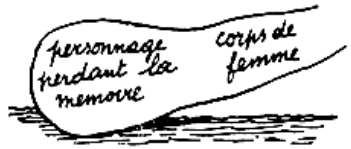
Un mot peut prendre la place d'un objet dans la réalité :

Une image peut prendre la place d'un mot dans une proposition ·

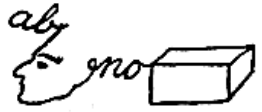
Un objet fait supposer qu'il y en a d'autres derrière lui :

Tout tend à faire penser qu'il y a peu de relation entre un objet et ce qui le représente

Les mots qui servent à désigner deux objets différents ne montrent pas ce qui peut séparer ces objets l'un de l'autre

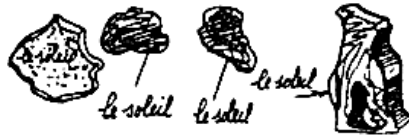


Dans un tableau, les mots sont de la même substance que les images

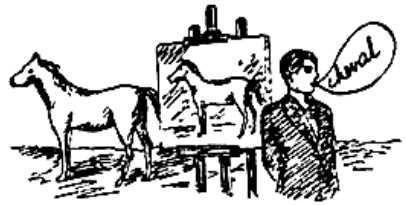


On voit autrement les images et les mots dans un tableau ·

Une forme quelconque peut remplacer l'image d'un objet



Un objet ne fait jamais le même office son nom ou que son image



Or, les contours visibles des objets, dans la réalité, se touchent comme s'ils formaient une mosaïque :

Les figures vagues ont une signification aussi nécessaire aussi parfaite que les précises ·

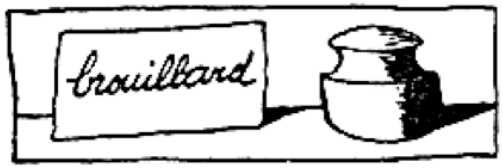
Parfois, les noms écrits dans un tableau désignent des choses précises, et les images des choses vagues ·

Ou bien le contraire :

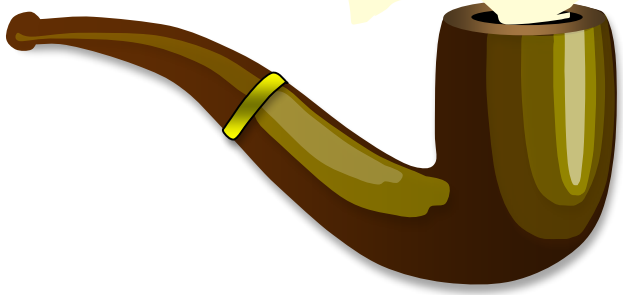
Un mot peut prendre la place d'un objet dans la réalité :

On voit autrement les images et les mots dans un tableau ·

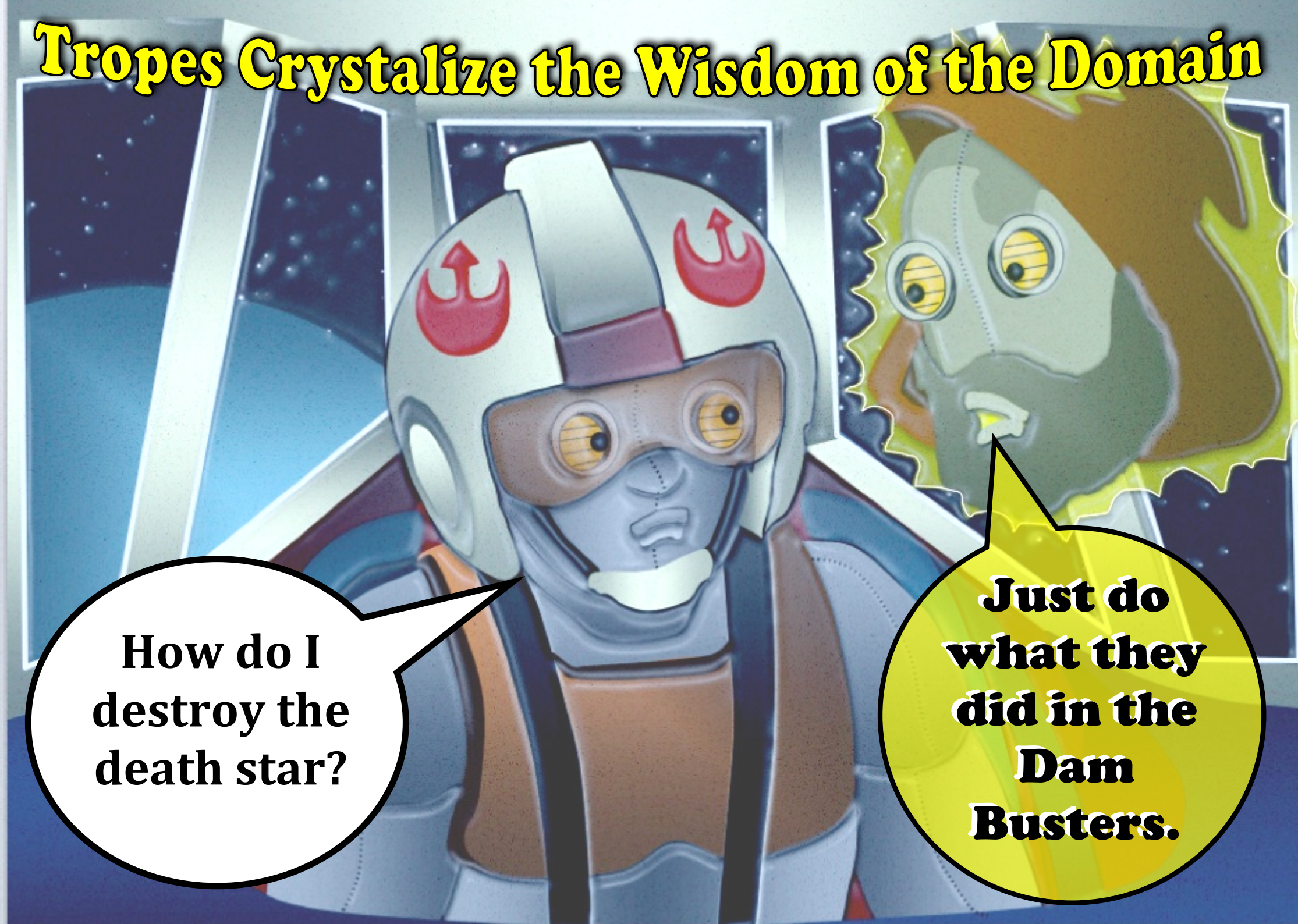
Ou bien le contraire :



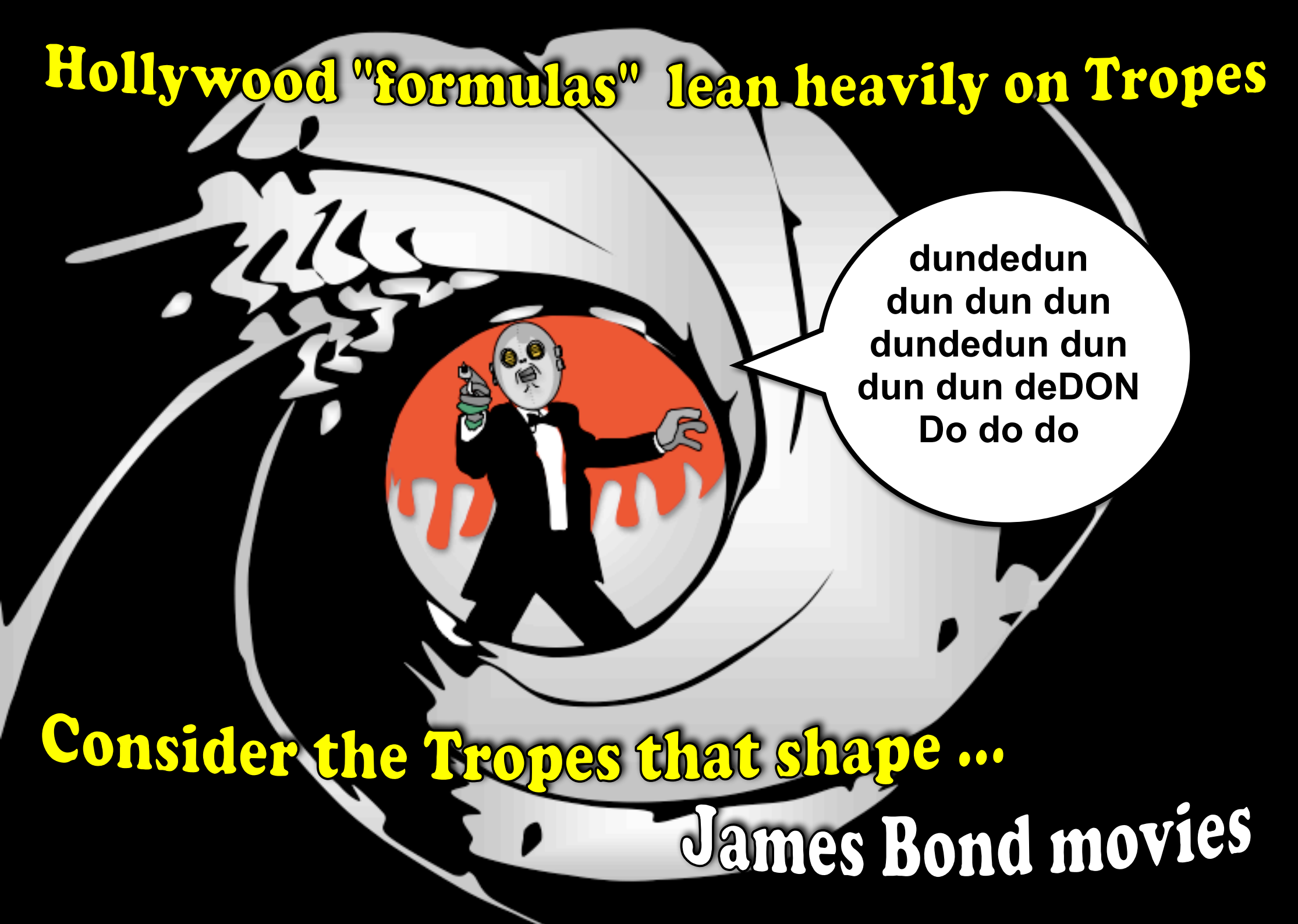
Rene Magritte's *Surrealist Tropes*



Tropes Crystalize the Wisdom of the Domain



Hollywood "formulas" lean heavily on Tropes



Consider the Tropes that shape ...

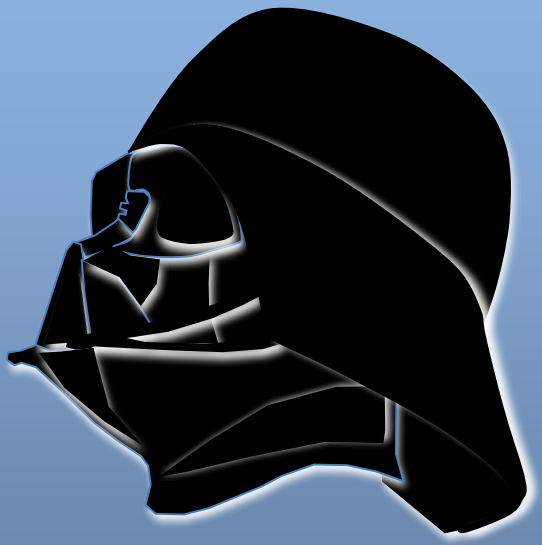
James Bond movies



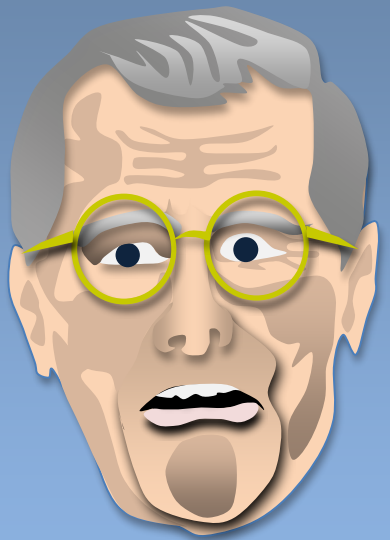
First, Choose a Dastardly Villain ...







Next, Choose a Fearsome Henchman ...



Then, Pick a Beautiful or Exotic Setting ...
... and threaten it with imminent destruction by an inventive means.

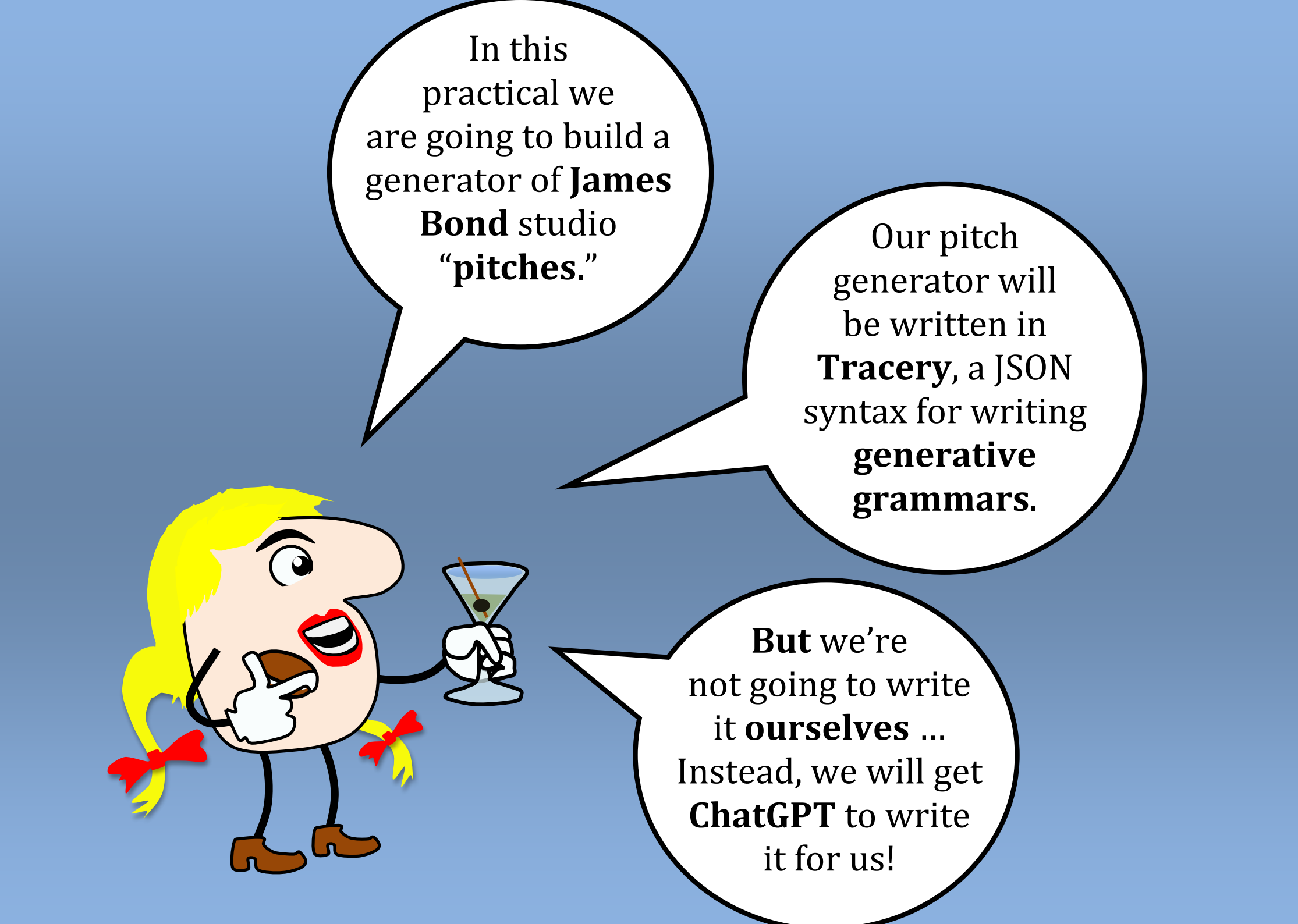

Add some fancy gadgets and a "McGuffin" ...

And Beautiful Sidekicks with Suggestive Names ...

(and deadly skills)

Top with Deadly Traps and Edge-of-Seat Escapes


The Bond "Formula" Ties All Of These Tropes Together
Into A Competent No-Surprises Movie
dundedun dun dun dun dundedun dun dun dun deDON Do do do


In this practical we are going to build a generator of **James Bond** studio "**pitches**."
Our pitch generator will be written in **Tracery**, a JSON syntax for writing **generative grammars**.
But we're not going to write it **ourselves** … Instead, we will get **ChatGPT** to write it for us!

My Grammar

login landscape json


```
{
"origin":["[myPlace:#path#]#line#"],
"line":["#mood.capitalize# and #mood#, the
#myPlace# was #mood# with #substance#",
"#nearby.capitalize# #myPlace.a# #move.ed# through
the #path#, filling me with #substance#"],
"nearby":["beyond the #path#", "far away", "ahead",
"behind me"],
"substance":["light", "reflections", "mist", "shadow",
"darkness", "brightness", "gaiety", "merriment"],
"mood":["overcast", "alight", "clear", "darkened", "blue",
"shadowed", "illuminated", "silver", "cool", "warm",
"summer-warmed"],
"path":["stream", "brook", "path", "ravine", "forest",
"fence", "stone wall"],
"move":["spiral", "twirl", "curl", "dance", "twine",
"weave", "meander", "wander", "flow"]
}
```


expansion r, the path was summer-warmed with shadow

origin

line

myPlace:#path#	mood		mood		myPlace	mood		substance
path	darkened	and	silver	, the	path	was	summer-warmed	with	shadow

Set your browser to **tracery.io/editor** for today's practical

origin 1 reroll step 27592628

Darkened and silver, the path was summer-warmed with shadow

This site was built by the creator of **Tracery**, Kate Compton.

Start your grammar with a { to begin in JSON format …

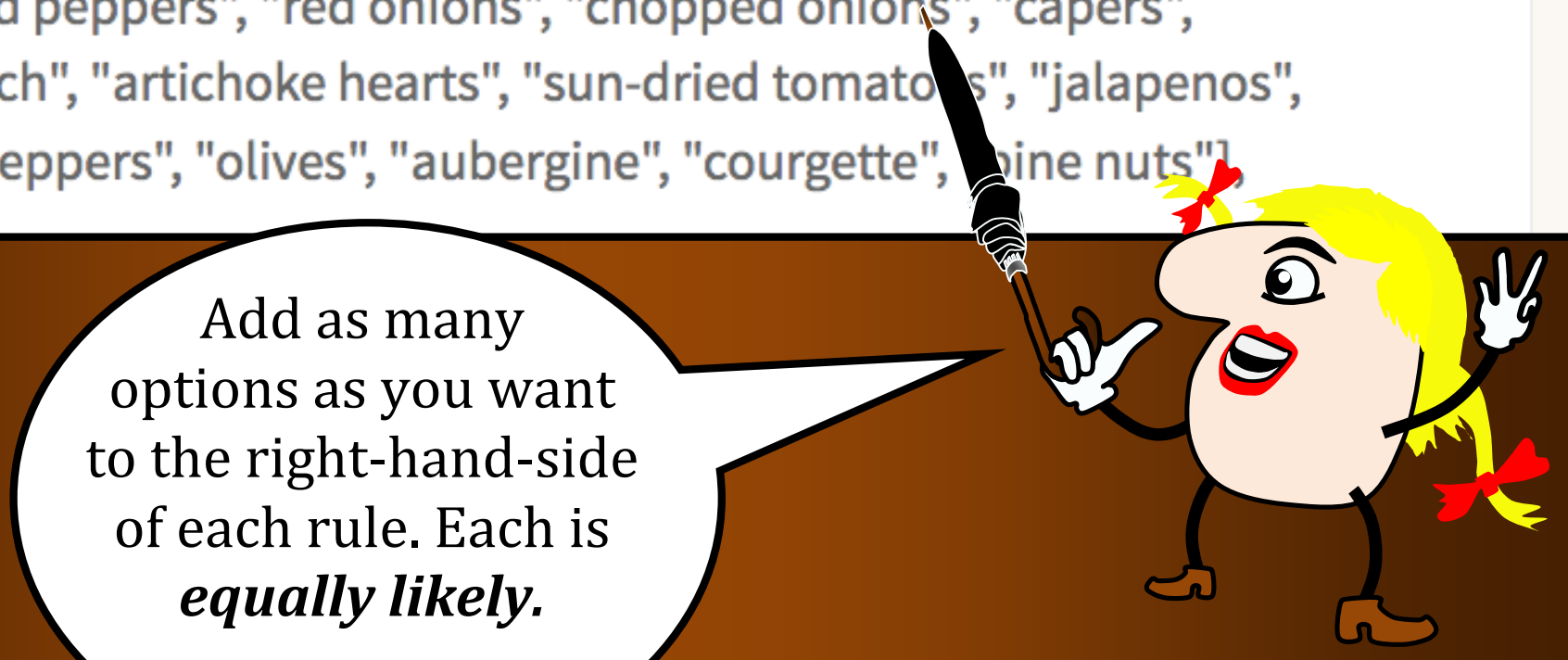
Tracery JSON

```
{
"origin": ["#pizza#"],

"pizza": ["#meat# and #veggie# with #cheese# and #sauce# on #base#."],
```

The main rule of your **Tracery** grammar is called "**origin**." The name of each rule is a ***non-terminal symbol*** that can be expanded using the right-hand-side. We refer to non-terminals by encasing them in **# … #** on the right-hand side of a rule.

Next, write rules to expand the non-terminals **#meat#** and **#veggie#** in our earlier rule for making a pizza

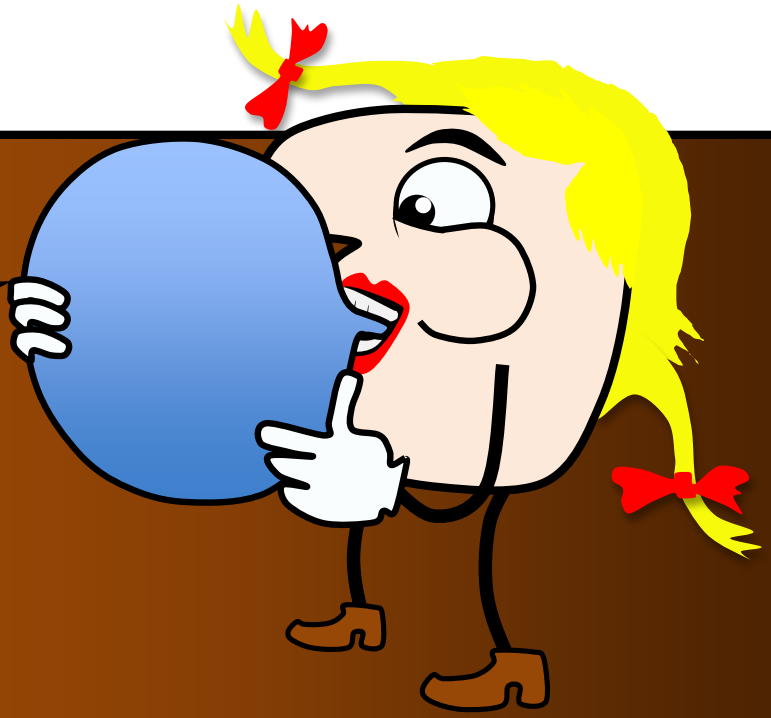
```
"meat": ["sausage", "meatballs", "grilled chicken", "lamb kofta",
         "Tandoori chicken", "roast beef", "Korean beef", "steak tartare",
         "tiger prawns", "king prawns", "fish balls", "tuna chunks",
         "smoked salmon", "anchovies", "ham", "Parma ham", "bacon",
         "pastrami"],

"veggie": ["grilled peppers", "red onions", "chopped onions", "capers",
           "spinach", "artichoke hearts", "sun-dried tomatoes", "jalapenos",
           "bell peppers", "olives", "aubergine", "courgette", "pine nuts"],
```



Ditto for the non-terminals **#cheese#** and **#sauce#**
Add as much variety as your imagination will allow …

```
"cheese": ["feta cheese", "ricotta cheese", "buffalo mozzarella", "mature cheddar",
           "cottage cheese", "Swiss cheese", "Queso Fresca", "goats' cheese",
           "smoked tofu", "mascarpone", "Romano cheese", "provolone cheese",
           "camembert", "gorgonzola", "Stilton", "chevre"],

"sauce": ["marinara sauce", "tomato sauce", "barbecue sauce", "wine sauce",
          "ragu sauce", "bechamel"],
```

Whenever ***Tracery*** expands a rule, it **randomly picks** from the right-hand side options for each **non-terminal**.



We now write a rule for the last non-terminal, **#base#**, and close then our grammar with a final **}**

```
"base": ["a thin crust", "a thick crust", "a gluten-free crust", "a deep-pan base",
         "focaccia", "ciabatta", "puff pastry crust", "a stuffed crust",
         "a folded crust", "a wholegrain crust", "a polenta crust"]

}
```

Every **non-terminal** in every rule now has its own **explicit rule** definition, so our grammar is ready to **start making** pizzas.

My Grammar login landscape json


```
{
"origin":["#pizza#"],
"pizza":["I made this pizza with #Meat# and #Vegetable# as toppings on #Cheese# and #Sauce# on a #Base#."],
"Meat":["Swedish meatballs", "Filet Mignon", "Beef jerky", "Roast beef", "Korean beef", "Taco beef", "Chicken Fajitas", "Salami", "Pepperoni", "sausage", "meatballs", "Chorizo", "King prawns", "grilled chicken", "tuna", "chicken Tikka", "curried chicken", "Tandoori chicken", "sweet and sour chicken", "Mongolian beef", "falafel", "smoked salmon", "pastrami", "black pudding", "blood sausage", "Spam", "ham hock", "Parma ham", "pulled pork", "Chilli con carne", "bacon", "anchovies", "soy chunks", "Quorn chunks", "Tofu cubes"],
"Vegetable":["Canadian bacon", "bell peppers", "baby spinach", "red onions", "French onions", "French fries", "red peppers", "sun-dried tomatoes", "sliced aubergine", "sliced courgettes", "carrot strips", "shallots", "artichoke hearts", "beetroot slices", "pine nuts", "pumpkin chunks", "raisins", "capers", "chickpeas", "avocado", "jalapenos", "habanero chillis", "sauerkraut", "kimchi", "pickled onions", "chopped garlic", "cherry tomatoes", "sliced almonds", "grilled apricots", "black olives", "green olives", "Shitake mushrooms", "button mushrooms", "Portobello mushrooms", "porcini mushrooms", "wilted greens", "sliced green tomatoes"],
"Cheese":["feta cheese", "Gorgonzola", "Stilton", "Camembert", "Cheddar", "goat's cheese", "haloumi cheese", "smoked tofu", "bean curd", "Quark", "Mozzarella", "Danish blue cheese", "Provolone", "Swiss cheese", "Red Leicester", "Wensleydale", "Port Salut cheese", "Monterey Jack", "Gruyere", "Edam", "Gouda", "Munster cheese", "soy cheese", "paneer cheese",
```


expansion th chicken Tikka and grilled apricots as toppings on paneer cheese and Marinara sauce on a deep dish.

origin

pizza

	Meat		Vegetable		Cheese		Sauce	
I made this pizza with	chicken Tikka	and	grilled apricots	as toppings on	paneer cheese	and	Marinara sauce	on a

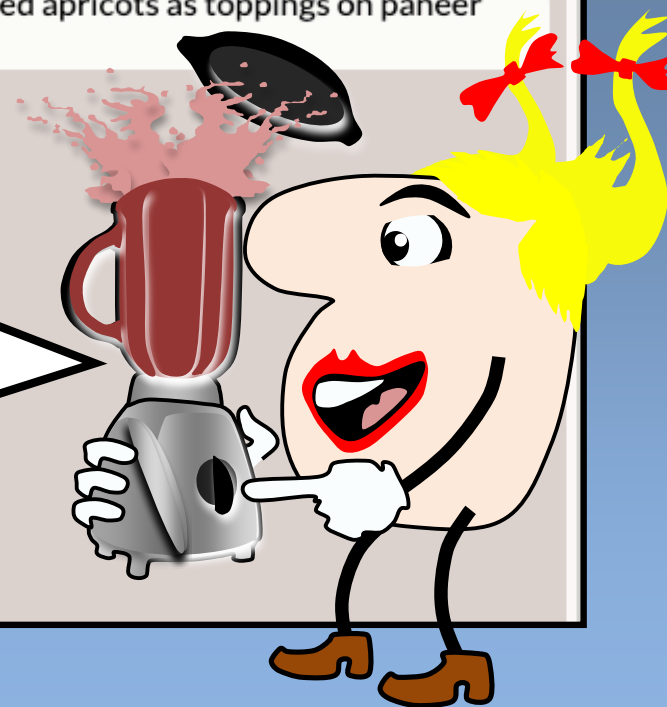
Base: deep dish .

This panel shows which **non-terminals** were used, and where, in the output.

origin 1 reroll step 51386112

I made this pizza with chicken Tikka and grilled apricots as toppings on paneer cheese and Marinara sauce on a deep dish.

Now we're ready to **blend** our ingredients and **generate** our outputs.



1: Generating text!

Tracery uses **grammars** to generate text. A replacement grammar takes a starting **symbol**, and replaces it with one of several **rules**

On the right (or below, depending on your screen size), you can see the json object that represents this very simple grammar. Farther right is the current start symbol ('animal') and a list of possible texts that it can generate.

Press the reroll button a few times to generate more. Not very interesting, yet, so lets get more complicated!

```
{
    "animal": ["unicorn","raven","sparrow","scorpion","coyote","eagle","owl","lizard","zeb
}
```

start symbol: animal

- lizard
- coyote
- kitten
- unicorn
- owl
- duck
- unicorn

reroll

2: Rules within rules

Rules can call rules!

Take a look at the rule for 'sentence': 'The #color# #animal# of the #natureNoun# is called #name#'. Each of the words between two hashtags is a **symbol** to be replaced.

Try adding your name to the list of names, surrounded by quotation marks like all the other options. It will now appear in some of the generated text!

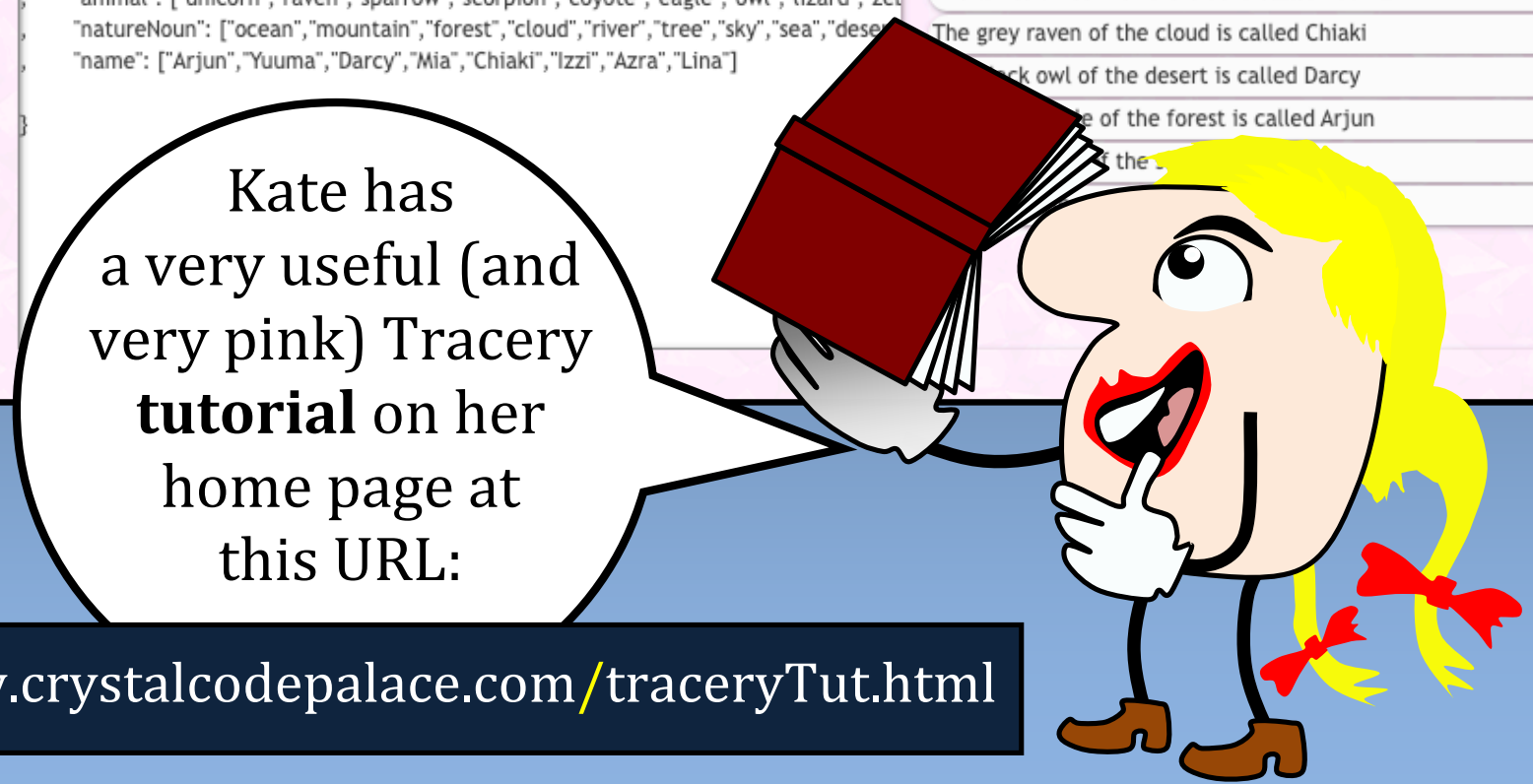
Note: JSON is very very fussy. There must be a comma between every option, but none after the last option

```
{
    "sentence": ["The #color# #animal# of the #natureNoun# is called #name#"]
    "color": ["orange","blue","white","black","grey","purple","indigo","turquoise"]
    "animal": ["unicorn","raven","sparrow","scorpion","coyote","eagle","owl","lizard","zeb
    "natureNoun": ["ocean","mountain","forest","cloud","river","tree","sky","sea","dese
    "name": ["Arjun","Yuuma","Darcy","Mia","Chiaki","Izzi","Azra","Lina"]
}
```

start symbol: sentence

- The black kitten of the sky is called Azra
- The white unicorn of the tree is called Yuuma
- The grey raven of the cloud is called Chiaki
- ck owl of the desert is called Darcy
- e of the forest is called Arjun
- the

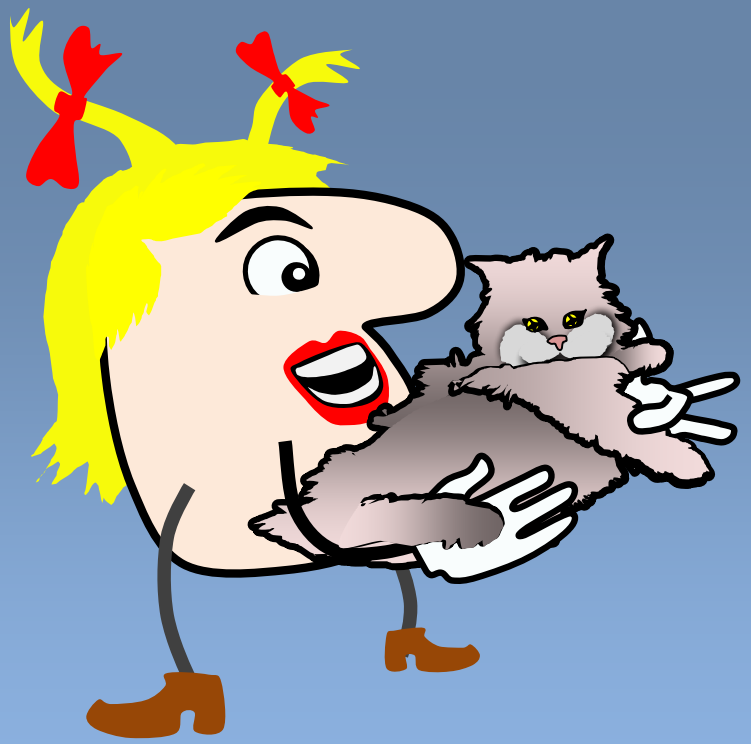
reroll



http://www.crystalcodepalace.com/traceryTut.html

Now, use a **step-by-step** chat with ChatGPT to elicit key **Bond tropes** and to suggest new instances.

You must get **ChatGPT** to **write** the **Tracery grammar** that puts these instances together as a new studio "**pitch**."



Submit a working grammar and an **annotated transcript** of your dialogue with ChatGPT.