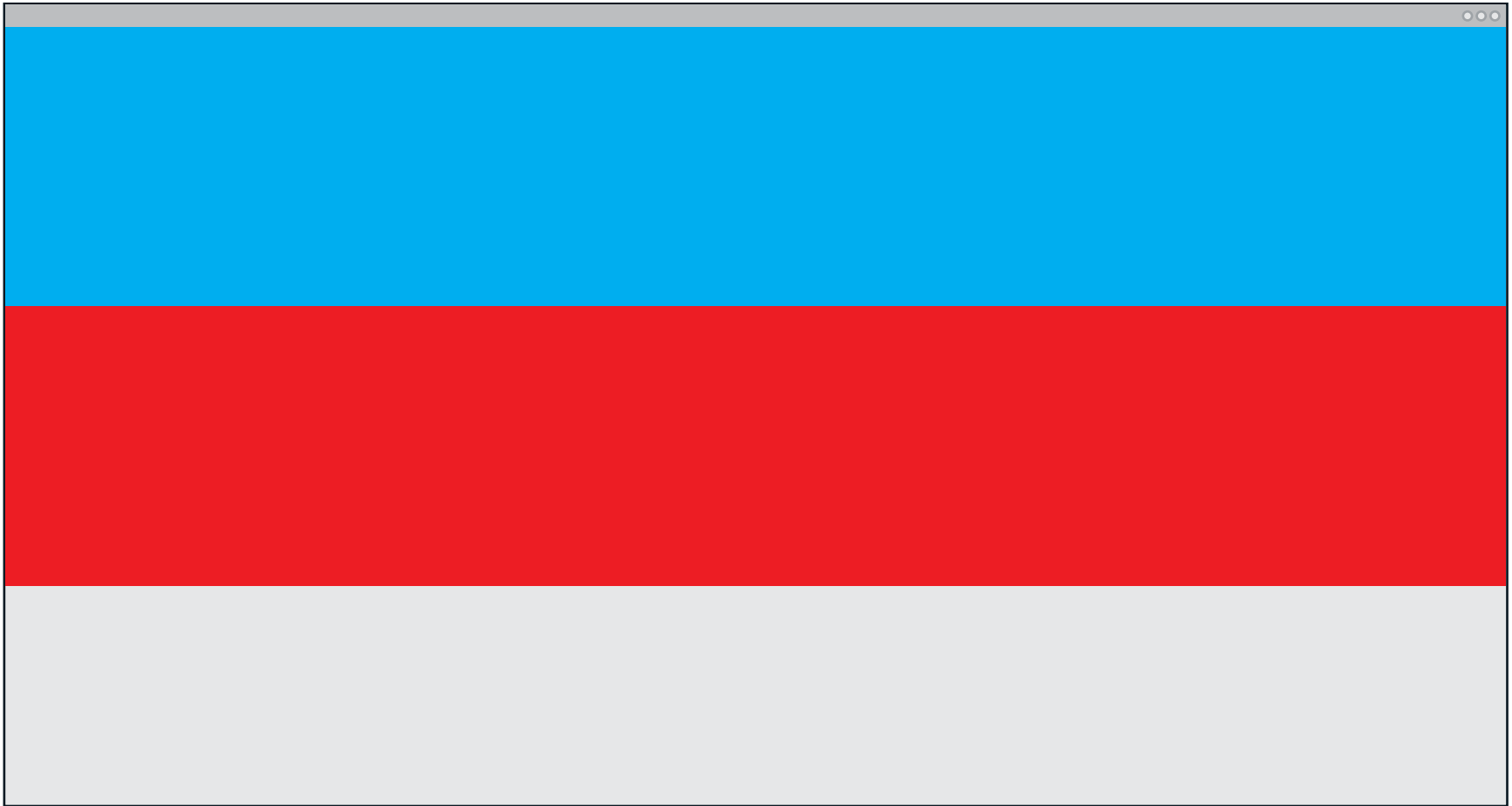


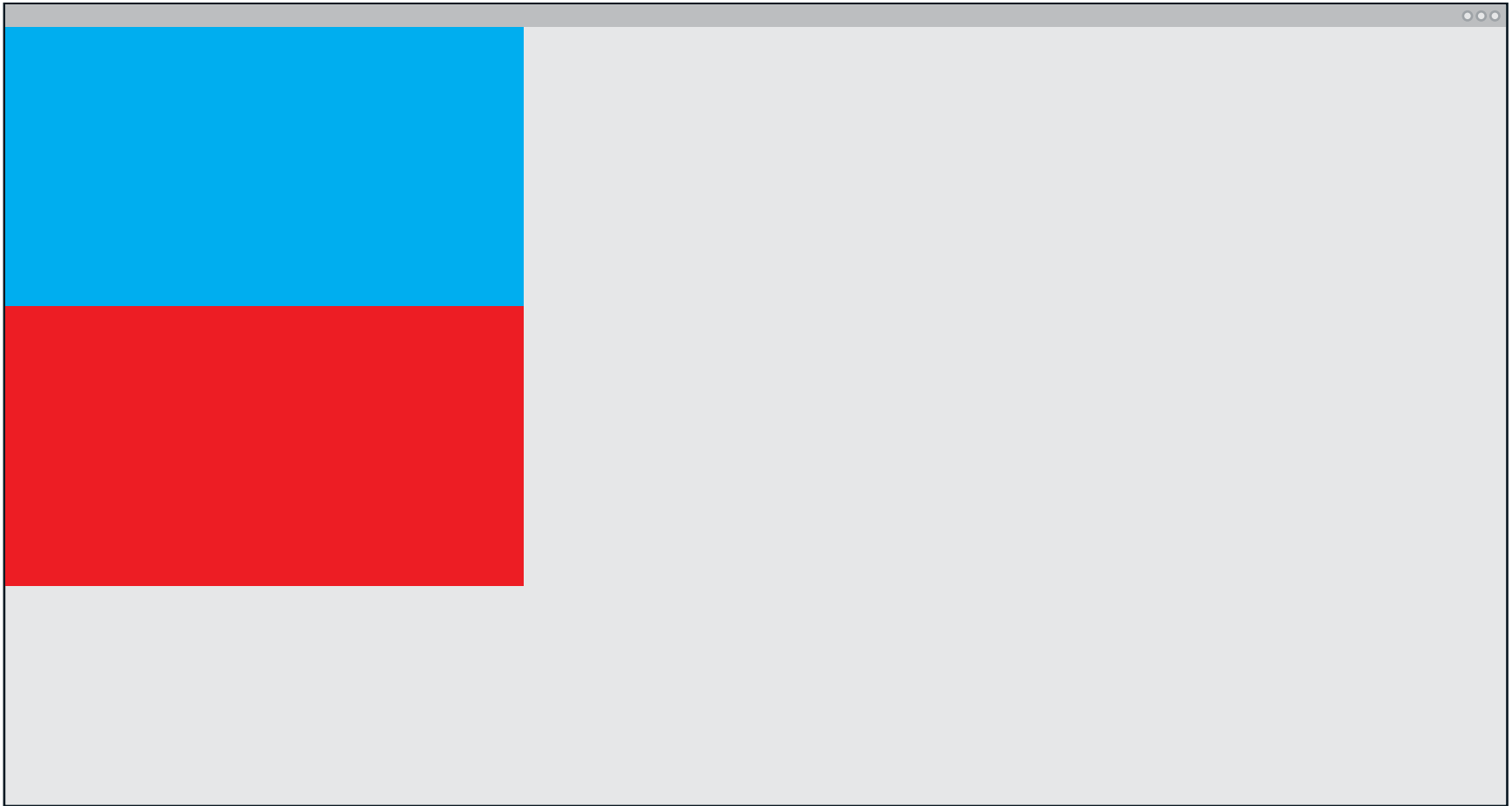
BLOCK FORMATTING CONTEXT

**IN A BLOCK FORMATTING CONTEXT,
THE CONTENT INSIDE THAT CONTEXT IS
HANDLED IN A VERY SPECIFIC WAY**

**BOXES ARE LAID OUT ONE ON TOP OF
EACH OTHER.**

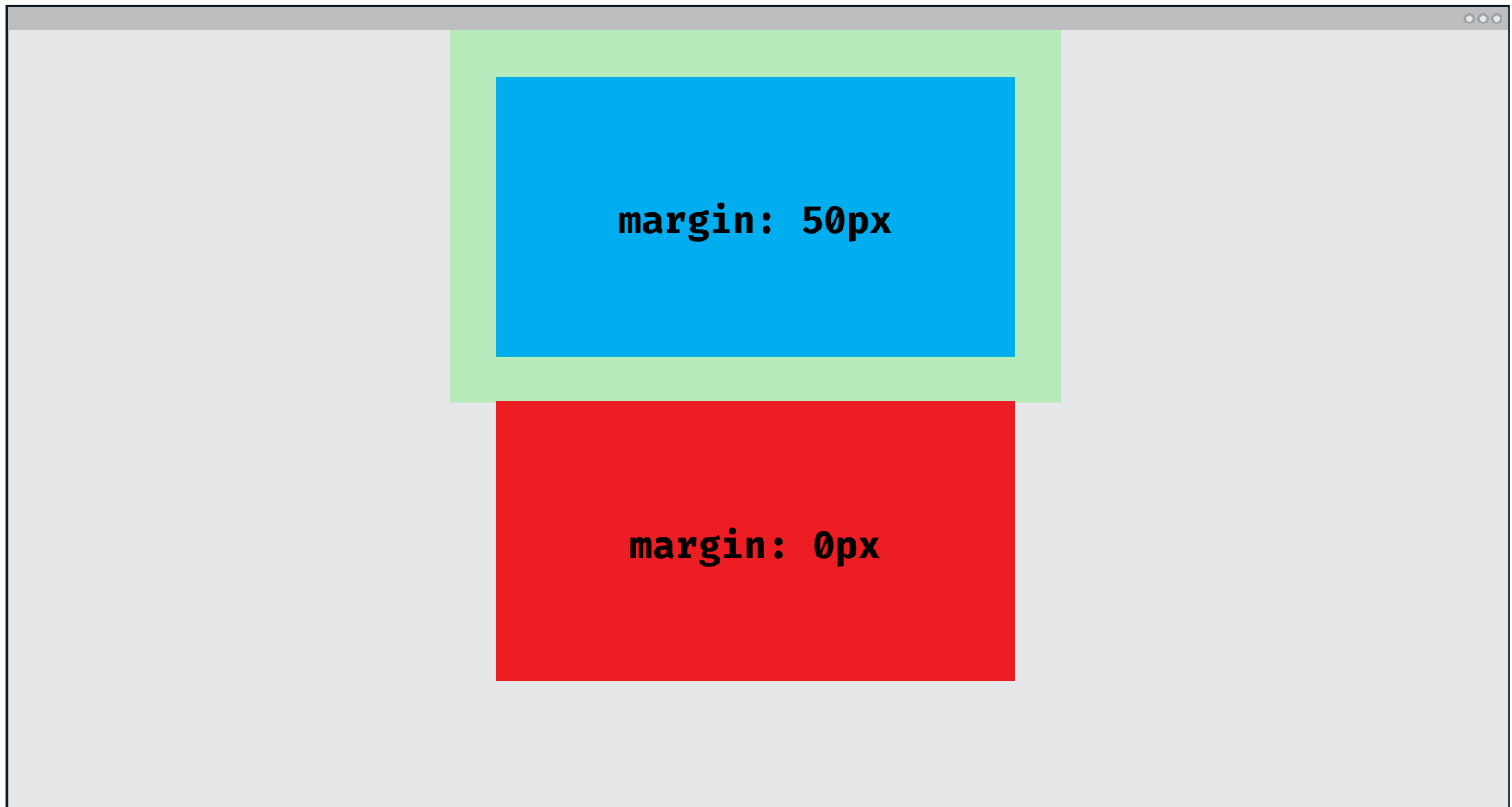
**EVEN IF THERE IS ROOM, THEY WON'T
TRY TO SQUEEZE IN.**

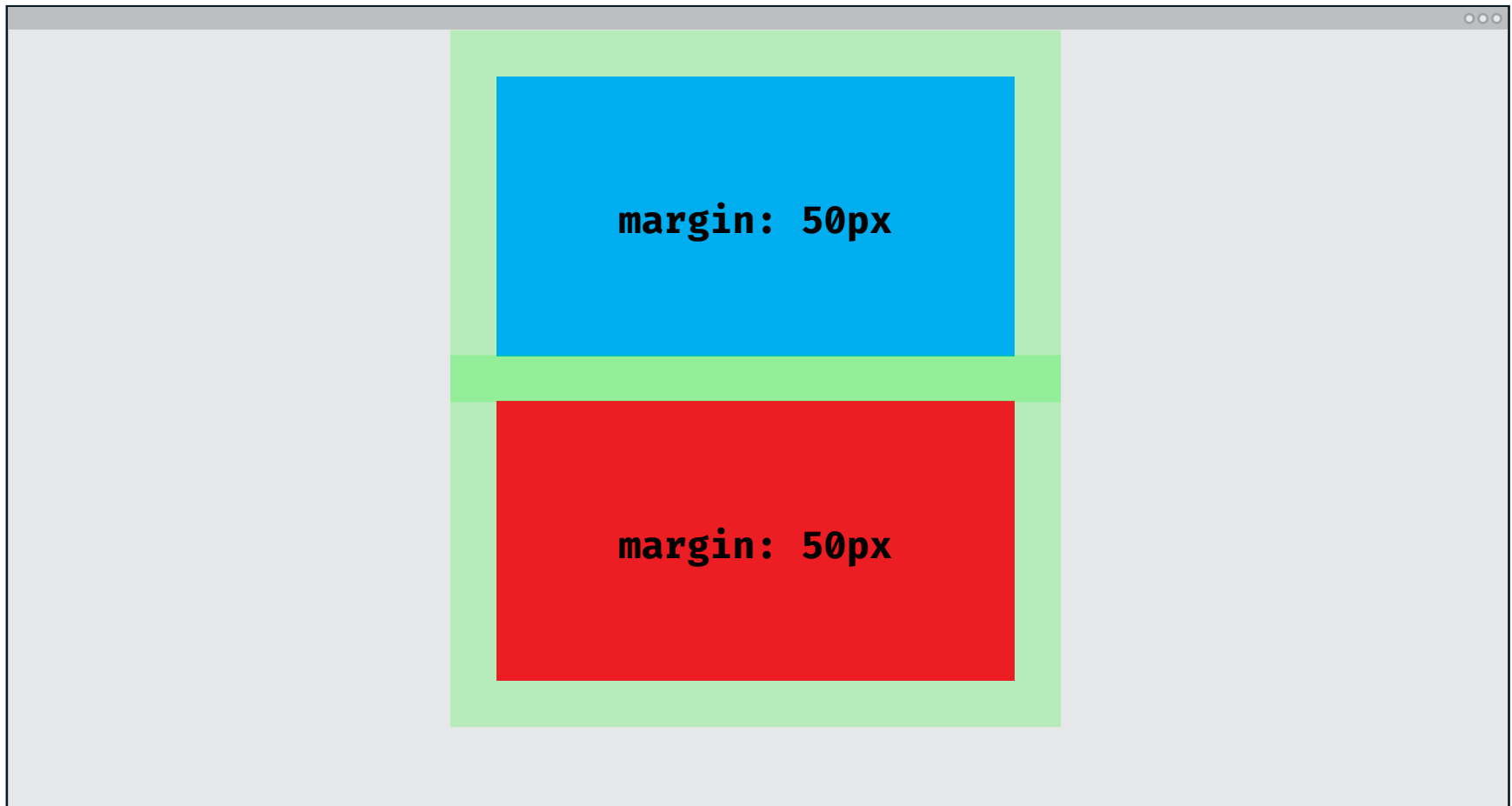


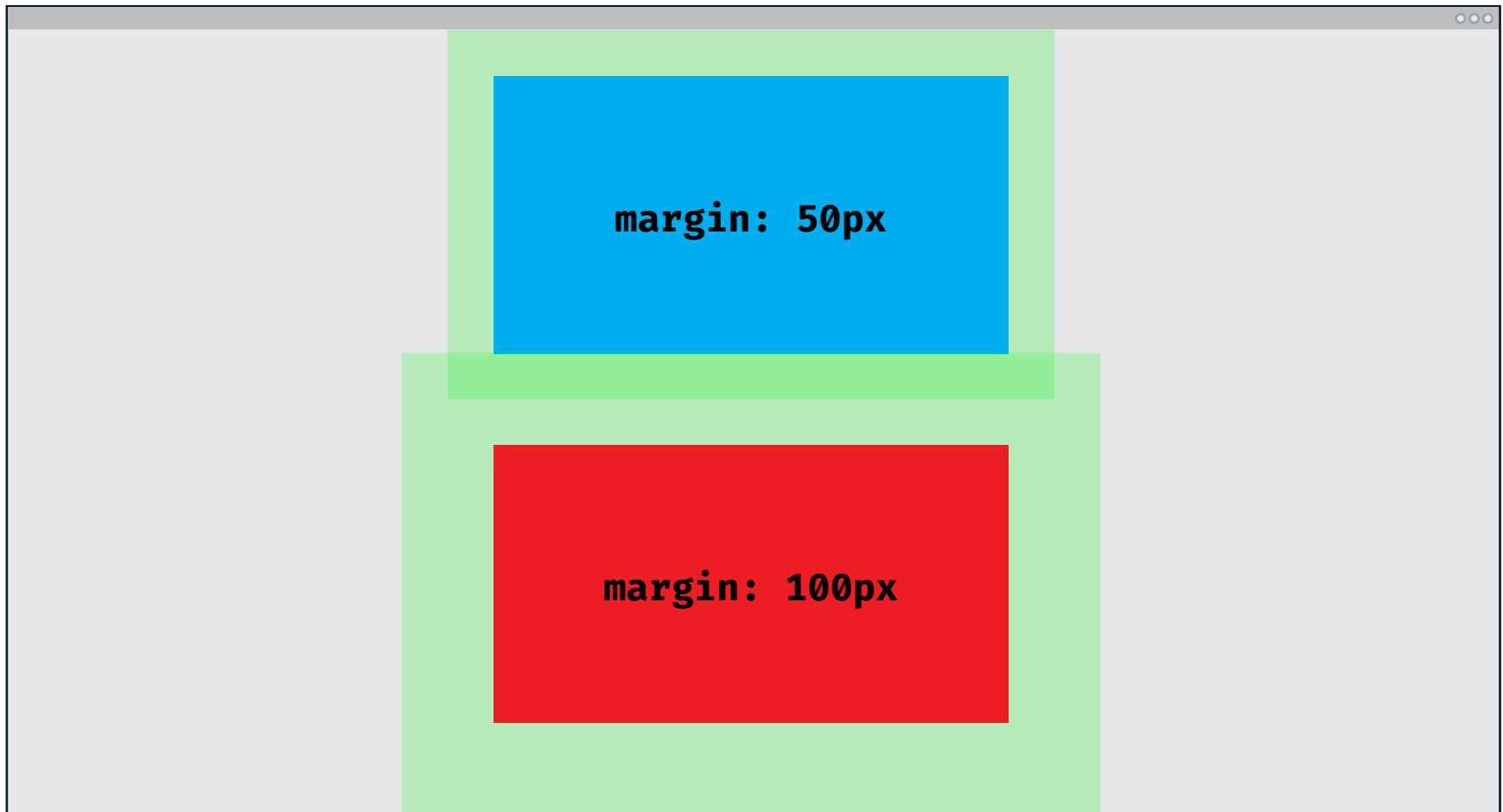


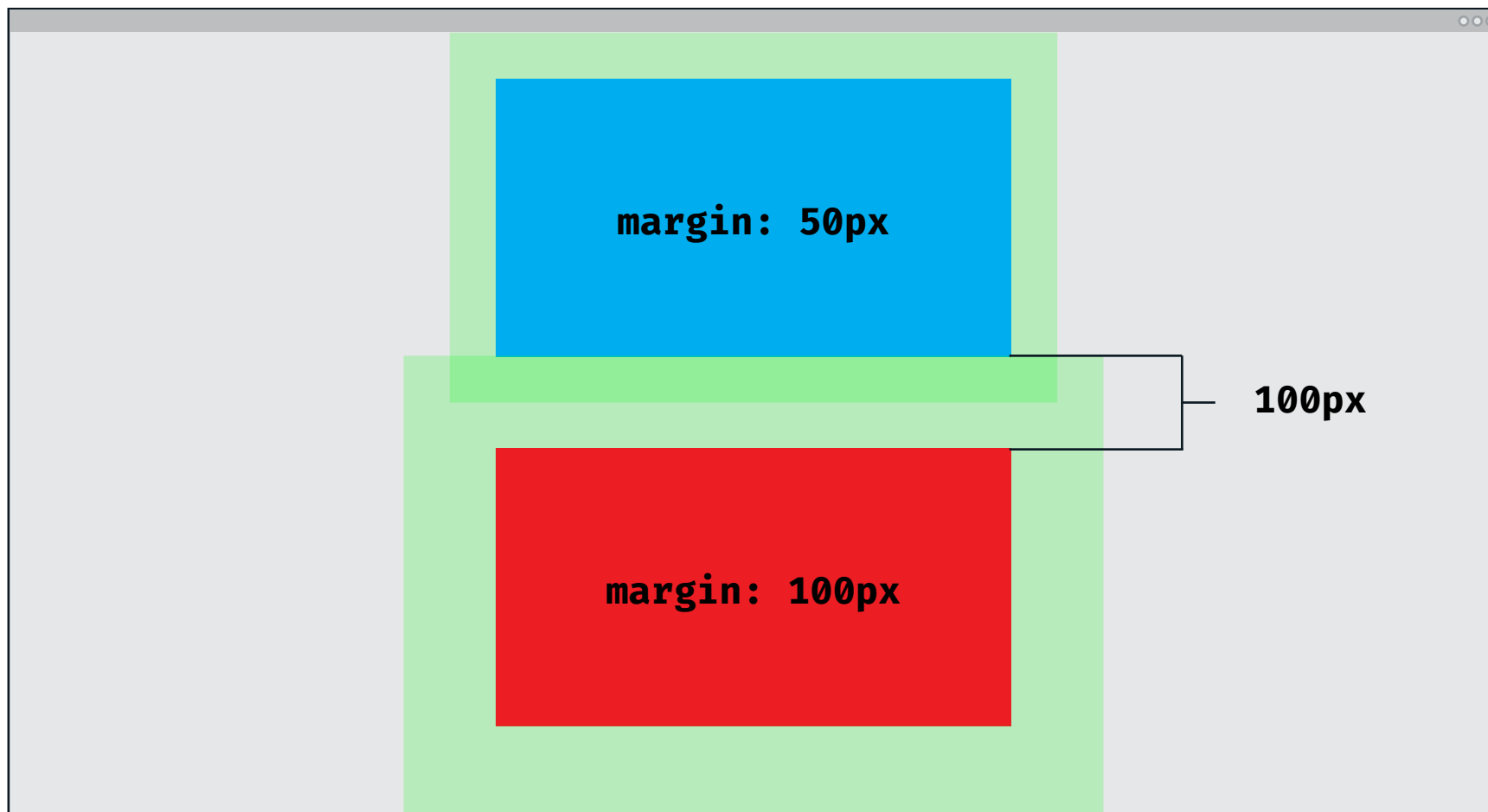
**WE CAN CREATE SPACE BETWEEN
THESE ELEMENTS WITH VERTICAL
MARGINS.**

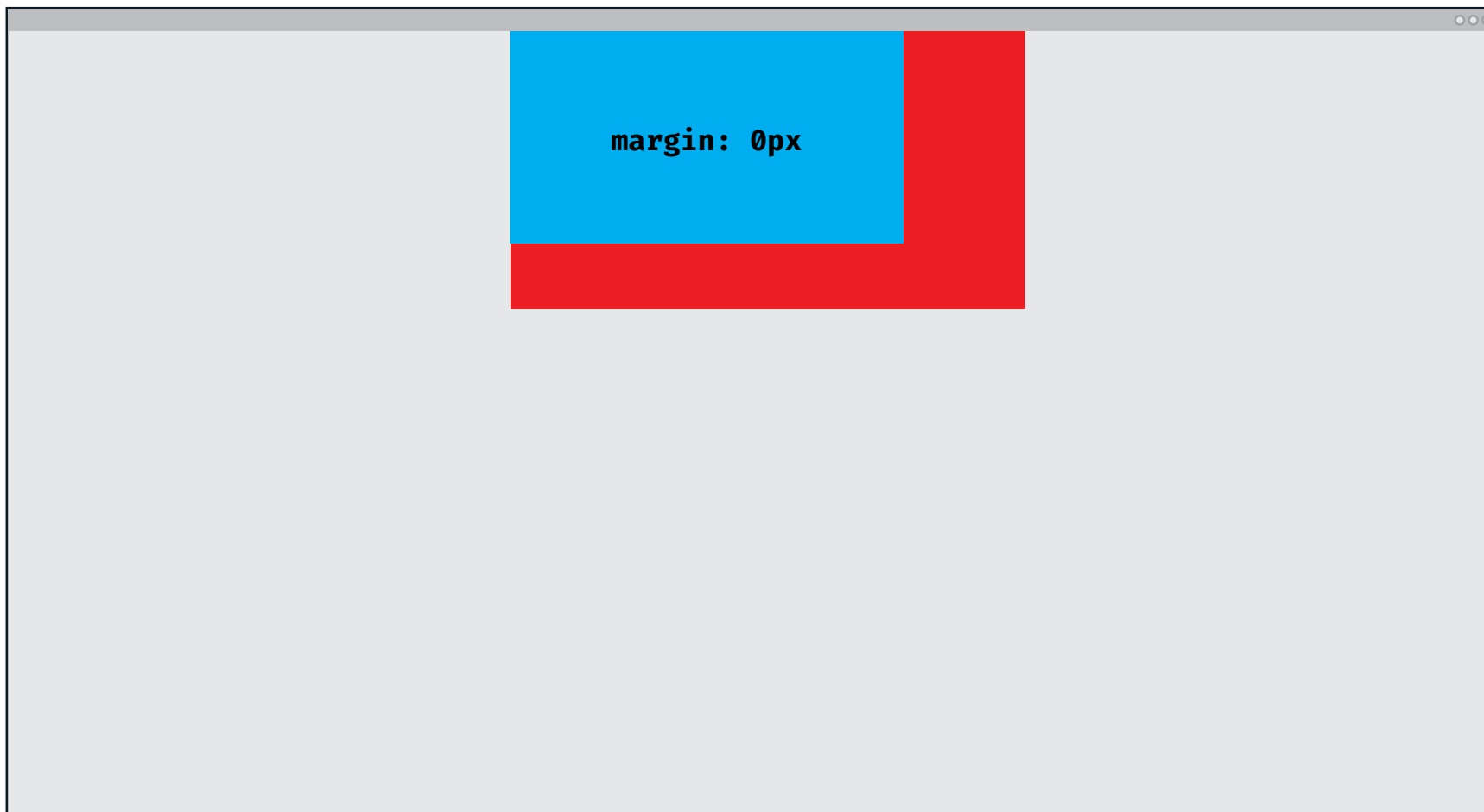
**AND VERY IMPORTANTLY, THOSE
MARGINS COLLAPSE.**

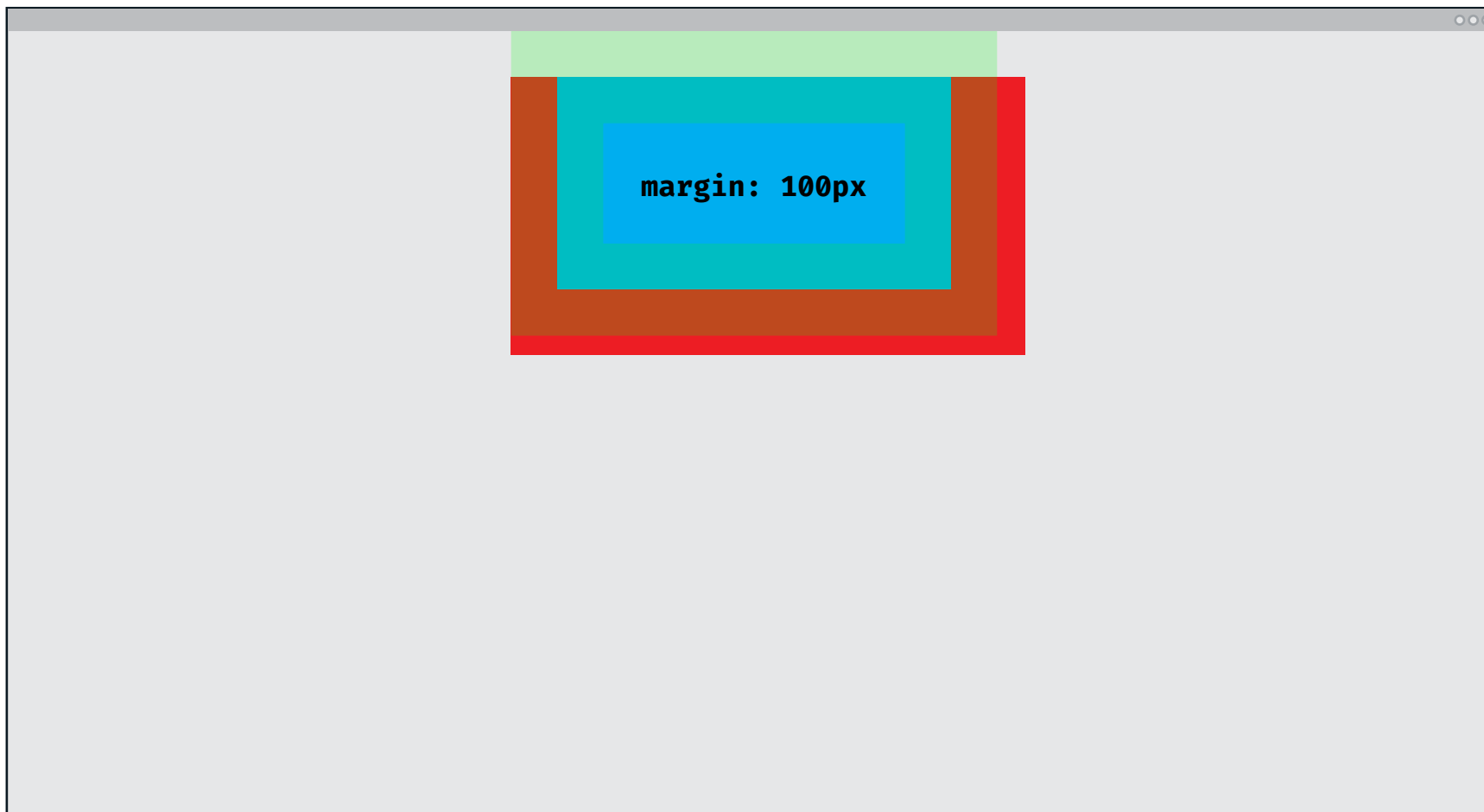












**THE ROOT OF OUR DOCUMENT
(THE HTML ELEMENT) CREATES A BLOCK
FORMATTING CONTEXT, WHICH ALL OF
OUR CONTENT LIVES INSIDE**

**WE CAN CREATE NEW BLOCK
FORMATTING CONTEXTS WITHIN OUR
DOCUMENT AS WELL.**

**BLOCK FORMATTING CONTEXTS ARE A
BIG DEAL IF YOU USE FLOATS.**

**BUT WE DON'T DO THAT MUCH
ANYMORE.**

**THEY ALSO HAVE AN IMPACT ON
COLLAPSING MARGINS.**

**THIS IS REALLY IMPORTANT THESE
DAYS.**

WHEN A BLOCK CREATES A NEW BLOCK FORMATTING CONTEXT, IT:

- contains all internally floated elements
- excludes all externally floated elements
- stops margins from collapsing

WHEN A BLOCK CREATE A NEW BLOCK FORMATTING CONTEXT, IT:

- contains all internally floated elements
- excludes all externally floated elements
- **stops margins from collapsing**

YOU CAN CREATE A NEW BLOCK FORMATTING CONTEXT IN A LOT OF WAYS (THIS IS JUST A FEW)

- Using `float`
- position `absolute`, `fixed`, and `inline-block`
- `overflow` of anything except `visible`
- using `columns`
- flex and grid items
- `display: flow-root`

- float example:
- margin collapsing example:

WHY DOES THIS MATTER?

**WITH SO MANY THINGS CREATING
A NEW FORMATING CONTEXT, AND
BECAUSE A NEW BLOCK FORMATTING
CONTEXT STOPS MARGINS FROM
COLLAPSING, IT CAN BE HARD TO BE
CONSISTENT.**