

# Design Styleguide

## Principle Typeface

Averta

Regular

Semibold

Bold

Extra Bold

## Wordmark

res( )nate

The wordmark can be typed.  
A vector or bitmap file isn't  
necessary.

- only use Averta Bold
- the gap between parentheses  
can be made with two spaces

## Wordmark

res( )nate

res( )nate

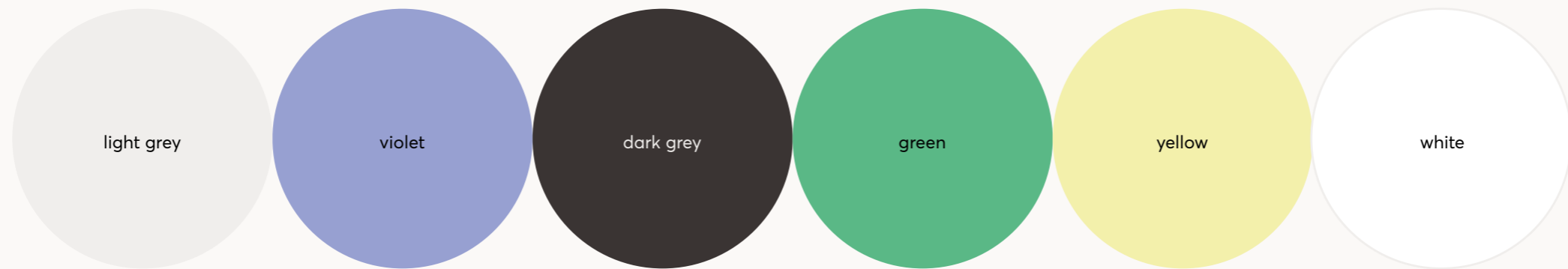
res( )nate

res( )nate

res( )nate

res( )nate

## Resonate colour swatches



CMYK	7 6 7 0	46 35 0 0	0 12 7 90	65 0 59 0	8 0 43 0	0 0 0 0
RGB	213 218 218	151 167 220	65 65 65	84 235 128	246 253 172	255 255 255
HEX	D5 DA DA	97 A7 DC	41 41 41	54 EB 80	F6 FD AC	FF FF FF

## The blockchain key-visual

### Building the blockchain key-visual

The key-visual is an abstract interpretation of the blockchain data structure. It is not meant as an accurate technical description, although several blockchain terms have been used in the description. The key-visual can, however, be thought of as a dynamic graphic algorithm. Its final form is derived from a sequence of interdependent rules.

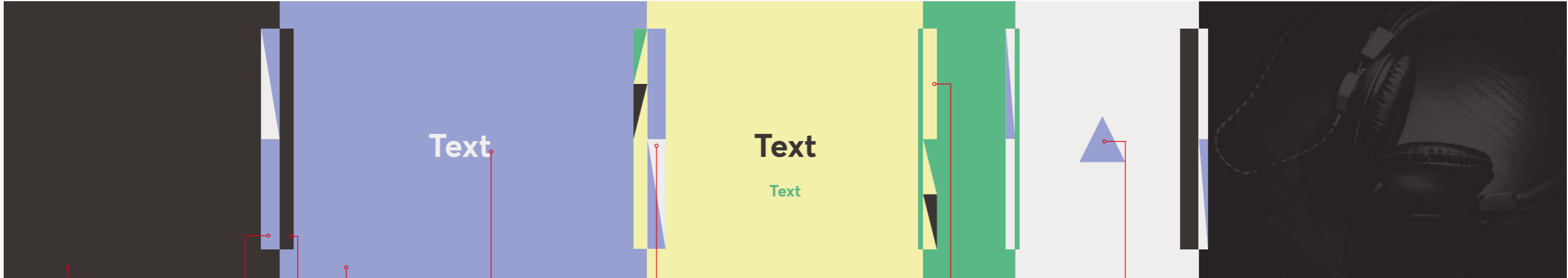
More information about the real blockchain structure can be found at:  
<http://chimera.labs.oreilly.com/books/1234000001802/ch07.html>



# Building the blockchain key-visual

block height →

↑  
block width



**This is a block**

- it can be any resonate swatch-colour
- it can be blank, contain text or contain an image

**This is another block**

- adjacent blocks cannot have the same colour

**This is a previous block-hash**

- it represents the previous block in the stack

**This is block content [text]**

- text might typically be the cooperative's name, a URL, a slogan etc

**This is a hash function**

- it is a triangle representing the colour of the block content to which its block-hash refers
- the triangle's width is 50% of the block-hash width
- the triangle points towards the block to which its block-hash refers
- adjacent hash functions should be offset by half the width of the block-hash

**This is another hash function**

- it has two triangles, referring to the two texts in the previous block
- multiple triangles always share the same allocated 50% of the block-hash

**This is block content [ornament]**

- there are four ornaments: diamond, circle, triangle, and the brackets from the Resonate logo

**This is a stack**

- the stack is a collection of blocks
- the stack fills the available format

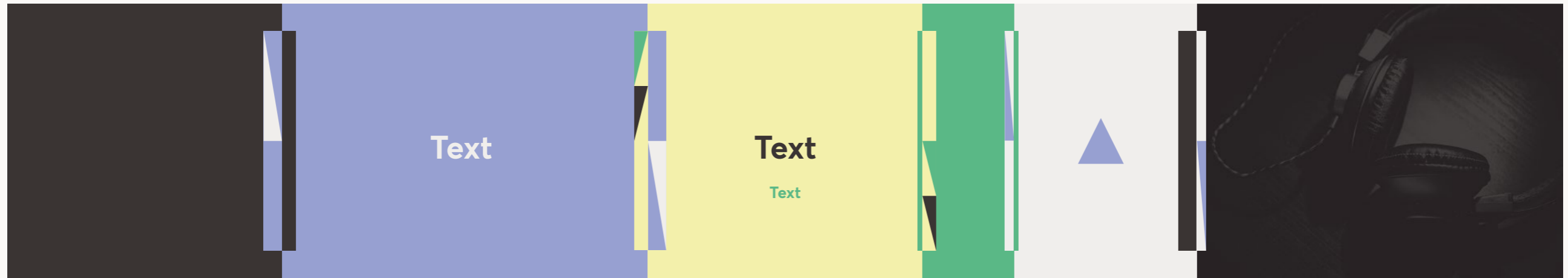
**This is a next block-hash**

- it represents the next block in the stack
- its height = 5% of the longest side of the block it represents
- its width = 80% block width
- it has the same background colour as the block it represents



# Building the blockchain key-visual

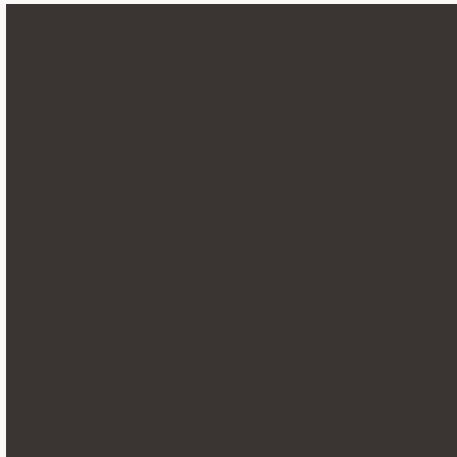
## Step 1





# Building the blockchain key-visual

## Step 1

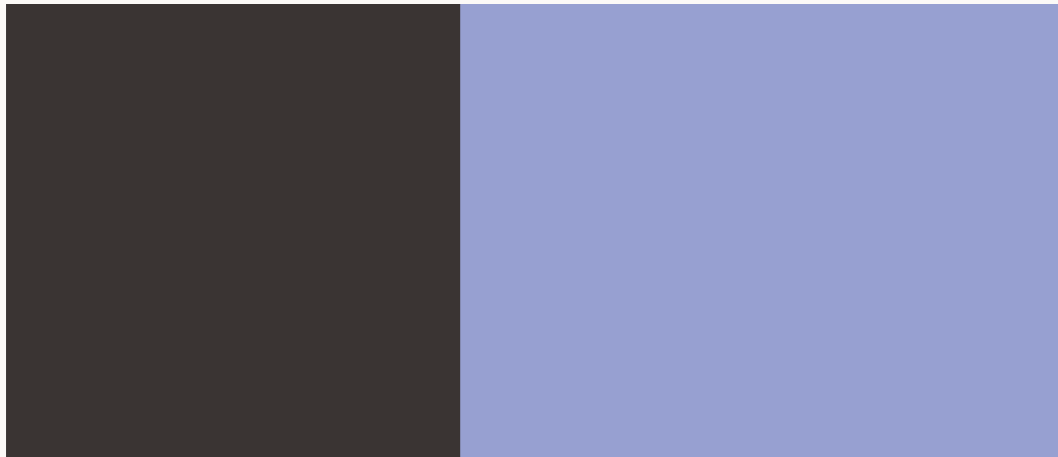


**Create a block.**

- Any size you like.
- Any resonate-colour you like.

## Building the blockchain key-visual

### Step 2



#### Add a block.

- Any size you like.
- Any resonate-colour you like, as long as it is different to the previous block-colour.

# Building the blockchain key-visual

## Step 3



**Create a  
*previous-block-hash.***

- **it can be aligned to the center**

# Building the blockchain key-visual

## Step 3



**Create a  
*previous-block-hash.***

- it can be aligned to the top

# Building the blockchain key-visual

## Step 3



**Create a  
*previous-block-hash.***

- **it can be aligned to the  
bottom**

# Building the blockchain key-visual

## Step 3



**Create a  
*previous-block-hash.***

**We're going to stick with  
the centered version**

# Building the blockchain key-visual

## Step 3

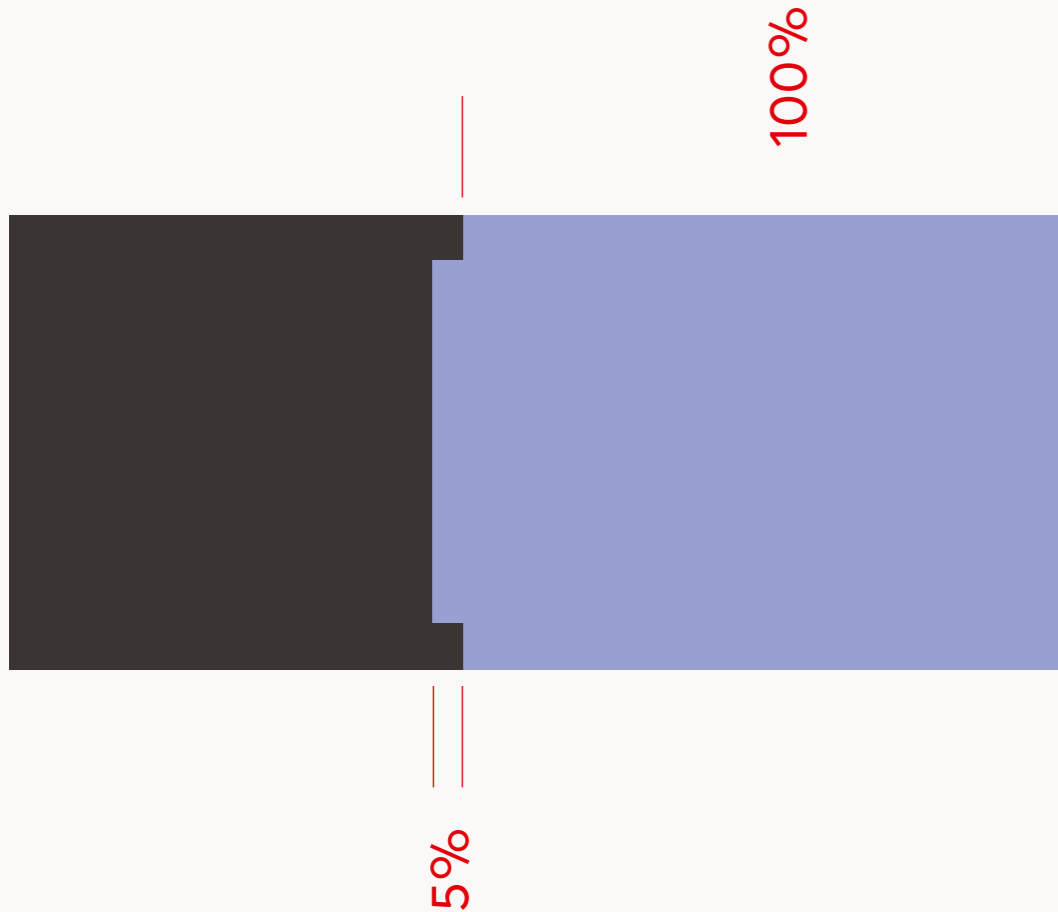


Create a  
*previous-block-hash*.

- its width is 80% of the  
block width

# Building the blockchain key-visual

## Step 3



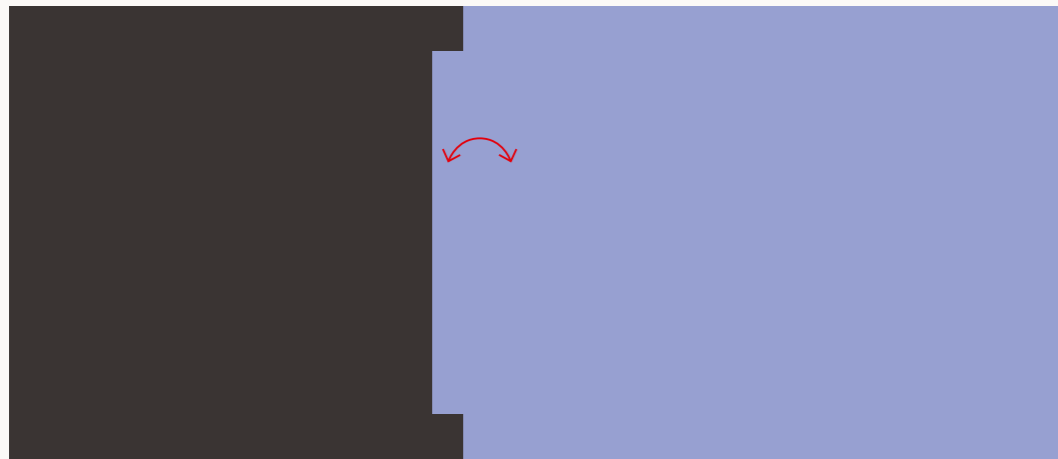
**Create a  
*previous-block-hash.***

- its height is 5% of the block width



## Building the blockchain key-visual

### Step 3

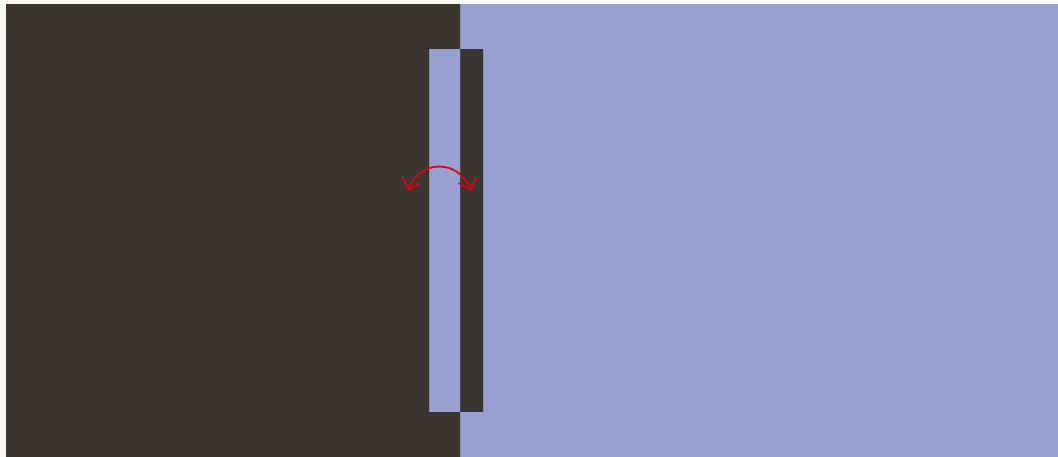


**Create a  
*previous-block-hash.***

- its colour is the same as the block it represents

## Building the blockchain key-visual

### Step 4

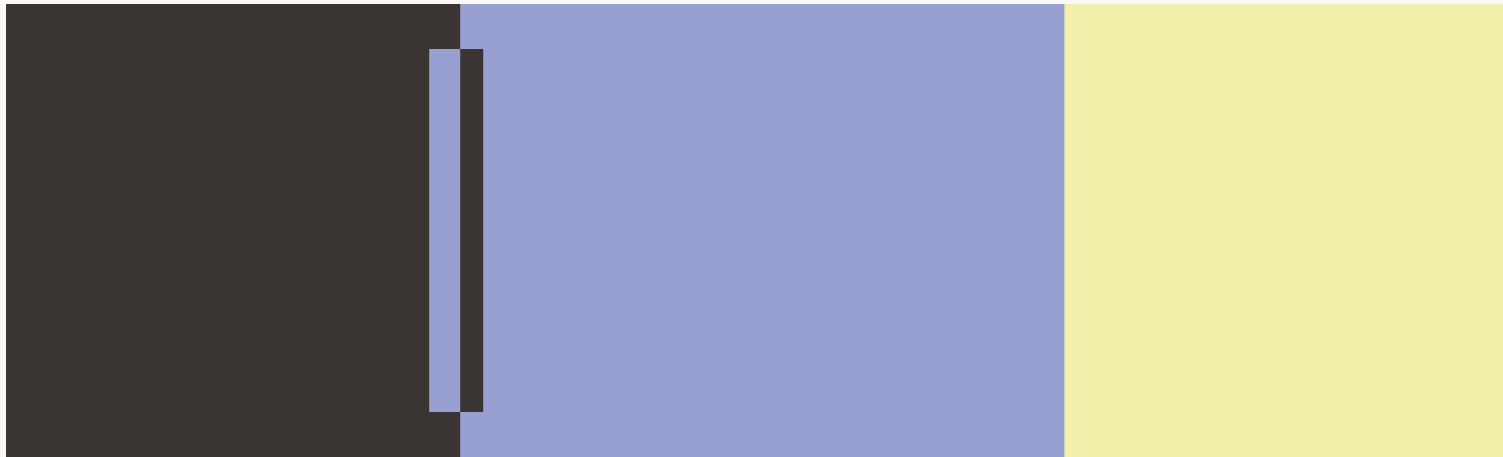


Now create a  
*next-block-hash*.

- its colour is the same as the block it represents

## Building the blockchain key-visual

### Step 5

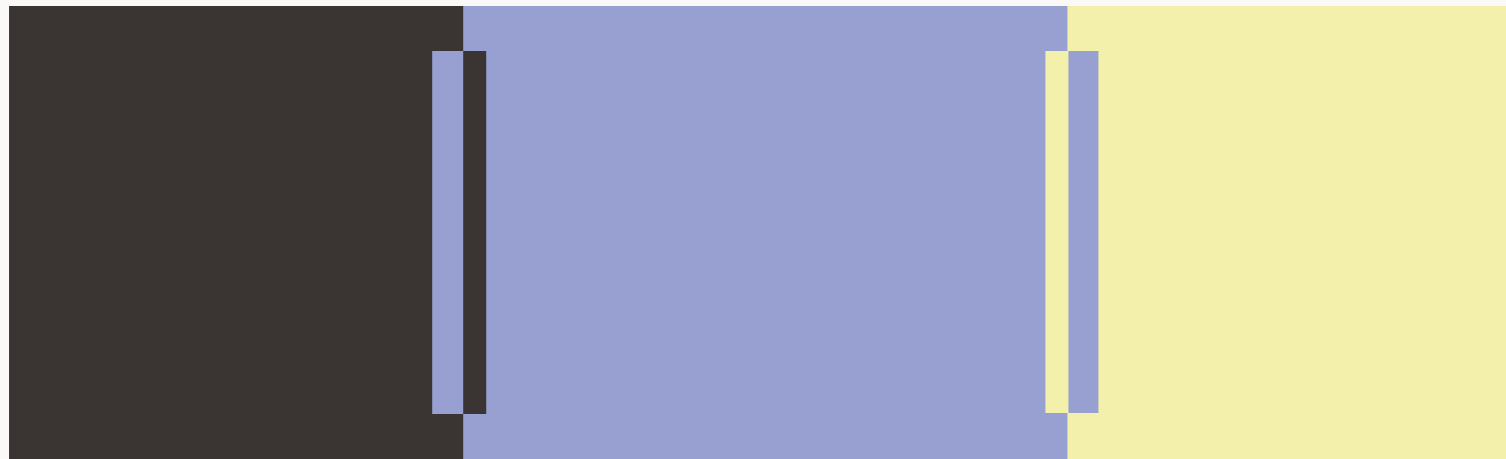


#### Add a block.

- Any size you like.
- Any resonate-colour you like, as long as it is different to the previous block-colour.

## Building the blockchain key-visual

### Step 6

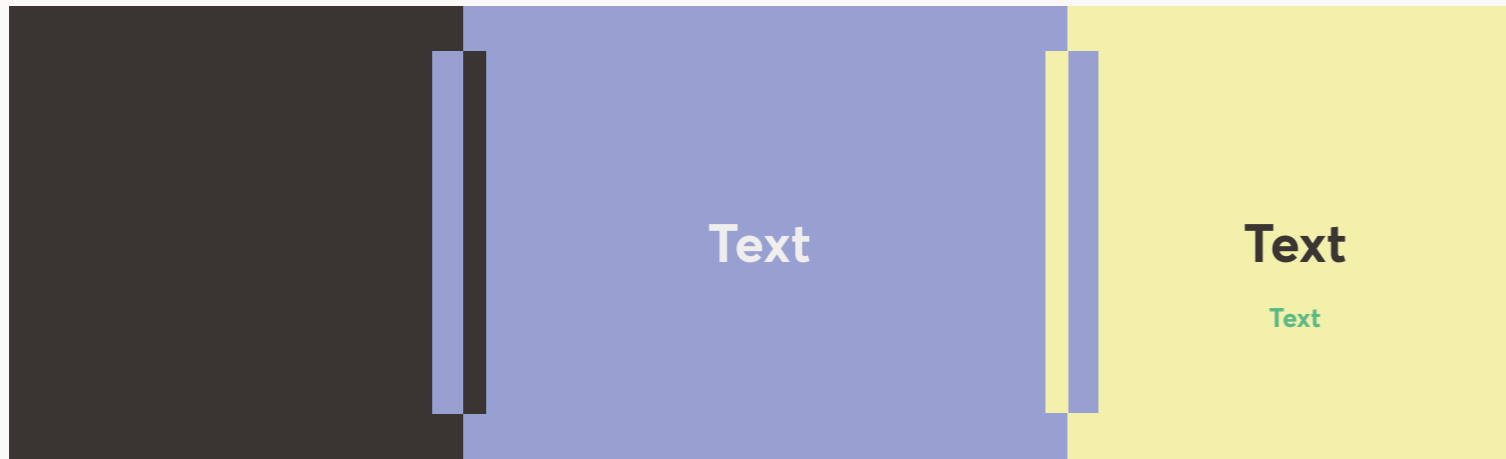


**Create more  
*block-hashes***

- **apply same rules  
as in steps 3 and 4**

# Building the blockchain key-visual

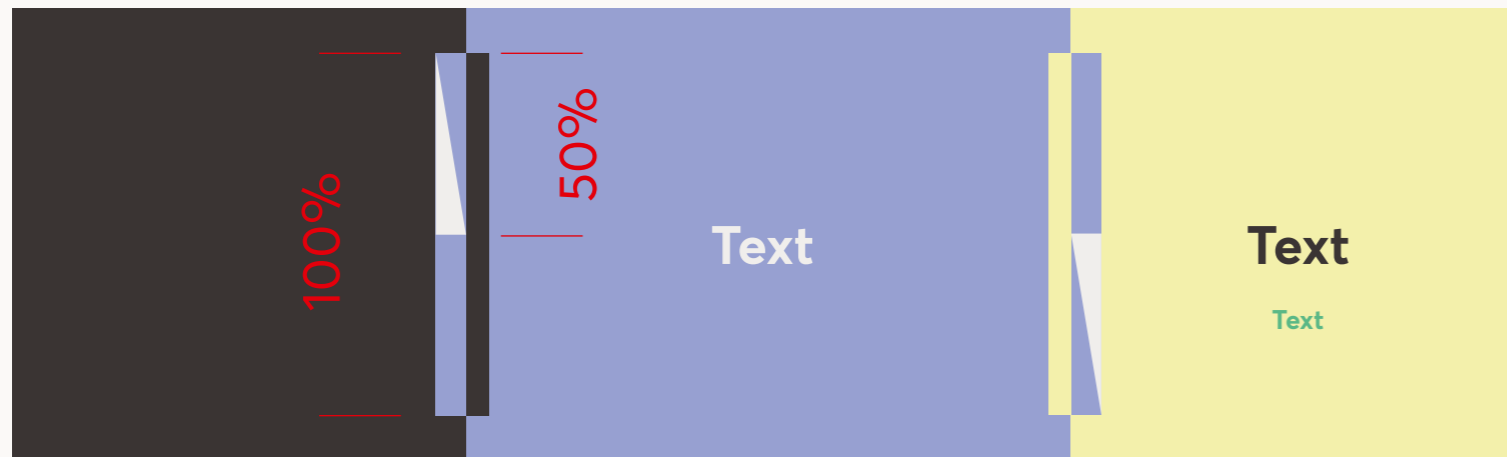
## Step 7



**Add some text**

# Building the blockchain key-visual

## Step 8

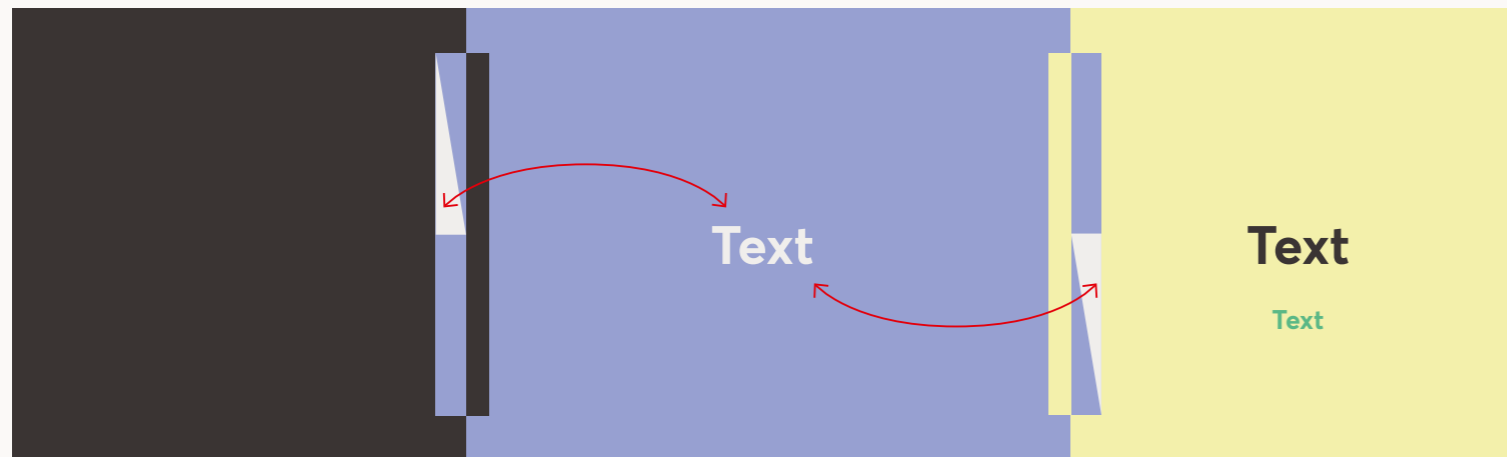


### Create *hash-functions*

- width is 50% of the width of its *block-hash*

# Building the blockchain key-visual

## Step 8

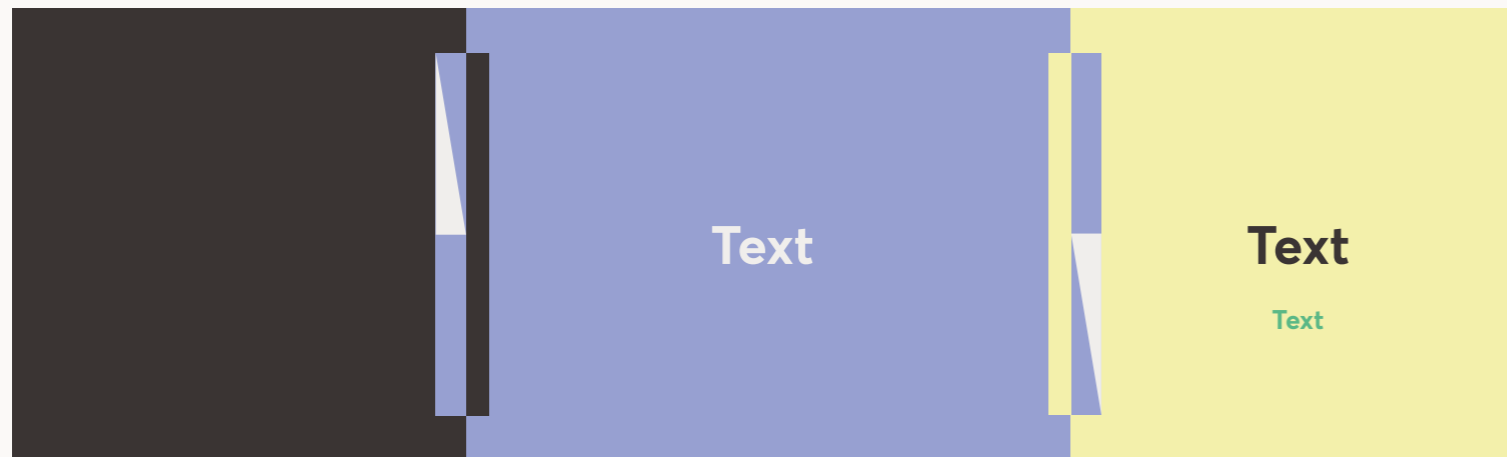


### Create *hash-functions*

- *hash-function*  
colour is the same  
as the *block-content*  
it represents

# Building the blockchain key-visual

## Step 8



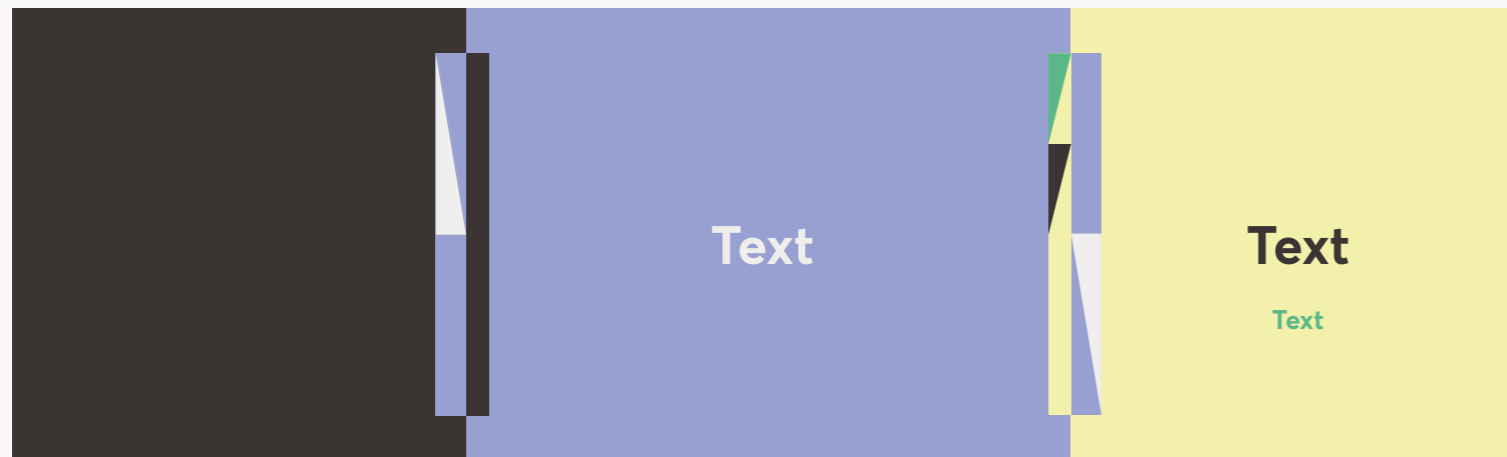
### Create *hash-functions*

- *hash-functions*  
point towards their  
block content



# Building the blockchain key-visual

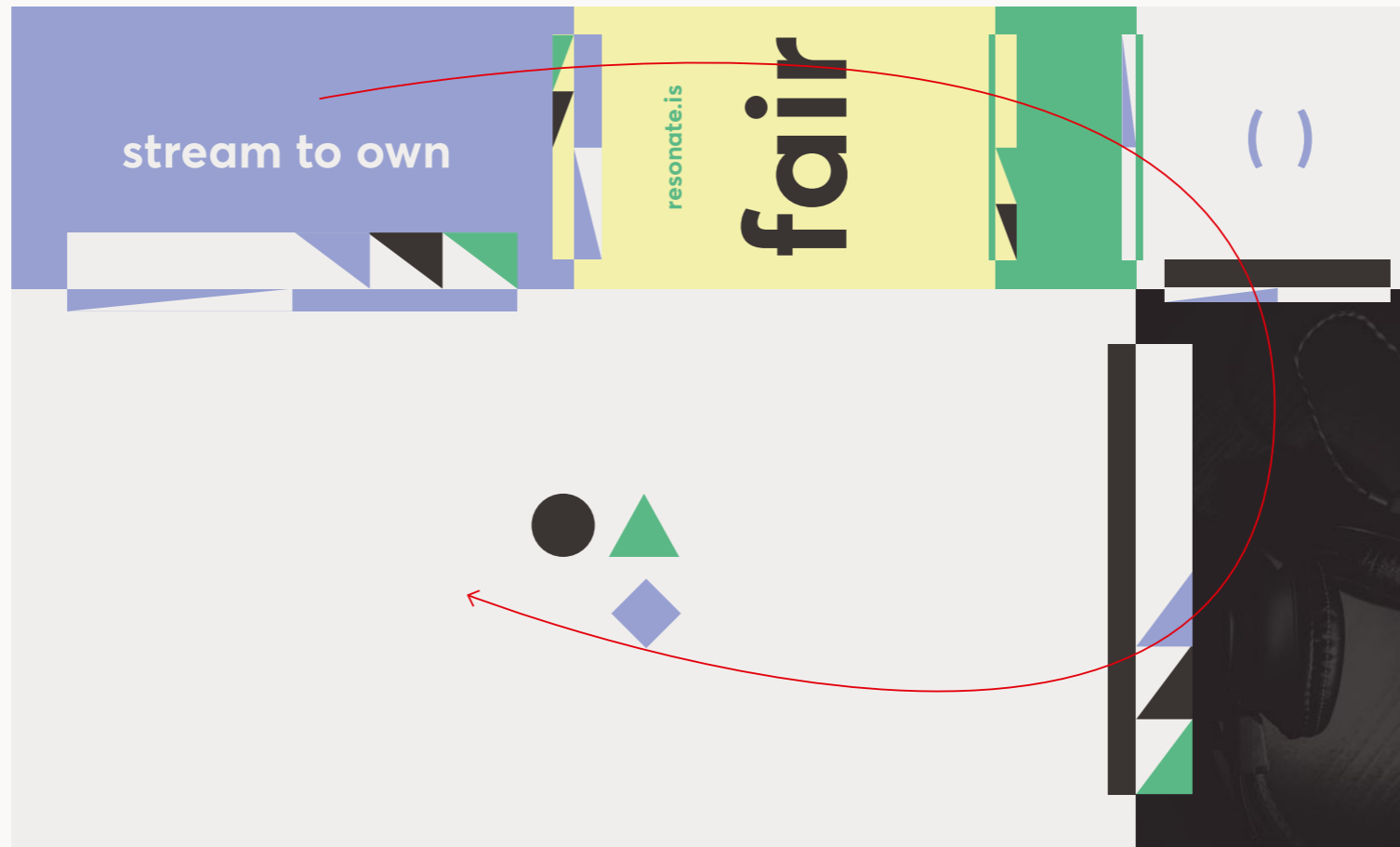
## Step 9



**Create next set of  
*hash-functions***

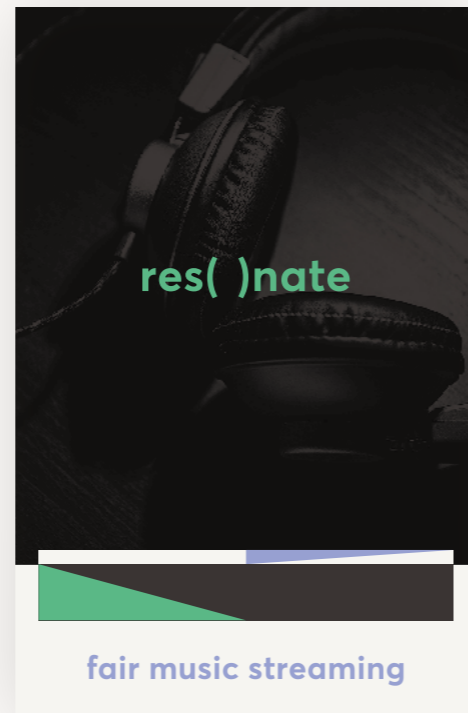
- **same rules apply**

## Key-visual as layout



The blockchain works as a linear form, but can also be "wrapped" around a format in to a loop.

## Examples in use Business card



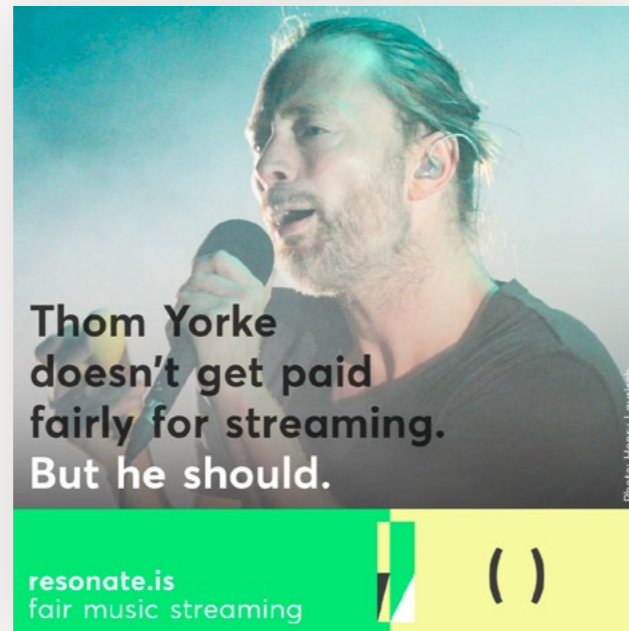
## Examples in use

### Postcard



## Examples in use

### Promo image



Images can be used  
independently from the  
blockchain