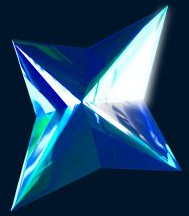
BRO Academy



CSS++

Lesson 15

Lesson Plan

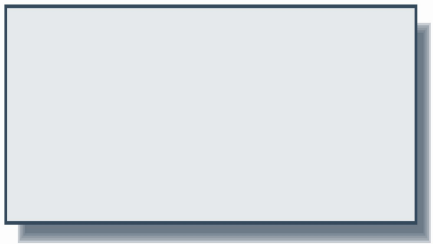
1
Shadows

2
Filters

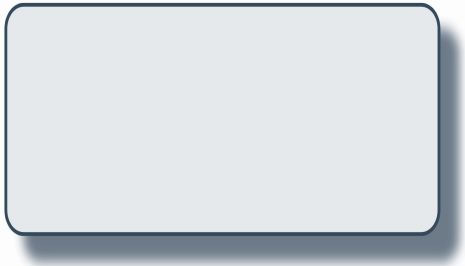
3
Masking & clipping

Shadows

BOX-SHADOW



```
p { box-shadow: red 5px 8px 15px 18px inset; }
```



External shadows

– Placed outside the `border`
– Mimics the shape of the block, including rounded corners, etc.

Internal shadows (inset)

– Placed above `background-images`, `background-color`
– but below content

Syntax box-shadow

```
p { box-shadow: inset 5px 8px 15px 18px red; }
```

offset-x, offset-y, blur-radius, spread-radius – order matters

inset internal shadow optional order does not matter	**offset-x** horizontal offset positive = right, negative = left, 0 = no offset	**offset-y** vertical offset positive = down, negative = up, 0 = no offset
blur-radius optional, default is 0 which means sharp edges, values only greater than 0	**spread-radius** optional, default is 0, increases or decreases the size of the shadow	**color** optional, if not specified, it inherits the element's color order does not matter

```
p { box-shadow: inset offset-x, offset-y, blur-radius, spread-radius color; }
```

examples

box-shadow

TEXT-SHADOW

- Only external shadows.
- Follows the shape of the text.
- Multiple shadows can be applied.
- Syntax is similar to **box-shadow**:
  - offset-x
  - offset-y
  - blur-radius
  - color

```
p {
  text-shadow: 2px 2px 8px #FF0000;
}
```

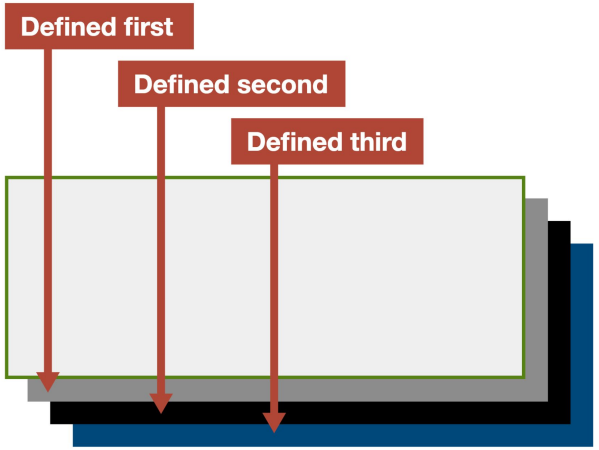
Multiple box/text-shadow

can be applied to a single element by separating each shadow with a comma:

```
p { box-shadow: inset 10px 10px 10px 10px red,    /* all possible values specified */
                red 10px 10px 10px 10px inset,    /* different order */
                10px 10px 10px 10px red,          /* no inset specified */
                10px 10px 10px 10px,              /* no color specified */
                10px 10px 10px,                   /* no spread-radius specified */
                10px 10px; }                      /* no blur-radius specified */
```

example →

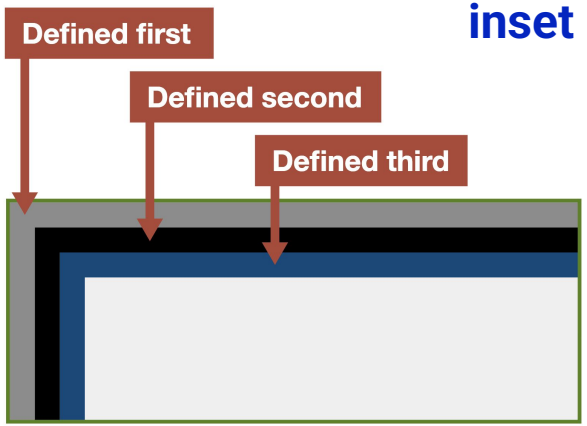
order in box/text-shadow



The order of the shadows matters

```
p { box-shadow: first, second, third; }
```

the first shadow in the list will be on top, and subsequent shadows will be layered below it.



Filters

filter

– Applies visual effects to elements
(like in instagram)

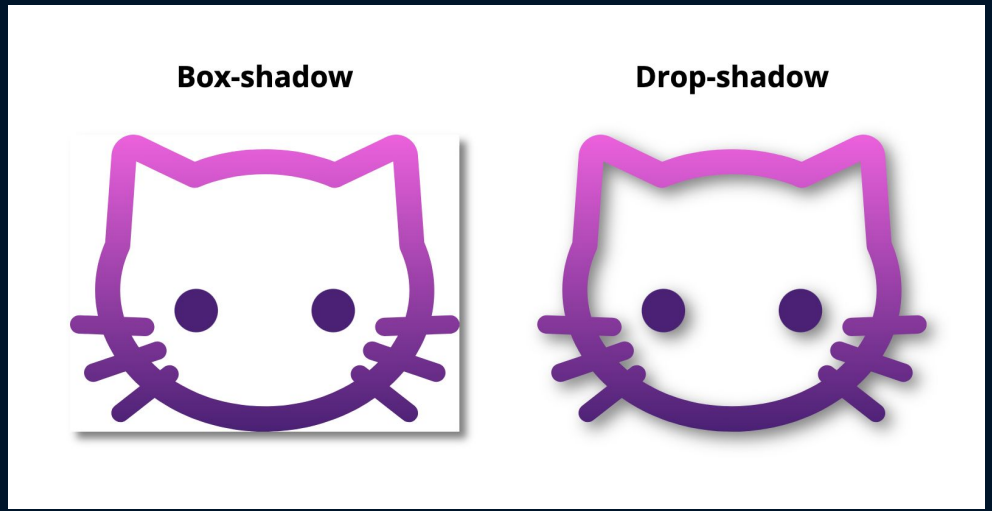
Values:

- **Keyword –** `blur`
- **`url` –** `url("filters.svg#filter-id")`

```
filter: blur(5px);
```

filter

- `drop-shadow(x y blur color)` – inner shadow
- `blur(px)`



color filters

`brightness(0-1)`

`contrast(%)`

`grayscale(%)` – black and white

`hue-rotate(deg)` – shifts the color palette around the color wheel

`invert(%)`

`opacity(%)`

`saturate(%)`

`sepia(%)` – like vintage photo

No Filter Applied

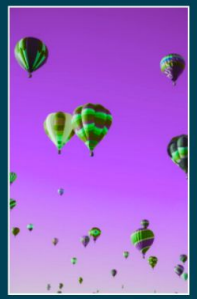
filter: blur(2px);

filter: brightness(0.4);

filter: contrast(200%);

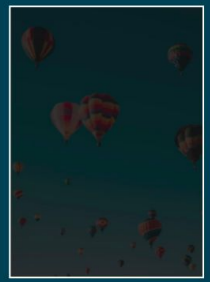
filter: drop-shadow(16px red);

filter: grayscale(80%);



filter: hue-rotate(90deg);

filter: invert(85%);



filter: opacity(15%);

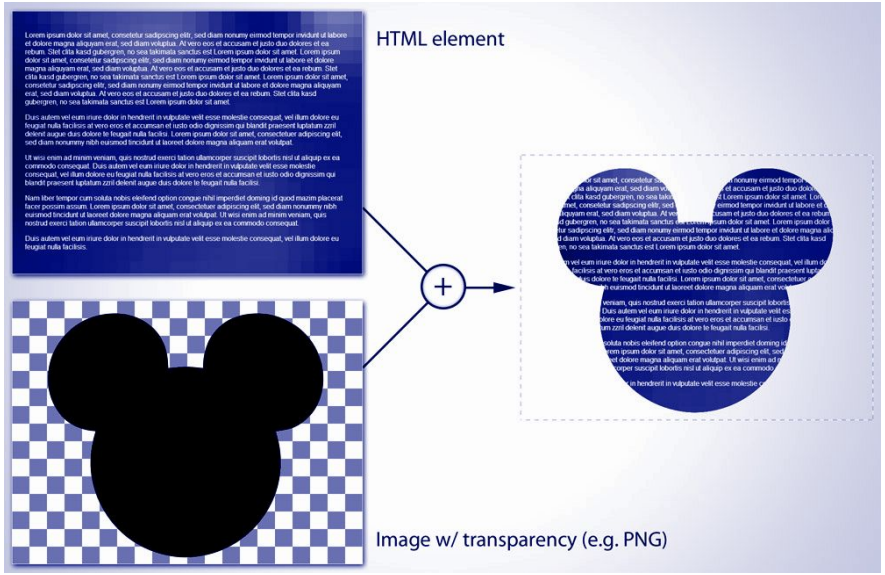
filter: saturate(400%);

filter: sepia(560%);

example →

Masking & clipping

CSS mask



Imagine cutting out a circle in a sheet of paper and placing it over a picture; you've applied a mask.

Used to create complex shapes for elements.

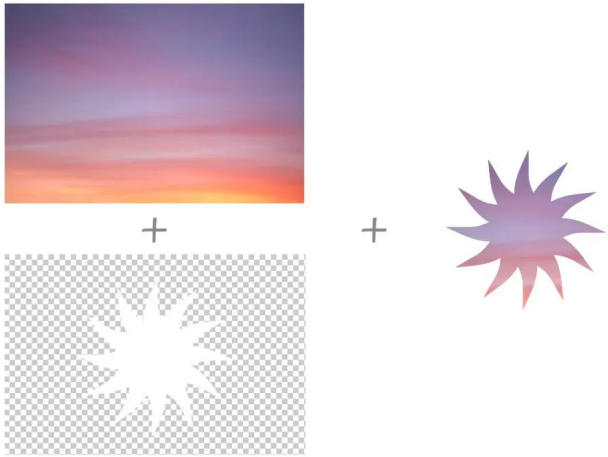
Masks operate based on the alpha channel:

- ➔ **Black –** full invisibility
- ➔ **White –** full visibility
- ➔ **Gray –** partial transparency

CSS mask

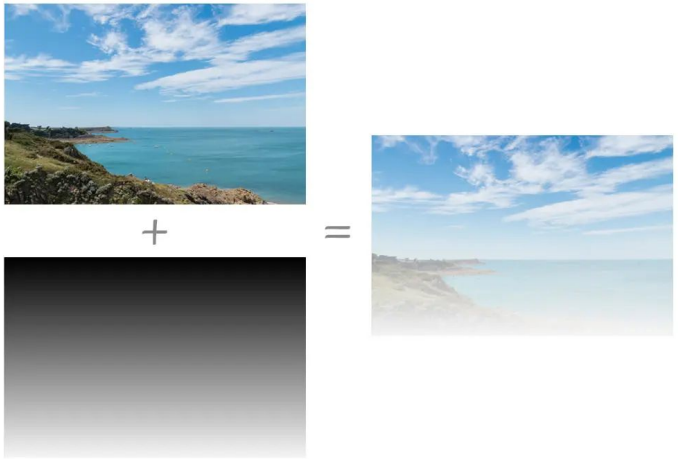
Image:

```
mask: url(mask.png);
```



Gradient:

```
mask: linear-gradient(from, to);
```



example →

CSS mask properties

`mask-image` – the image used as the mask

`mask-mode` – chooses the mask based on transparent or opaque areas

`mask-position` – mask position relative to the element

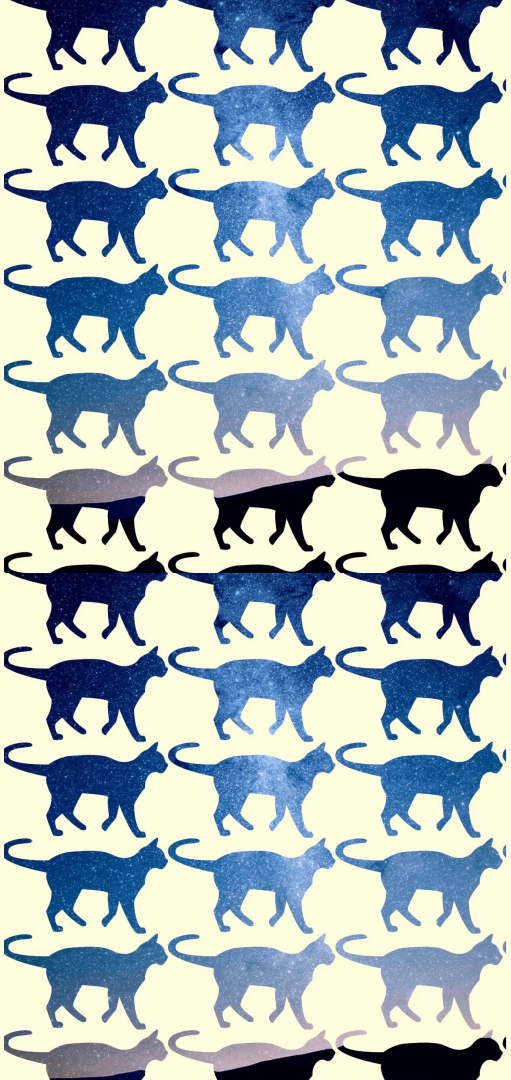
`mask-size`

`mask-repeat` – whether the mask repeats

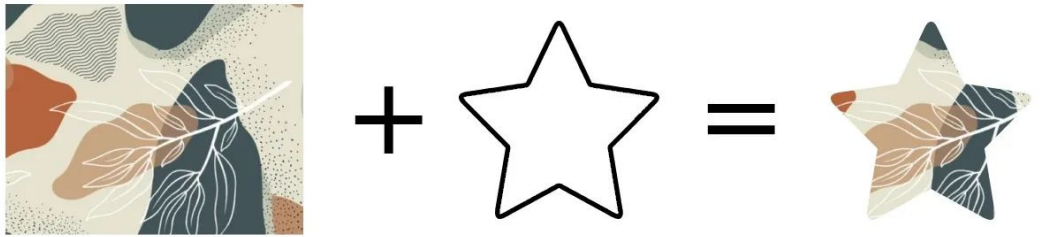
`mask-origin` – defines the starting point of the mask – **`border`, `padding`, `content`**

`mask-clip` – the area to which the mask is applied

`mask-composite` – allows combining mask layers



CSS clip-path



– Defines the area to show or hide

– Consists of shapes or coordinates

Generator clip-path – [Clippy]

example →

CSS clip-path shapes

```
clip-path: inset(width height);
```

```
clip-path: inset(width height round border-radius);
```

```
clip-path: circle(radius at x, y);
```

```
clip-path: ellipse(radius-x, radius-y at x, y);
```

```
clip-path: polygon(vertex, vertex ...);
```

inset – rectangle

circle

ellipse

polygon – any shape with any number of corners

path – SVG path with coordinates

```
clip-path: path("M0.5,1 C0.5 ... “)
```

example →

MASK VS CLIP-PATH

Raster or vector	Vector
Partial Transparency	Opacity Only
Pre-drawn Images	Custom Shapes
More Complex Settings (mask-* properties)	Limits Element Shape (no additional properties)
Static Shape	Animatable Shape Changes
Text Wraps Shape Perimeter	Text Wraps Around Original Rectangle

shape-outside

Text Wrapping Around a Shape.

Shapes:

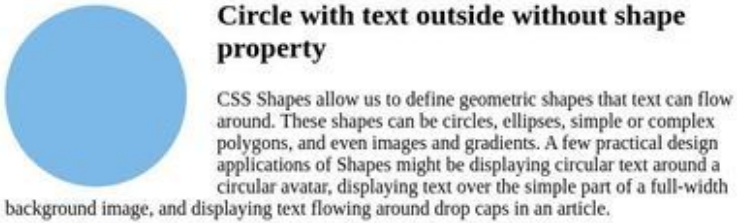
- **circle()** – Creates a circular shape for the text to wrap around.
- **ellipse()**
- **inset()** – Defines a rectangular area.
- **polygon()** – Creates any shape with three or more corners.
- **url()** – Uses an image as the shape for text wrapping.

Applied to an element that the text should wrap around.

example →

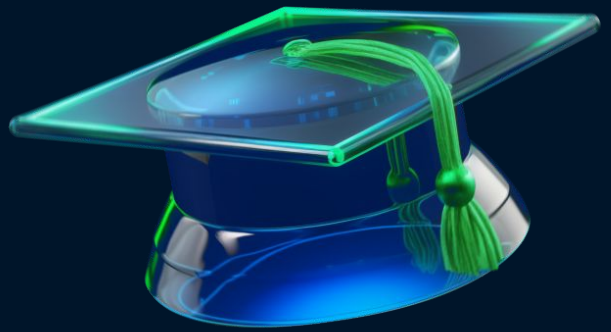
shape-outside

```
clip-path: circle(70% at 0% 50%)
|
shape-outside: circle(70% at 0% 50%)
+
float: left/right
+
margin-left/right
```

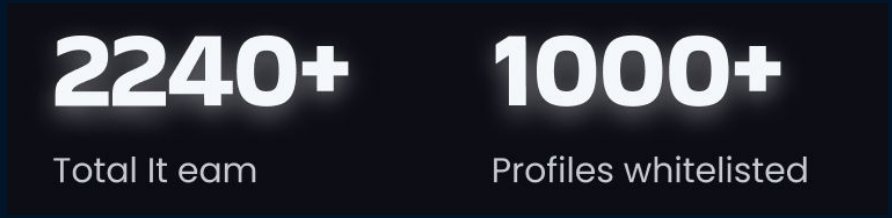


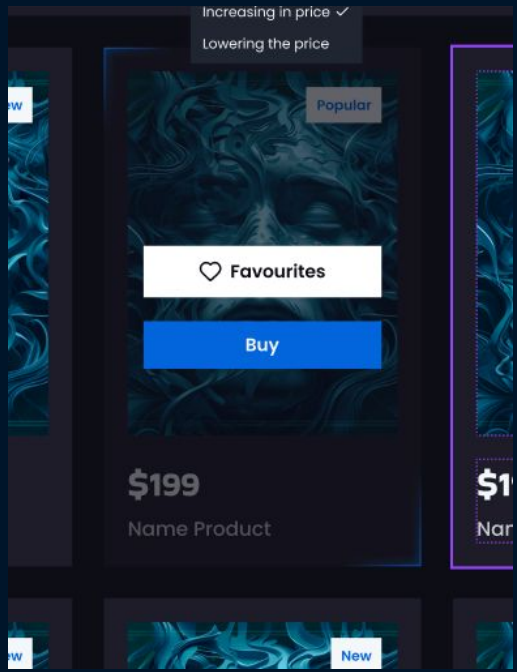
Summary

1. Text and box Shadows
2. Filters
3. Clip path vs Mask



Homework



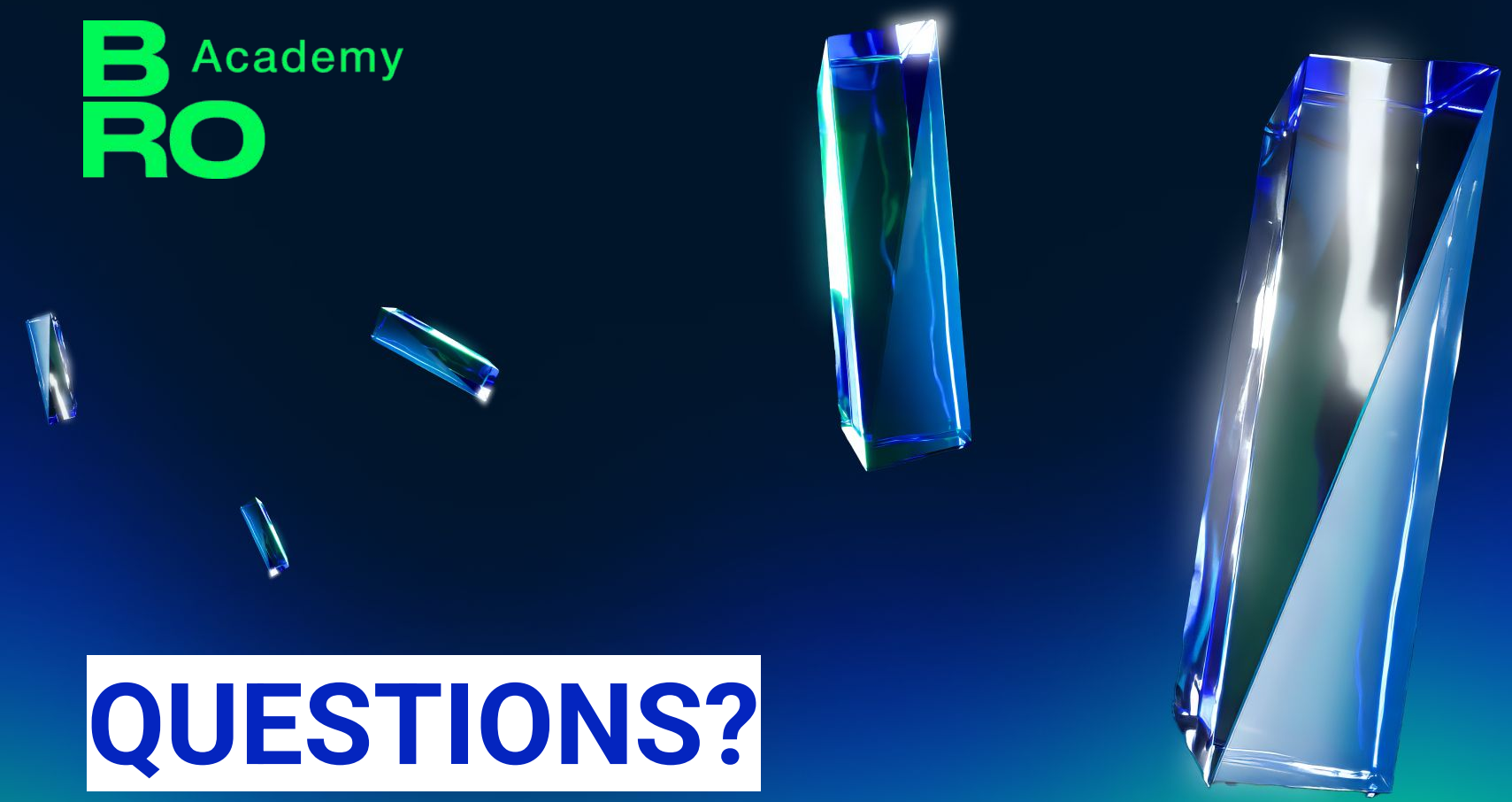


1. **Achieve the highest level of accuracy with the design mockup:**
   - **Apply visual effects such as**

     shadows, shapes, filters and etc

Your website should look exactly like the design mockup.
This is the final stage of work on the website. Next, we will only be adding animations.

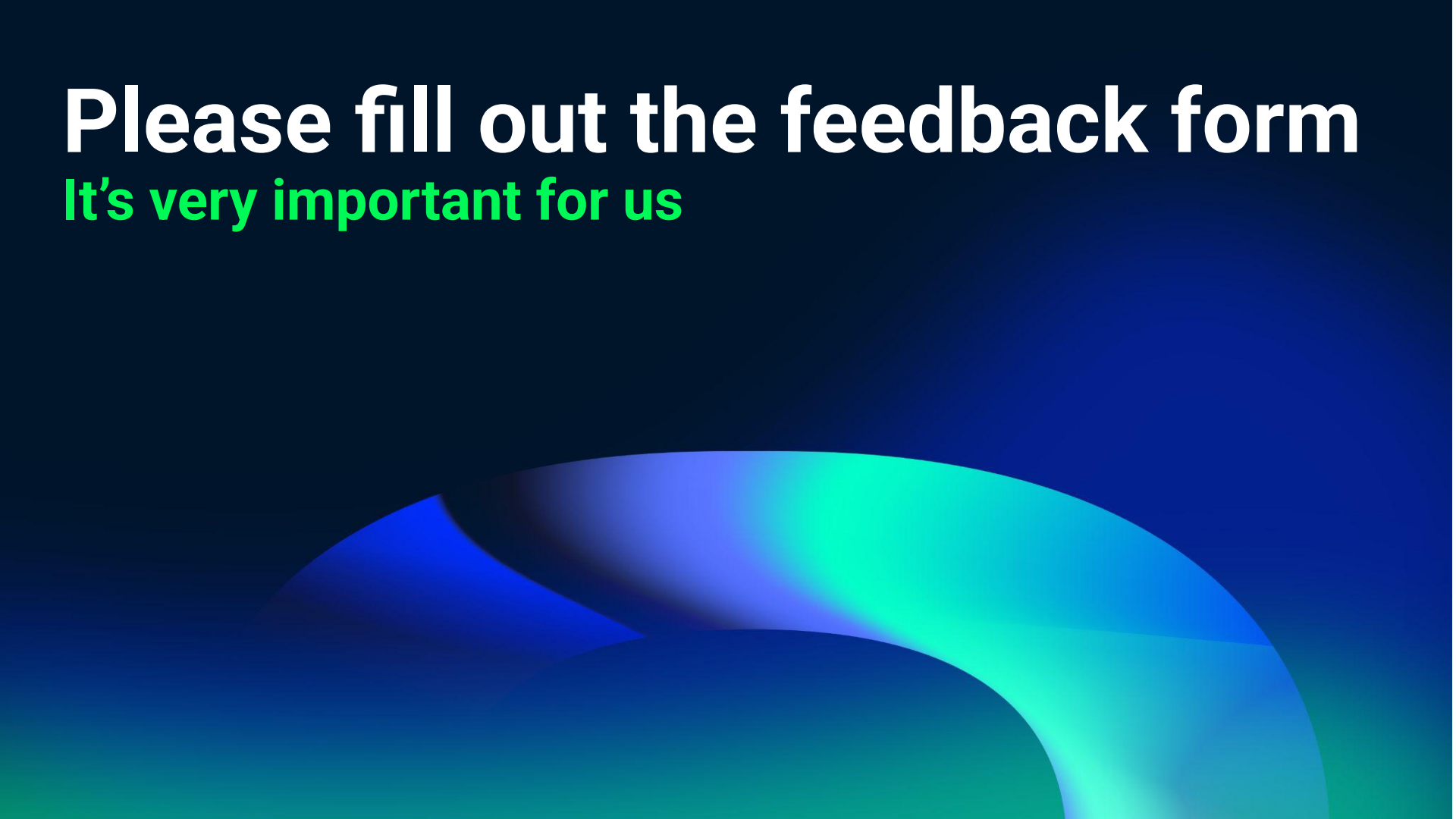
BRO Academy



QUESTIONS?

Please fill out the feedback form

It’s very important for us




THANK YOU!
Have a good evening!